

**FABIANE DE OLIVEIRA**

**EFEITO DE MALHAS ANISOTRÓPICAS BIDIMENSIONAIS SOBRE  
O DESEMPENHO DO MÉTODO *MULTIGRID* GEOMÉTRICO**

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

Orientador: Prof. Dr. Carlos Henrique Marchi.  
Co-orientador: Prof. Dr. Marcio Augusto Villela Pinto.

**CURITIBA**

**2010**

Oliveira, Fabiane de

Efeito de malhas anisotrópicas bidimensionais sobre o desempenho do método multigrid geométrico / Fabiane de Oliveira. – Curitiba, 2010. 204 f. : il. ; graf., tab.

Orientador: Carlos Henrique Marchi

Co-orientador: Marcio Augusto Villela Pinto

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

1. Sistemas lineares. 2. Dinâmica dos fluidos. 3. Métodos de redes múltiplas (Análise numérica). I. Marchi, Carlos Henrique. II. Pinto, Marcio Augusto Villela. III. Título.

CDD 003.74

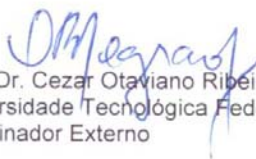
## TERMO DE APROVAÇÃO

FABIANE DE OLIVEIRA

### EFEITO DE MALHAS ANISOTRÓPICAS BIDIMENSIONAIS SOBRE O DESEMPENHO DO MÉTODO MULTIGRID GEOMÉTRICO

Tese aprovada como requisito parcial à obtenção de grau de Doutor em Engenharia Mecânica, área de concentração Fenômenos de Transporte e Mecânica dos Sólidos, no Programa de Pós-Graduação em Engenharia Mecânica, Setor de Tecnologia da Universidade Federal do Paraná.

Banca Examinadora:



Prof. Dr. Cezar Otaviano Ribeiro Negrão  
Universidade Tecnológica Federal do Paraná  
Examinador Externo



Prof. Dr. Francisco Marcondes  
Universidade Federal do Ceará  
Examinador externo



Profª. Drª. Viviana Cocco Mariani  
Pontifícia Universidade Católica do Paraná  
Examinadora Externa



Prof. Dr. Luciano Kiyoshi Araki  
Universidade Federal do Paraná  
Examinador Interno



Prof. Dr. Carlos Henrique Marchi  
Universidade Federal do Paraná  
Presidente

Curitiba, 26 de maio de 2010.

*A Deus, por tudo.*

*Aos meus pais e irmã.*

*Aos meus padrinhos e primos.*

## **AGRADECIMENTOS**

Agradeço ao meu orientador, Prof. Dr. Carlos Henrique Marchi e ao meu co-orientador Prof. Dr. Marcio Augusto Villela Pinto por terem aceitado me orientar neste trabalho e por todo o conhecimento recebido.

Agradeço também, a todos meus professores pelos conhecimentos por eles transmitidos.

Agradeço aos membros da banca examinadora, Cezar Otaviano Ribeiro Negrão, Francisco Marcondes, Luciano Kiyoshi Araki e Viviana Cocco Mariani pelo tempo dispendido na leitura deste trabalho e pelas importantes sugestões apontadas.

Agradeço ao Programa de Pós-Graduação em Engenharia Mecânica (PG-MEC) 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 minha família pelo apoio e incentivo durante a realização deste curso.

Agradeço aos meus amigos que direta ou indiretamente me ajudaram no desenvolvimento deste trabalho, de uma forma especial aos amigos do LENA.

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

## RESUMO

O objetivo deste trabalho é reduzir o tempo de CPU (*Central Processing Unit*) necessário para resolver problemas difusivos bidimensionais, discretizados com malhas anisotrópicas. Os modelos matemáticos considerados referem-se a três problemas bidimensionais lineares de condução de calor, governados pelas equações de Laplace e Poisson, com condições de contorno de Dirichlet. O método de diferenças finitas é usado para discretizar as equações diferenciais com esquema de diferença central (CDS) de segunda ordem. Os sistemas de equações algébricas são resolvidos usando-se os métodos Gauss-Seidel lexicográfico e *red-black*, associados ao método *multigrid* geométrico com esquema de correção (CS) e ciclo V. Foram resolvidos problemas com anisotropia geométrica, diversas malhas e razões de aspecto. O número de iterações internas ( $v$ ) foi verificado em um intervalo de 1 a 3.000. A análise do número de níveis foi realizada utilizando-se o número máximo de níveis ( $L_{\text{máximo}}$ ) e  $L_{\text{máximo}} - 1, L_{\text{máximo}} - 2, L_{\text{máximo}} - 3, L_{\text{máximo}} - 4$ . São feitas comparações entre diversos algoritmos de engrossamento: engrossamento padrão (EP), semi-engrossamento (SE), semi-engrossamento completo (SEC), semi-engrossamento seguido de engrossamento padrão (SE-EP) e engrossamento padrão seguido de semi-engrossamento (EP-SE). Também são realizadas comparações entre alguns operadores de restrição: injeção, meia ponderação e ponderação completa. São propostos três tipos de restrição para problemas anisotrópicos: meia ponderação geométrica, ponderação geométrica completa e ponderação parcial. O processo de prolongação utilizado é a interpolação bilinear. Também foi investigado o efeito sobre o tempo de CPU causado por: número de pontos na malha ( $N$ ); número de iterações internas no *solver* ( $v$ ); e número de malhas ( $L$ ). Verificou-se que: o algoritmo SE-EP é o mais rápido entre os cinco algoritmos testados; e confirmou-se que, para problemas isotrópicos e anisotrópicos, o *solver* Gauss-Seidel *red-black* com restrição por ponderação parcial resulta em menor tempo de CPU em relação ao Gauss-Seidel lexicográfico.

**Palavras-chave:** Algoritmos de engrossamento. Diferenças finitas. Laplace. Operadores de restrição. *Multigrid* geométrico. Poisson.

## ABSTRACT

The purpose of this work is to reduce the CPU (*Central Processing Unit*) time necessary to solve two-dimensional diffusive problems, discretized with anisotropic grids. The mathematical models considered are related to three two-dimensional linear heat diffusion problems, governed by Laplace and Poisson equations, with Dirichlet boundary conditions. The differential equations are discretized by the finite difference method, with second order approximations, given by central differences scheme (CDS) for derivatives. The systems of equations are solved with the lexicographical and red-black Gauss-Seidel methods, associated to geometric multigrid with correction scheme (CS) and V-cycle. For problems with geometric anisotropic, several grids and aspect ratios are considered. The inner iterations are verified for the interval between 1 and 3.000. The analyses for the number of levels are accomplished using the maximum level number ( $L_{m\acute{a}ximo}$ ) and  $L_{m\acute{a}ximo} - 1, L_{m\acute{a}ximo} - 2, L_{m\acute{a}ximo} - 3, L_{m\acute{a}ximo} - 4$ . Comparisons among several semi-coarsening algorithms are made: standard coarsening (EP), semi-coarsening (SE), full semi-coarsening (SEC), semi-coarsening followed by standard coarsening (SE-EP) and standard coarsening followed by semi-coarsening (EP-SE). Comparisons are also made among some restrictions schemes: injection, half weighting and full weighting. Three restriction schemes for anisotropic problems are proposed: geometric half weighting, geometric full weighting and partial weighting. The prolongation process used is the bilinear interpolation. The CPU time changes caused by the following effects are also studied: the number of nodes of a grid ( $N$ ); the number of inner iterations of the solver ( $\nu$ ); and the number of grids used ( $L$ ). It was verified that: the algorithm SE-EP is the fastest of the five algorithms tested; for isotropic and anisotropic problems, red-black Gauss-Seidel solver with partial weighting restriction results in smaller CPU time compared to lexicographical Gauss-Seidel.

**Keywords:** Coarsening algorithm. Finite difference. Geometric multigrid. Laplace. Poisson. Restriction operators.

## LISTA DE ALGORITMOS

Algoritmo 2.1: Procedimento de Gauss-Seidel lexicográfico para $Au = f$ .....	40
Algoritmo 2.2: Procedimento de Gauss-Seidel <i>red-black</i> para $Au = f$ .....	41
Algoritmo 2.3: Ciclo V com esquema CS para várias malhas. ....	51
Algoritmo 2.4: <i>Multigrid</i> com ciclo V, Esquema CS para diversas malhas e diversos ciclos. ....	52



## LISTA DE FIGURAS

Figura 1.1: Discretização.....	25
Figura 1.2: Exemplos de malhas estruturadas: (a) uniforme e (b) não-uniforme.....	27
Figura 1.3: Exemplo de malha não-estruturada.....	27
Figura 1.4: Ordenação lexicográfica. ....	27
Figura 1.5: Ordenação <i>red-black</i> .....	28
Figura 2.1: Modos de Fourier.....	44
Figura 2.2: Comportamento da suavização do erro em métodos iterativos.....	44
Figura 2.3: Comportamento da suavização do erro em métodos iterativos.....	44
Figura 2.4: Processo de engrossamento e geração de malhas. ....	45
Figura 2.5: Engrossamento padrão com razão de engrossamento $r = 2$ .....	47
Figura 2.6: Restrição por injeção (Trottenberg et al. 2001). ....	47
Figura 2.7: Prolongação bilinear (Extraída de Trottenberg et al. 2001).....	49
Figura 2.8: Diagramas: (a) ciclo V, (b) F, (c) dente-de-serra, (d) W. ....	50
Figura 2.9: Anisotropia tipo I ( $h_x \neq h_y, N_x = N_y, C_x \neq C_y$ ). ....	56
Figura 2.10: Anisotropia tipo II ( $h_x = h_y, N_x \neq N_y$ e $C_x \neq C_y$ ). ....	56
Figura 2.11: Anisotropia tipo III ( $h_x \neq h_y, N_x \neq N_y, C_x = C_y$ ). ....	57
Figura 2.12: Anisotropia tipo IV ( $h_x \neq h_y, N_x \neq N_y, C_x \neq C_y$ ). ....	57
Figura 2.13: Algoritmos de engrossamento: a) EP, b) SE, c) SE-EP, d) EP-SE.....	59
Figura 3.1: Domínio bidimensional de cálculo para a equação de Laplace. ....	62
Figura 4.1: Erros envolvidos nos métodos de engenharia (adaptada de Marchi e Schneider, 2004).....	68
Figura 4.2: Ordem efetiva e ordem aparente para a temperatura no nó central. ....	72
Figura 4.3: Ordem efetiva e ordem aparente para a temperatura média.....	72
Figura 4.4: Ordem efetiva e ordem aparente para a norma $l_\infty$ . ....	73
Figura 4.5: Ordem efetiva e ordem aparente para a norma $\bar{l}_1$ .....	73
Figura 4.6: Ordem efetiva e ordem aparente para a temperatura média em uma malha anisotrópica.....	74
Figura 4.7: Erro numérico para a equação de Laplace senoidal.....	75
Figura 4.8: Erro numérico para a equação de Laplace linear. ....	75

Figura 4.9: Erro numérico para a equação de Poisson. ....	76
Figura 4.10: Comportamento da norma $l_2$ do resíduo em função do número de ciclos $V$ para uma maha $8193 \times 8193$ .....	77
Figura 4.11: Norma infinito ( $l_\infty$ ) do erro numérico <i>versus</i> número de incógnitas para o método <i>multigrid</i> . ....	78
Figura 4.12: Temperatura média <i>versus</i> número de incógnitas ( $N$ ) para a equação de Laplace senoidal.....	80
Figura 5.1: Exemplos de malhas: a) isotrópica e b) anisotrópica com $Q = 2$ .....	88
Figura 5.2: Algoritmo semi-engrossamento completo (SEC).....	88
Figura 5.3: Tempo de CPU <i>versus</i> $\nu$ para $E = 262.144$ , SE-EP, equação de Laplace senoidal e $Q < 1$ .....	91
Figura 5.4: Tempo de CPU <i>versus</i> $\nu$ para $E = 262.144$ , SE-EP, equação de Laplace senoidal e $Q > 1$ . ....	91
Figura 5.5: Tempo de CPU <i>versus</i> $\nu$ para $Q = 16$ , $E = 262.144$ e EP.....	93
Figura 5.6: Tempo de CPU <i>versus</i> $\nu$ para $Q = 16$ , $E = 262.144$ e SE.....	93
Figura 5.7: Tempo de CPU <i>versus</i> $\nu$ para $Q = 16$ , $E = 262.144$ e SEC. ....	94
Figura 5.8: Tempo de CPU <i>versus</i> $\nu$ para $Q = 16$ , $E = 262.144$ e EP-SE.....	94
Figura 5.9: Tempo de CPU <i>versus</i> $\nu$ para $Q = 1$ , $E = 262.144$ , EP e <i>solver</i> GS-LEX. ....	97
Figura 5.10: Tempo de CPU <i>versus</i> $\nu$ para $Q = 1/64$ , $E = 262.144$ , SE-EP.....	99
Figura 5.11: Tempo de CPU <i>versus</i> $\nu$ para $Q = 1/16$ , $E = 262.144$ , SE-EP.....	99
Figura 5.12: Tempo de CPU <i>versus</i> $\nu$ para $Q = 16$ , $E = 262.144$ , SE-EP.....	100
Figura 5.13: Tempo de CPU <i>versus</i> $\nu$ para $Q = 64$ , $E = 262.144$ , SE-EP.....	100
Figura 5.14: Tempo de CPU <i>versus</i> número de níveis ( $L$ ) para $E = 262.144$ , EP e equação de Laplace senoidal. ....	102
Figura 5.15: Tempo de CPU <i>versus</i> número de níveis para $E = 262.144$ , SE e equação de Laplace senoidal. ....	103
Figura 5.16: Tempo de CPU <i>versus</i> número de níveis para $E = 262.144$ , SEC e equação de Laplace senoidal. ....	103
Figura 5.17: Tempo de CPU <i>versus</i> número de níveis para $E = 262.144$ , EP-SE e equação de Laplace senoidal. ....	104
Figura 5.18: Tempo de CPU <i>versus</i> número de níveis para $E = 262.144$ , algoritmo SE-EP e equação de Laplace senoidal. ....	104

Figura 5.19: Tempo de CPU <i>versus</i> razão de aspecto para diversas razões de aspecto, $E = 262.144$ e equação de Laplace senoidal. ....	106
Figura 5.20: Tempo de CPU <i>versus</i> razão de aspecto para diversas razões de aspecto, $E = 262.144$ e equação de Laplace linear. ....	106
Figura 5.21: Tempo de CPU <i>versus</i> razão de aspecto para diversas razões de aspecto, $E = 262.144$ e equação de Poisson. ....	107
Figura 5.22: Tempo de CPU <i>versus</i> $E$ para $Q = 64$ e equação de Laplace senoidal. ....	108
Figura 5.23: Tempo de CPU <i>versus</i> $E$ , para $Q = 1/64$ e equação de Laplace senoidal. ....	108
Figura 6.1: Nó utilizado para a restrição por injeção. ....	116
Figura 6.2: Nós utilizados para a restrição por meia ponderação. ....	117
Figura 6.3: Nós utilizados para a restrição por ponderação completa. ....	118
Figura 6.4: Tempo de CPU <i>versus</i> razão de aspecto para o algoritmo SE-EP, $E = 262.144$ para as equações de Laplace senoidal, Laplace linear e Poisson com $\varepsilon = 10^{-10}$ . ....	121
Figura 6.5: Tempo de CPU <i>versus</i> razão de aspecto o algoritmo SE-EP, $E = 262.144$ para as equações de Laplace senoidal, Laplace linear e Poisson com $\varepsilon = 10^{-7}$ . ....	122
Figura 6.6: Tempo de CPU <i>versus</i> número de elementos para $Q \leq 1$ , algoritmo SE-EP e equação de Laplace senoidal com $\varepsilon = 10^{-10}$ . ....	123
Figura 6.7: Tempo de CPU <i>versus</i> número de elementos para $Q \geq 1$ , algoritmo SE-EP e equação de Laplace senoidal com $\varepsilon = 10^{-10}$ . ....	123
Figura 6.8: Tempo de CPU <i>versus</i> número de elementos para $Q \leq 1$ , algoritmo SE-EP e equação de Laplace linear com $\varepsilon = 10^{-10}$ . ....	124
Figura 6.9: Tempo de CPU <i>versus</i> número de elementos para $Q \geq 1$ , algoritmo SE-EP e equação de Laplace linear com $\varepsilon = 10^{-10}$ . ....	124
Figura 6.10: Tempo de CPU <i>versus</i> número de elementos para $Q \leq 1$ , algoritmo SE-EP e equação de Poisson com $\varepsilon = 10^{-10}$ . ....	125
Figura 6.11: Tempo de CPU <i>versus</i> número de elementos para $Q \geq 1$ , algoritmo SE-EP e equação de Poisson com $\varepsilon = 10^{-10}$ . ....	125
Figura 6.12: Tempo de CPU <i>versus</i> $v_{EP}$ para $Q = 1/64$ , $E = 262.144$ , SE-EP ótimo e equação de Laplace senoidal. ....	132
Figura 6.13: Tempo de CPU <i>versus</i> $v_{EP}$ para $Q = 64$ , $E = 262.144$ , SE-EP ótimo e equação de Laplace senoidal. ....	132

Figura 6.14: Tempo de CPU <i>versus</i> $\nu$ para $Q = 1$ , $E = 262.144$ , EP ótimo para as equações de Laplace senoidal, Laplace linear e Poisson. ....	133
Figura 6.15: Tempo de CPU <i>versus</i> número de níveis ( $L$ ) para $E = 262.144$ , algoritmo SE-EP ótimo e equação de Laplace senoidal. ....	134
Figura 6.16: Tempo de CPU <i>versus</i> número de níveis ( $L$ ) para $E = 262.144$ , algoritmo SE-EP ótimo e equação de Laplace senoidal. ....	134
Figura 6.17: Tempo de CPU <i>versus</i> número de elementos ( $E$ ) para o algoritmo SE-EP, $Q = 1/64$ e equação de Laplace senoidal. ....	135
Figura 6.18: Tempo de CPU <i>versus</i> número de elementos ( $E$ ) para o algoritmo SE-EP, $Q = 64$ , equação de Laplace senoidal. ....	136
Figura 6.19: Tempo de CPU <i>versus</i> número de elementos ( $E$ ) para o algoritmo EP e equação de Laplace senoidal e $Q = 1$ . ....	136
Figura 6.20: Tempo de CPU <i>versus</i> número de elementos ( $E$ ) para um problema isotrópico e equação de Laplace senoidal. ....	137
Figura 6.21: Tempo de CPU <i>versus</i> número de elementos ( $E$ ) para um problema anisotrópico com $Q = 1/64$ , e equação de Laplace senoidal. ....	138
Figura A.1: Solução analítica da equação de Laplace senoidal. ....	156
Figura A.2: Solução analítica da equação de Laplace linear. ....	156
Figura A.3: Solução analítica da equação de Poisson. ....	157
Figura A.4: Perfil da temperatura em $x = 1/2$ . ....	157
Figura A.5: Perfil da temperatura em $y = 1/2$ . ....	158
Figura D.1: Tempo de CPU <i>versus</i> $\nu$ para $E = 4.194.304$ , SE-EP e equação de Laplace senoidal. ....	169
Figura D.2: Tempo de CPU <i>versus</i> $\nu$ para $E = 67.108.864$ , SE-EP e equação de Laplace senoidal. ....	169
Figura D.3: Tempo de CPU <i>versus</i> $\nu$ para $E = 262.144$ , SE-EP e equação de Laplace linear. ....	170
Figura D.4: Tempo de CPU <i>versus</i> $\nu$ para $E = 262.144$ , SE-EP e equação de Poisson. ....	170
Figura D.5: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace senoidal, algoritmo EP e $E = 65.536$ . ....	172
Figura D.6: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace senoidal, algoritmo EP e $E = 262.144$ . ....	173

Figura D.7: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace senoidal, algoritmo EP e $E = 4.194.304$ .....	174
Figura D.8: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace senoidal, algoritmo SE e $E = 65.536$ .....	176
Figura D.9: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace senoidal, algoritmo SE e $E = 262.144$ .....	177
Figura D.10: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace senoidal, algoritmo SEC e $E = 4.096$ .....	178
Figura D.11: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace senoidal, algoritmo SEC e $E = 65.536$ .....	179
Figura D.12: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace senoidal, algoritmo SEC e $E = 262.144$ .....	180
Figura D. 13: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace senoidal, algoritmo SEC e $E = 1.048.576$ .....	181
Figura D.14: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace senoidal, algoritmo EP-SE e $E = 65.536$ .....	183
Figura D.15: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace senoidal, algoritmo EP-SE e $E = 262.144$ .....	184
Figura D.16: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace linear, algoritmo EP e $E = 262.144$ .....	185
Figura D.17: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace linear, algoritmo SE e $E = 262.144$ .....	185
Figura D.18: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace linear, algoritmo SEC e $E = 262.144$ .....	186
Figura D.19: Tempo de CPU <i>versus</i> $\nu$ para a equação de Laplace linear, algoritmo EP-SE e $E = 262.144$ .....	186
Figura D.20: Tempo de CPU <i>versus</i> $\nu$ para a equação de Poisson, algoritmo EP e $E = 262.144$ .....	187
Figura D.21: Tempo de CPU <i>versus</i> $\nu$ para a equação de Poisson, algoritmo SE e $E = 262.144$ .....	187
Figura D.22: Tempo de CPU <i>versus</i> $\nu$ para a equação de Poisson, algoritmo SEC e $E = 262.144$ .....	188

Figura D.23: Tempo de CPU <i>versus</i> $\nu$ para a equação de Poisson, algoritmo EP-SE e $E = 262.144$ .....	188
Figura D.24: Tempo de CPU <i>versus</i> número de elementos para $Q = 1/16$ e equação de Laplace linear. ....	189
Figura D.25: Tempo de CPU <i>versus</i> número de elementos para $Q = 16$ e equação de Laplace linear. ....	189
Figura D.26: Tempo de CPU <i>versus</i> número de elementos para $Q = 1/16$ e equação de Poisson.....	190
Figura D.27: Tempo de CPU <i>versus</i> número de elementos para $Q = 16$ e equação de Poisson. ....	190
Figura E. 1: Tempo de CPU <i>versus</i> $\nu_{EP}$ para $Q = 1/6$ , $E = 262.144$ e SE-EP ótimo e equação de Laplace linear.....	195
Figura E. 2: Tempo de CPU <i>versus</i> $\nu_{EP}$ para $Q = 16$ , $E = 262.144$ e SE-EP ótimo e equação de Laplace linear.....	196
Figura E. 3: Tempo de CPU <i>versus</i> $\nu_{EP}$ para $Q = 1/16$ , $E = 262.144$ e SE-EP ótimo e equação de Poisson.....	196
Figura E. 4: Tempo de CPU <i>versus</i> $\nu_{EP}$ para $Q = 16$ , $E = 262.144$ e SE-EP ótimo e equação de Poisson.....	197
Figura E.5: Tempo de CPU <i>versus</i> número de elementos ( $E$ ) para o algoritmo SE-EP (padrão) e SE-EP (ótimo) para $Q = 1/16$ e equação de Laplace linear. ....	198
Figura E.6: Tempo de CPU <i>versus</i> número de elementos ( $E$ ) para o algoritmo SE-EP (padrão) e SE-EP (ótimo) para $Q = 16$ e equação de Laplace linear. ....	198
Figura E. 7: Tempo de CPU <i>versus</i> número de elementos ( $E$ ) para o algoritmo EP <i>versus</i> EP (ótimo) para $Q = 1$ e equação de Laplace linear.....	199
Figura E.8: Tempo de CPU <i>versus</i> número de elementos ( $E$ ) para o algoritmo SE-EP (padrão) e SE-EP (ótimo) para $Q = 1/16$ e equação de Poisson. ....	200
Figura E.9: Tempo de CPU <i>versus</i> número de elementos ( $E$ ) para o algoritmo SE-EP padrão <i>versus</i> SE-EP (ótimo) para $Q = 16$ e equação de Poisson. ....	200
Figura E.10: Tempo de CPU <i>versus</i> número de elementos ( $E$ ) para o algoritmo EP <i>versus</i> EP (ótimo) para $Q = 1$ e equação de Poisson. ....	201

## LISTA DE TABELAS

Tabela 1.1: Comparação entre os métodos de solução de problemas em engenharia (adaptada de Tannehill et al. 1997).....	24
Tabela 3.1: Problemas resolvidos.....	61
Tabela 3.2: Soluções analíticas para as equações em estudo. ....	62
Tabela 4.1: Comparação entre as normas obtidas no método <i>singlegrid</i> . ....	79
Tabela 4.2: Comparação entre as normas obtidas no método <i>multigrid</i> . ....	79
Tabela 4.3: Comparação entre as normas para uma malha 17x17. ....	81
Tabela 4.4: Comparação entre as normas para uma malha 33x33. ....	82
Tabela 4.5: Comparação entre as normas para uma malha 65x65. ....	82
Tabela 4.6: Comparação entre as normas para uma malha 129x129. ....	83
Tabela 4.7: Comparação entre as variáveis de interesse para a anisotropia IV, razão de aspecto 1/16 e malha 4097x257 e equação de Laplace senoidal.....	86
Tabela 5.1: Malhas utilizadas para determinar $v_{\text{ótimo}}$ para o algoritmo SE-EP para a equação de Laplace senoidal. ....	90
Tabela 5.2: $v_{\text{ótimo}}$ para o algoritmo SE-EP e equação de Laplace senoidal.....	92
Tabela 5.3: Malhas utilizadas para determinar $v_{\text{ótimo}}$ para os algoritmos: EP, SE, SEC e EP-SE e equação de Laplace senoidal. ....	92
Tabela 5.4: $v$ com melhor desempenho médio para cada razão de aspecto, algoritmo SE e equação de Laplace senoidal. ....	95
Tabela 5.5: Parâmetros para o intervalo de $v$ com melhor desempenho médio para o algoritmo de SE.....	95
Tabela 5.6: $v$ com melhor desempenho médio para cada razão de aspecto e equação de Laplace senoidal. ....	96
Tabela 5.7: $v$ com melhor desempenho médio para cada razão de aspecto para as equações de Laplace linear (LL) e Poisson (Po).....	98
Tabela 5.8: Malhas utilizadas para determinar $L_{\text{ótimo}}$ .....	102
Tabela 5.9: Malhas utilizadas para análise de $E$ .....	107
Tabela 5.10: Valor da ordem ( $p$ ) da Eq. (5.1) para os algoritmos de engrossamento. ....	109
Tabela 5.11: Valor da ordem ( $c$ ) da Eq. (5.1) para os algoritmos de engrossamento. ....	110

Tabela 5.12: <i>Speed-up</i> do SE-EP em relação aos algoritmos: EP, SE, SEC, EP-SE para a equação de Laplace senoidal. ....	111
Tabela 5.13: <i>Speed-up</i> do SE-EP em relação aos algoritmos: EP, SE, SEC, EP-SE para a equação de Laplace linear. ....	111
Tabela 5.14: <i>Speed-up</i> do SE-EP em relação aos algoritmos: EP, SE, SEC, EP-SE para a equação de Poisson.....	111
Tabela 5.15: <i>Speed-up</i> do SEC em relação ao SE para as equações de Laplace senoidal, Laplace linear e Poisson. ....	112
Tabela 6.1: Valores de $c$ e $p$ para Eq. (5.1) e algoritmo SE-EP. ....	126
Tabela 6.2: Tempo de CPU (s) para a análise do bloco SE, malha 2049x129 e $Q = 1/16$ .....	128
Tabela 6.3: Tempo de CPU (s) para a análise do bloco SE, malha 129x2049 e $Q = 16$ .....	128
Tabela 6.4: Tempo de CPU (s) para a análise do bloco EP, malha 2049x129 e $Q = 1/16$ .....	128
Tabela 6.5: Tempo de CPU (s) para a análise do bloco EP, malha 129x2049 e $Q = 16$ .....	129
Tabela 6.6: Tempo de CPU (s) para a análise do bloco EP, malha 513x513 e $Q = 1$ .....	129
Tabela 6.7: Parâmetros para os algoritmos SE-EP padrão e SE-EP ótimo. ....	130
Tabela 6.8: Parâmetros para os algoritmos EP e EP ótimo. ....	130
Tabela 6.9: <i>Speed-up</i> do SG: GS-RB em relação SE-EP ótimo para a equação de Laplace senoidal.....	139
Tabela 6.10: Valores de $c$ e $p$ para Eq. (5.1) e algoritmo SE-EP ótimo. ....	140
Tabela 6.11: Valores de $c$ e $p$ para SE-EP padrão e SE-EP ótimo e equação de Laplace senoidal.....	140
Tabela 6.12: <i>Speed-up</i> do SE-EP padrão em relação SE-EP ótimo. ....	140
Tabela C. 1: Soluções analíticas para as equações de Laplace senoidal, Laplace linear e Poisson.....	161
Tabela C. 2: Resultados numéricos para as variáveis de interesse referentes à equação de Laplace senoidal. ....	162
Tabela C. 3: Resultados numéricos para as variáveis de interesse referentes à equação de Laplace linear. ....	162
Tabela C. 4: Resultados numéricos para as variáveis de interesse referentes à equação de Poisson.....	163
Tabela C. 5: Cálculo de $h$ para a discretização dos modelos numéricos. ....	164
Tabela C. 6: Ordem efetiva ( $p_E$ ), ordem aparente ( $p_U$ ) para a aproximação numérica da temperatura no nó central para a Equação de Laplace senoidal.....	164



Tabela C. 7: Ordem efetiva ( $p_E$ ), ordem aparente ( $p_U$ ) para a aproximação numérica da temperatura média e Equação de Laplace senoidal.....	165
Tabela C. 8: Ordem efetiva ( $p_E$ ), ordem aparente ( $p_U$ ) para a norma $l_\infty$ e Equação de Laplace senoidal.....	165
Tabela C. 9: Ordem efetiva ( $p_E$ ), ordem aparente ( $p_U$ ) para a norma $l_1$ média para a Equação de Laplace senoidal.....	165
Tabela C. 10: Equação de Poisson para a temperatura no nó central.....	166
Tabela C. 11: Equação de Poisson para a temperatura numérica média.....	166
Tabela C. 12: Equação de Poisson para norma $l_\infty$ .....	167
Tabela C. 13: Equação de Poisson para norma $l_1$ média.....	167
Tabela D.1: Malhas utilizadas para a determinação do $\nu_{\text{ótimo}}$ .....	168
Tabela D.2: $\nu_{\text{ótimo}}$ para o algoritmo SE-EP e equação de Poisson.....	171
Tabela D.3: $\nu_{\text{ótimo}}$ para cada razão de aspecto, algoritmo EP e equação de Laplace senoidal.....	175
Tabela D.4: Parâmetros para o intervalo recomendado de $\nu$ , $Q = 16$ , EP, e equação de Laplace senoidal.....	175
Tabela D. 5: $\nu_{\text{ótimo}}$ para cada razão de aspecto, algoritmo SEC e equação de Laplace senoidal.....	182
Tabela D.6: $\nu_{\text{ótimo}}$ para cada razão de aspecto, algoritmo EP-SE e equação de Laplace senoidal.....	182
Tabela D.7: Malhas utilizadas para análise de $E$ para as equações de Laplace linear e Poisson.....	189
Tabela D.8: Valor da ordem ( $p$ ) da Eq. (5.1) para os algoritmos de engrossamento para a equação de Laplace linear.....	191
Tabela D.9: Valor do coeficiente ( $c$ ) da Eq. (5.1) para os algoritmos de engrossamento para a equação de Laplace linear.....	191
Tabela D.10: Valor da ordem ( $p$ ) da Eq. (5.1) para os algoritmos de engrossamento para a equação de Poisson.....	192
Tabela D.11: Valor do coeficiente ( $c$ ) da Eq. (5.1) para os algoritmos de engrossamento para a equação de Poisson.....	192
Tabela E.1: Malhas utilizadas para análise da influência do número de elementos ( $E$ ) no tempo de CPU.....	194

## LISTA DE ABREVIATURAS E SIGLAS

CDS	<i>Central Differencing Scheme</i> – Esquema de Diferença Central
CFD	<i>Computational Fluid Dynamics</i> – Dinâmica dos Fluidos Computacional
CPU	<i>Central Processing Unit</i> – Unidade Central de Processamento
CS	<i>Correction Scheme</i> – Esquema de Correção
CV	Coeficiente de Variação
EP	Engrossamento padrão
EP-SE	Engrossamento padrão seguido de semi-engrossamento
FW	<i>Full weighting</i> - Ponderação completa
SE	Semi-engrossamento
SEC	Semi-engrossamento completo
SE-EP	Semi-engrossamento seguido de engrossamento padrão
SG	<i>Singlegrid</i>
FAS	<i>Full Approximation Scheme</i> – Esquema de Aproximação Completa
FMG	<i>Full Multigrid</i>
GHW	<i>Geometric half weighting</i> - Meia ponderação geométrica
GFW	<i>Geometric full weighting</i> - Ponderação geométrica completa
GS-LEX	Gauss-Seidel lexicográfico
GS-RB	Gauss-Seidel <i>red-black</i>
HW	<i>Half weighting</i> - Meia ponderação
I	<i>Injection</i> – Injeção
MG	<i>Multigrid</i>
MSI	<i>Modified Strongly Implicit Method</i>
LL	Laplace linear
LS	Laplace senoidal
PW	<i>Partial weighting</i> – Ponderação parcial
Pó	Poisson
TDMA	<i>Tridiagonal Matrix Algorithm</i>

## LISTA DE SÍMBOLOS

$a_{ij}$	Coefficientes da matriz $A$
$A$	Matriz dos coeficientes
$C_x$	Comprimento do domínio 2D na direção $x$
$C_y$	Comprimento do domínio 2D na direção $y$
$C$	Coefficiente do ajuste de curvas
$D$	Desvio de um ponto em relação à média
$e$	Erro
$E$	Número de elementos
$E_x$	Número de elementos na malha na direção $x$
$E_y$	Número de elementos na malha na direção $y$
$Ed$	Erro de discretização
$E_\pi$	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 $x$
$ITE_{\max}$	Número máximo de iterações externas ou ciclos.
$J$	Contador na direção coordenada $y$
$N$	Número de nós da malha
$N_x$	Número de nós na direção $x$
$N_y$	Número de nós na direção $y$
$L$	Número de níveis
$L_{\max}$	Número máximo de níveis
$L_{\text{ótimo}}$	Número ótimo de níveis
$\bar{l}_1$	Norma $l_1$ média

$l_2$	Norma $l_2$
$l_\infty$	Norma infinito
$P$	Ordem do <i>solver</i>
$p_E$	Ordem efetiva
$p_U$	Ordem aparente
$Q$	Razão de refino
$Q$	Razão de aspecto
$R$	Resíduo
$R^2$	Domínio real bidimensional
$R$	Razão de engrossamento
$S$	Segundos
$S$	Desvio padrão amostral
$S_P$	<i>Speed-up</i>
$t_{CPU}$	Tempo de processamento
$\bar{t}_{CPU}$	Tempo médio de processamento
$T$	Temperatura
$\overline{TA}$	Temperatura analítica média
$\overline{TN}$	Temperatura numérica média
$TN$	Temperatura numérica
$U$	Solução exata do sistema
$\bar{u}$	Solução aproximada do sistema
$u_{xx}$	Derivada segunda de $u$ em relação à $x$
$u_{yy}$	Derivada segunda de $y$ em relação à $y$
$x,y$	Variáveis do sistema coordenado cartesiano 2D

### ***Letras gregas***

$\varepsilon$	Tolerância
$\Phi$	Solução analítica exata da variável de interesse
$\phi$	Solução numérica da variável de interesse
$\kappa$	Coefficiente de anisotropia física
$\nu$	Número de iterações internas

$\nu_{\text{ótimo}}$	Número ótimo de iterações internas
$\nu_{EP}$	Número de iterações internas para o bloco do engrossamento padrão
$\nu_{SE}$	Número de iterações internas para o bloco do semi-engrossamento
$\nu_1$	Número de pré-suavização
$\nu_2$	Número de pós-suavização
$\mu$	Número de ciclos internos
$\rho$	Constante para a solução da equação diferencial
$\Omega^h$	Malha fina
$\Omega^{2h}$	Malha grossa com engrossamento padrão

### ***Sub-índices***

1	Malha fina
2	Malha grossa
3	Malha super-grossa

### ***Símbolos***

$\approx$	Aproximadamente igual
$\ll$	Muito menor
$\gg$	Muito maior

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>23</b>
1.1	GENERALIDADES EM DINÂMICA DOS FLUIDOS COMPUTACIONAL .....	23
1.2	REVISÃO BIBLIOGRÁFICA .....	29
1.3	RELEVÂNCIA DO PROBLEMA .....	33
1.4	OBJETIVOS .....	34
1.5	DELINEAMENTO DESTE TEXTO .....	36
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>38</b>
2.1	MÉTODOS DE RESOLUÇÃO PARA SISTEMAS LINEARES .....	38
2.1.1	Gauss-Seidel lexicográfico .....	39
2.1.2	Gauss-Seidel <i>red-black</i> .....	40
2.2	EQUAÇÃO RESIDUAL .....	41
2.3	MÉTODO <i>MULTIGRID</i> .....	42
2.3.1	A filosofia do método <i>multigrid</i> .....	43
2.3.2	Restrição .....	46
2.3.3	Prolongação .....	48
2.3.4	Ciclos .....	49
2.3.5	Esquemas de correção .....	49
2.4	PROBLEMAS ANISOTRÓPICOS.....	52
2.4.1	Introdução .....	52
2.4.2	Tipos de anisotropia geométrica.....	55
2.4.3	Algoritmos de engrossamento .....	58
2.5	RESUMO DO CAPÍTULO 2 .....	59
<b>3</b>	<b>MODELOS MATEMÁTICOS E NUMÉRICOS.....</b>	<b>61</b>
3.1	MODELOS MATEMÁTICOS .....	61
3.2	MODELOS NUMÉRICOS .....	62
3.3	DADOS DE IMPLEMENTAÇÃO .....	65
<b>4</b>	<b>VERIFICAÇÃO NUMÉRICA DO CÓDIGO COMPUTACIONAL.....</b>	<b>67</b>
4.1	INTRODUÇÃO.....	67
4.2	ORDEM EFETIVA E ORDEM APARENTE .....	71
4.3	ERROS NUMÉRICOS.....	74
4.4	TESTES DE COERÊNCIA.....	76

4.4.1	Definição da tolerância e da maior malha a ser utilizada .....	76
4.4.2	Erro de discretização .....	78
4.4.3	Comparação entre os erros máximos .....	78
4.4.4	Análise da temperatura em função do número de incógnitas .....	80
4.5	COMPARAÇÃO COM A LITERATURA .....	80
4.6	PROBLEMAS ANISOTRÓPICOS .....	83
4.7	CONCLUSÃO DO CAPÍTULO .....	84
<b>5</b>	<b>ALGORITMOS DE ENGROSSAMENTO .....</b>	<b>87</b>
5.1	INTRODUÇÃO .....	87
5.2	RESULTADOS NUMÉRICOS .....	89
5.2.1	Estudo do número de iterações internas ( $\nu$ ) .....	90
5.2.2	Estudo do número de níveis ( $L$ ) .....	101
5.2.3	Comparação entre os algoritmos de engrossamento .....	105
5.2.4	Influência do número de elementos ( $E$ ) .....	107
5.2.5	Análise de complexidade para a equação de Laplace senoidal .....	109
5.3	CONCLUSÃO DO CAPÍTULO .....	112
5.4	PARÂMETROS COM MELHOR DESEMPENHO MÉDIO .....	114
<b>6</b>	<b>SEMI-ENGROSSAMENTO SEGUIDO DE ENGROSSAMENTO PADRÃO .....</b>	<b>115</b>
6.1	OPERADORES DE RESTRIÇÃO .....	115
6.1.1	Injeção (I) .....	116
6.1.2	Meia ponderação (HW) .....	116
6.1.3	Ponderação completa (FW) .....	117
6.1.4	Meia ponderação geométrica (GHW) .....	118
6.1.5	Ponderação geométrica completa (GFW) .....	119
6.1.6	Ponderação parcial (PW) .....	119
6.2	RESULTADOS NUMÉRICOS .....	120
6.2.1	Razão de aspecto ( $Q$ ) .....	120
6.2.2	Número de elementos ( $E$ ) .....	123
6.2.3	Análise de complexidade para o algoritmo SE-EP .....	126
6.2.4	Análise dos parâmetros ótimos para o algoritmo dois estágios .....	126
6.3	COMPARAÇÃO COM A LITERATURA .....	141
6.4	CONCLUSÃO DO CAPÍTULO .....	142
6.5	PARÂMETROS COM O MELHOR DESEMPENHO MÉDIO .....	143
<b>7</b>	<b>CONCLUSÃO .....</b>	<b>144</b>

7.1	CONCLUSÃO GERAL .....	144
7.2	CONTRIBUIÇÕES .....	145
7.3	PARÂMETROS COM MELHOR DESEMPENHO MÉDIO .....	146
7.4	EXTRAPOLAÇÕES .....	147
7.5	TRABALHOS FUTUROS .....	147
	<b>REFERÊNCIAS .....</b>	<b>149</b>
	<b>APÊNDICE A: SOLUÇÕES ANALÍTICAS .....</b>	<b>154</b>
	<b>APÊNDICE B: MEDIDAS DE DISPERSÃO.....</b>	<b>159</b>
	<b>APÊNDICE C: TESTES DE COERÊNCIA.....</b>	<b>161</b>
	<b>APÊNDICE D: ALGORITMOS DE ENGROSSAMENTO .....</b>	<b>168</b>
	D1. ESTUDO DO NÚMERO DE ITERAÇÕES INTERNAS ( $\nu$ ): .....	168
	D2. ESTUDO DA INFLUÊNCIA DO NÚMERO DE ELEMENTOS ( $E$ ) .....	188
	D3. ANÁLISE DE COMPLEXIDADE .....	191
	<b>APÊNDICE E: SEMI-ENGROSSAMENTO SEGUIDO DE ENGROSSAMENTO</b>	
	<b>PADRÃO .....</b>	<b>193</b>
	E1. MALHAS UTILIZADAS PARA A ANÁLISE DO NÚMERO DE ELEMENTOS ..	193
	E2. ESTUDO DOS <i>SOLVERS</i> E TIPO DE RESTRIÇÃO.....	193
	E3. ESTUDO DO NÚMERO DE ITERAÇÕES INTERNAS ( $\nu$ ) .....	195
	E4. COMPARAÇÃO ENTRE OS ALGORITMOS .....	197
	<b>APÊNDICE F: GERAÇÃO DE MALHAS .....</b>	<b>202</b>



# 1 INTRODUÇÃO

Neste capítulo introdutório é apresentado um estudo sobre as generalidades da dinâmica dos fluidos computacional. A seguir é apresentada uma revisão bibliográfica do método *multigrid*, a relevância do problema e os objetivos desta tese. Por último é apresentado um delineamento do texto.

## 1.1 GENERALIDADES EM DINÂMICA DOS FLUIDOS COMPUTACIONAL

É cada vez mais crescente o emprego de métodos numéricos para a solução de problemas em engenharia. Uma das áreas em que seu uso é muito difundido é a Dinâmica dos Fluidos Computacional (CFD). Segundo Fortuna (2000), a CFD é uma área da computação científica que estuda métodos computacionais para simulação de fenômenos que envolvem fluidos em movimento com ou sem troca de calor, onde o interesse principal é obter as distribuições de velocidade, pressão e temperatura na região de escoamento.

Três tipos de métodos podem ser empregados na solução de um problema de engenharia: experimentais, analíticos e/ou numéricos. Cada método possui vantagens e desvantagens dependendo das características do problema a ser resolvido. A Tab. 1.1, adaptada de Tannehill et al. (1997), faz uma comparação entre estes tipos de métodos. Os métodos analíticos e numéricos também podem ser denominados de métodos teóricos porque ambos trabalham com modelos matemáticos (MALISKA, 2004).

Para determinar a solução de um modelo matemático por meio de um método numérico encontram-se diversas fontes de erro, sendo classificado em (MARCHI, 2001): erros de truncamento, erros de iteração, erros de arredondamento e erros de programação. Estas fontes estão descritas a seguir:

- Erros de truncamento: resultam das aproximações numéricas feitas na discretização das derivadas envolvidas no modelo matemático. Quando o erro da solução numérica é gerado apenas por erros de truncamento da série de Taylor, ele é denominado de erro de discretização.

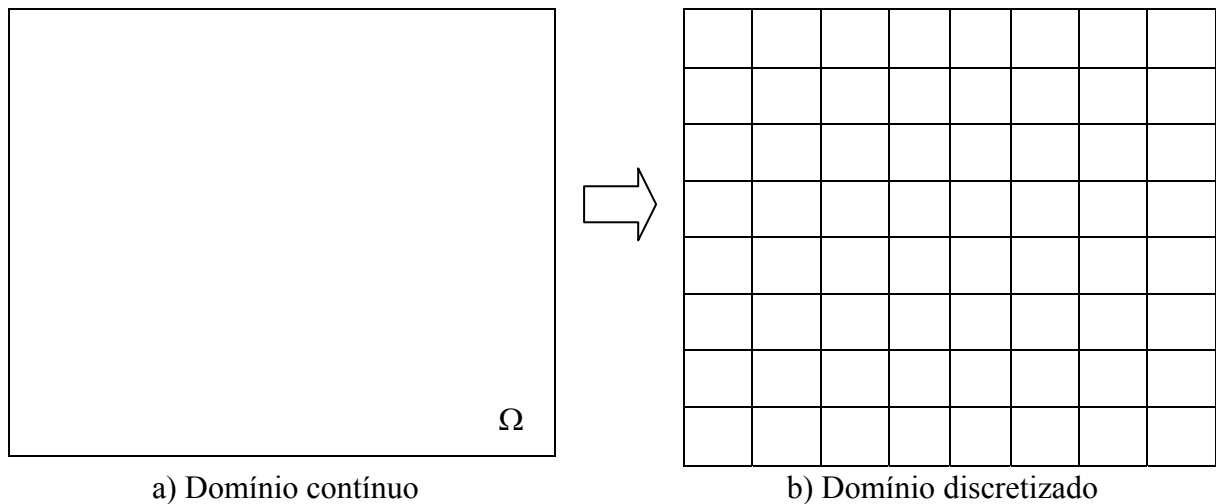
**Tabela 1.1: Comparação entre os métodos de solução de problemas em engenharia (adaptada de Tannehill et al. 1997).**

<b>Método</b>	<b>Vantagens</b>	<b>Desvantagens</b>
Experimental	- mais realístico.	- equipamento exigido; - erros experimentais; - dificuldades de medição; - custo operacional.
Analítico	- representação matemática do fenômeno real.	- restrita a geometria e problemas simples e lineares.
Numérico	- não há restrição à linearidade; - não há restrição a geometrias e processos complicados; - evolução temporal do processo.	- erros de truncamento; - erros de modelagem; - custo computacional.

- Erros de iteração: segundo Ferziger e Peric (2002) o erro de iteração é a diferença entre a solução exata das equações discretizadas e a solução numérica em uma determinada iteração. Esse erro pode ser originado por diferentes fatores (MARTINS, 2002), como: o emprego de métodos iterativos para solução de equações discretizadas, o uso de métodos segregados na obtenção de modelos constituídos por várias equações diferenciais, ou ainda pela existência de não-linearidades no modelo matemático. Em geral estes tipos de erros diminuem com o aumento do número de iterações. O erro de máquina ( $E_{\pi}$ ) é obtido ao executar o programa até que os erros de iteração sejam eliminados.
- Erros de arredondamento: são os erros que ocorrem devido à representação finita dos números reais nas computações. Estes erros dependem do compilador (*software*) usado para gerar o código computacional e do computador (*hardware*) empregado na sua execução.
- Erros de programação: são inerentes ao programador e à utilização do código implementado, incluindo basicamente (ROACHE, 1998): os erros resultantes do uso incorreto de um modelo numérico na aproximação de um modelo matemático, os erros gerados na implementação do modelo numérico em um programa computacional e os erros cometidos no uso do programa computacional durante a obtenção da solução numérica.

A idéia do método numérico é resolver as equações diferenciais, substituindo as derivadas nela existentes por expressões algébricas envolvendo a função incógnita. Ao

contrário do método analítico que permite calcular os valores das variáveis dependentes em um número infinito de pontos, a aproximação numérica fornece a solução em um número discreto de pontos (pontos nodais ou nós) definido pela malha computacional, (MESQUITA, 2000). Em geral, se o sistema numérico for consistente, quanto maior for o número de pontos, mais próxima estará a solução numérica da analítica. A Fig. 1.1 (a) representa um domínio contínuo. O domínio discretizado é representado pela malha da Fig. 1.1 (b).



**Figura 1.1: Discretização.**

O passo seguinte está relacionado com a maneira de se obter as equações algébricas, comumente denominado de discretização. Três classes merecem destaque, a saber:

- Elementos finitos (HUGHES, 2000; REDDYE e GARTLING, 1994);
- Diferenças finitas (FORTUNA, 2000; TANNEHILL et al., 1997);
- Volumes finitos (MALISKA, 2004; VERSTEEG e MALALASEKERA, 2007);
- Elementos de contorno (BREBBIA et al., 1984).

Na discretização pelo método das diferenças finitas (GOLUB e ORTEGA, 1992; TANNEHILL et al., 1997) em problemas bidimensionais, o domínio  $\{(x, y) \in \mathbb{R}^2 : 0 \leq x \leq C_x \text{ e } 0 \leq y \leq C_y\}$  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 (incluindo os contornos) e  $C_x$  e  $C_y$

determinam o comprimento do domínio de cálculo. Isto introduz uma malha uniforme com os pontos:

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

onde  $i = 1, \dots, N_x$ ,  $j = 1, \dots, N_y$  e  $h_x$  e  $h_y$  são os incrementos 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  $\Omega^h$ . Em problemas unidimensionais, dois nós consecutivos da malha determinam um subintervalo, neste trabalho denominado por “elemento” ( $E$ ). A distância entre estes dois nós é definida por  $h_x$ . Para problemas bidimensionais,  $h_x$  e  $h_y$  são os comprimentos de cada elemento nas direções  $x$  e  $y$ , respectivamente. O número total de elementos é dado por  $E$ . Embora este conceito não seja comumente empregado no método de diferenças finitas, ele é muito adequado neste trabalho. A razão entre os comprimentos  $h_x$  e  $h_y$  é denominada de razão de aspecto ( $Q$ ) da malha e é definida por:

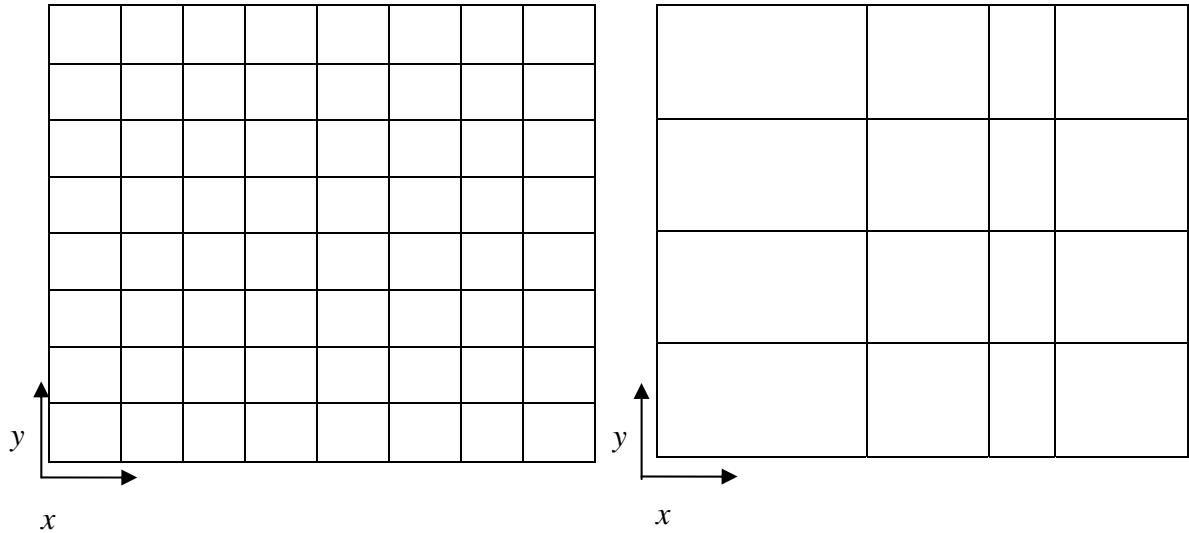
$$Q = \frac{h_x}{h_y} \quad (1.2)$$

Se  $Q = 1$ , a malha chama-se isotrópica, caso contrário, anisotrópica (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 2.6.

Dependendo da distribuição dos pontos discretos no domínio, a malha pode ser classificada em uniforme ou não-uniforme e estruturada ou não-estruturada conforme as Figs. 1.2 e 1.3, respectivamente.

Depois de particionado o domínio usa-se algum tipo de ordenação para identificar os nós na malha: lexicográfica, *red-black*, *backward* entre outras. As ordenações lexicográfica e *red-black* estão representadas nas Figs. 1.4 e 1.5, respectivamente. Os quadrados representam os pontos vermelhos e os círculos os pontos pretos. A ordenação é feita apenas nos pontos internos, pois para os modelos envolvidos nesta tese, a solução nos contornos é conhecida. Mais detalhes sobre os tipos de ordenação podem ser encontrados em Trottenberg et al.

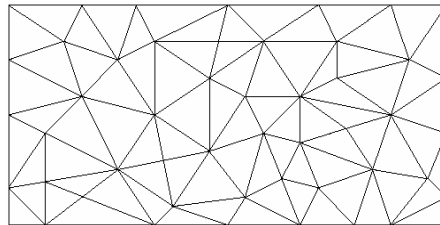
(2001) e Wesseling (1992). Em Wesseling (1992) a ordenação *red-black* é denominada de *white-black*.



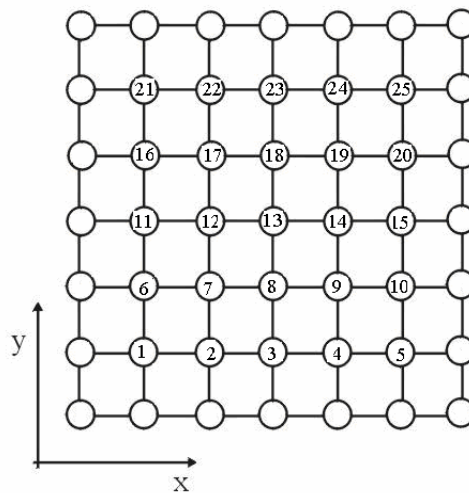
a) malha estruturada e uniforme.

b) malha estruturada e não-uniforme em  $x$ .

**Figura 1.2: Exemplos de malhas estruturadas: (a) uniforme e (b) não-uniforme.**



**Figura 1.3: Exemplo de malha não-estruturada.**



**Figura 1.4: Ordenação lexicográfica.**



O método *multigrid* (MG) refere-se à família de algoritmos iterativos utilizados para resolver com eficiência equações diferenciais parciais discretizadas (ROACHE, 1998). O objetivo do método *multigrid* é acelerar a convergência de um esquema iterativo (TANNEHILL et al., 1997). O método *multigrid* é um dos métodos iterativos mais eficientes e gerais conhecidos até hoje (HIRSCH, 1988; TANNEHILL et al., 1997; YAN et al., 2007).

O método *multigrid* foi desenvolvido inicialmente para equações elípticas como a equação de Poisson discretizada, mas tem sido aplicado com grande sucesso a uma variedade de problemas como equações de Euler e Navier-Stokes discretizadas (GHIA et al., 1982). Atualmente pode também ser aplicado em diversas áreas, tais como: engenharia do petróleo (reservatório ou simulação de oleoduto), estudos ambientais (localização de poluição e ciclones, previsão de tempo), entre outros.

Na próxima seção é feita uma revisão bibliográfica do método *multigrid*.

## 1.2 REVISÃO BIBLIOGRÁFICA

Os primeiros estudos na área de *multigrid* foram feitos por Fedorenko (1964) e Bakhvalov (1966). Ambos investigaram a convergência de problemas de valor de fronteira de segunda ordem. Fedorenko usou a equação de Poisson nos seus artigos enquanto Bakhvalov trabalhou com a equação de advecção-difusão iniciando o estudo da convergência para as equações elípticas.

Em meados da década de 70, Brandt (1977) foi o primeiro a reconhecer o potencial do *multigrid*, apontando as suas vantagens. Brandt é conhecido como o pai dos métodos *multigrid* e muitos consideram este artigo como sendo a origem do *multigrid* moderno (CRAIG, 1996).

Segundo Craig (1996) uma grande quantidade de artigos foi publicada na década de 80, e assim o *multigrid* tornou-se um método de solução padrão em muitas áreas de aplicação. Seu uso foi estendido para a solução de equações diferenciais em muitas áreas como, por exemplo, equações diferenciais parabólicas e hiperbólicas. Várias ferramentas teóricas tornaram-se comuns neste período, entre elas o *multigrid* algébrico (desenvolvido na Universidade do Colorado em Denver), (CRAIG, 1996).

Na década de 90 uma parte da pesquisa sobre o método *multigrid* foi motivada por profundas mudanças na área de matemática aplicada em geral, devido a três desenvolvimentos computacionais: disponibilidade de computadores mais rápidos, surgimento da computação paralela e a grande quantidade de informações através da internet.

Uma introdução aos métodos *multigrid*, com detalhes na análise de Fourier, o esquema CS (esquema de correção), FAS (esquema de aproximação completa), FMG (*full multigrid*), operadores, algoritmos, complexidade, propriedades variacionais, análise espectral, aplicações, etc, pode ser encontrada nos livros de Briggs et al. (2000), Trottenberg et al. (2001) e Wesseling (1992). Esses livros fazem uma introdução ao método *multigrid* e são considerados literatura básica. No capítulo 2 desta tese é feito um estudo de seus conceitos básicos.

O método *multigrid* pode ser aplicado na resolução de sistemas lineares e não-lineares, bem como em malhas estruturadas (*multigrid* geométrico), e em malhas não-estruturadas (*multigrid* algébrico). Wesseling e Oosterlee (2001) fazem uma revisão do desenvolvimento do método *multigrid* geométrico, com ênfase nas aplicações em problemas de dinâmica dos fluidos computacional descrevendo os estudos realizados em relação aos métodos empregados na indústria, bem como a eficiência do *multigrid* obtido nas aplicações acadêmicas. Stüben (2001) faz o mesmo para o *multigrid* algébrico.

No *multigrid* geométrico pode-se considerar diversos tipos de ciclos (ordem na qual as malhas são visitadas): ciclo V, W, dente-de-serra, entre outros. Aplicações destes ciclos podem ser encontradas em:

- Ciclo V: Briggs et al. (2000); Mesquita e De-Lemos (2004); Wesseling e Oosterlee, (2001) e Yan et al. (2007).
- Ciclo W: Chisholm (1997) e Manzano (1999);
- Ciclo dente-de-serra: Fletcher (1991); Gerolymos e Vallet (2005); Zeeuw (1996) e Zeeuw (2005).

O *multigrid* pode ter início na malha mais grossa, tendo em vista que a convergência é melhor nestas malhas. Esse processo é denominado *full multigrid*. Diversos trabalhos com o uso de *full multigrid* foram desenvolvidos. Entre eles pode-se citar: Chisholm (1997); Hortmann et al. (1990) e Manzano (1999). Yan e Thiele (1998) propõem um algoritmo *full multigrid* com esquema FAS modificado onde somente os resíduos são restritos para a malha mais grossa e as soluções iniciais na malha grossa são tomadas do ciclo precedente.



Uma outra forma de acelerar a convergência do método *multigrid* é a análise e otimização de parâmetros. Alguns parâmetros estudados são:

- Tipos de ciclos: Chisholm (1997), Mesquita e De-Lemos (2004);
- Iterações internas: Chisholm (1997), Mesquita (2000); Mesquita e De-Lemos (2004); Pinto e Marchi (2006) e Rabi e De-Lemos (1998),
- Operadores de restrição: Chisholm (1997);
- *Solvers*: Pinto e Marchi (2006);
- Número de níveis: Pinto e Marchi (2006).

A literatura dispõe de diversas estratégias para determinar o momento de se mudar a malha no método *multigrid*. Estas estratégias são denominadas de critério dinâmico e critério de ciclo (RABI, 1998). O critério dinâmico consiste em monitorar a taxa de convergência da solução numérica, a qual pode ser determinada pela razão das normas dos resíduos de duas iterações sucessivas, por exemplo. O critério de ciclo consiste em especificar o número de iterações internas em cada nível. O critério de ciclo pode ser encontrado em diversos trabalhos: Briggs et al. (2000); Hortmann et al. (1990); Mesquita e De-Lemos (2004); Trottenberg et al. (2001); Wesseling (1992); Yan et al. (2007) entre outros. Mais detalhes sobre o critério dinâmico pode ser encontrado no artigo de Brandt (1977).

O método *multigrid* também pode ser aplicado em problemas anisotrópicos, ou seja, com  $Q \neq 1$ . A anisotropia aparece com frequência na área de CFD, por exemplo, em problemas de camada limite e escoamentos turbulentos. Aplicar o método *multigrid* com engrossamento padrão (engrossamento em ambas as direções) faz com que as taxas de convergência se degenerem, surgindo a necessidade de métodos mais eficazes para este tipo de problema. Vários estudos têm dado ênfase para algoritmos de semi-engrossamento, algoritmos que combina pontos ou células na direção de forte acoplamento. Mulder (1989) e Naik e Van Rosendale (1993) utilizam o semi-engrossamento múltiplo no lugar do semi-engrossamento padrão, e concluíram que o último apresenta melhores resultados. O semi-engrossamento direcional combinado por relaxação por linhas foi desenvolvido por Dendy et al. (1989). Radespiel e Swanson (1995) apresentam uma variante do semi-engrossamento para as equações de Navier-Stokes. Oosterlee e Wesseling (1993) utilizaram semi-engrossamento múltiplo na equação de difusão anisotrópica e equação de convecção-difusão. Larsson et al.

(2005) resolveram problemas anisotrópicos com semi-engrossamento condicional, mantendo altas taxas de convergência com alta razão de aspecto.

Em Zhang (2002) são abordados problemas anisotrópicos acoplando-se a técnica de semi-engrossamento seguida de engrossamento padrão intitulado como “*partial semicoarsening*”. Foram feitas comparações para a equação de Poisson bidimensional com a técnica denominada de engrossamento padrão (*full coarsening*). Zhang (2002) comparou estas técnicas para as razões de aspecto  $Q = 1/2, 1/4, 1/8$  e  $1/16$  (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.

Pinto (2006) estudou problemas anisotrópicos envolvendo a equação de Laplace bidimensional e os algoritmos: engrossamento padrão, semi-engrossamento, semi-engrossamento seguido de engrossamento padrão e engrossamento padrão seguido de semi-engrossamento. Pinto (2006) comparou estas técnicas para as razões de aspecto  $Q = 1/1.024, 1, 2, 16, 128, 1024$  e  $8192$ . Pinto constatou que o algoritmo semi-engrossamento seguido de engrossamento padrão com suavizador Gauss-Seidel lexicográfico é eficiente para a anisotropia estudada.

Segundo Susie (2007), Hutchinson e Raithby (1986) desenvolveram um novo método *multigrid* baseado no método de correções por bloco de Settari e Aziz (1973). No método *multigrid* de correções aditivas as equações da malha grossa são obtidas sem o uso de operadores de restrição e interpolação fixos, ao invés disso é determinada uma correção constante para cada bloco de células da malha grossa forçando que o somatório dos resíduos seja zero após a correção ser aplicada. Este método apresenta muitas vantagens, podendo citar-se que ao gerar as equações das malhas grossas não necessita de considerações especiais para as condições de contorno, coeficientes anisotrópicos ou malhas irregulares.

Segundo Cordazzo (2006), estudos realizados por Elias et al. (1997), mostram a perda de eficiência nos métodos iterativos quando as equações discretizadas têm coeficientes anisotrópicos<sup>1</sup>. Isso acontece porque os métodos iterativos, além de reduzir com eficiência apenas os erros cujo comprimento de onda são equivalentes ao tamanho da malha, fazem-no preferencialmente na direção dos grandes coeficientes. Nesses casos a anisotropia dos coeficientes origina diferentes escalas de tempo de propagação da informação, afetando diretamente o comportamento da convergência. Os autores mostram que uma aglomeração

---

<sup>1</sup> Esses coeficientes surgem devido à razão de aspecto da malha diferente da unidade e/ou a propriedades físicas diferentes em cada direção.

dos volumes de controle da malha refinada que vise reduzir a anisotropia dos coeficientes das malhas grosseiras torna o método *multigrid* ainda mais eficiente. A razão é a diminuição da variação das escalas de tempo de propagação da informação nas malhas grosseiras.

Fischer e Huckle (2006) fazem uma análise de problemas no contexto de sistemas anisotrópicos. São geradas funções e suas curvas de nível, permitindo o desenvolvimento de métodos *multigrid* para sistemas onde a anisotropia ocorre em direções arbitrárias e não ao longo dos eixos coordenados. Fischer e Huckle (2008) dão continuidade a este estudo incluindo o uso de técnicas de suavização sofisticadas que podem ser combinadas com engrossamento padrão e semi-engrossamento. Gee et al. (2009) propõem um novo tipo de prolongação para problemas anisotrópicos com o uso do método *multigrid* algébrico.

Em CFD, geralmente são resolvidos problemas com perturbações singulares, fortes anisotropias, alta razão de aspecto, equações governantes que podem exibir um comportamento em uma parte do domínio e outro em outra parte do domínio. Quanto mais distante a razão de aspecto ( $Q$ ) está da unidade, ou seja,  $0 < Q \ll 1$  ou  $Q \gg 1$  (forte anisotropia), 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. Por exemplo, em problemas de camada limite, onde a razão de aspecto ( $Q$ ) pode ser da 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. Isto motiva o estudo das propriedades do método *multigrid* geométrico, tanto em sua fundamentação como em aplicações, para que se possam elaborar algoritmos mais eficientes para classes de problemas mais abrangentes.

### 1.3 RELEVÂNCIA DO PROBLEMA

A resolução de problemas de mecânica dos fluidos e transferência de calor através de métodos numéricos requer um custo computacional demasiadamente alto e muitas vezes inviável devido ao grande número de equações a serem resolvidas em cada passo iterativo. No início dos cálculos a taxa de convergência é grande, passando a decair sensivelmente à medida que o processo iterativo evolui. Existem muitos trabalhos de pesquisa que visam

melhorar a taxa de convergência dos métodos numéricos. Um método usado para melhorar a taxa de convergência é o *multigrid* (BRIGGS et al., 2000).

A eficiência do método *multigrid* não tem sido totalmente alcançada em aplicações realísticas da engenharia na área de CFD. (WESSELING e OOSTERLEE, 2001). Com a crescente complexidade das aplicações em CFD, é crescente também a demanda por métodos mais eficientes e robustos. Espera-se que esses métodos tenham uma boa redução do tempo computacional, usem pouca memória e possam abordar não-linearidades e acoplamentos sem grandes prejuízos em seu desempenho.

Segundo Trottenberg et al. (2001), experiências com métodos *multigrid* mostram que as escolhas de parâmetros (estrutura da malha grossa, o suavizador, o número de iterações internas em cada malha, ciclos e os esquemas de restrição e interpolação) podem ter uma forte influência na eficiência do algoritmo. Não há regras gerais na escolha destes parâmetros, porém certos valores podem ser recomendados para determinadas situações. A taxa de convergência depende das escolhas feitas. Uma simples modificação no algoritmo pode resultar em uma redução significativa no tempo computacional o que justifica a importância de estudar os diversos parâmetros do método *multigrid*.

A anisotropia geométrica ocorre com certa frequência em problemas de dinâmica dos fluidos computacional principalmente em problemas de camada limite. Nestes problemas, aplicar o método *multigrid* com engrossamento padrão não conduz a resultados satisfatórios (TROTTEMBERG et al., 2001). Uma alternativa é usar técnicas de semi-engrossamento. Vários estudos têm sido realizados em busca de técnicas de semi-engrossamento mais eficientes: Cordazzo (2006); Dendy et al. (1989); Fischer e Huckle (2008); Gee et al. (2009); Larsson et al. (2005); Mesquita e De-Lemos (2004); Mulder (1989); Naik e van Rosendale (1993); Pinto e Marchi (2006); Radespiel e Swanson (1995); Zeeuw (1996) e Zhang (2002).

## 1.4 OBJETIVOS

Neste trabalho são investigados os seguintes parâmetros do método *multigrid* geométrico: número de elementos, número de iterações internas (número de iterações do método numérico a fim de suavizar as componentes de erro), número de níveis (número de

malhas percorridas), métodos de resolução de sistemas de equações algébricas (aqui chamados de *solvers*), razões de aspecto e operadores de restrição.

Este trabalho tem por objetivo otimizar o método *multigrid* geométrico utilizado para a resolução de problemas difusivos bidimensionais, discretizados com malhas isotrópicas e anisotrópicas com alta razão de aspecto. Os objetivos gerais do trabalho são:

- Analisar os parâmetros do método *multigrid* em malhas isotrópicas e anisotrópicas;
- Minimizar o tempo de CPU para o método *multigrid* em malhas isotrópicas e anisotrópicas para as equações de Laplace e Poisson.

Os objetivos específicos são:

- Fazer uma análise do número de suavizações e do número de níveis no ciclo V *multigrid*;
- Verificar a influência da anisotropia no método *multigrid*;
- Implementar algumas técnicas de semi-engrossamento existentes na literatura;
- Propor uma nova técnica para resolver problemas anisotrópicos;
- Propor operadores de restrição para problemas anisotrópicos.
- Verificar o efeito dos parâmetros do método *multigrid* sobre o tempo de CPU.

No presente trabalho comparam-se cinco algoritmos de engrossamento para problemas anisotrópicos: engrossamento padrão (EP) proposto por Brandt (1977); semi-engrossamento (SE) proposto por Mulder (1989); semi-engrossamento completo (SEC) proposto neste trabalho; engrossamento padrão seguido de semi-engrossamento (EP-SE) proposto por Pinto e Marchi (2006); e semi-engrossamento seguido de engrossamento padrão (SE-EP) proposto por Zhang (2002). Também são realizadas comparações entre alguns tipos de restrições: injeção, meia ponderação e ponderação completa. São propostos três tipos de restrição para problemas anisotrópicos: meia ponderação geométrica, ponderação geométrica completa e ponderação parcial.

O algoritmo *multigrid* adotado é o esquema de correção (*Correction Scheme*) com ciclo V e prolongação por interpolação bilinear. Os modelos matemáticos considerados referem-se a três problemas bidimensionais lineares de condução de calor, governados pelas equações de Laplace e Poisson, com condições de contorno de Dirichlet. Neste trabalho são

estudadas equações lineares. Pretende-se, posteriormente fazer um estudo sobre as equações não-lineares.

## 1.5 DELINEAMENTO DESTE TEXTO

Esta tese apresenta sete capítulos, referências bibliográficas e apêndices constituídos da seguinte forma:

- No segundo capítulo encontra-se a fundamentação teórica. São descritos os métodos de resolução para sistemas lineares utilizados, apresenta as características principais do método *multigrid* e problemas anisotrópicos.
- No terceiro capítulo apresenta-se os modelos matemáticos e numéricos utilizados.
- No quarto capítulo é apresentada uma verificação do código computacional utilizado.
- No capítulo cinco é mostrada uma comparação entre os diversos algoritmos de engrossamento.
- No capítulo seis é feita uma análise detalhada do algoritmo de semi-engrossamento seguido de engrossamento padrão. São apresentados alguns operadores de restrição da literatura e alguns são propostos.
- No sétimo capítulo é apresentada a conclusão do trabalho com as contribuições para a literatura, parâmetros com melhor desempenho médio, extrapolações dos resultados e recomendações para trabalhos futuros.
- O apêndice A apresenta as soluções analíticas obtidas para as equações envolvidas no trabalho (Laplace senoidal, Laplace linear e Poisson).
- O apêndice B mostra algumas medidas de dispersão.
- O apêndice C contém alguns dados complementares referentes ao quarto capítulo para a verificação do código computacional: soluções analíticas, resultados numéricos para as variáveis de interesse, cálculo de  $h$ , ordem efetiva, ordem aparente e erro numérico para as equações envolvidas no trabalho.
- O apêndice D apresenta alguns dados complementares referentes ao capítulo cinco. Estudo do número de iterações internas ( $\nu$ ), estudo do número de elementos ( $E$ ) e análise de complexidade.

- O apêndice E contém alguns dados complementares referentes ao capítulo seis: as malhas utilizadas, estudo dos *solvers* e operadores de restrição, estudo do número de iterações internas ( $\nu$ ) e uma comparação entre os algoritmos.
- O apêndice F apresenta a forma pela qual as malhas foram geradas.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentada uma fundamentação teórica para o desenvolvimento do método *multigrid*. São descritos os métodos para a resolução de sistemas lineares: Gauss-Seidel lexicográfico e Gauss-Seidel *red-black*. São apresentadas as definições de equação residual e os aspectos principais do método *multigrid*.

### 2.1 MÉTODOS DE RESOLUÇÃO PARA SISTEMAS LINEARES

Os métodos numéricos para a resolução de sistemas lineares podem ser divididos em dois grupos: métodos diretos e métodos iterativos.

Os métodos diretos fornecem, através de um processo finito de passos previamente conhecidos, a solução exata do problema, se esta existir, com exceção dos erros de arredondamento (CUNHA, 2003). Entre os métodos diretos mais utilizados podem-se citar: Eliminação de Gauss, TDMA (*Tridiagonal Matrix Algorithm*), fatoração LU e fatoração Cholesky. Estes métodos apresentam algumas desvantagens em relação aos métodos iterativos, tais como:

- Provocam o preenchimento de uma matriz esparsa, isto é, poderão surgir elementos não-nulos em posições originariamente nulas;
- São sensíveis a erros de arredondamento, principalmente quando a matriz for mal-condicionada;
- Tempo de processamento elevado, principalmente em sistemas de grande porte.

Para que um método seja consistente, é necessário que o erro de truncamento torne-se desprezível para malhas numéricas bastante refinadas, ou seja, a equação discretizada deve tender à equação original quando o espaçamento da malha tender a zero, (MALISKA, 2004) dando origem a sistemas de grande porte onde os métodos iterativos são mais adequados.

Os métodos iterativos geram uma seqüência de vetores a partir de uma aproximação inicial. Sob certas condições, esta seqüência converge para a solução exata, caso ela exista.



Podem resolver sistemas de grande porte e não provocam o preenchimento de matrizes esparsas. Alguns dos métodos iterativos são: Jacobi, Gauss-Seidel, Fatoração LU incompleta, MSI (*Modified Strongly Implicit Method*). Detalhes sobre métodos de resolução de sistemas lineares (métodos diretos e métodos iterativos) podem ser encontrados em Burden e Faires (2008) e Cunha (2003). Neste trabalho optou-se pelo método Gauss-Seidel como *solver* padrão. O *solver* Gauss-Seidel tem sido bastante utilizado em trabalhos envolvendo o método *multigrid* entre eles: Trottenberg et al., (2001); Zhang (2002); Fischer e Huckle (2008) entre outros.

### 2.1.1 Gauss-Seidel lexicográfico

Matrizes pentadiagonais ocorrem com frequência em dinâmica dos fluidos computacional, principalmente em problemas bidimensionais onde são utilizadas aproximações de segunda ordem para as derivadas com o uso do método das diferenças finitas.

Com a equação diferencial governante para o caso bidimensional forma-se uma equação algébrica para cada nó. Um exemplo para esquemas de 5 pontos está apresentado na Eq. (2.1).

$$a_{i,j} u_{i,j} + a_{i-1,j} u_{i-1,j} + a_{i+1,j} u_{i+1,j} + a_{i,j-1} u_{i,j-1} + a_{i,j+1} u_{i,j+1} = f_{i,j} \quad (2.1)$$

onde

$$A_{i,j}^p = a_{i,j}; A_{i,j}^n = a_{i,j+1}; A_{i,j}^e = a_{i+1,j}; A_{i,j}^s = a_{i,j-1}; A_{i-1,j}^w = a_{i+1,j} \quad (2.2)$$

$$u_{i,j} = u_P; u_{i+1,j} = u_E; u_{i-1,j} = u_W; u_{i,j-1} = u_S; u_{i,j+1} = u_N \quad (2.3)$$

O conjunto de equações na forma da Eq. (2.1) escrita para cada par  $(i, j)$  do domínio, conduz a uma equação matricial da forma da Eq. (1.3). O método de Gauss-Seidel (BURDEN e FAIRES, 2008) 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 Eq. (2.1) na forma:

$$u_p^k = -(A_w u_w^k + A_e u_e^k + A_s u_s^k + A_n u_n^k + f_p) / A_p \quad (2.4)$$

onde o super-índice  $k$  representa a  $k$ -ésima iteração e o sub-índice a posição do nó na malha computacional. Para o método Gauss-Seidel com ordenamento lexicográfico, ou seja, de oeste ( $W$ ) para leste ( $E$ ) e do sul ( $S$ ) para o norte ( $N$ ), pode-se considerar como conhecidas, na mesma iteração, as variáveis  $u_w$  e  $u_s$ . Para as variáveis  $u_e$  e  $u_n$  são utilizados os valores obtidos na iteração anterior. O *solver* Gauss-Seidel lexicográfico utiliza a ordenação lexicográfica apresentada por Wesseling (1992). O algoritmo 2.1 apresentado a seguir descreve um procedimento para o *solver* Gauss-Seidel lexicográfico. Mais detalhes deste algoritmo podem ser encontrados em Burden e Faires (2008) e Versteeg e Malalasekera (2007).

**Algoritmo 2.1: Procedimento de Gauss-Seidel lexicográfico para  $Au = f$ .**

---

GAUSS-SEIDEL LEXICOGRÁFICO ( $k = 0, k_{\max}$  e  $u_0$ )

1. Calcular  $u_p^{k+1}$  pela Eq. (2.2) para todo  $P$ .
2. Faça  $k$  receber  $k+1$ .
3. Volte ao passo 1 até convergir ou até atingir  $k_{\max}$ .

Fim

---

### 2.1.2 Gauss-Seidel *red-black*

Verifica-se na Eq. (2.4) que para atualizar cada ponto são necessários os quatro vizinhos mais próximos. Se reorganizarmos a malha utilizando a ordenação *red-black* definida na Fig. 1.5, a relaxação pode ser realizada simultaneamente nos pontos definidos como vermelhos e pretos. Esta idéia beneficia a computação paralela, mas também pode apresentar bons resultados em computadores seriais (Zhang, 1996). O *solver* Gauss-Seidel *red-black* lexicográfico utiliza a ordenação *red-black* apresentada por Wesseling (1992). O

algoritmo 2.2 apresentado a seguir descreve um procedimento para o *solver* Gauss-Seidel *red-black*. Mais detalhes deste algoritmo podem ser encontrados em Parter (1988) e Zhang (1996).

**Algoritmo 2.2: Procedimento de Gauss-Seidel *red-black* para  $Au = f$ .**

---

GAUSS-SEIDEL RED-BLACK ( $k = 0, k_{\max}$  e  $u_0$ )

1. Calcular  $u_p^{k+1}$  pela Eq. (2.4) para todos os pontos pretos (círculos da Fig. 1.5).
2. Calcular  $u_p^{k+1}$  pela Eq. (2.4) para todos os pontos vermelhos (quadrados da Fig. 1.5).
3. Faça  $k$  receber  $k+1$ .
4. Volte ao passo 1 até convergir ou até atingir  $k_{\max}$ .

Fim

---

## 2.2 EQUAÇÃO RESIDUAL

Considere o sistema representado pela Eq. (1.3). Supondo que o sistema tenha solução única  $u$  e seja  $\bar{u}$  sua aproximação. O erro algébrico é dado por:

$$e = u - \bar{u} \quad (2.5)$$

A magnitude do vetor do erro pode ser medida através de normas de vetores (BRIGGS et al., 2000 e BURDEN e FAIRES, 2008). As normas mais utilizadas são a norma infinito ( $l_\infty$ ) e a norma Euclidiana ou norma-2, definidas pelas Eqs. (2.6) e (2.7), respectivamente.

$$\|e\|_\infty = \max_{1 \leq j \leq N} |e_j| \quad (2.6)$$

$$\|e\|_2 = \left\{ \sum_{j=1}^N e_j^2 \right\}^{1/2} \quad (2.7)$$

onde  $e_j$  é a  $j$ -ésima componente do vetor erro.

O erro, assim como a solução exata, são geralmente desconhecidos. Uma forma de verificar o quão próximo  $\bar{u}$  está de  $u$  é através do resíduo, dado por:

$$R = f - A\bar{u} \quad (2.8)$$

Nota-se que  $e \approx 0$  implica que  $R \approx 0$ , porém  $R \approx 0$  não implica que  $e \approx 0$ . (BRIGGS et al., 2000). Para matrizes bem condicionadas, se  $R \approx 0$  implica em  $e \approx 0$ .

O resíduo é uma medida de quanto a aproximação  $\bar{u}$  falha ao satisfazer o problema original. Sua norma pode ser medida pelas mesmas normas utilizadas para o erro. Lembrando que  $Au = f$  e utilizando as definições de  $R$  e  $e$ , pode-se deduzir uma importante relação entre o erro e o resíduo denominada de equação residual:

$$Ae = R \quad (2.9)$$

Esta equação residual nos informa que o erro satisfaz o mesmo conjunto de equações com as variáveis  $u$  quando  $f$  é substituída pelo resíduo  $R$ . A equação residual tem uma grande vantagem. Supondo que uma aproximação  $\bar{u}$  seja determinada através de um método iterativo, calcula-se o resíduo através da Eq. (2.8). Para melhorar a aproximação  $\bar{u}$ , resolve-se a equação residual para  $e$  e então calcula-se uma nova aproximação usando a definição do erro:

$$u = \bar{u} + e \quad (2.10)$$

A equação residual é de grande importância no contexto do método *multigrid*.

### 2.3 MÉTODO *MULTIGRID*

Nesta seção pretende-se explorar o método *multigrid* e seus princípios fundamentais, os operadores de transferência entre malhas, algoritmos e ciclos.

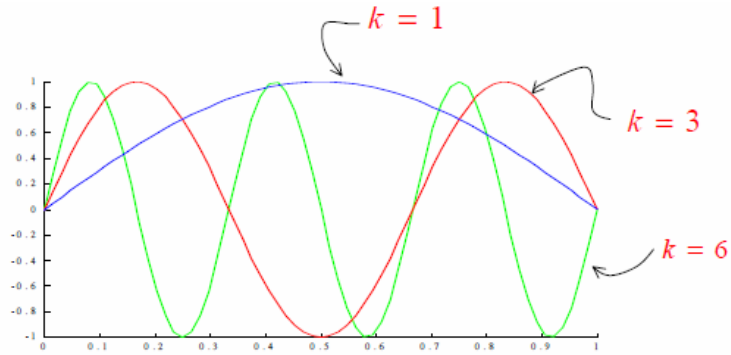
### 2.3.1 A filosofia do método *multigrid*

Uma técnica eficiente, usada para aliviar as fortes oscilações do resíduo em cada malha é suavizá-los por um método de sub-relaxação (método iterativo). Um dos métodos iterativos utilizados na literatura é o método *multigrid*. Este método tem como base a observação das propriedades dos métodos iterativos. A taxa de convergência depende dos autovalores da matriz de iteração associada ao método. O autovalor de maior magnitude é denominado de raio espectral da matriz e determina quão rápida a solução é alcançada. O autovetor associado a esse autovalor determina a distribuição do erro de iteração, variando consideravelmente de método para método (FERZIGER e PERIC, 2002).

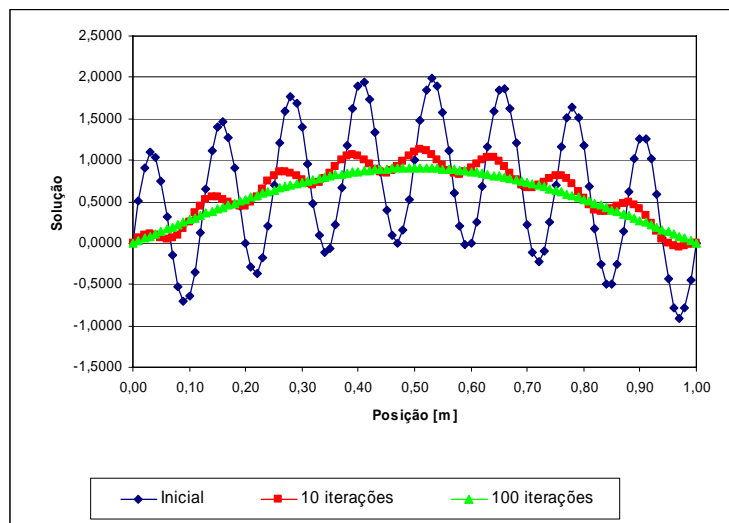
A maior parte dos métodos iterativos padrão (como por exemplo, o método de Gauss-Seidel) apresenta propriedades de suavização de erros locais de alta frequência (componentes oscilatórias do erro), enquanto as baixas frequências são mantidas praticamente inalteradas. Deste modo as primeiras iterações deste processo, geralmente, têm rápida convergência, caracterizando a presença de modos oscilatórios de erro. Porém, após algumas iterações o processo torna-se lento, sinalizando a predominância de modos suaves (BRANDT, 1977, BRIGGS, 2000).

Para ilustrar este procedimento, é apresentada a figura dos modos de Fourier extraída de Briggs et al. (2000). Os modos de Fourier são dados por  $v_j = \text{sen}(jk\pi/n)$  com  $0 \leq j \leq n$  e  $k \leq n-1$ , onde  $j$  é a componente do vetor  $v$ ,  $n$  é o número de elementos e  $k$  é o número de ondas ou modos de Fourier. Na Fig. 2.1 podem ser observados os modos de Fourier para  $k=1$ ,  $k=3$  e  $k=6$ . Os valores para  $k$  denotam quantos “meio-senos” constituem a curva no domínio do problema. A notação indica um vetor que contém  $k$  ondas.

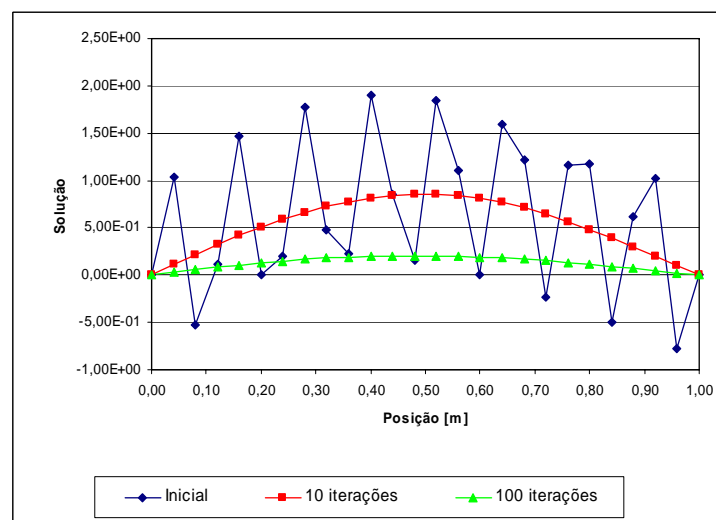
Nota-se na Fig. 2.1 que pequenos valores de  $k$  correspondem a ondas longas e suaves, enquanto valores maiores de  $k$  referem-se a ondas mais oscilatórias. De acordo com Briggs et al. (2000) e Trottenberg et al. (2001), os métodos iterativos, como Jacobi e Gauss-Seidel, possuem propriedades de suavização, ou seja, são capazes de reduzir rapidamente as componentes de erros oscilatórios (ou de alta frequência).



**Figura 2.1: Modos de Fourier.**



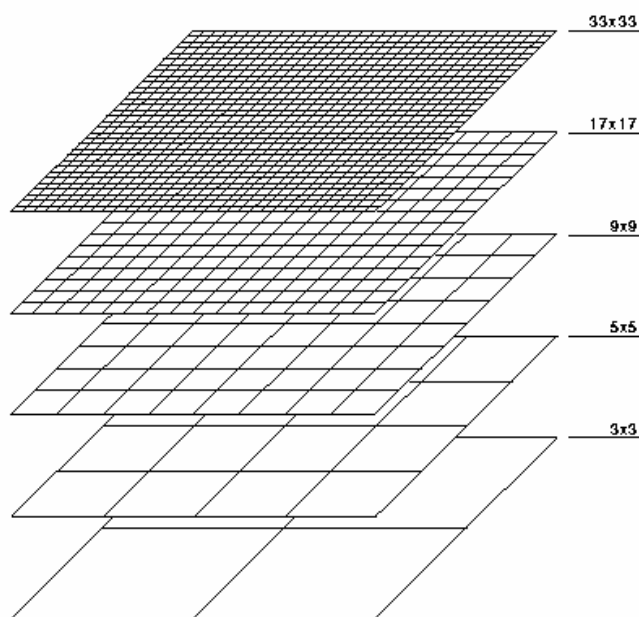
**Figura 2.2: Comportamento da suavização do erro em métodos iterativos. Malha de 100 elementos (101 nós), (adaptada de Araki (2007)).**



**Figura 2.3: Comportamento da suavização do erro em métodos iterativos. Malha de 25 elementos (26 nós), (adaptada de Araki (2007)).**

A idéia do método *multigrid* é cobrir um espectro maior de comprimentos de onda através da iteração não apenas em uma única malha, mas em uma seqüência de malhas cada vez mais grossas, ou seja, uma malha com menor quantidade de pontos que a malha original. Nesta malha mais grossa o processo de relaxação é repetido até que as componentes de erro tornem-se suaves. Quando isto ocorrer o problema é novamente transferido para uma malha mais grossa e assim sucessivamente até chegar à malha mais grossa estabelecida inicialmente ou na mais grossa possível.

A Fig. 2.4 ilustra uma seqüência de malhas que serão visitadas no processo de engrossamento. Neste exemplo, considera-se uma malha de  $33 \times 33$  nós (malha mais fina). O processo de engrossamento foi realizado até atingir a malha  $3 \times 3$  (a malha mais grossa possível). A quantidade de malhas utilizadas recebe o nome de número de níveis e será representado por  $L$ . No problema definido na Fig. 2.4 são considerados 5 níveis.



**Figura 2.4: Processo de engrossamento e geração de malhas.**  
Fonte: adaptada de <http://www.mgnet.org/mgnet/tutorials/xwb/mg.html>.

A taxa de convergência ideal (teórica) do método *multigrid* é independente do tamanho da malha, isto é, independe do número de pontos da malha (FERZIGER e PERIC; 2002; ROACHE 1998). Para obter um bom desempenho do *multigrid*, diversos níveis de malha devem ser usados (TANNEHILL et al., 1997). Pinto e Marchi (2007) e Santiago e Marchi (2008) recomendam usar todos os níveis.

Com o problema representado em uma malha mais fina é necessário transferir as informações da malha fina para a malha imediatamente mais grossa (operador de restrição) e da malha grossa para a malha imediatamente mais fina (operador de prolongação). Estes operadores são denominados de operadores de transferência entre malhas.

A transferência das informações para uma malha mais grossa é realizada através de operadores de transferência de malhas, os quais serão definidos na subseção 2.3.2. A razão de engrossamento, para o caso bidimensional, considerando malhas uniformes (os elementos possuem o mesmo tamanho  $h$ ), é definida como  $r = H/h$ , onde  $h$  representa a dimensão dos elementos da malha fina  $\Omega^h$ ,  $H$  o tamanho do elemento da malha imediatamente mais grossa  $\Omega^H$ .

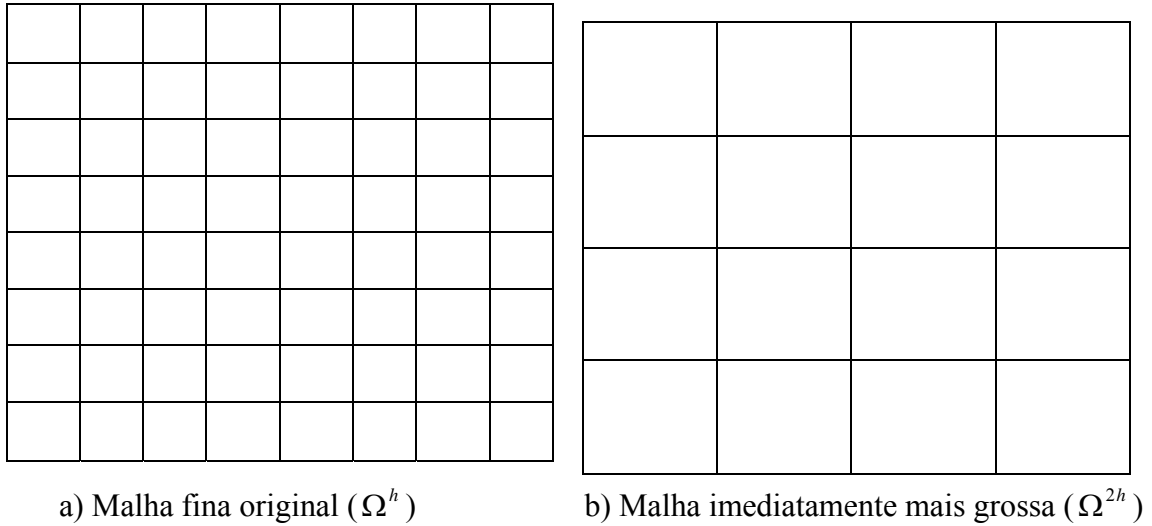
Diversas razões de engrossamento podem ser utilizadas ( $r = 2$ ,  $r = 3$ ,  $r = 4$  entre outras). Brandt (1977) constatou que a razão  $r = 2$  (caso onde a malha grossa tem o dobro do espaçamento da malha fina, ou seja,  $H = 2h$ ) é a recomendável, por estar mais próxima da razão de engrossamento 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. A Fig. 2.5 (a) representa uma malha fina original com  $N_x = N_y = 9$ , onde  $N_x$  e  $N_y$  representam indicam o número de nós nas direções  $x$  e  $y$ , respectivamente. Com o uso de  $r = 2$ , em ambas as direções, determina-se a malha imediatamente mais grossa com  $N_x = N_y = 5$  representada na Fig. 2.5 (b). A malha mais fina é denotada por  $\Omega^h$ , significando que cada elemento da malha tem comprimento  $h$ . Para o caso particular com razão de engrossamento  $r = 2$ , tem-se que  $H = 2h$ , logo a malha imediatamente mais grossa pode ser representada por  $\Omega^{2h}$ . O engrossamento pode ainda ser realizado em uma única direção (semi-engrossamento) ou nas duas direções simultaneamente (engrossamento padrão).

### 2.3.2 Restrição

Os operadores que transferem informações da malha fina  $\Omega^h$  para a malha grossa  $\Omega^{2h}$  são denominados de operadores de restrição e são representados genericamente por  $I_h^{2h}$  e definidos por  $\bar{u}^{2h} = [I_h]^{2h} \bar{u}^h$ .



Entre os operadores de restrição conhecidos, um dos mais utilizados é o operador de restrição por injeção (BRIGGS et al., 2000; TROTTEBERG et al., 2001; WESSELING, 1992).

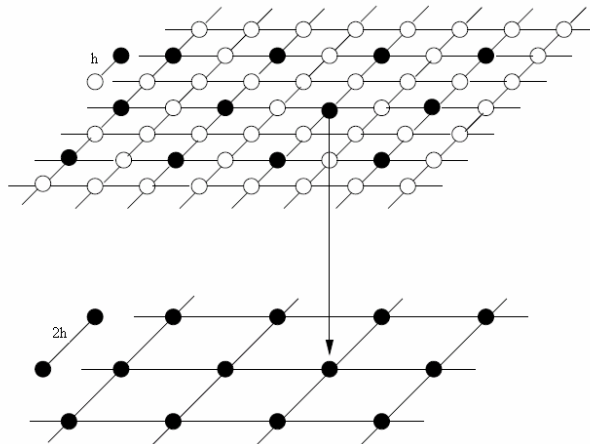


**Figura 2.5: Engrossamento padrão com razão de engrossamento  $r = 2$ .**

$$\bar{u}_{i,j}^{2h} = \bar{u}_{2i,2j}^h, \quad 1 \leq i \leq \frac{N_x}{2} - 1, \quad 1 \leq j \leq \frac{N_y}{2} - 1 \quad (2.11)$$

onde  $N_x$  e  $N_y$  são os números dos nós da malha fina nas direções coordenadas  $x$  e  $y$ , respectivamente e os sub-índices estão no intervalo  $1 \leq i \leq N_x$  e  $1 \leq j \leq N_y$ .

A Fig. 2.6 extraída de Trottenberg et al. (2001), mostra o uso do operador de restrição por injeção para o caso bidimensional.



**Figura 2.6: Restrição por injeção (Trottenberg et al. 2001).**

Outros operadores de restrição utilizados são meia ponderação e ponderação completa que serão descritos na seção 5.2. Mais detalhes sobre os operadores de restrição podem ser encontrados em Trottenberg et al. (2001).

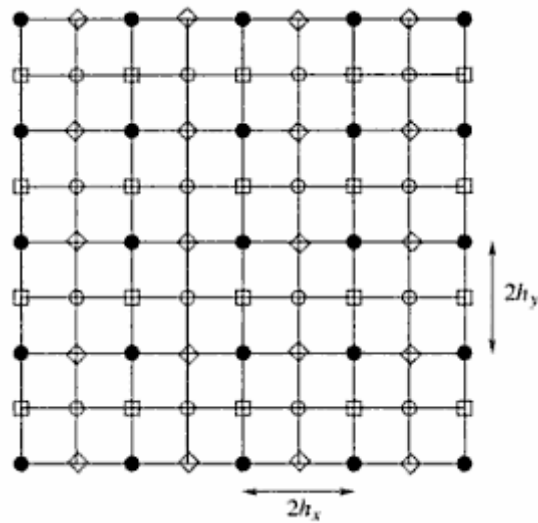
### 2.3.3 Prolongação

Os operadores que transferem informações da malha grossa  $\Omega^{2h}$  para a malha fina  $\Omega^h$  são chamados de operadores de prolongação, também conhecidos como interpolação e muito utilizados na Álgebra Linear. São representados genericamente por  $I_{2h}^h$  e definidos por  $\bar{u}^h = [I]_{2h}^h \bar{u}^{2h}$ . Uma forma de interpolação utilizada na literatura para problemas bidimensionais é a interpolação bilinear (BRIGGS et al., 2000; FERZIGER e PERIC, 2002; TANNEHILL et al., 1997; TROTTEBERG et al., 2001; WESSELING, 1992) dada por:

$$\begin{cases} \bar{u}_{2i,2j}^h = \bar{u}_{i,j}^{2h} & \text{para } \bullet \\ \bar{u}_{2i+1,2j}^h = \frac{1}{2}(\bar{u}_{i,j}^{2h} + \bar{u}_{i+1,j}^{2h}) & \text{para } \square \\ \bar{u}_{2i,2j+1}^h = \frac{1}{2}(\bar{u}_{i,j}^{2h} + \bar{u}_{i,j+1}^{2h}) & \text{para } \diamond \\ \bar{u}_{2i+1,2j+1}^h = \frac{1}{4}(\bar{u}_{i,j}^{2h} + \bar{u}_{i+1,j}^{2h} + \bar{u}_{i,j+1}^{2h} + \bar{u}_{i+1,j+1}^{2h}) & \text{para } \circ \end{cases} \quad (2.12)$$

onde  $i$  e  $j$  são índices da malha grossa com  $1 \leq i \leq N_x/2 - 1$ ,  $1 \leq j \leq N_y/2 - 1$ . A Fig. 2.7 extraída de Trottenberg et al. (2001) apresenta a malha fina com os símbolos referenciados na Eq. 2.12.

Outros operadores de prolongação podem ser vistos em Wesseling (1992), como o operador de interpolação quadrática ou o operador de prolongação de De Zeeuw (1990). Neste trabalho foi utilizado-se prolongação por interpolação bilinear.



**Figura 2.7: Prolongação bilinear (Extraída de Trottenberg et al. 2001).**

### 2.3.4 Ciclos

A ordem na qual as malhas são visitadas é chamada de ciclo *multigrid*. Existe uma generalização do ciclo V conhecida como ciclo  $\mu$  (WESSELING, 1992). Desta generalização pode-se obter outros tipos de ciclos. Devido aos seus formatos, são chamados de ciclo W, ciclo dente-de-serra e ciclo F. A Fig. 2.8 apresenta alguns exemplos.

Nesta figura o símbolo  $\bullet$  representa as suavizações realizadas em cada nível de malha e os traços que os unem representam os operadores de transferência entre malhas (operador de restrição e de prolongação).

### 2.3.5 Esquemas de correção

Dois tipos de esquemas podem ser usados com o método *multigrid* (BRIGGS et al., 2000): esquema de correção (CS) e o esquema de aproximação completa (FAS). O esquema CS transfere o resíduo para a malha imediatamente mais grossa e a Eq. (1.3) é resolvida somente na malha mais fina. Nas malhas mais grossas resolve-se a equação do resíduo, Eq. (2.7).

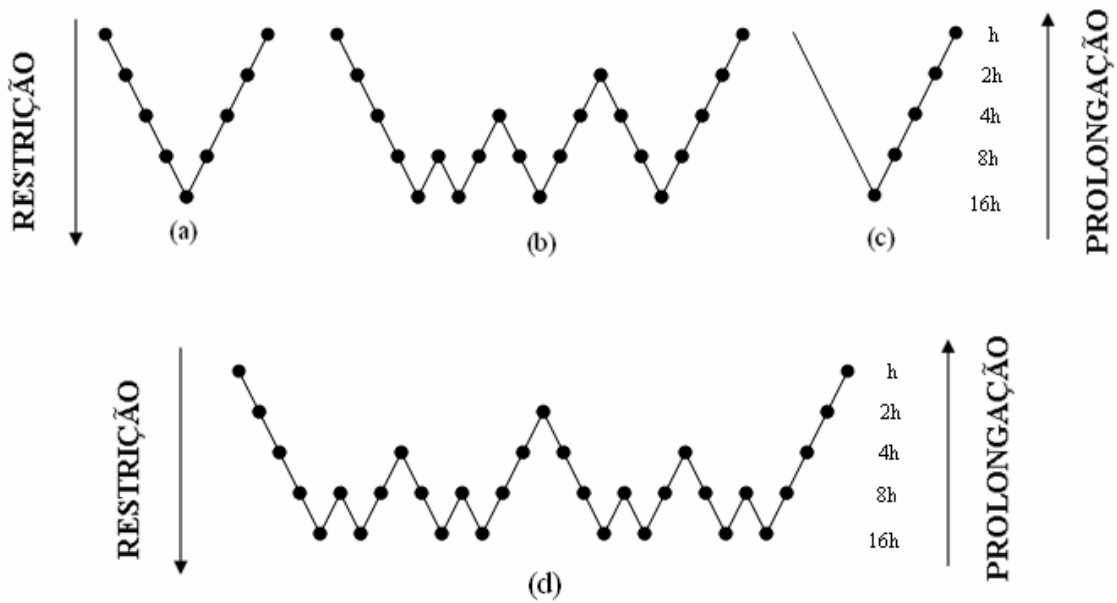


Figura 2.8: Diagramas: (a) ciclo V, (b) F, (c) dente-de-serra, (d) W.

O procedimento do esquema CS para várias malhas está descrito no algoritmo 2.3. Na prática,  $\nu_1$  e  $\nu_2$  representam o número de suavizações realizadas no processo de restrição e prolongação. Eles são chamados de pré e pós-suavização, respectivamente. Os super-escritos  $h$ ,  $2h$  e  $4h$  indicam a malha onde se definem os vetores ou matrizes. Os operadores  $A^{2h}$ ,  $A^{4h}$ ,  $A^{8h}$ , ... podem ser tomados por discretização (BRIGGS et al., 2000).

O esquema FAS transfere o resíduo e a solução, e a Eq. (1.3) é resolvida em todas as malhas. O esquema de aproximação completa (FAS) recebe este nome pelo fato de que o problema na malha grossa é resolvido para a solução e o resíduo e não apenas para o resíduo. A característica básica do FAS é que ele restringe o resíduo e a solução para a malha grossa. Isto nos permite a manipulação precisa de funções em malhas grossas. Mais detalhes sobre o esquema FAS podem ser encontrados em Briggs et al. (2000).

O algoritmo 2.3 descreve o algoritmo para um único ciclo V. O procedimento a seguir descrito no algoritmo 2.4 desenvolve diversas chamadas de CSMG percorrendo vários ciclos V até se atingir um critério de parada ou alcançar o número máximo de ciclos escolhidos  $ITE_{\max}$ .

**Algoritmo 2.3: Ciclo V com esquema CS para várias malhas.**  
(adaptado de Briggs et al. 2000).

---

CSMG( $\bar{u}_0, \bar{u}, f, h, \nu_1, \nu_2$ )

Início

Esquema de Correção (CS)

- 1) Iterar  $(A^h \bar{u}^h = f^h)$   $\nu_1$  vezes em  $\Omega^h$  com estimativa inicial  $\bar{u}_0^h$ ;
- 2) Calcular o resíduo:  $R^h = f^h - A^h \bar{u}^h$ ;
- 3) Restringir o resíduo da malha  $\Omega^h$  para a malha  $\Omega^{2h}$ :  $f^{2h} = I_h^{2h} R^h$ ;
  - a. Iterar  $(A^{2h} \bar{u}^{2h} = f^{2h})$   $\nu_1$  vezes em  $\Omega^{2h}$  com estimativa inicial  $\bar{u}_0^{2h} = 0$ ;
  - b. Calcular o resíduo:  $R^{2h} = f^{2h} - A^{2h} \bar{u}^{2h}$ ;
  - c. Restringir o resíduo da malha  $\Omega^{2h}$  para a malha  $\Omega^{4h}$ :  $f^{4h} = I_{2h}^{4h} R^{2h}$ ;
    - i. Iterar  $(A^{4h} \bar{u}^{4h} = f^{4h})$   $\nu_1$  vezes em  $\Omega^{4h}$  com estimativa inicial  $\bar{u}_0^{4h} = 0$ ;
    - ii. Calcular o resíduo:  $R^{4h} = f^{4h} - A^{4h} \bar{u}^{4h}$ ;
    - iii. Restringir o resíduo da malha  $\Omega^{4h}$  para a malha  $\Omega^{8h}$ :  $f^{8h} = I_{4h}^{8h} R^{4h}$ ;
    - iv. ...
    - v. ...
- iv. Corrigir  $\bar{u}^{4h} \leftarrow \bar{u}^{4h} + I_{8h}^{4h} \bar{u}^{8h}$ ;
- v. Iterar  $(A^{4h} \bar{u}^{4h} = f^{4h})$   $\nu_2$  vezes em  $\Omega^{4h}$  com estimativa inicial  $\bar{u}^{4h}$ ;
- d. Corrigir  $\bar{u}^{2h} \leftarrow \bar{u}^{2h} + I_{4h}^{2h} \bar{u}^{4h}$ ;
- e. Iterar  $(A^{2h} \bar{u}^{2h} = f^{2h})$   $\nu_2$  vezes em  $\Omega^{2h}$  com estimativa inicial  $\bar{u}^{2h}$ ;
- 4) Corrigir  $\bar{u}^h \leftarrow \bar{u}^h + I_{2h}^h \bar{u}^{2h}$ ;
- 5) Iterar  $(A^h \bar{u}^h = f^h)$   $\nu_2$  vezes com estimativa inicial  $\bar{u}^h$ .

Fim de CSMG

---

---

**Algoritmo 2.4: Multigrid com ciclo V, Esquema CS para diversas malhas e diversos ciclos.**

---

Escolher  $\bar{u}_0^h$  e  $ITE_{\max}$

1.  $i = 1$

Enquanto (não atingir a convergência) ou ( $i \leq ITE_{\max}$ ), faça:

2. CSMG( $\bar{u}_0, \bar{u}, f, h$ )

3.  $\bar{u}_0 = \bar{u}$

4.  $i = i + 1$

Fim

---

## 2.4 PROBLEMAS ANISOTRÓPICOS

### 2.4.1 Introdução

Segundo Montero et al. (2001), a anisotropia ocorre naturalmente no campo de dinâmica dos fluidos computacional onde a simulação de fenômenos físicos de pequena escala, tais como camadas limites com alto número de Reynolds causam uma malha altamente alongada, conduzindo a uma lenta convergência em métodos *multigrid*.

Em problemas bidimensionais a anisotropia é caracterizada pelos coeficientes dos termos  $u_{xx}$  e  $u_{yy}$ , os quais podem diferir por ordens de magnitude. Considere problemas bidimensionais, tais que:

- 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. Os coeficientes dos operadores discretos variam ao longo do domínio.
- A discretização da malha tenha elementos com tamanho constante, 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:

$$-\left|g^2 + \kappa w^2\right|u_{xx} + 2(1 - \kappa)g w u_{xy} - \left|w^2 + \kappa g^2\right|u_{yy} = f \quad (2.13)$$

onde  $g = \cos(\alpha)$ ,  $w = \sin(\alpha)$ ,  $0 \leq \alpha \leq \frac{\pi}{2}$  e  $0 < \kappa \ll 1$  ou  $\kappa \gg 1$ , ou seja  $\kappa$  distinto da unidade.

Para o caso particular onde  $\alpha = 0$ , a expressão dada pela Eq. (2.13) torna-se:

$$-u_{xx} - \kappa u_{yy} = f \quad (2.14)$$

e diz-se que a anisotropia está alinhada com o eixo de coordenada  $x$ .

Para o caso particular onde  $\alpha = \frac{\pi}{2}$ , a Eq. (2.13) torna-se:

$$-\kappa u_{xx} - u_{yy} = f \quad (2.15)$$

e diz-se que a anisotropia está alinhada com o eixo de coordenada  $y$ .

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

$$Q = \frac{h_x}{h_y} = \sqrt{\kappa} \quad (2.16)$$

**Exemplo de anisotropia física:** A equação diferencial dada pela Eq. (2.14) discretizada em uma malha de tamanho  $h$  em ambas as direções ( $h_x = h_y = h$ ), com  $\kappa$  bem distinto da unidade. Nesse caso a razão de aspecto ( $Q$ ), é redefinida em função das equações Eq. (2.13), Eq. (2.14) e Eq. (2.15) por:

$$Q = \sqrt{\kappa} \quad (2.17)$$

Os dois casos são exemplos de problemas anisotrópicos, ou seja, problemas onde o acoplamento entre os pontos vizinhos é muito forte em alguma das direções (anisotropia física) ou quando a discretização é baseada em malhas com razão de aspecto distinta da unidade (anisotropia geométrica).

Nesta tese escolheu-se como ponto de partida a anisotropia geométrica. Pretende-se posteriormente realizar um estudo com a anisotropia física utilizando as técnicas já estudadas na anisotropia geométrica.

Toma-se como problema modelo o caso onde  $\alpha = 0$ , ou seja, anisotropia alinhada na direção  $x$ , dado pela Eq. (2.14). Neste caso tem-se dois tipos de anisotropia em função de  $\kappa$ :

- 1º tipo: Para  $0 < \kappa \ll 1$  o forte acoplamento está na direção  $x$ .
- 2º tipo: Para  $\kappa \gg 1$  o forte acoplamento está na direção  $y$ .

Se  $\kappa = 1$  tem-se um problema isotrópico. As taxas de convergência de problemas envolvendo  $\kappa \approx 0$  se deterioram com o uso do método *multigrid* com o engrossamento padrão (TROTTEMBERG et al., 2001). Por exemplo, considere um caso extremo com  $\kappa = 0$  na Eq. (2.14). Neste caso tem-se um problema unidimensional (equação de Poisson) representado por

$$-u_{xx} = f \quad (2.18)$$

Portanto, os métodos de relaxação suavizarão as componentes de erro apenas na direção  $x$ , o que significa que o controle do erro na direção  $y$  será totalmente aleatório.

Existem diversas estratégias para resolver problemas que envolvem este tipo de anisotropia. Quando a anisotropia é conhecida antecipadamente pode ser utilizado um suavizador implícito por bloco para melhorar a eficiência do *multigrid* (TROTTEMBERG et al., 2001). Usualmente isto é feito aplicando um *solver* implícito em direções de forte acoplamento. De um modo geral a natureza da anisotropia não é conhecida antecipadamente. Assim não há nenhum modo de saber quais variáveis estão acopladas (TROTTEMBERG et



al., 2001). No caso em que a natureza da anisotropia não é conhecida pode-se utilizar a técnica de semi-engrossamento direcional (MULDER, 1989).

Em problemas bidimensionais o problema de alinhamento pode ser superado usando relaxação por linha, mas possui como desvantagem a linearização global do resíduo restringindo-se a domínios simples (TROTTEBERG et al., 2001).

Outro modo de resolver o problema é usando semi-engrossamento, que consiste em combinar pontos ou células na direção de forte acoplamento. Por exemplo, considere um problema onde o forte acoplamento esteja na direção  $x$ . Como a boa convergência do método *multigrid* é esperada na direção  $x$ , 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  $y$ . Na subseção 2.4.3. será feita uma descrição mais detalhada sobre os algoritmos de semi-engrossamento.

Briggs et al. (2000) e Wesseling (1992) afirmam que a estratégia de semi-engrossamento não é tão rápida quanto o engrossamento padrão. No semi-engrossamento os pontos são reduzidos pela metade durante o engrossamento e no engrossamento padrão, são reduzidos por um quarto no processo de engrossamento. Mas mesmo assim, em problemas anisotrópicos, aplicar o método *multigrid* com engrossamento padrão não conduz a resultados satisfatórios o que justifica o fato de utilizar estratégias de semi-engrossamento (TROTTEBERG et al., 2001).

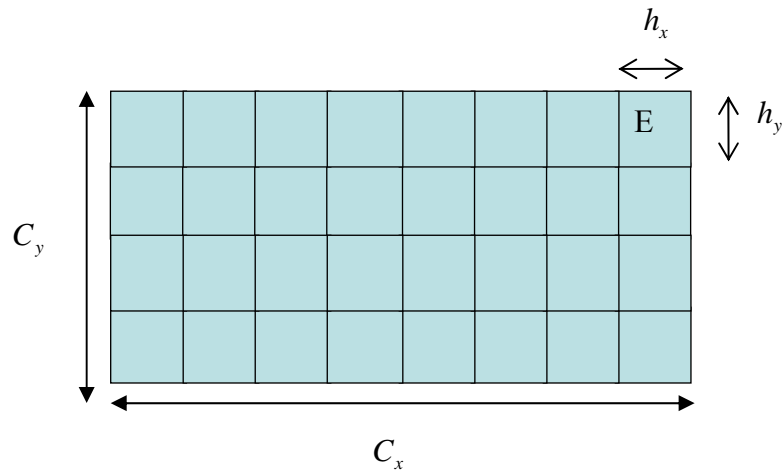
## 2.4.2 Tipos de anisotropia geométrica

A anisotropia geométrica pode ser dividida em quatro tipos, caracterizadas pelo tamanho da malha  $(h_x, h_y)$ , comprimento do domínio de cálculo  $(C_x, C_y)$  e número de nós  $(N_x, N_y)$  nas direções  $x$  e  $y$ , respectivamente. Neste trabalho elas serão denominadas por anisotropia tipo I, II, III e IV de acordo com as suas características a seguir. A anisotropia tipo II é denominada anisotropia no domínio de cálculo.

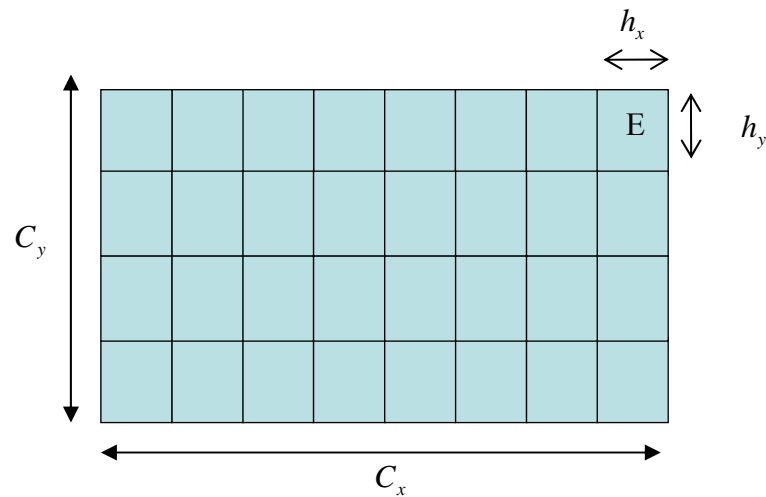
- Anisotropia tipo I:  $h_x \neq h_y$ ,  $N_x = N_y$ ,  $C_x \neq C_y$ .

- Anisotropia tipo II:  $h_x = h_y$ ,  $N_x \neq N_y$ ,  $C_x \neq C_y$ .
- Anisotropia tipo III:  $h_x \neq h_y$ ,  $N_x \neq N_y$ ,  $C_x = C_y$ .
- Anisotropia tipo IV:  $h_x \neq h_y$ ,  $N_x \neq N_y$ ,  $C_x \neq C_y$ .

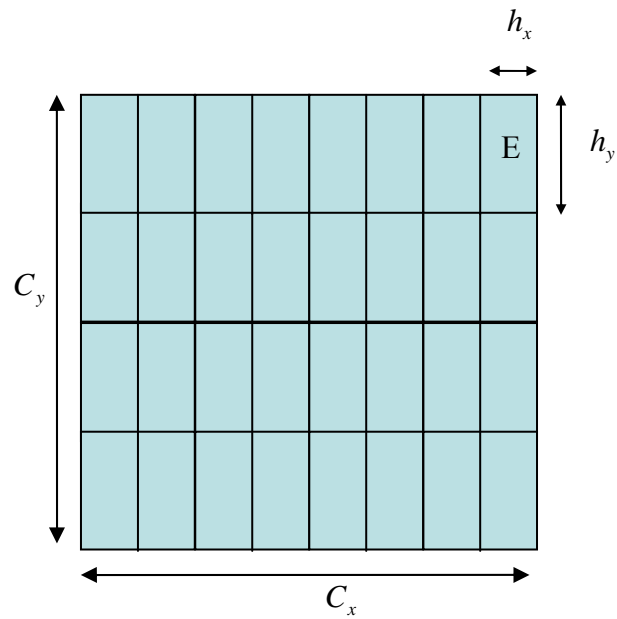
As Figs 2.9 a 2.12 exemplificam estas anisotropias.



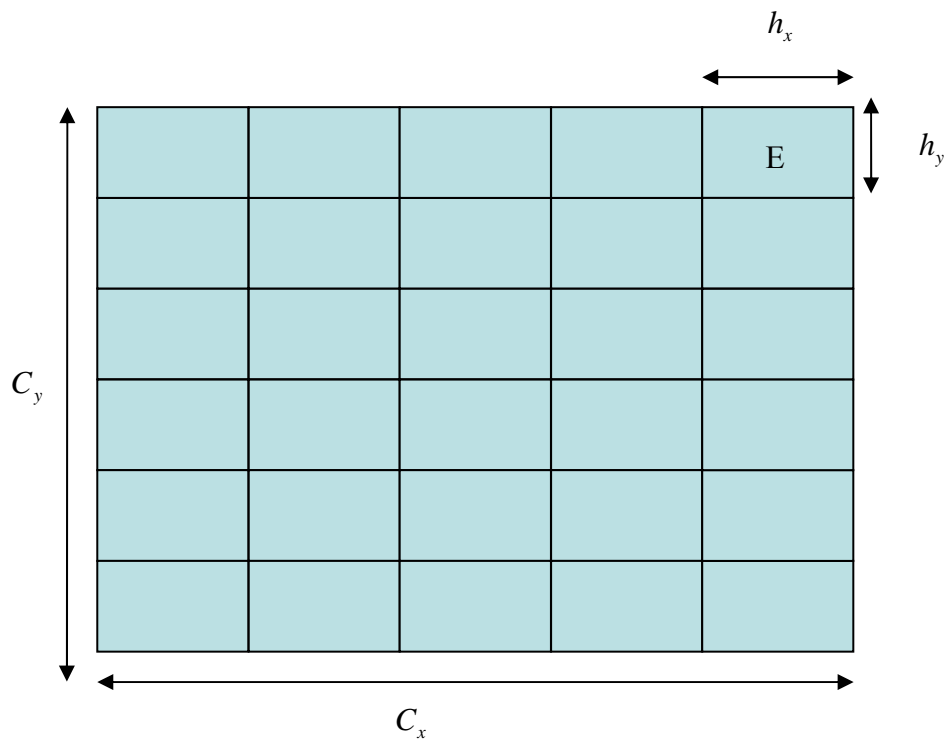
**Figura 2.9: Anisotropia tipo I ( $h_x \neq h_y$ ,  $N_x = N_y$ ,  $C_x \neq C_y$ ).**



**Figura 2.10: Anisotropia tipo II ( $h_x = h_y$ ,  $N_x \neq N_y$  e  $C_x \neq C_y$ ).**



**Figura 2.11: Anisotropia tipo III ( $h_x \neq h_y$ ,  $N_x \neq N_y$ ,  $C_x = C_y$ ).**



**Figura 2.12: Anisotropia tipo IV ( $h_x \neq h_y$ ,  $N_x \neq N_y$ ,  $C_x \neq C_y$ ).**

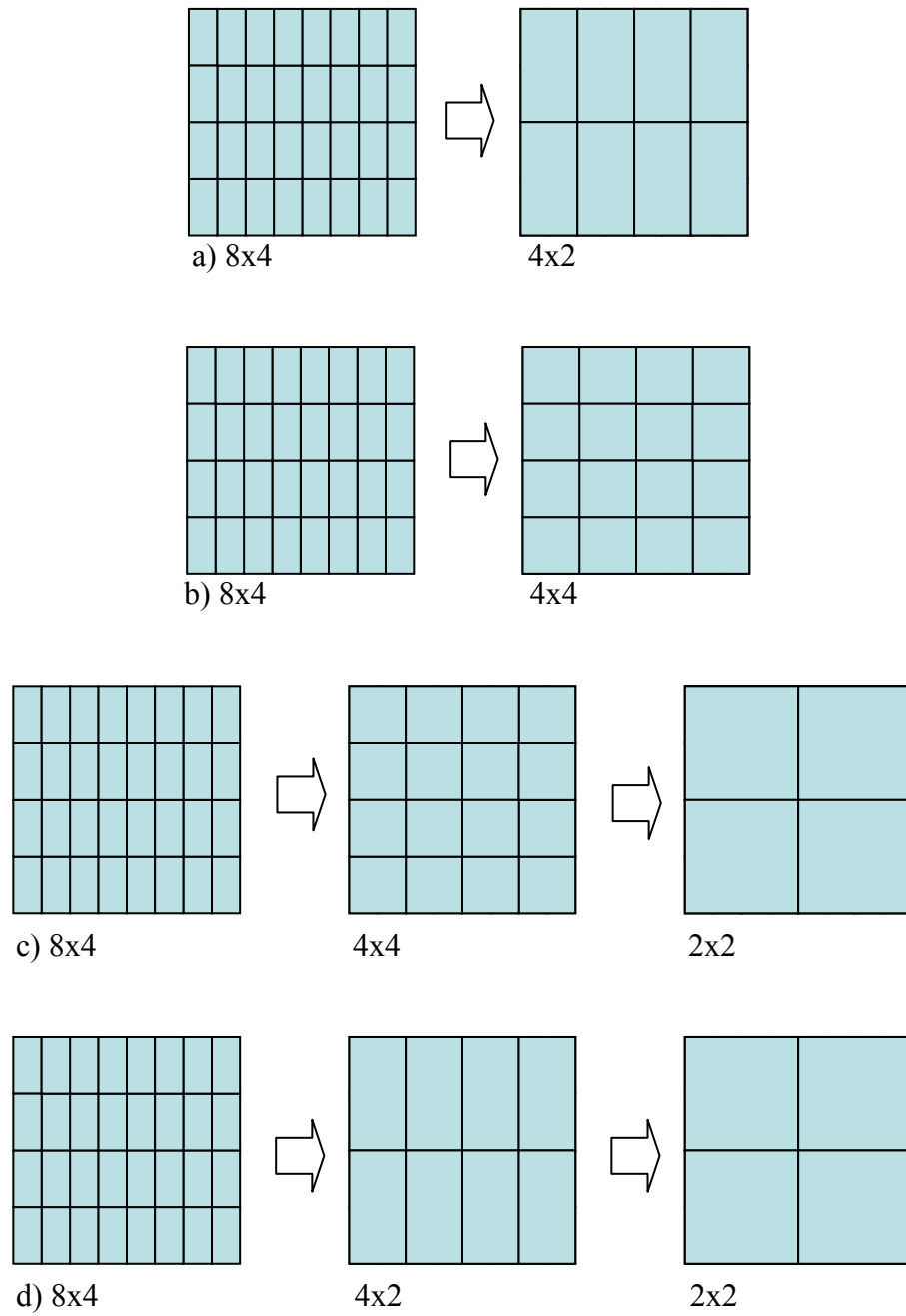
Neste trabalho optou-se pelo estudo da anisotropia tipo III. Este tipo de anisotropia já foi utilizada em alguns trabalhos, como por exemplo, Pinto (2006) e Zhang (2002). Desta forma, os resultados obtidos nesta tese podem ser comparados com os obtidos naqueles

trabalhos. Posteriormente pretende-se estudar outros tipos de anisotropia com as técnicas utilizadas para a anisotropia tipo III.

### 2.4.3 Algoritmos de engrossamento

Algoritmos que envolvem engrossamento e semi-engrossamento podem ser implementados de diversas maneiras. Algumas destas abordagens estão descritas a seguir.

- Engrossamento padrão (EP): apresentado por Brandt (1977). Realiza o engrossamento simultaneamente em ambas as direções. O procedimento padrão do engrossamento é dobrar o tamanho do elemento em cada direção. O procedimento utilizado neste algoritmo está apresentado na Fig. 2.12a.
- Semi-engrossamento (SE): apresentado em Mulder (1989). Realiza o engrossamento em apenas uma das direções coordenadas, onde isto seja possível ou desejado. O semi-engrossamento é realizado até que a malha torne-se isotrópica. Pode ser aplicado em problemas anisotrópicos nos quais seja conhecida a direção de forte acoplamento. O procedimento utilizado neste algoritmo, para o semi-engrossamento em  $x$ , está apresentado na Fig. 2.12b.
- Semi-engrossamento seguido de engrossamento padrão (SE-EP): apresentado em Zhang (2002). Aplica-se o semi-engrossamento até que a malha torne-se isotrópica, e a seguir aplica-se o engrossamento padrão. O procedimento utilizado neste algoritmo, para o semi-engrossamento em  $x$ , está apresentado na Fig. 2.12c.
- Engrossamento padrão seguido de semi-engrossamento (EP-SE): proposto por Pinto e Marchi (2006). Aplica-se primeiro o engrossamento padrão e a seguir o semi-engrossamento. O procedimento utilizado neste algoritmo, para o semi-engrossamento em  $x$ , está apresentado na Fig. 2.12d.



**Figura 2.13:** Algoritmos de engrossamento: a) EP, b) SE, c) SE-EP, d) EP-SE.

## 2.5 RESUMO DO CAPÍTULO 2

Neste capítulo foi apresentada a fundamentação teórica para o estudo do método *multigrid*. Foram descritos os métodos para a resolução de sistemas lineares: Gauss-Seidel

lexicográfico e Gauss-Seidel *red-black*. Definiu-se equação residual e na seqüência foram apresentados os aspectos principais do método *multigrid*: a filosofia do método, operadores de restrição e de prolongação, tipos de ciclos, esquemas de correção (CS e FAS). Também foi apresentada uma introdução a problemas anisotrópicos com as definições de anisotropia física e geométrica, tipos de anisotropia geométrica e ilustrado o funcionamento dos algoritmos de engrossamento.

### 3 MODELOS MATEMÁTICOS E NUMÉRICOS

Este capítulo descreve os modelos matemáticos e numéricos utilizados neste trabalho e os dados de implementação. Utilizou-se as equações de Laplace e Poisson. Trottenberg et al. (2001) recomendam a utilização de equações simples, como por exemplo, Laplace e Poisson para a análise de parâmetros ótimos.

#### 3.1 MODELOS MATEMÁTICOS

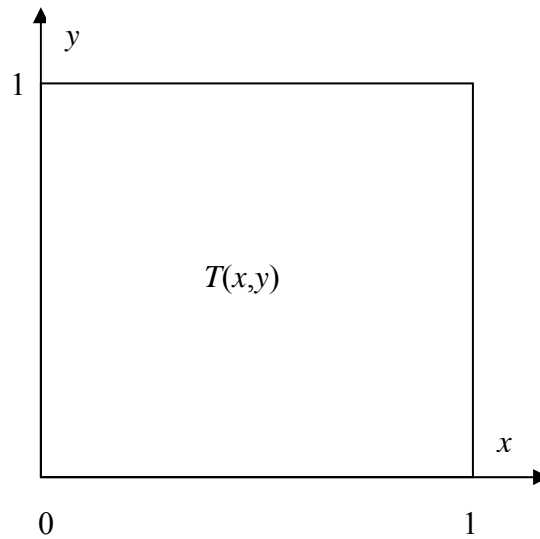
Os modelos matemáticos considerados neste trabalho referem-se a três problemas bidimensionais lineares de condução de calor, governados pelas equações de Laplace e Poisson bidimensionais (INCROPERA e DEWITT,1998):

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = f \quad (3.1)$$

onde  $x$  e  $y$  são as direções coordenadas (variáveis independentes);  $T$  representa a variável dependente (temperatura). Os termos fonte ( $f$ ) e as condições de contorno de Dirichlet nos domínios  $\{(x, y) \in R^2 : 0 \leq x \leq 1 \text{ e } 0 \leq y \leq 1\}$ , conforme Fig. 3.1 estão na Tab. 3.1.

**Tabela 3.1: Problemas resolvidos.**

<b>Equação</b>	<b>Termo Fonte</b>	<b>Condições de Contorno</b>
Laplace linear	$f = 0$	$T(0, y) = T(x, 0) = 0,$ $T(x, 1) = x, T(1, y) = y$
Laplace senoidal	$f = 0$	$T(0, y) = T(x, 0) = T(1, y) = 0$ $T(x, 1) = \text{sen}(\pi x)$
Poisson	$f = -2[(1 - 6x^2)y^2(1 - y^2) + (1 - 6y^2)x^2(1 - x^2)]$	$T(0, y) = T(x, 0) = 0$ $T(1, y) = T(x, 1) = 0$



**Figura 3.1: Domínio bidimensional de cálculo para a equação de Laplace.**

As soluções analíticas para as equações de Laplace senoidal, Laplace linear e Poisson estão apresentadas na Tab. 3.2. Mais detalhes sobre obtenção dessas soluções encontram-se no apêndice A.

**Tabela 3.2: Soluções analíticas para as equações em estudo.**

<b>Equação</b>	<b>Solução Analítica</b>
Laplace linear	$T(x, y) = xy$
Laplace senoidal	$T(x, y) = \text{sen}(\pi x) \frac{\text{senh}(\pi y)}{\text{senh}(\pi)}$
Poisson	$T(x, y) = (x^2 - x^4)(y^4 - y^2)$

### 3.2 MODELOS NUMÉRICOS

A discretização do domínio será descrita somente para a equação de Laplace senoidal. Para as equações de Laplace linear e Poisson ela é feita de forma análoga, variando apenas o termo fonte e os contornos. A discretização é feita com malhas uniformes e malhas estruturadas anisotrópicas, onde o domínio  $\{(x, y) \in \mathbb{R}^2 : 0 \leq x \leq 1 \text{ e } 0 \leq y \leq 1\}$  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 (incluindo os contornos).

Para cada um dos  $(N_x - 2).(N_y - 2)$  pontos interiores da malha, a Eq. (3.1) é discretizada com o método das diferenças finitas (MDF) com diferença central (CDS) (BURDEN e FAIRES, 2008; TANNEHILL et al., 1997). Este método consiste em aproximar diretamente cada termo da equação diferencial separadamente dos demais. Portanto, a aproximação do primeiro termo da Eq. (3.1) é dada por:

$$\left( \frac{\partial^2 T}{\partial x^2} \right)_{i,j} \approx \frac{(T_{i-1,j} - 2T_{i,j} + T_{i+1,j})}{h_x^2} \quad (3.2)$$

e para o segundo termo, tem-se:

$$\left( \frac{\partial^2 T}{\partial y^2} \right)_{i,j} \approx \frac{(T_{i,j-1} - 2T_{i,j} + T_{i,j+1})}{h_y^2} \quad (3.3)$$

onde  $T_{i,j}$  é a solução numérica em cada nó  $(x_i, y_j)$ .

Substituindo as equações Eq. (3.2) e Eq. (3.3) na Eq. (3.1) obtém-se:

$$\frac{(T_{i-1,j} - 2T_{i,j} + T_{i+1,j})}{h_x^2} + \frac{(T_{i,j-1} - 2T_{i,j} + T_{i,j+1})}{h_y^2} = 0 \quad (3.4)$$

Rearranjando os termos da Eq. (3.4) tem-se:

$$\left( \frac{2}{h_x^2} + \frac{2}{h_y^2} \right) T_{i,j} = \left( \frac{1}{h_x^2} \right) T_{i-1,j} + \left( \frac{1}{h_x^2} \right) T_{i+1,j} + \left( \frac{1}{h_y^2} \right) T_{i,j-1} + \left( \frac{1}{h_y^2} \right) T_{i,j+1} \quad (3.5)$$

ou ainda:

$$a_{i,j} T_{i,j} + a_{i-1,j} T_{i-1,j} + a_{i+1,j} T_{i+1,j} + a_{i,j-1} T_{i,j-1} + a_{i,j+1} T_{i,j+1} = b_{i,j} \quad (3.6)$$

Considerando-se as equações Eq. (3.5) e Eq. (3.6), tem-se:

$$a_{i-1,j} = a_{i+1,j} = -\frac{1}{h_x^2}; \quad a_{i,j-1} = a_{i,j+1} = -\frac{1}{h_y^2}; \quad a_{i,j} = \frac{2}{h_x^2} + \frac{2}{h_y^2}; \quad b_{i,j} = 0 \quad (3.7)$$

Estes coeficientes são válidos para os nós internos da malha, que envolvem:

$$\begin{cases} i = 2, 3, \dots, N_x - 1 \\ j = 2, 3, \dots, N_y - 1 \end{cases} \quad (3.8)$$

Para a determinação dos coeficientes nos nós do contorno da malha utilizou-se condições de contorno de Dirichlet, em coordenadas cartesianas, resultando em:

$$a_{i,j} = 1; \quad a_{i-1,j} = a_{i+1,j} = a_{i,j-1} = a_{i,j+1} = 0; \quad b_{i,j} = T_{C.C} \quad (3.9)$$

onde  $T_{C.C}$  representa as temperaturas nos contornos.

tem-se  $T(x,1) = \text{sen}(\pi x)$ ,  $T(1,0) = T(1,y) = T(0,y) = 0$ , onde  $T$  representa os valores conhecidos da temperatura em cada nó dos contornos, que envolvem:

$$\begin{cases} i = 1 \text{ e } N_x \\ j = 1 \text{ e } N_y \end{cases} \quad (3.10)$$

Se  $T$  e  $f$  são denotados por  $T = (T_1, \dots, T_N)^t$  e  $f = (f_1, \dots, f_N)^t$ , respectivamente, onde  $f$  é o vetor independente formado pelos termos  $b_p$ , então o sistema da Eq. (3.6) pode ser representado por um sistema de equações algébricas do tipo:

$$AT = f \quad (3.11)$$

onde a matriz dos coeficientes  $A$  é pentadiagonal  $N$  por  $N$ , simétrica e definida positiva (Briggs et al., 2000),  $T$  é um vetor de incógnitas e  $f$  o termo fonte. Uma matriz  $A$  é definida positiva se ela é simétrica e se  $x^t Ax > 0$  para todo vetor  $n$ -dimensional  $x \neq 0$ , (BURDEN e Faires, 2008).

### 3.3 DADOS DE IMPLEMENTAÇÃO

Os dados desta seção definem as características básicas do método *multigrid* padrão utilizado nesta tese. Novos parâmetros utilizados serão definidos ao longo do texto. O sistema de equações algébricas representado pela Eq. (3.11) é resolvido com o método *multigrid* geométrico, conforme descrito por Wesseling (1992), usando-se o esquema de correção CS. Conforme Fletcher (1991), o esquema de aproximação completa FAS é de 5% a 10% mais caro, em relação ao tempo de CPU, que o CS por restringir o resíduo e a solução para as malhas mais grossas. O tipo de ciclo utilizado é o V, pois o W é cerca de 50% mais caro em relação ao número de operações envolvidas (HIRSCH, 1988).

Utiliza-se restrição por injeção e prolongação através de interpolação bilinear (TROTTEBERG et al., 2001). Entre os operadores de restrição conhecidos (BRIGGS et al., 2000 e TROTTEBERG et al. 2001), o mais comum é o operador de restrição por injeção. O operador de restrição por injeção também é utilizado em diversos trabalhos entre eles Gerolymos e Vallet (2005) e Pinto (2006). O operador de prolongação bilinear também é muito utilizado na literatura. Entre eles pode-se citar: BRIGGS et al. (2000); FERZIGER e PERIC (2002); TANNEHILL et al. (1997); TROTTEBERG et al. (2001) e WESSELING (1992). A razão de engrossamento de malhas ( $r$ ) usada é a padrão, ou seja,  $r = 2$ . (BRANDT, 1977 e BRIGGS et al., 2000).

O suavizador ou *solver* considerado é o método de Gauss-Seidel lexicográfico (GS) descrito em Burden e Faires (2008). No presente trabalho, o número de iterações internas feitas em cada malha é representado por  $\nu$ .

O número de vezes que o ciclo V é repetido é denominado de iterações externas. O critério de convergência usado para interromper as iterações externas é a norma  $l_2$  do resíduo adimensionalizada pela norma  $l_2$  do resíduo na estimativa inicial (norma freqüentemente utilizada na literatura), dada por:

$$l_2 = \frac{\|R(it)\|_2}{\|R(0)\|_2} \quad (3.12)$$

onde  $R(it)$  é o resíduo na iteração atual e  $R(0)$  o resíduo na estimativa inicial. Neste trabalho, para simplificar a notação esta norma será denotada por norma  $l_2$ . Entre os trabalhos que

usam esta norma pode-se citar: BRIGGS et al. (2000); CHISHOLM (1993); TROTTEBERG et al. (2001); ZHANG (2002) e WESSELING (1992). O processo iterativo é interrompido quando a norma  $l_2$  é menor ou igual à tolerância ( $\varepsilon = 10^{-10}$ ). Esta tolerância também já tem sido usada em outros trabalhos entre os quais pode-se citar Zhang, (2002). Mais detalhes sobre a escolha da tolerância encontram-se na subseção 4.4.1.

Em todas as simulações, considerou-se um número de níveis de malha  $L$  tal que  $1 \leq l \leq L \leq L_{máximo}$ , onde  $l$  é o número do nível de uma malha em particular; e  $L_{máximo}$  representa o número máximo possível de malhas que se pode usar para uma dada malha mais fina, com a malha mais grossa tendo apenas um nó interno. Por exemplo, se  $N = 513 \times 513$  nós as malhas são de  $513 \times 513$ ,  $257 \times 257$ ,  $129 \times 129$ ,  $65 \times 65$ ,  $33 \times 33$ ,  $17 \times 17$ ,  $9 \times 9$ ,  $5 \times 5$  e  $3 \times 3$  nós; neste exemplo, portanto,  $L_{máximo} = 9$ .

O foco deste trabalho é a minimização do tempo de CPU ( $t_{CPU}$ ). Entende-se por tempo de CPU o tempo gasto para realizar a geração de malhas, atribuir a estimativa inicial, calcular os coeficientes e resolver o sistema linear representado pela Eq. (3.11) até atingir a tolerância estabelecida com base no critério de convergência. Este tempo é medido em segundos (s) usando-se a sub-rotina CPU\_TIME do Fortran 2003. Em algumas simulações o tempo de CPU foi muito pequeno, menor do que um segundo. Quando isto ocorre aumenta-se a imprecisão da função CPU\_TIME. Por este motivo, criou-se um processo no qual as simulações de um mesmo problema são repetidas até que o tempo seja superior a um valor fixo. O procedimento foi realizado da seguinte forma: Para todas as malhas cujo tempo de CPU foi inferior a 10 segundos, acrescentou-se ao programa principal um ciclo externo. Este ciclo faz com que o número de simulações se repita automaticamente até que seja obtido um tempo de CPU igual ou superior a 10 segundos. Nestes casos, o tempo de CPU de uma simulação é uma média do tempo obtido em todas as repetições. Esse procedimento foi adotado para reduzir a incerteza da sub-rotina CPU\_TIME na medida do tempo de CPU.

O valor ótimo de um parâmetro é determinado quando a solução do problema é obtida no menor tempo de CPU quando os demais parâmetros estão fixos. Assim, denota-se por  $v_{ótimo}$  o número de iterações internas no *solver* que resulta no menor tempo de CPU e por  $L_{ótimo}$  o número de níveis de malhas que resulta no menor tempo de CPU.

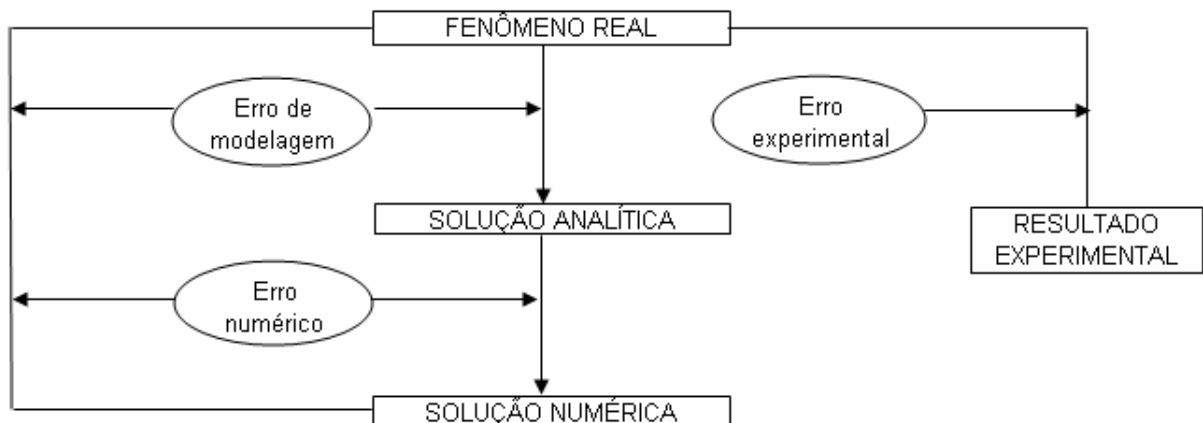
## 4 VERIFICAÇÃO NUMÉRICA DO CÓDIGO COMPUTACIONAL

Este capítulo apresenta uma análise do código computacional empregado na obtenção dos resultados numéricos obtidos neste trabalho. Esta análise foi feita através do estudo de erros numéricos. São analisadas as equações de Laplace senoidal, Laplace linear e Poisson em malhas isotrópicas e a equação de Laplace senoidal em malhas anisotrópicas. O objetivo deste capítulo é verificar a coerência do código computacional utilizado na tese.

A seção 4.1. apresenta uma introdução do capítulo com definições de interesse. Apresenta também alguns detalhes numéricos utilizados nas simulações. A seção 4.2 faz um estudo das ordens efetiva e aparente e na seção 4.3 o estudo dos erros numéricos. A seção 4.4 apresenta alguns testes para mostrar a coerência do código computacional. Primeiramente verificou-se qual a maior malha a ser utilizada sem o uso de memória de HD (*hard disk*) e a tolerância máxima que pode ser utilizada. Na seqüência foi realizado um estudo do erro de discretização e uma comparação entre os erros máximos obtidos com os métodos em estudo. Realizou-se também, uma análise da temperatura em função do número de incógnitas. A seção 4.5 apresenta algumas comparações dos resultados obtidos pelo código computacional e os resultados obtidos na literatura. Na seção 4.6 é apresentado um estudo dos algoritmos de engrossamento para problemas anisotrópicos.

### 4.1 INTRODUÇÃO

Para determinar a solução de um modelo matemático por meio de um método numérico encontram-se diversas fontes de erros. Estes erros estão ligados ao processo de análise e resolução do problema. Há dois grupos de métodos empregados para a solução de um problema, os métodos experimentais e os métodos teóricos. Os métodos teóricos podem ser divididos em analíticos e numéricos (MARCHI, 2001). A Fig. 4.1 apresenta a divisão entre os métodos de solução e os erros gerados em cada um deles.



**Figura 4.1: Erros envolvidos nos métodos de engenharia (adaptada de Marchi e Schneider, 2004).**

Como este trabalho trata de soluções numéricas de equações diferenciais parciais, os erros experimentais, apresentados na Fig. 4.1, não são descritos aqui, uma vez que estão sendo considerados apenas métodos de soluções numéricas de problemas, ou seja, métodos teóricos. A magnitude aceitável para o erro numérico depende, dentre outros fatores, da finalidade da solução numérica, dos recursos financeiros envolvidos, do tempo permitido para realizar as simulações e dos recursos computacionais disponíveis. Sabendo-se que as soluções numéricas contêm erros, é importante estimá-los pelos seguintes motivos (MARCHI, 2001):

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

A diferença entre a solução analítica exata ( $\Phi$ ) de uma variável de interesse e a sua solução numérica ( $\phi$ ) é denominado por Ferziger e Peric (2002) de erro da solução numérica ( $Ed$ ), ou simplesmente erro numérico, isto é:

$$\text{Erro numérico}(\phi) = \Phi - \phi \quad (4.1)$$

O valor do erro numérico independe de resultados experimentais, mas só pode ser obtido quando a solução analítica de um modelo matemático é conhecida. Porém, em termos práticos, isto é, para soluções numéricas de modelos matemáticos cuja solução analítica é desconhecida, não é possível obter o erro numérico. Nestes casos é necessário estimar qual seria o valor da solução analítica. Assim, em vez do erro numérico calcula-se o erro estimado que também é chamado de incerteza por Mehta (1996) e Chapra e Canale (1994). Mais detalhes sobre incerteza podem ser encontrados em Marchi (2001).

O erro numérico é causado por várias fontes de erro, conforme definido na seção 1.1. Quando o erro da solução numérica é gerado apenas por erros de truncamento da série de Taylor, ele é denominado de erro de discretização ( $Ed$ ).

As estimativas dos erros de discretização, gerados pelos erros de truncamento podem ser divididas em dois grupos: estimativas *a priori* e *a posteriori* da solução numérica (SZABÓ e BABUSKA, 1991).

As estimativas de erro *a priori* proporcionam uma análise qualitativa do erro de discretização antes mesmo de se obter uma solução numérica. O objetivo de uma estimativa *a priori* é obter a ordem assintótica da equação diferencial discretizada. Com esta estimativa, antes de se obter qualquer solução numérica, é possível prever o comportamento assintótico do erro de discretização com relação ao refinamento da malha e à ordem aparente<sup>2</sup>. Também é possível avaliar qual é o efeito da redução do tamanho dos elementos da malha sobre o erro de discretização ( $Ed$ ) da solução numérica (MARCHI, 2001).

As estimativas de erro *a posteriori* são usadas para estimar a magnitude do erro de discretização. Existem vários métodos que podem ser empregados. Eles podem ser divididos em dois grandes conjuntos. No primeiro, as estimativas de erro são baseadas na solução numérica obtida numa única malha; em geral, o método dos elementos finitos se enquadra neste conjunto. No segundo conjunto, as estimativas de erro são baseadas nas soluções numéricas obtidas em malhas múltiplas, em geral, os métodos de diferenças finitas e de

---

<sup>2</sup> A ordem aparente é definida posteriormente na Eq. 4.3.

volumes finitos se enquadram neste conjunto. Alguns estimadores deste tipo são: delta, Richardson, GCI, multicoeficientes e convergente (MARCHI, 2001). Detalhes de como se determinar a estimativa *a priori* podem ser obtidos em Marchi (2001).

Neste trabalho foram consideradas as ordens efetiva e aparente dos erros de discretização. Para a obtenção da ordem efetiva é necessário o erro verdadeiro (solução analítica menos a solução numérica) em duas malhas. Para a obtenção da ordem aparente são necessárias três soluções numéricas. Estas duas ordens permitem verificar *a posteriori* se a ordem assintótica, obtida da discretização das equações diferenciais, ou seja, obtida *a priori* da solução numérica, é atingida. Será analisada também a queda do erro numérico (diferença entre solução analítica e numérica), com o refino da malha.

A ordem efetiva ( $p_E$ ) é definida como a inclinação local da curva do erro de discretização ( $Ed$ ) da solução numérica  $\phi$  versus o tamanho ( $h$ ) dos elementos da malha num gráfico em escala logarítmica. Seu cálculo permite verificar na prática, isto é, *a posteriori* das soluções numéricas, se à medida que  $h$  é reduzido, a ordem do erro de discretização das soluções numéricas tende à ordem assintótica dos erros de truncamento. A ordem efetiva pode ser calculada por:

$$p_E = \frac{\log\left[\frac{Ed(\phi_2)}{Ed(\phi_1)}\right]}{\log(q)} \quad (4.2)$$

onde  $\phi_1$  e  $\phi_2$  são as soluções numéricas obtidas em malhas fina e grossa, respectivamente e  $q$  é a razão de refino de malha, dada por:  $q = h_2/h_1$ .

A ordem aparente ( $p_U$ ) permite verificar na prática, isto é *a posteriori* das soluções numéricas, se à medida que  $h$  é reduzido, a ordem da incerteza das soluções numéricas tende à ordem assintótica dos erros de truncamento, ordem esta que é um resultado teórico, obtido *a priori* das soluções numéricas. A ordem aparente pode ser calculada por:

$$p_U = \frac{\log\left(\frac{\phi_2 - \phi_3}{\phi_1 - \phi_2}\right)}{\log(q)} \quad (4.3)$$



onde  $\phi_1$ ,  $\phi_2$  e  $\phi_3$  são as soluções numéricas obtidas para as malhas consideradas fina, grossa e super grossa, respectivamente e  $q$  é a razão de refino de malha, dada por:  $q = (h_2/h_1) = (h_3/h_2)$  (refino uniforme).

Neste capítulo foram consideradas quatro variáveis de interesse para a análise da ordem efetiva, ordem aparente e erro numérico: temperatura numérica no nó central ( $T(0,5;0,5)$ ); temperatura numérica média ( $\overline{TN}$ ) obtida pela regra do trapézio; norma  $l_\infty$  do erro numérico dada por  $\|TA - TN\|_\infty$  e norma  $\bar{l}_1$  do erro numérico dada por  $\|TA - TN\|_1 / N$ . Onde  $TA$  e  $TN$  são as temperaturas analíticas e numéricas respectivamente. As normas  $l_\infty$  e  $\bar{l}_1$  do erro numérico foram utilizadas somente para a verificação do código computacional.

Foram consideradas as equações de Laplace senoidal e linear conforme definidas na seção 3.1. A discretização foi feita através do método das diferenças finitas e com condições de contorno de Dirichlet. Para as simulações deste capítulo utilizou-se o *solver* Gauss-Seidel *red-black*, restrição por ponderação completa e interpolação bilinear. Foram realizadas 3 iterações internas, ou seja,  $\nu = 3$ . Foi utilizado um número de iterações externas suficientemente grande, de forma que o erro de máquina seja atingido. Foram obtidos resultados para as malhas 3x3, 5x5, 9x9, 17x17, 33x33, 65x65, 129x129, 257x257, 513x513, 1025x1025, 2049x2049, 4097x4097 e 8193x8193 nós. A Tab. C.5, apresentada no apêndice C, mostra o cálculo de  $h$  utilizado nas discretizações, juntamente com o número de níveis, nós e elementos correspondentes.

## 4.2 ORDEM EFETIVA E ORDEM APARENTE

Nesta seção faz-se um estudo das ordens efetivas, aparentes e do erro de discretização ( $Ed$ ), obtidos para as equações de Laplace senoidal, Laplace linear e Poisson. Foram utilizadas malhas isotrópicas, com razão de aspecto  $Q = 1$ . Utilizando-se os parâmetros definidos na seção 3.1 foram determinados os resultados analíticos e numéricos obtidos para cada equação e cada variável de interesse. Estes resultados encontram-se nas Tabs. C.1 a C.4, apresentadas no apêndice C. Com estes valores foram calculadas as ordens efetiva e aparente para cada uma das variáveis de interesse.

Para a equação de Laplace linear as ordens efetiva e aparente não têm seus valores tendendo para a ordem teórica. Este comportamento para o problema linear é esperado, pois não se tem a presença de erros de discretização, a solução analítica é igual a solução numérica, o que afeta o cálculo das ordens para o erro numérico. (FERZIGER E PERIC, 2002). As tabelas com os resultados obtidos para cada variável de interesse e equações de Laplace senoidal e Poisson encontram-se no apêndice C (Tabs. C.5 a C.13).

As Figs. 4.2 a 4.5 mostram os gráficos da ordem efetiva e da ordem aparente para cada variável de interesse para as equações de Laplace senoidal e Poisson.

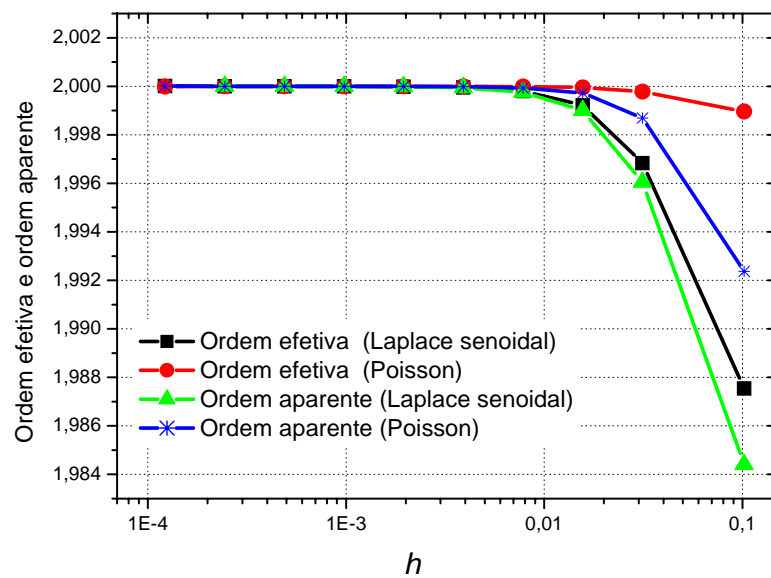


Figura 4.2: Ordem efetiva e ordem aparente para a temperatura no nó central.

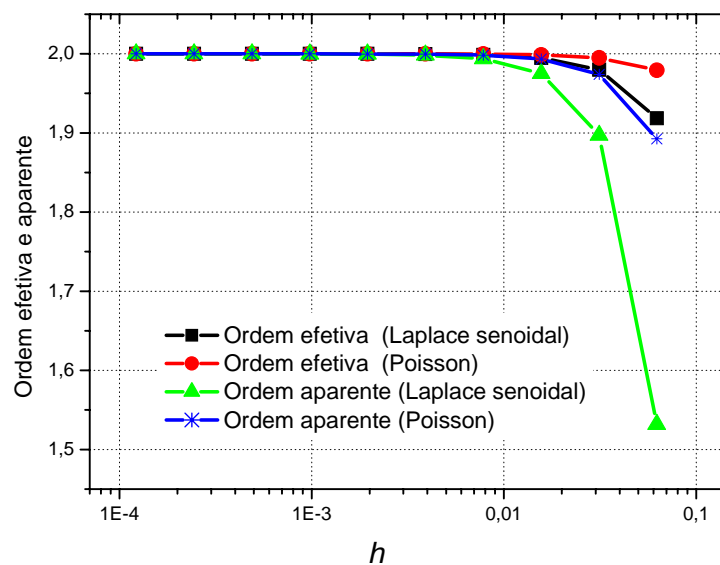


Figura 4.3: Ordem efetiva e ordem aparente para a temperatura média.

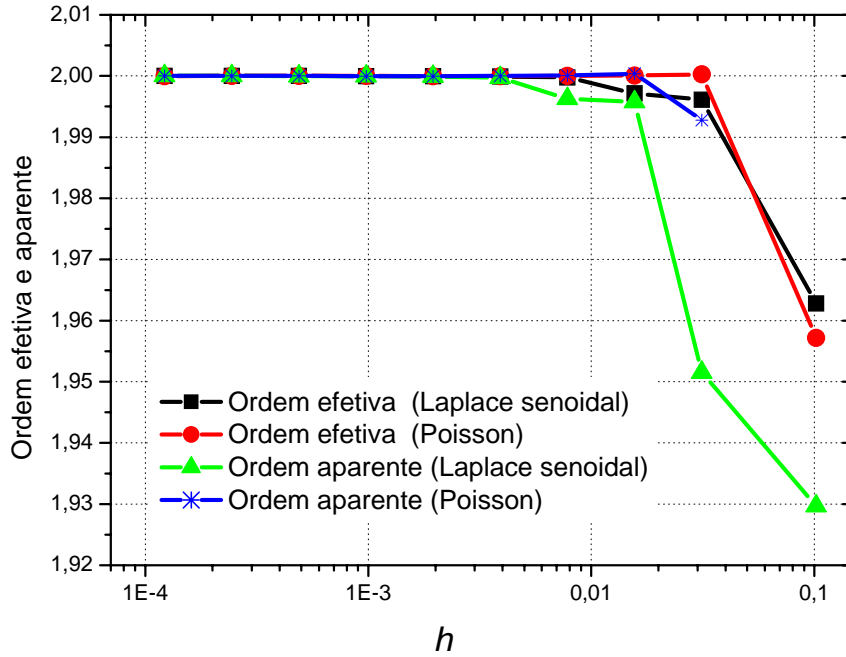


Figura 4.4: Ordem efetiva e ordem aparente para a norma  $l_{\infty}$ .

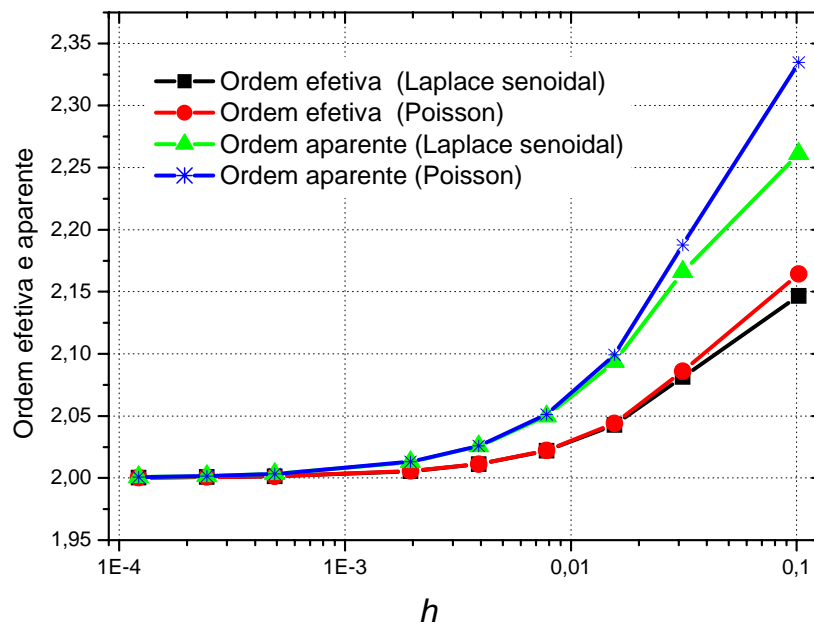
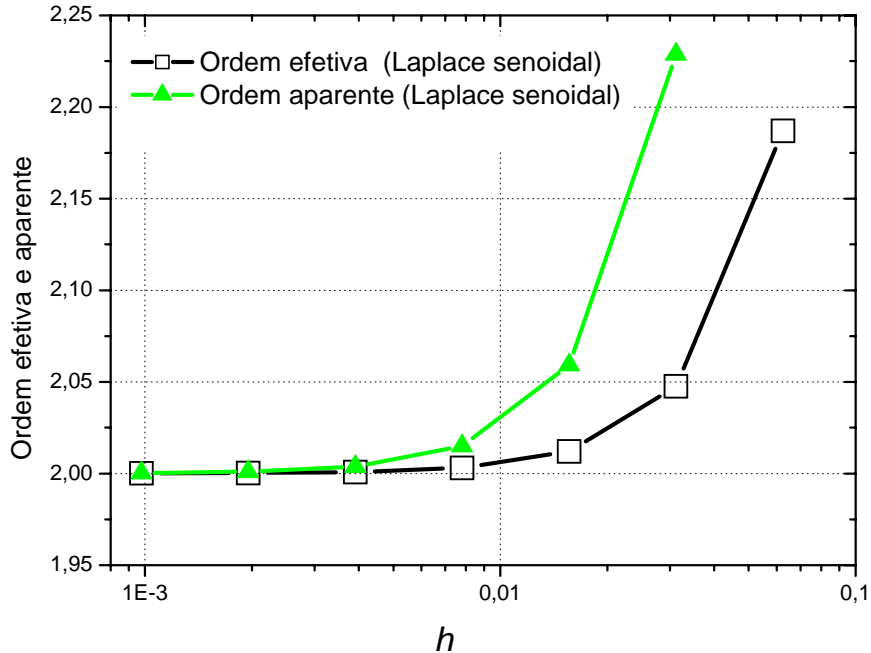


Figura 4.5: Ordem efetiva e ordem aparente para a norma  $l_1$ .

As ordens efetiva e aparente foram calculadas também para a equação de Laplace senoidal em malhas anisotrópicas. Considerou-se a razão de aspecto  $Q = 16$  e como variável de interesse a temperatura média. A Fig. 4.6 apresenta um gráfico com as normas efetiva e

aparente para a temperatura média. Verifica-se na figura que tanto a ordem efetiva como a ordem aparente tendem para 2, conforme o esperado.



**Figura 4.6: Ordem efetiva e ordem aparente para a temperatura média em uma malha anisotrópica.**

A ordem efetiva é calculada com o uso da solução analítica (com base no erro verdadeiro) e deve tender a 2 com o refino da malha. Na estimativa *a priori* esta ordem também é 2. Esse comportamento ocorre para todas as variáveis de interesse conforme pode-se verificar nas Figs. 4.2 a 4.5 para as equações de Laplace senoidal e Poisson. A ordem aparente é calculada com base na solução numérica, também deve tender à 2, com o refino da malha. Esse comportamento ocorre para todas as variáveis de interesse. Para a equação de Laplace senoidal e malhas anisotrópicas a norma efetiva e aparente também tendem a 2 conforme pode ser visto na Fig. 4.6.

### 4.3 ERROS NUMÉRICOS

Esta seção faz um estudo dos erros numéricos obtidos para as equações de Laplace senoidal, Laplace linear e Poisson. As Tabs. C.5 a C.12 do apêndice C apresenta os erros

numéricos obtidos para cada equação e cada variável de interesse. As Figs. 4.7 a 4.9 apresentam os erros numéricos obtidos para todas as variáveis de interesse e equações de Laplace senoidal, Laplace linear e Poisson, respectivamente.

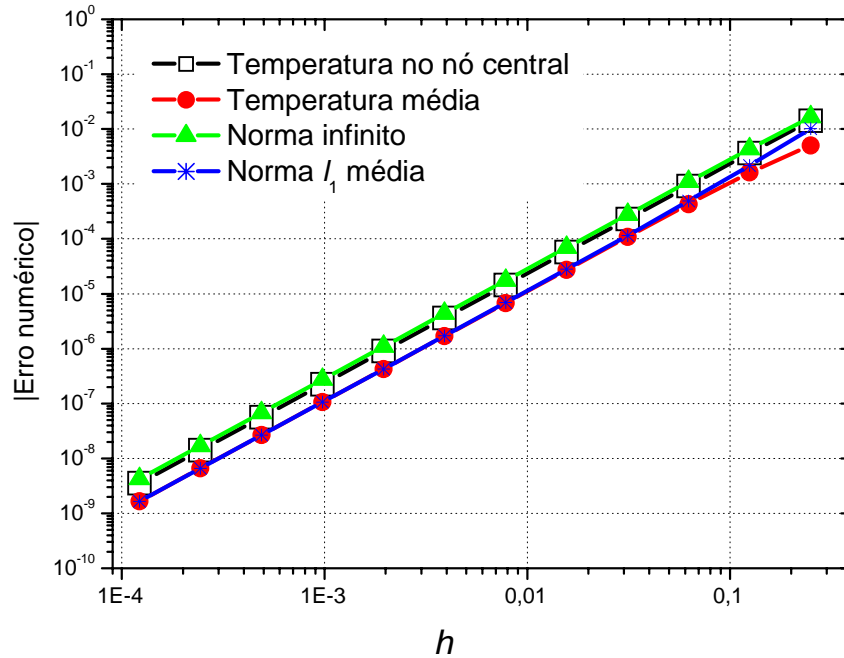


Figura 4.7: Erro numérico para a equação de Laplace senoidal.

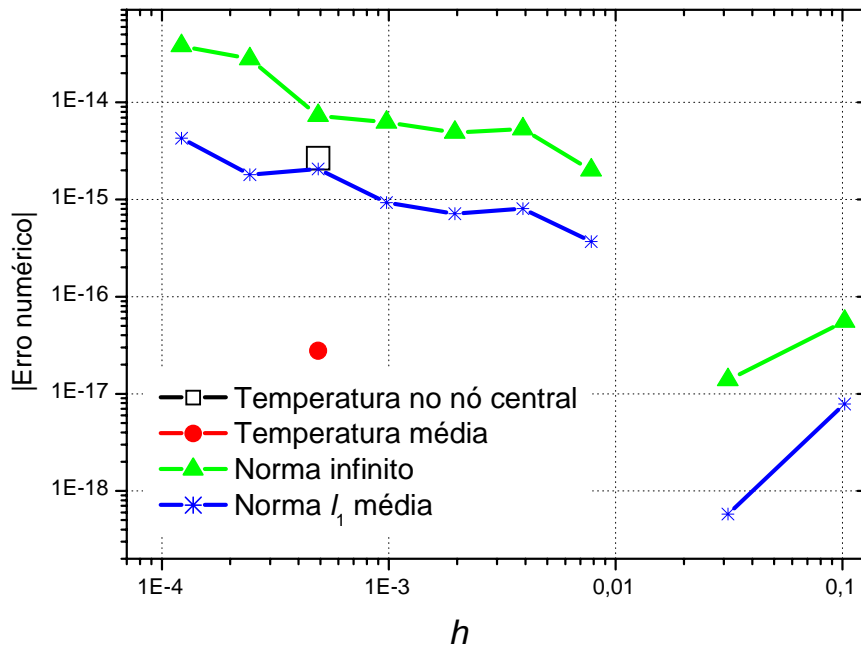


Figura 4.8: Erro numérico para a equação de Laplace linear.

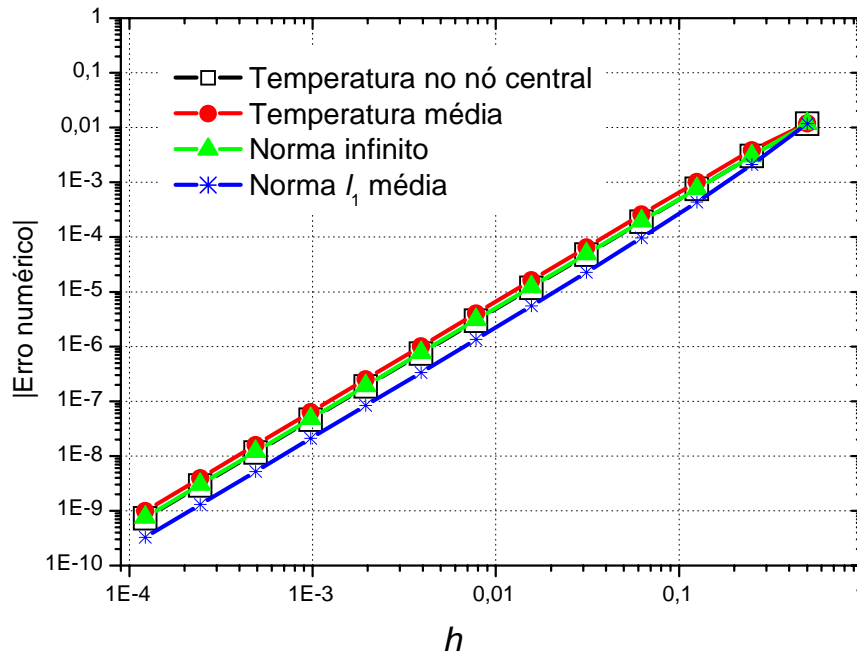


Figura 4.9: Erro numérico para a equação de Poisson.

Nas Figs. 4.7 e 4.9 verifica-se que para as equações de Laplace senoidal e Poisson o erro numérico tende a zero à medida que a malha é refinada, conforme o esperado. Na Fig. 4.8, referente à equação de Laplace linear, verifica-se que para as variáveis, norma  $l_\infty$  e norma  $\bar{l}_1$ , o erro numérico aumenta à medida que a malha é refinada. Este comportamento é atribuído a influência dos erros de arredondamento, já que a equação de Laplace linear não apresenta erros de discretização. O erro numérico para a temperatura numérica no nó central e para a temperatura numérica média só foi obtido em um único ponto. Para estas variáveis, estes valores também sofrem influência dos erros de arredondamento. Observa-se na Fig. 4.8 que os erros numéricos são bem próximos de zero.

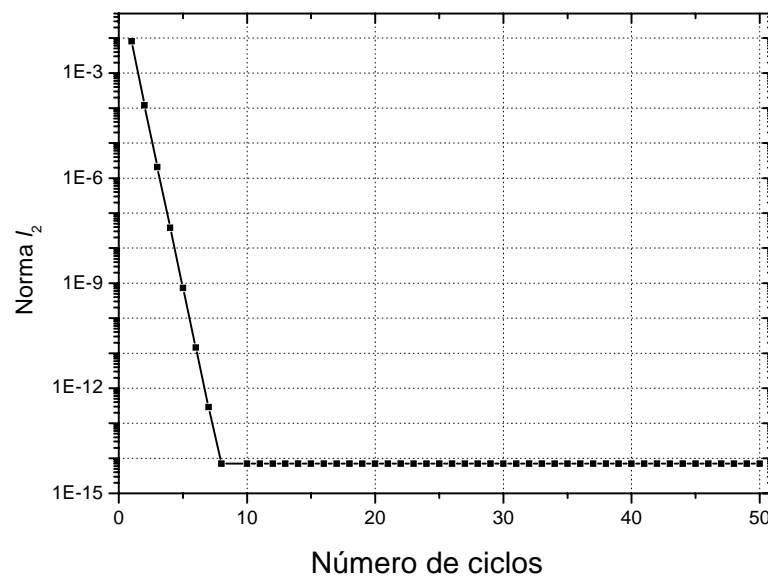
## 4.4 TESTES DE COERÊNCIA

### 4.4.1 Definição da tolerância e da maior malha a ser utilizada

Durante a execução das simulações foi monitorada a memória computacional empregada na resolução do problema tendo como objetivo a determinação do maior problema que pode ser resolvido sem o uso de memória virtual. Para a obtenção da memória empregada

para a resolução da equação de Laplace senoidal foi utilizado o Gerenciador de Tarefas do Windows, onde o uso de memória pode ser observado. Concluiu-se que, para a maior malha isotrópica, o maior problema a ser resolvido possui 67.108.864 elementos, ou seja, em um problema isotrópico corresponde a uma malha 8193x8193. O computador utilizado para as simulações desta tese possui 8 GB de memória RAM. Para a malha 8193x8193, utilizou-se 3 GB de memória RAM. Ao se utilizar mais um nível, ou seja, para uma malha 16384x1638, já seriam necessários 12 GB de memória RAM, o que ultrapassaria a memória disponível.

Para determinar a tolerância a ser utilizada nas simulações utilizou-se a maior malha possível (8193x8193). O processo iterativo foi executado até que o erro de máquina fosse atingido, isto é, até que o resíduo comece a oscilar em torno de um valor. A Fig. 4.10 apresenta o comportamento da norma  $l_2$  do resíduo para uma malha 8193x8193. Verifica-se que a norma começa a decrescer nos primeiros ciclos e começa a estabilizar em torno de  $10^{-15}$ . O valor mínimo da tolerância a ser empregada nas simulações deve ser de pelo menos uma ordem de grandeza maior do que a norma  $l_2$ , arredondando-se para cima. Neste caso, como  $l_2 = 10^{-15}$ , o valor mínimo a empregar para a tolerância será  $10^{-14}$ , que arredondando resulta em  $10^{-13}$ . Desta forma, a tolerância a ser recomendada para a equação de Laplace senoidal deve ser  $\geq 10^{-13}$ , pois com este valor, garante-se que o resíduo não ficará oscilando em torno do erro de máquina. Por garantia foi utilizada uma tolerância um pouco maior ( $\varepsilon = 10^{-10}$ ), considerando-se que a oscilação da norma  $l_2$  em problemas anisotrópicos pode ocorrer antes que a tolerância de  $10^{-13}$  seja atingida.



**Figura 4.10: Comportamento da norma  $l_2$  do resíduo em função do número de ciclos  $V$  para uma malha 8193x8193.**

#### 4.4.2 Erro de discretização

Algumas simulações foram realizadas a fim de verificar se a solução numérica do problema melhora ao utilizar um maior número de nós. Para isto o programa foi executado até atingir o erro de máquina. A Fig. 4.11 apresenta a norma infinito do erro numérico em função do número de incógnitas para o método *multigrid* e *solvers* Gauss-Seidel lexicográfico (GS-LEX) e Gauss-Seidel *red-black* (GS-RB). Verifica-se que, para ambos os *solvers*, a norma infinito diminui à medida que se aumenta o tamanho do problema.

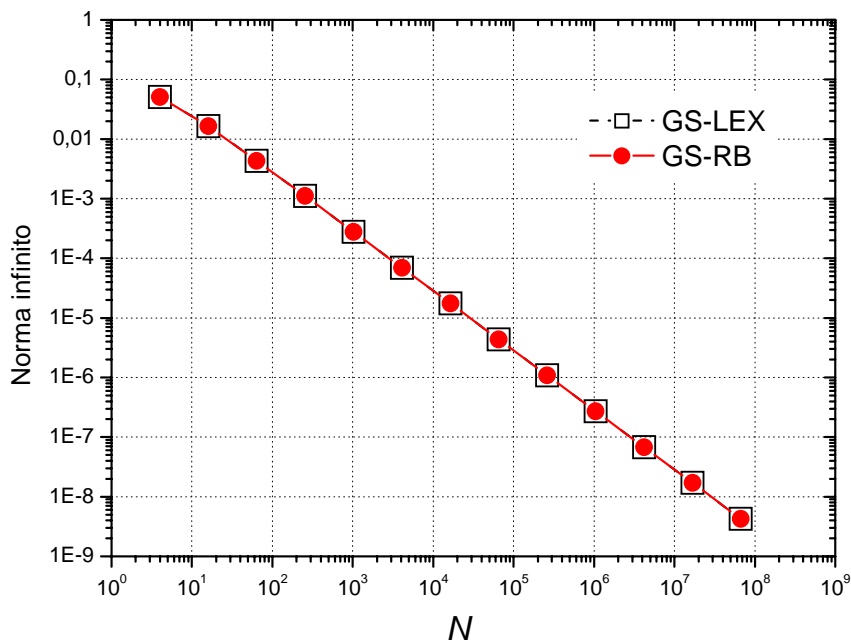


Figura 4.11: Norma infinito ( $l_\infty$ ) do erro numérico *versus* número de incógnitas para o método *multigrid*.

#### 4.4.3 Comparação entre os erros máximos

O objetivo desta seção é comparar o erro máximo obtido pelos métodos *singlegrid* e *multigrid* com o uso da norma  $l_\infty$  para o erro numérico e das normas  $\bar{l}_1$  e a norma  $l_2$ . A norma  $\bar{l}_1$  será usada somente nos testes de coerência. A norma  $l_2$  será utilizada como critério de parada e está definida na Eq. 3.12. O erro foi medido através da norma infinito definida pela diferença entre a temperatura numérica e a temperatura analítica. Utilizou-se uma



tolerância de  $\varepsilon = 10^{-10}$  e uma malha 17x17 nós, para ambos os métodos (*singlegrid* e *multigrid*) e *solvers* (GS-LEX e GS-RB).

A Tab. 4.1 apresenta uma comparação entre as normas para o método *singlegrid* e *solvers* GS-LEX e GS-RB. Verifica-se que a norma  $l_\infty$  é igual para ambos os *solvers* e que a diferença entre a norma  $l_2$  encontra-se na décima segunda casa decimal, sendo bem próximas da tolerância estabelecida ( $\varepsilon = 10^{-10}$ ).

**Tabela 4.1: Comparação entre as normas obtidas no método *singlegrid*.**

<i>Solver</i>	Norma	$l_\infty$	$l_2$
GS-LEX		1,1088415789510253E-03	9,960225E-11
GS-RB		1,1088415722596556E-03	9,872027E-11

A Tab. 4.2 apresenta uma comparação entre as normas obtidas para o método *multigrid* e *solvers* GS-LEX e GS-RB. Verifica-se que a norma  $l_\infty$  é igual para ambos os *solvers* e que a diferença entre a norma  $l_2$  encontra-se na décima primeira casa decimal, sendo bem próximas da tolerância estabelecida ( $\varepsilon = 10^{-10}$ ).

Posteriormente comparou-se as normas dos métodos *singlegrid* e *multigrid*. Analisando as Tabs. 4.1 e 4.2 conclui-se que a diferença entre as normas  $l_\infty$  encontram-se na décima casa decimal (igual a tolerância pré-estabelecida). Isto ocorre para os métodos *multigrid* e *singlegrid* e com ambos os *solvers* em estudo. Pode-se concluir que as normas não apresentam diferenças significativas. O mesmo procedimento também foi realizado para a norma  $\bar{l}_1$  e os resultados encontrados foram similares.

**Tabela 4.2: Comparação entre as normas obtidas no método *multigrid*.**

<i>Solver</i>	Norma	$l_\infty$	$l_2$
GS-LEX		1.1088416700695825E-03	4,517291E-12
GS-RB		1.1088416700695825E-03	1,478271E-11

#### 4.4.4 Análise da temperatura em função do número de incógnitas

Esta seção tem como objetivo comparar as soluções numéricas com a solução analítica obtida para a equação de Laplace senoidal. As soluções numéricas foram obtidas através dos métodos *singlegrid* e *multigrid*, ambos com o uso dos *solvers* Gauss-Seidel lexicográfico e Gauss-Seidel *red-black*. Para estas simulações utilizou-se como critério de parada o erro de máquina. A Fig. 4.12 apresenta o gráfico das temperaturas médias em função do número de variáveis. Cada curva representa uma temperatura média (analítica e numéricas) obtidas através dos métodos *singlegrid* e *multigrid* e com os *solvers* Gauss-Seidel lexicográfico (GS-LEX) e Gauss-Seidel *red-black* (GS-RB). Verifica-se que à medida que se aumenta o número de nós na malha discretizada, as temperaturas numéricas médias se aproximam cada vez mais da temperatura analítica média. Verifica-se também que as temperaturas numéricas médias são as mesmas para todos os métodos e *solvers* utilizados.

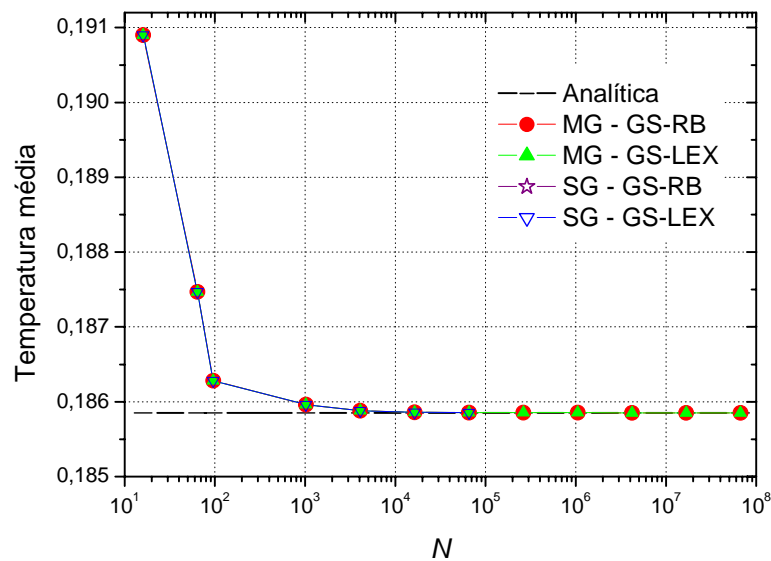


Figura 4.12: Temperatura média *versus* número de incógnitas ( $N$ ) para a equação de Laplace senoidal.

#### 4.5 COMPARAÇÃO COM A LITERATURA

Esta seção tem como objetivo comparar os resultados obtidos pelo programa a ser utilizado nesta tese com os resultados obtidos por Briggs et al. (2000). Utilizou-se a equação

de Poisson 2D definida na seção 3.1. O termo fonte é o mesmo utilizado por Briggs, definido na Tab. 3.1. Foram considerados os seguintes parâmetros no método *multigrid*: restrição por ponderação completa, *solver* Gauss-Seidel *red-black*, ciclo V com  $v = 2$  na restrição e  $v = 1$  na prolongação. O critério de parada é baseado na norma  $l_2$ . Executou-se 15 iterações externas (ciclos V) para cada malha.

As Tab. 4.3 a 4.6 apresentam os resultados obtidos através dos dois programas para malhas 17x17, 33x33, 65x65 e 129x129, respectivamente (malhas utilizadas por Briggs et al., 2000). A primeira coluna apresenta o número de ciclos utilizados. A segunda e a terceira coluna “Briggs” apresentam os resultados obtidos por Briggs et al. (2000), a quarta e a quinta coluna “Tese” apresentam os resultados obtidos através do programa utilizado nesta tese. As colunas referentes à “taxa” mostram a taxa de variação do resíduo em dois ciclos consecutivos, obtida através da razão entre a  $l_2$  obtida no ciclo atual e o ciclo anterior. Verifica-se também que a diferença entre as normas obtidas através dos dois programas não são significativas, exceto para a o resíduo na aproximação inicial de uma malha 33x33 referente à Tab. 4.4. Ao comparar as taxas obtidas verifica-se que ambos os programas convergem rapidamente.

**Tabela 4.3: Comparação entre as normas para uma malha 17x17.**

Ciclos V	Briggs		Tese	
	$l_2$	taxa	$l_2$	taxa
0	6,75E+02	-	1,63E+01	-
1	4,01E+00	0,01	9,13E-02	0,01
2	1,11E-01	0,03	5,94E-03	0,07
3	3,96E-03	0,04	4,05E-04	0,07
4	1,63E-04	0,04	2,85E-05	0,07
5	7,45E-06	0,05	2,05E-06	0,07
6	3,75E-07	0,05	1,51E-07	0,07
7	2,08E-08	0,06	1,13E-08	0,07
8	1,24E-09	0,06	8,50E-10	0,08
9	7,74E-11	0,06	6,48E-11	0,08
10	4,99E-12	0,06	4,97E-12	0,08
11	3,27E-13	0,07	3,84E-13	0,08
12	2,18E-14	0,07	2,98E-14	0,08
13	2,33E-15	0,11	3,10E-15	0,10
14	1,04E-15	0,45	1,68E-15	0,54
15	6,61E-16	0,64	1,31E-15	0,78

Tabela 4.4: Comparação entre as normas para uma malha 33x33.

Ciclos V	Briggs		Tese	
	$l_2$	taxa	$l_2$	taxa
0	2,60E+03	-	3,39E+01	-
1	1,97E+01	0,01	9,85E-02	0,00
2	5,32E-01	0,03	6,93E-03	0,07
3	2,06E-02	0,04	5,05E-04	0,07
4	9,79E-04	0,05	3,76E-05	0,07
5	5,20E-05	0,05	2,85E-06	0,08
6	2,96E-06	0,06	2,19E-07	0,08
7	1,77E-07	0,06	1,69E-08	0,08
8	1,10E-08	0,06	1,32E-09	0,08
9	7,16E-10	0,07	1,04E-10	0,08
10	4,79E-11	0,07	8,22E-12	0,08
11	3,29E-12	0,07	6,52E-13	0,08
12	2,31E-13	0,07	5,31E-14	0,08
13	1,80E-14	0,08	6,88E-15	0,13
14	6,47E-15	0,36	5,89E-15	0,86
15	5,11E-15	0,79	5,30E-15	0,90

Tabela 4.5: Comparação entre as normas para uma malha 65x65.

Ciclos V	Briggs		Tese	
	$l_2$	taxa	$l_2$	taxa
0	1,06E+04	-	6,90E+01	-
1	7,56E+01	0,01	1,00E-01	0,00
2	2,07E+00	0,03	7,26E-03	0,07
3	8,30E-02	0,04	5,39E-04	0,07
4	4,10E-03	0,05	4,09E-05	0,08
5	2,29E-04	0,06	3,16E-06	0,08
6	1,39E-05	0,06	2,47E-07	0,08
7	8,92E-07	0,06	1,95E-08	0,08
8	5,97E-08	0,07	1,55E-09	0,08
9	4,10E-09	0,07	1,24E-10	0,08
10	2,87E-10	0,07	9,96E-12	0,08
11	2,04E-11	0,07	8,03E-13	0,08
12	1,46E-12	0,07	6,73E-14	0,08
13	1,08E-13	0,07	2,17E-14	0,32
14	2,60E-14	0,24	2,11E-14	0,97
15	2,30E-14	0,88	2,06E-14	0,98

Tabela 4.6: Comparação entre as normas para uma malha 129x129.

Ciclos V	Briggs		Tese	
	$l_2$	taxa	$l_2$	taxa
0	4,16E+04	-	1,39E+02	-
1	2,97E+02	0,01	1,01E-01	0,00
2	8,25E+00	0,03	7,36E-03	0,07
3	3,37E-01	0,04	5,51E-04	0,07
4	1,65E-02	0,05	4,20E-05	0,08
5	8,99E-04	0,05	3,26E-06	0,08
6	5,29E-05	0,06	2,56E-07	0,08
7	3,29E-06	0,06	2,03E-08	0,08
8	2,14E-07	0,07	1,63E-09	0,08
9	1,43E-08	0,07	1,31E-10	0,08
10	9,82E-10	0,07	1,06E-11	0,08
11	6,84E-11	0,07	8,71E-13	0,08
12	4,83E-12	0,07	1,16E-13	0,13
13	3,64E-13	0,08	8,46E-14	0,73
14	1,03E-13	0,28	8,24E-14	0,97
15	9,19E-14	0,89	8,35E-14	1,01

#### 4.6 PROBLEMAS ANISOTRÓPICOS

Esta seção tem por objetivo verificar se os diferentes algoritmos de engrossamento (EP, SE, EP-SE e SE-EP) utilizados nesta tese, produzem os mesmos resultados numéricos. As variáveis de interesse consideradas são: temperatura numérica no ponto  $T(0,5;0,5)$ , temperatura numérica média ( $\overline{TN}$ ), norma infinita do erro numérico ( $l_\infty$ ) e norma  $\overline{l}_1$  do erro numérico. Utilizou-se a norma  $\overline{l}_1$  do erro numérico apenas nos testes de coerência. O domínio de cálculo com  $C_x \neq C_y$  também só foi utilizado nos testes de coerência. Nas demais simulações utilizou-se sempre o domínio unitário, ou seja,  $C_x = C_y = 1$ .

Para estas simulações foi utilizado o tipo de anisotropia geométrica mais geral (Anisotropia tipo IV) com  $N_x \neq N_y$ ,  $C_x \neq C_y$  e  $h_x \neq h_y$ , definida na seção 2.4.2. Esta anisotropia é a única que possui todos os parâmetros diferentes, ou seja, se o programa estiver coerente para esta anisotropia, também estará coerente para as demais anisotropias, que são casos particulares desta.

Os testes foram realizados para a equação de Laplace senoidal. Os parâmetros fixos para o método *multigrid* são: *solver* Gauss-Seidel *red-black* com 10 iterações internas, restrição por ponderação completa, prolongação por interpolação bilinear. O programa foi executado até atingir o erro de máquina.

Os testes consistem em: dada uma razão de aspecto verificar se variáveis de interesse consideradas apresentam os mesmos resultados para os quatro algoritmos em estudo. Foi utilizado domínio retangular, ou seja,  $C_x = 256$  e  $C_y = 1$  com  $N_x = 4097$  e  $N_y = 257$  e um problema com 1.048.576 elementos e razão de aspecto 1/16.

A Tab. 4.7 traz uma comparação entre as variáveis de interesse para este problema. Para fins de comparação também são apresentadas a temperatura analítica média ( $\overline{TA}$ ) e temperatura analítica no ponto  $TA(0,5;0,5)$ . Verifica-se que os resultados obtidos em todas as variáveis de interesse são bem próximos. Portanto, um problema anisotrópico pode ser resolvido corretamente com qualquer um dos quatro algoritmos de engrossamento em estudo.

#### 4.7 CONCLUSÃO DO CAPÍTULO

Este capítulo apresentou uma análise do código computacional empregado na obtenção dos resultados numéricos obtidos neste trabalho. Esta análise foi feita através do estudo dos erros numéricos. Foram analisadas as equações de Laplace senoidal, Laplace linear e Poisson em malhas isotrópicas e a equação de Laplace senoidal em malhas anisotrópicas. O objetivo deste capítulo foi verificar a coerência do código computacional empregado nesta tese.

Para o cálculo da ordem efetiva, ordem aparente e erro numérico utilizou-se o *solver* Gauss-Seidel *red-black* com  $\nu = 3$ . Trottenberg et al. (2001) sugere utilizar  $\nu = \nu_1 + \nu_2 \leq 3$ . Foram consideradas quatro variáveis de interesse: temperatura no nó central, temperatura numérica média, norma  $l_\infty$  do erro numérico e norma  $\bar{l}_1$  do erro numérico para as equações Laplace senoidal, Laplace linear e Poisson. Verificou-se que, para todas as equações, o erro numérico tende a zero. A ordem efetiva e a ordem aparente tendem a 2, coincidindo com a ordem assintótica definida *a priori*, para todas as variáveis de interesse nas equações de

Laplace senoidal e Poisson. Conclui-se que a ordem assintótica dos erros de truncamento foram atingidas para todas as variáveis e equações em estudo.

Através do teste de coerência aplicado na equação de Laplace senoidal verificou-se que a maior malha possível (sem o uso de memória virtual) possui 8.193 x 8.193 nós.

A comparação entre as normas (norma  $l_\infty$  do erro numérico e norma  $l_2$  do erro numérico) obtidas para a equação de Laplace senoidal foi feita comparando os algoritmos *singlegrid* e *multigrid* e os *solvers* Gauss-Seidel lexicográfico e Gauss-Seidel *red-black*. Verificou-se que as normas obtidas não apresentam diferenças significativas. Concluiu-se também que a norma  $l_\infty$  e as normas  $\bar{l}_1$  e  $l_2$  estão bem próximas à tolerância pré-estabelecida.

Para a equação de Laplace senoidal, método *multigrid* e *solvers* Gauss-Seidel lexicográfico e *red-black*, foi verificado que a norma  $l_\infty$  do erro numérico diminui à medida que se aumenta o tamanho do problema. As temperaturas numéricas para diversas malhas foram obtidas através dos métodos: *singlegrid* e *multigrid* com Gauss-Seidel lexicográfico e Gauss-Seidel *red-black*. Todos os métodos vão até o erro de máquina. Observa-se que à medida que se aumenta o número de nós na malha discretizada a temperatura numérica média se aproxima cada vez mais da temperatura analítica média. As temperaturas numéricas são iguais, ao se utilizar os métodos *singlegrid* e *multigrid* com os *solvers* Gauss-Seidel lexicográfico e *red-black*. Verificou-se também que as temperaturas numéricas estão bem próximas da temperatura analítica, atingindo a tolerância estabelecida.

Os resultados obtidos foram comparados com os obtidos por Briggs et al. (2000) para a equação de Poisson. Foram utilizados para esta análise o método *multigrid* com *solver* Gauss-Seidel *red-black* com  $\nu = 2$  na restrição e  $\nu = 1$  na prolongação, restrição por ponderação completa e norma  $l_2$ . Verificou-se que a convergência obtida pelo programa está bem próxima da obtida por Briggs et al. (2000).

A análise do programa envolvendo problemas anisotrópicos e diversos algoritmos de engrossamento foi realizada utilizando o método *multigrid* com os seguintes parâmetros: *solver* Gauss-Seidel *red-black* e restrição por ponderação completa. Verificou-se que não há diferença significativa entre os resultados obtidos para todas as variáveis de interesse e todos os algoritmos. Portanto, um problema anisotrópico pode ser resolvido corretamente com o uso dos quatro algoritmos de engrossamento em estudo.

Conclui-se que o código computacional utilizado nesta tese está coerente e apresenta bons resultados.

**Tabela 4.7: Comparação entre as variáveis de interesse para a anisotropia IV, razão de aspecto 1/16 e malha 4097x257 e equação de Laplace senoidal.**

<b>Variáveis de interesse</b>	<b>EP</b>	<b>SE</b>	<b>EP-SE</b>	<b>SE-EP</b>
$\overline{TA}$	3,1830589150270650E-01	3,1830589150270650E-01	3,1830589150270650E-01	3,1830589150270650E-01
$\overline{TN}$	3,1830587595920062E-01	3,1830587595905996E-01	3,1830587595920046E-01	3,1830587595920062E-01
$TA(0,5;0,5)$	4,9999058775942240E-01	4,9999058775942240E-01	4,9999058775942240E-01	4,9999058775942240E-01
$T(0,5;0,5)$	4,9999058775989025E-01	4,9999058775955629E-01	4,9999058775998184E-01	4,9999058775989025E-01
$l_\infty$	5,2724491439448684E-13	1,5054624213917123E-13	5,7942539655186920E-13	5,2724491439448684E-13
$\overline{l}_1$	1,9892714354973051E-13	5,7629090208828183E-14	1,9840250336510515E-13	1,9892714354973051E-13



## 5 ALGORITMOS DE ENGROSSAMENTO

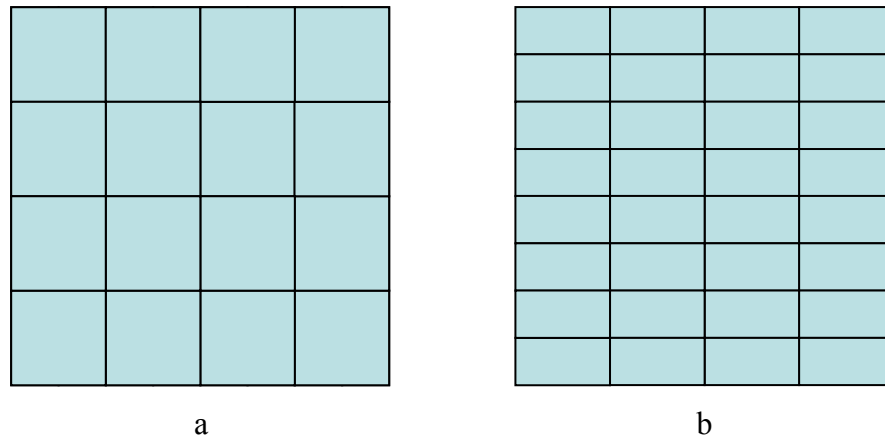
Neste capítulo o método *multigrid* geométrico é aplicado a três problemas bidimensionais lineares de condução de calor, governados pelas equações de Laplace e Poisson, com condições de contorno de Dirichlet dada pela Eq. (3.1) descritas detalhadamente na seção 3.1. Este capítulo tem por objetivo apresentar os testes utilizados para verificar qual o melhor algoritmo de engrossamento para problemas isotrópicos e anisotrópicos. Ele também apresenta o estudo de alguns parâmetros do método *multigrid*. O estudo preliminar está limitado a alguns tamanhos de problemas e razões de aspecto, entretanto um estudo mais detalhado será apresentado no capítulo 6.

O presente capítulo está dividido da seguinte forma: A primeira seção apresenta uma introdução a problemas anisotrópicos e define os algoritmos de engrossamento utilizados neste trabalho. A seção 5.2 apresenta os resultados numéricos obtidos: estudo do número de iterações internas ( $v$ ), número de níveis ( $L$ ), comparação entre os algoritmos de engrossamento, número de elementos ( $E$ ) e a análise de complexidade. As últimas seções apresentam a conclusão do capítulo e os parâmetros com melhor desempenho médio.

### 5.1 INTRODUÇÃO

Para as três equações em estudo consideram-se malhas anisotrópicas (BRIGGS et al., 2000; DENDY et al., 1989), onde a razão de aspecto  $Q$  é distinta da unidade, ou seja, anisotropia devido aos elementos da malha serem altamente alongados (anisotropia geométrica). A Fig. 5.1 apresenta um exemplo de malha isotrópica e anisotrópica com  $Q = 2$ .

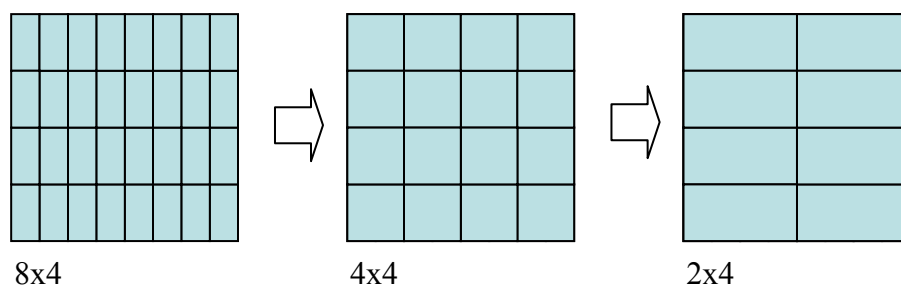
Em malhas anisotrópicas, quanto mais distinta da unidade é a razão de aspecto, mais o desempenho do método *multigrid* se deteriora (WESSELING e OOSTERLEE, 2001), com isso a taxa de convergência piora, podendo até mesmo ocorrer divergência (LARSSON et al., 2005). Este tipo de anisotropia é muito comum em problemas práticos de engenharia, como em problemas de camada limite, onde a razão de aspecto  $Q$  pode ser da ordem de  $10^3$ ,  $10^4$  ou mais (WESSELING e OOSTERLEE, 2001).



**Figura 5.1: Exemplos de malhas: a) isotrópica e b) anisotrópica com  $Q = 2$ .**

O objetivo deste capítulo é verificar a influência da razão de aspecto no tempo de CPU e a influência dos métodos de engrossamento para problemas anisotrópicos. Foi feita uma comparação entre o tempo de CPU com o uso de alguns algoritmos de engrossamento. Os algoritmos utilizados são: engrossamento padrão (EP); semi-engrossamento (SE); semi-engrossamento seguido de engrossamento padrão (SE-EP); engrossamento padrão seguido de semi-engrossamento (EP-SE) descritos na seção 2.4.3. Também é proposto um novo algoritmo de engrossamento para problemas anisotrópicos, o algoritmo de semi-engrossamento completo (SEC).

O semi-engrossamento completo (SEC) realiza o engrossamento em apenas uma das direções coordenadas como o algoritmo SE, porém o engrossamento é realizado até a malha mais grossa possível. Este algoritmo para o semi-engrossamento em  $x$ , está exemplificado na Fig. 5.2.



**Figura 5.2: Algoritmo semi-engrossamento completo (SEC).**

O algoritmo proposto neste capítulo, SEC, confronta o algoritmo SE proposto por Mulder (1989). O algoritmo SE de Mulder aplica o semi-engrossamento até as malhas

tornarem-se isotrópicas, nem sempre atingindo a malha mais grossa possível. O algoritmo SEC realiza o engrossamento até a malha mais grossa possível, portanto a malha anisotrópica torna-se isotrópica, voltando a ser anisotrópica conforme pode ser visto na Fig. 5.2. Deste modo podem-se considerar dois elementos a serem considerados em sua formulação:

- 1) Realizar o engrossamento até a malha mais grossa possível mesmo que a malha volte a ser anisotrópica.
- 2) Utilizar um menor número de malhas, na qual a malha mais grossa é uma malha isotrópica, (isto é, o semi-engrossamento de Mulder (1989)).

No primeiro caso considera que o método *multigrid* converge rapidamente ao utilizar-se o número máximo de níveis (PINTO et al., 2005). Já o segundo não se preocupa em utilizar o número máximo de níveis, mas o número necessário para que a malha torne-se isotrópica. Neste caso, continuar o engrossamento faz com que a malha volte a ser anisotrópica e a convergência do método *multigrid* se deteriora em problemas anisotrópicos (WESSELING e OOSTERLEE, 2001).

## 5.2 RESULTADOS NUMÉRICOS

Os algoritmos foram implementados na linguagem Fortran 2003, versão 9.1 da Intel com precisão dupla. As simulações foram realizadas num microcomputador com processador Intel Core 2 Duo de 2.66 GHz, 8 GB de RAM e sistema operacional Windows xp 64 bits.

Cerca de 2.000 simulações foram realizadas com as seguintes variantes: equações (Laplace senoidal, Laplace linear e Poisson); algoritmos (EP, SE, SEC, EP-SE, SE-EP); número de elementos (4.096, 65.536, 262.144 e 1.048.576); razões de aspecto ( $Q = 1/64, 1/16, 1/4, 1, 4, 16, 64$ ) e número de níveis ( $L_{máximo}, L_{máximo} - 1, L_{máximo} - 2, L_{máximo} - 3, L_{máximo} - 4$ ).

O número de iterações internas variou entre 1 a 3.000 com intervalos entre 1, 10, 100 ou 200, dependendo do tempo de CPU obtido em cada simulação e da variação entre o tempo obtido entre duas simulações consecutivas.

Nesta seção, adota-se o número máximo possível de níveis, exceto no momento onde é feita a análise do número de níveis. Os parâmetros considerados para a análise da anisotropia são:  $h_x \neq h_y$ ,  $N_x \neq N_y$  e  $C_x = C_y = 1$ . São apresentados a seguir somente os resultados mais representativos do estudo realizado.

### 5.2.1 Estudo do número de iterações internas ( $\nu$ )

A Tab. 5.1 mostra as dimensões das malhas finas ( $N = N_x \times N_y$ ) e as respectivas razões de aspecto ( $Q$ ) utilizadas para determinar o número ótimo de iterações internas ( $\nu_{\text{ótimo}}$ ), primeiramente para a equação de Laplace senoidal e algoritmo SE-EP. Considerou-se malhas com o mesmo número de elementos, por exemplo, a malha  $N_x \times N_y = 4097 \times 65$  possui  $4096 \times 64 = 262.144$  elementos; a malha  $N_x \times N_y = 2049 \times 129$  possui  $2048 \times 128 = 262.144$  elementos. O processo utilizado para a geração das malhas está apresentado no apêndice F1. Foi utilizado o mesmo número de elementos para que fosse possível comparar a variação do tempo de CPU em função da razão de aspecto. Não foi possível utilizar o mesmo número de nós, pois quando este número é ímpar, não é possível escrevê-lo na forma  $2^a$ , onde  $a \in \mathbb{Z}^+$ , forma necessária para a determinação de malhas com mesmo número de nós. O mesmo processo se repete para as equações de Laplace linear e Poisson, outras razões de aspecto e tamanhos de problemas.

**Tabela 5.1: Malhas utilizadas para determinar  $\nu_{\text{ótimo}}$  para o algoritmo SE-EP para a equação de Laplace senoidal.**

$Q$	Malhas		
1/64	4097x65	16385x257	65537x1025
1/16	2049x129	8193x513	32769x2049
1/4	1025x257	4097x1025	16385x4097
4	257x1025	1025x4097	4097x16385
16	129x2049	513x8193	2049x32769
64	65x4097	257x16385	1025x65537

A Fig. 5.3 apresenta o tempo de CPU em função do número de iterações internas para a equação de Laplace senoidal, algoritmo SE-EP, razões de aspecto menores que a unidade ( $Q = 1/64$ ,  $1/16$  e  $1/4$ ) em um problema com 262.144 elementos. Verifica-se que para as razões de aspecto  $Q = 1/64$  e  $Q = 1/16$ ,  $\nu_{\text{ótimo}} = 2$ . Para  $Q = 1/4$  tem-se  $\nu_{\text{ótimo}} = 3$ . Ao utilizar  $\nu = 2$ , para a razão de aspecto  $Q = 1/4$  o acréscimo no tempo de CPU será de 3%.

A Fig. 5.4 apresenta o tempo de CPU em função do número de iterações internas para a equação de Laplace senoidal, algoritmo SE-EP, razões de aspecto maiores que a unidade

( $Q = 4, 16$  e  $64$ ) em um problema com 262.144 elementos. Verifica-se na Fig. 5.4 que para as razões de aspecto maiores que a unidade  $\nu_{\text{ótimo}} = 4$ .

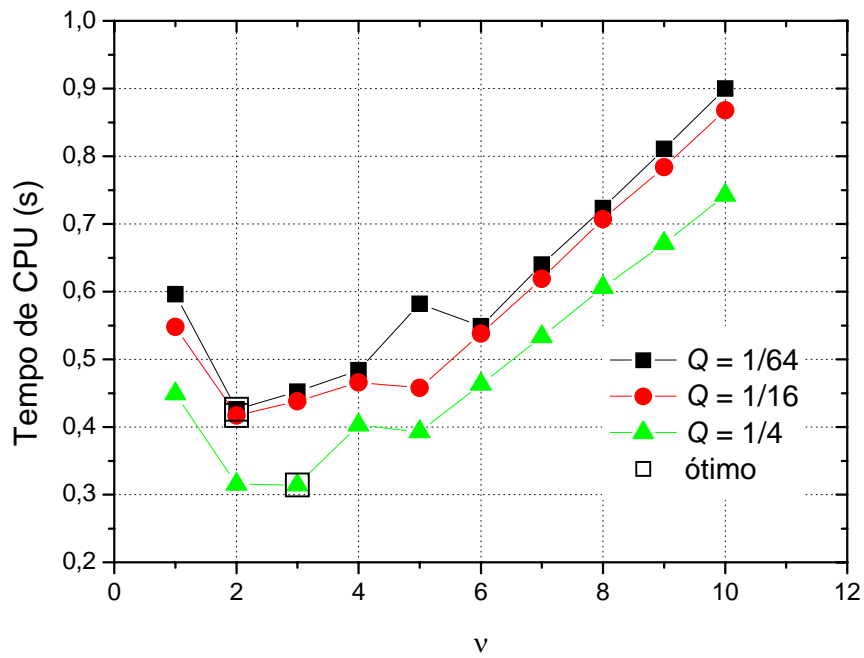


Figura 5.3: Tempo de CPU versus  $\nu$  para  $E = 262.144$ , SE-EP, equação de Laplace senoidal e  $Q < 1$ .

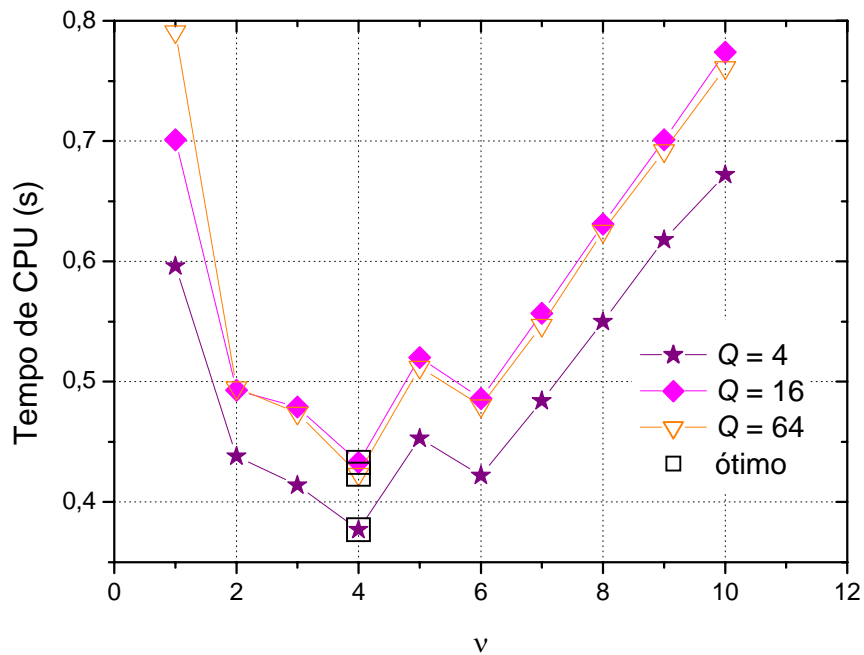


Figura 5.4: Tempo de CPU versus  $\nu$  para  $E = 262.144$ , SE-EP, equação de Laplace senoidal e  $Q > 1$ .

Observa-se nas Figs. 5.3 e 5.4 um valor diferenciado de  $\nu_{\text{ótimo}}$  para  $Q < 1$  e  $Q > 1$ . Isto é devido ao comportamento da equação de Laplace senoidal ser mais oscilatória em uma das direções. Os gráficos para as equações de Laplace linear e Poisson, outros algoritmos e tamanhos de problemas encontram-se no Apêndice D. A Tab. 5.2 apresenta o número ótimo de iterações internas ( $\nu_{\text{ótimo}}$ ) para os dados da Tab. 5.1. Observa-se que para o algoritmo SE-EP o tamanho do problema não influencia o  $\nu_{\text{ótimo}}$ . Para a razão de aspecto  $Q = 1$ , não se aplica o algoritmo SE-EP, pois sendo uma malha isotrópica não é necessário utilizar o SE.

**Tabela 5.2:  $\nu_{\text{ótimo}}$  para o algoritmo SE-EP e equação de Laplace senoidal.**

$E$	$Q$	1/64	1/16	1/4	4	16	64
262.144		2	2	2 e 3	4	4	4
4.194.304		2	2	2	4	4	4
67.108.864		2	2	2	4	4	4

Para a análise do  $\nu_{\text{ótimo}}$  nos algoritmos EP, SE, SEC e EP-SE foram utilizadas malhas com menor número de nós. Isto foi necessário devido ao fato destes algoritmos não apresentarem boa convergência para problemas anisotrópicos, o tempo de CPU ser bem maior comparado ao SE-EP e serem necessárias mais simulações para a determinação do  $\nu_{\text{ótimo}}$ . As malhas utilizadas estão descritas na Tab. 5.3.

**Tabela 5.3: Malhas utilizadas para determinar  $\nu_{\text{ótimo}}$  para os algoritmos: EP, SE, SEC e EP-SE e equação de Laplace senoidal.**

$Q$	Malhas	
1/64	2049x33	4097x65
1/16	1025x65	2049x129
1/4	513x129	1025x257
1	257x257	513x513
4	129x513	257x1025
16	65x1025	129x2049
64	33x1025	65x4097

Para os algoritmos EP, SE, SEC, e EP-SE o  $\nu_{\text{ótimo}}$  variou significativamente em relação à razão de aspecto, além de apresentar valores muito altos. Procurou-se contemplar várias iterações internas para a determinação do ótimo. Variou-se o  $\nu$  entre 1 a 3.000, com intervalos de 1, 10, 100 ou 200 de acordo com o tempo de CPU obtido em cada simulação e

também considerando a variação de tempo entre duas simulações consecutivas. Em alguns algoritmos verificou-se que o tempo de CPU é bem próximo para diversas iterações internas.

As Figs. 5.5 à 5.8 apresentam o tempo de CPU em função do número de iterações internas para a equação de Laplace senoidal e para os diversos algoritmos. Apresenta-se um gráfico para cada algoritmo de engrossamento: EP, SE, SEC e EP-SE, razão de aspecto  $Q = 16$  e um problema com 262.144 elementos. Os gráficos para as equações de Laplace linear e Poisson, outros algoritmos e tamanhos de problemas encontram-se no Apêndice D.

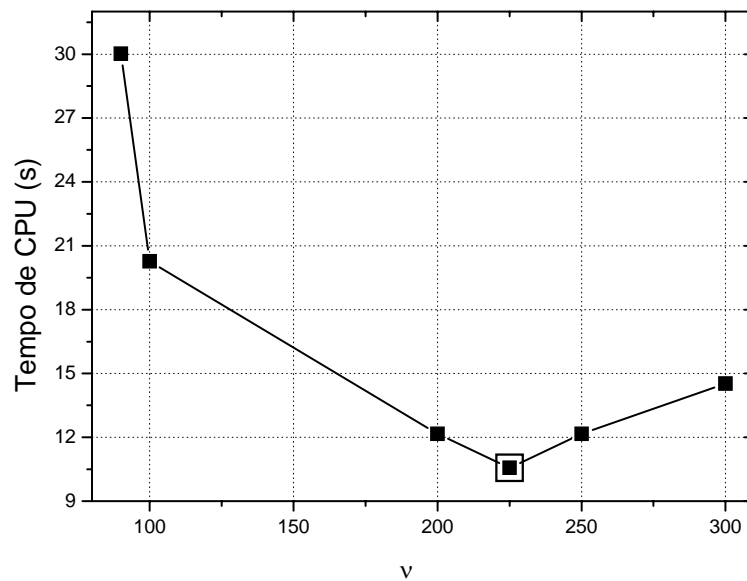


Figura 5.5: Tempo de CPU *versus*  $v$  para  $Q = 16$ ,  $E = 262.144$  e EP.

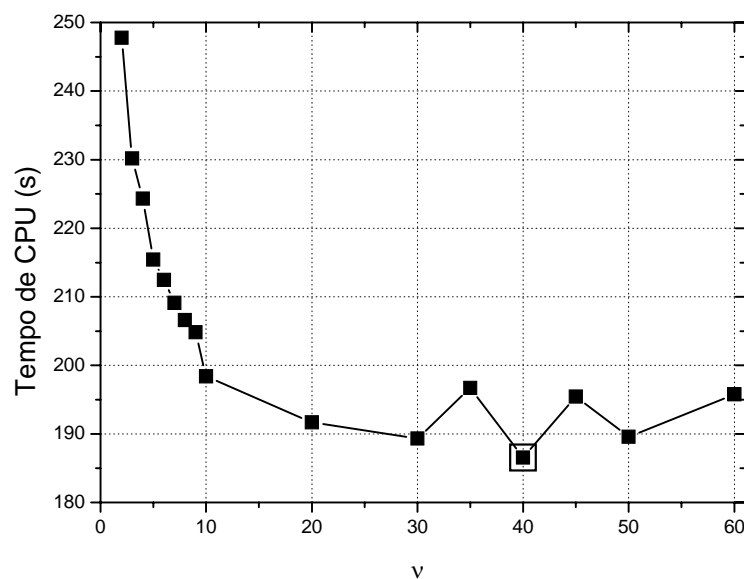


Figura 5.6: Tempo de CPU *versus*  $v$  para  $Q = 16$ ,  $E = 262.144$  e SE.

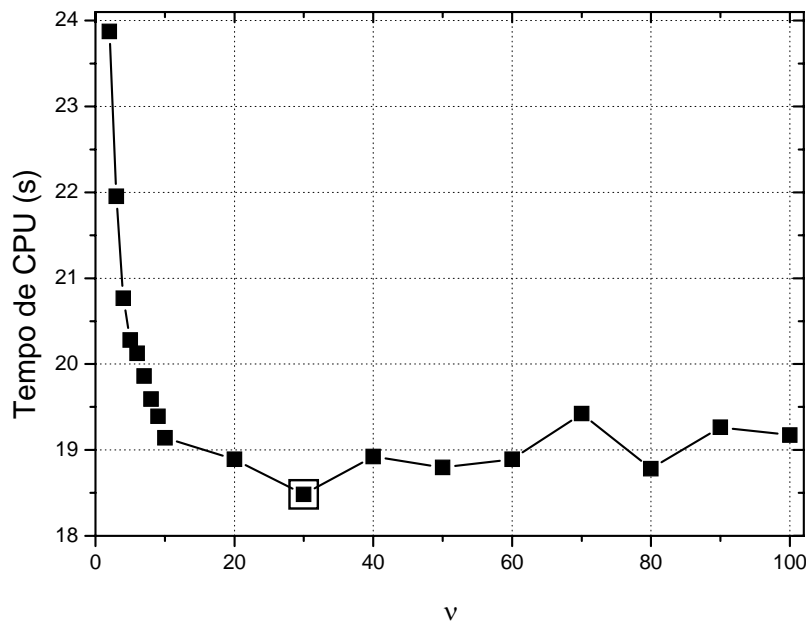


Figura 5.7: Tempo de CPU *versus*  $\nu$  para  $Q = 16$ ,  $E = 262.144$  e SEC.

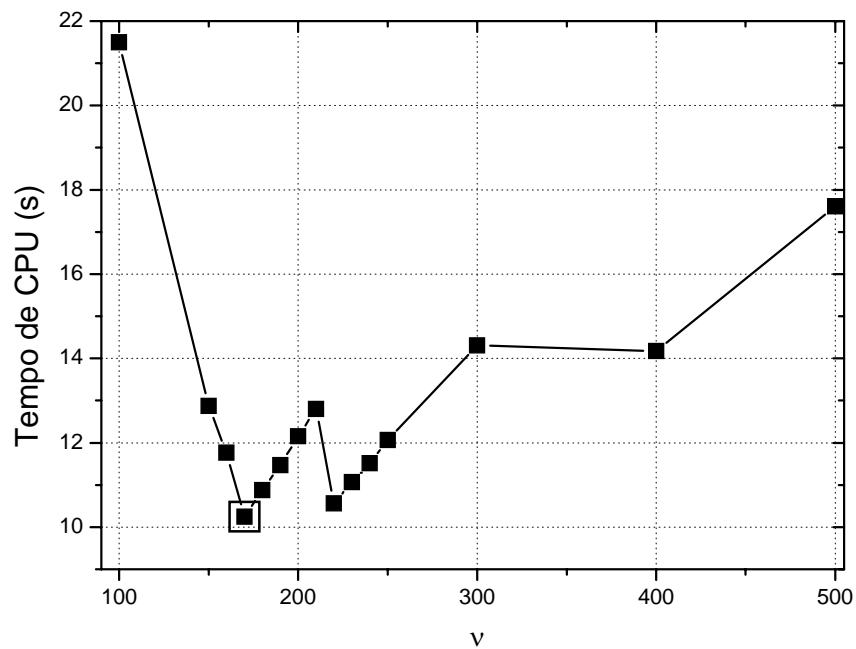


Figura 5.8: Tempo de CPU *versus*  $\nu$  para  $Q = 16$ ,  $E = 262.144$  e EP-SE.

Verifica-se nas Figs. 5.5 à 5.8 que são necessárias muitas iterações internas para se obter um menor tempo de CPU. Os gráficos apresentados não mostram todos os pontos simulados, mas sim o intervalo no qual se encontra o  $\nu_{\text{ótimo}}$ . Verifica-se também que poucas iterações internas resultam em um tempo de CPU muito alto ou até mesmo a divergência.



Com isto pode-se concluir que os algoritmos EP, SE, SEC e EP-SE não são muito estáveis para razões de aspecto diferentes da unidade, no que diz respeito ao valor do  $\nu_{\text{ótimo}}$ .

Através da análise das tabelas e gráficos determinou-se um único  $\nu_{\text{ótimo}}$  ou um intervalo contendo o  $\nu_{\text{ótimo}}$ . Por exemplo, para a equação de Laplace senoidal e algoritmo SE, determinou-se um intervalo contendo o  $\nu_{\text{ótimo}}$  para as malhas de 65.536 e 262.144 elementos. Através da intersecção dos intervalos, obtida para cada tamanho de problema, determinou-se um intervalo no qual se encontram os melhores desempenhos médios para o  $\nu$ . Os intervalos estão apresentados na Tab. 5.4.

**Tabela 5.4:  $\nu$  com melhor desempenho médio para cada razão de aspecto, algoritmo SE e equação de Laplace senoidal.**

$Q$	$E$	65.536	262.144	$\nu$
1/64		$50 \leq \nu \leq 60$	$\cong 60$	$55 \leq \nu \leq 65$
1/16		$10 \leq \nu \leq 150$	$96 \leq \nu \leq 99$	$96 \leq \nu \leq 99$
1/4		$20 \leq \nu \leq 190$	$\cong 120$	120
4		$40 \leq \nu \leq 200$	3	$3^3$
16		$20 \leq \nu \leq 200$	$20 \leq \nu \leq 40$	$20 \leq \nu \leq 40$
64		$10 \leq \nu \leq 140$	$10 \leq \nu \leq 70$	$10 \leq \nu \leq 70$

Para cada intervalo apresentado na Tab. 5.4 determinou-se o tempo médio de CPU ( $\bar{t}_{CPU}$ ), o desvio padrão amostral ( $s$ ) e o coeficiente de variação ( $CV$ ). Estes valores encontram-se na Tab. 5.5. O apêndice B apresenta as definições de desvio padrão e coeficiente de variação.

**Tabela 5.5: Parâmetros para o intervalo de  $\nu$  com melhor desempenho médio para o algoritmo de SE.**

$Q$	$E$	65.536			262.144		
		$\bar{t}_{CPU}$ (segundos)	$s$ (segundos)	$CV$	$\bar{t}_{CPU}$ (segundos)	$s$ (segundos)	$CV$
1/64		5,117	0	0	79,672	0	0
1/16		18,62	0,165	0,89%	18,844	0	0
1/4		58,781	0	0	874,984	0	0
4		47,87	0,173	0,36%	574,891	0	0
16		12,699	0,177	1,39%	189,19	2,566	1,36%
64		2,947	0,07	2,52%	43,31	0,844	1,95%

<sup>3</sup> Não houve intersecção entre os intervalos. Ao se utilizar  $\nu = 3$ , o acréscimo no tempo de CPU para problemas de 65.536 elementos será de 10,7%, equivalente a 5,47 segundos.

Verifica-se na Tab. 5.5 que os coeficientes de variação ( $CV$ ) são pequenos, menores que 3%, significando que a variabilidade do tempo de CPU em relação à média é pequena. Portanto, ao utilizar qualquer  $\nu$  pertencente ao intervalo da Tab. 5.4, o acréscimo no tempo de CPU será pequeno. Para  $Q = 4$ , não foi possível determinar um intervalo ótimo para os dois tamanhos de problema. Pode-se utilizar  $\nu = 3$ , com isso o acréscimo no tempo de CPU para problemas de 65.536 elementos será de 10,7% equivalente a 5,47 segundos. Se utilizar-se o intervalo  $40 \leq \nu \leq 200$  o acréscimo no tempo de CPU será bem maior: 26%. As tabelas para os demais algoritmos encontram-se no Apêndice D. Os valores de  $\nu$  com melhor desempenho médio para cada algoritmo e razão de aspecto encontram-se na Tab. 5.6.

**Tabela 5.6:  $\nu$  com melhor desempenho médio para cada razão de aspecto e equação de Laplace senoidal.**

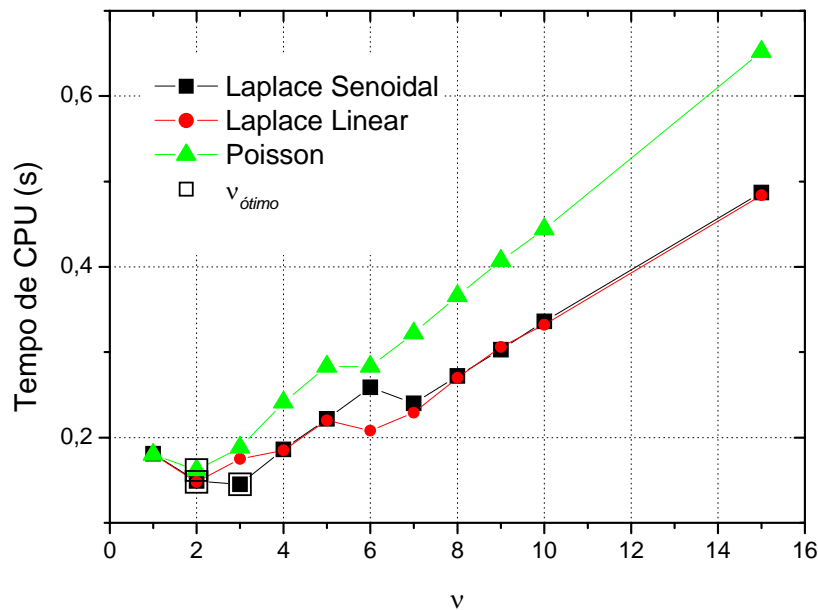
$Q$	Algoritmo	EP	SE	SEC	EP-SE	SE-EP
	1/64	44	$\cong 60$	$70 \leq \nu \leq 90$	5000	2
	1/16	221	$96 \leq \nu \leq 99$	$30 \leq \nu \leq 60$	200	2
	1/4	25	$\cong 120$	$40 \leq \nu \leq 60$	25	2
	1	2	-	-	-	-
	4	14	3	$30 \leq \nu \leq 80$	13	4
	16	$200 \leq \nu \leq 204$	$20 \leq \nu \leq 40$	$20 \leq \nu \leq 80$	170	4
	64	10	$10 \leq \nu \leq 70$	$20 \leq \nu \leq 50$	2500	4

Agora vamos tratar dos problemas Laplace linear e Poisson. As malhas e razões de aspecto a serem utilizadas para as equações de Laplace linear e Poisson foram escolhidas com base nos resultados obtidos para a equação de Laplace senoidal. Para o algoritmo SE-EP verificou-se que o  $\nu_{\text{ótimo}}$  é o mesmo para qualquer tamanho de problema. Em relação à razão de aspecto, o  $\nu_{\text{ótimo}}$  varia em função desta ser menor, igual ou maior que a unidade.

Para os algoritmos EP, SE, SEC e EP-SE e equação de Laplace senoidal verificou-se que o  $\nu_{\text{ótimo}}$  não teve um comportamento padrão, sendo fortemente afetado pela razão de aspecto e pelo algoritmo utilizado. Para comprovar este mesmo comportamento para as equações de Laplace linear e Poisson optou-se pelo uso de malhas com 262.144 elementos e razões de aspecto 1/16, 1 e 16.

A Fig. 5.9 apresenta uma comparação do número de iterações internas para o algoritmo EP nas equações de Laplace senoidal, Laplace linear e Poisson. Analisou-se o caso isotrópico, ou seja,  $Q = 1$  em um problema com 262.144 elementos. Verifica-se que para as equações de Laplace linear e Poisson  $\nu_{\text{ótimo}} = 2$ . Para a equação de Laplace senoidal tem-se

$\nu_{\text{ótimo}} = 3$ . Neste caso ao utilizar  $\nu = 2$  o acréscimo no tempo de CPU será de aproximadamente 0,004 segundo (2,8%), o qual pode ser considerado como uma imprecisão da função CPU\_TIME. Portanto, pode-se utilizar  $\nu = 2$ , para problemas isotrópicos resolvidos com o algoritmo EP e equações de Laplace senoidal, Laplace linear e Poisson.



**Figura 5.9:** Tempo de CPU *versus*  $\nu$  para  $Q = 1$ ,  $E = 262.144$ , EP e *solver* GS-LEX.

A Tab. 5.7 mostra o  $\nu$  com melhor desempenho médio para as equações: Laplace linear (LL) e Poisson (Po), para os algoritmos de engrossamento: EP, SE, SEC, EP-SE, SE-EP e razões de aspecto: 1/16, 1 e 16. Para a razão de aspecto  $Q = 1$  (problema isotrópico) utilizou-se somente o EP, pois todos os outros algoritmos recaem no EP quando a razão de aspecto é igual a 1. Verifica-se que os valores de  $\nu$  com melhor desempenho médio são muito altos, exceto para problemas isotrópicos. Verifica-se também que para problemas isotrópicos resolvidos através do algoritmo EP,  $\nu_{\text{ótimo}} = 2$  para as equações de Laplace linear e Poisson. Vale lembrar que o  $\nu_{\text{ótimo}}$  para a Equação de Laplace senoidal também é igual a dois, conforme Tab. 5.6. Para os problemas anisotrópicos o algoritmo que apresentou melhores resultados foi o SE-EP. Para este algoritmo o  $\nu_{\text{ótimo}} = 4$  para a equação de Laplace linear. Para a equação de Poisson  $\nu_{\text{ótimo}} = 3$ .

**Tabela 5.7:  $\nu$  com melhor desempenho médio para cada razão de aspecto para as equações de Laplace linear (LL) e Poisson (Po).**

$Q$	EP		SE		SEC		EP-SE		SE-EP	
	LL	Po	LL	Po	LL	Po	LL	Po	LL	Po
1/16	250	200	250	150	250	200	250	200	4	2
1	2	2	-	-	-	-	-	-	-	-
16	250	200	150	300	250	200	250	200	4	3

As Figs. 5.10 a 5.13 fazem uma comparação do número ótimo de iterações internas para as equações de Laplace senoidal, Laplace linear e Poisson com o uso do algoritmo SE-EP, respectivamente. Foram consideradas quatro razões de aspecto:  $Q = 1/64$  (Fig. 5.10),  $Q = 1/16$  (Fig. 5.11),  $Q = 16$  (Fig. 5.12) e  $Q = 64$  (Fig. 5.13). Verifica-se nas Fig. 5.10 e 5.11 que para a equação de Laplace senoidal,  $\nu_{\text{ótimo}} = 2$  para as razões de aspecto  $Q = 1/64$  e  $Q = 1/16$ . Para a equação de Laplace linear,  $\nu_{\text{ótimo}} = 4$  para as razões de aspecto  $Q = 1/64$  e  $Q = 1/16$ . Para a equação de Poisson e razão de aspecto  $Q = 1/64$ , o  $\nu_{\text{ótimo}}$  ocorre em dois pontos  $\nu = 2$  e  $\nu = 4$ . Para a razão  $Q = 1/16$ ,  $\nu_{\text{ótimo}} = 3$ .

Na Fig. 5.12 e 5.13 verifica-se que para as três equações em estudo e razões de aspecto 16 e 64 o  $\nu_{\text{ótimo}} = 4$ , exceto para a equação de Poisson com  $Q = 16$ .

Fazendo uma comparação entre os resultados obtidos para as duas razões de aspecto apresentadas verifica-se que para todas as equações em estudo e razão de aspecto  $Q = 64$  tem-se  $\nu_{\text{ótimo}} = 4$ . Para a razão de aspecto  $Q = 1/64$  o  $\nu_{\text{ótimo}}$  varia de acordo com a equação. Para a equação de Laplace linear o  $\nu_{\text{ótimo}}$  para as razões de aspecto 1/64 e 64 são iguais. Isto aconteceu devido à equação possuir a mesma oscilação em ambas as direções  $x$  e  $y$ . O gráfico destas funções, no qual pode-se observar a diferença entre as oscilações de cada função encontram-se no Apêndice A. Portanto, não importa se a razão de aspecto é maior ou menor que a unidade, ela não influenciará no  $\nu_{\text{ótimo}}$ . As equações de Laplace senoidal e Poisson não possuem a mesma oscilação nas duas direções coordenadas, logo o  $\nu_{\text{ótimo}}$  varia com a razão de aspecto menor ou maior que a unidade.

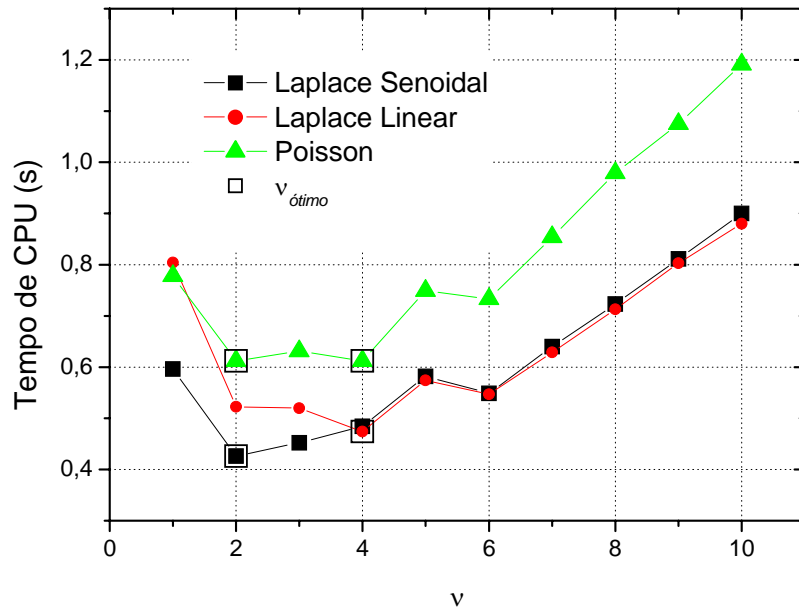


Figura 5.10: Tempo de CPU versus  $\nu$  para  $Q = 1/64$ ,  $E = 262.144$ , SE-EP.

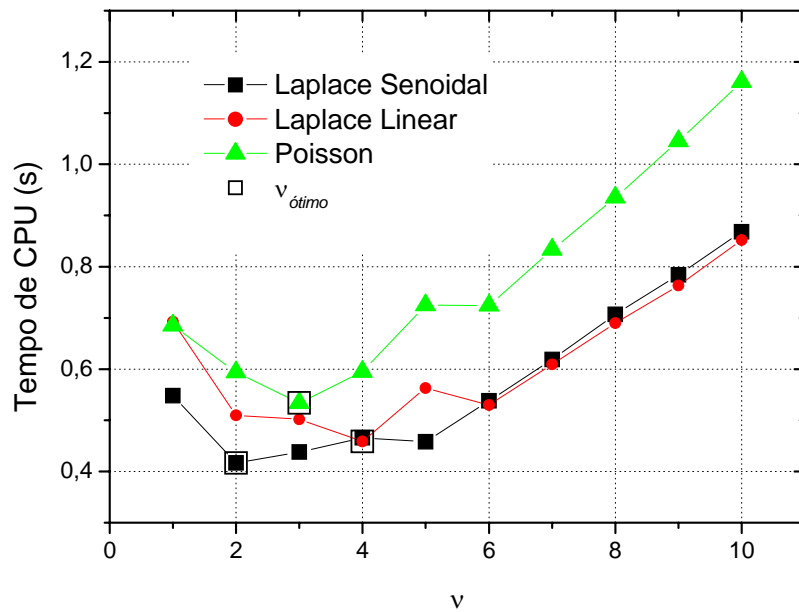


Figura 5.11: Tempo de CPU versus  $\nu$  para  $Q = 1/16$ ,  $E = 262.144$ , SE-EP.

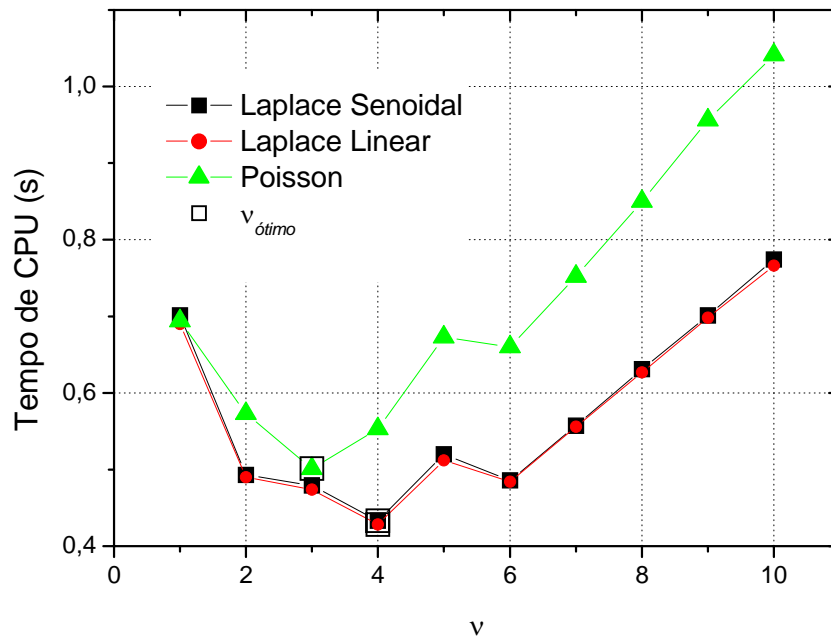


Figura 5.12: Tempo de CPU *versus*  $\nu$  para  $Q = 16$ ,  $E = 262.144$ , SE-EP.

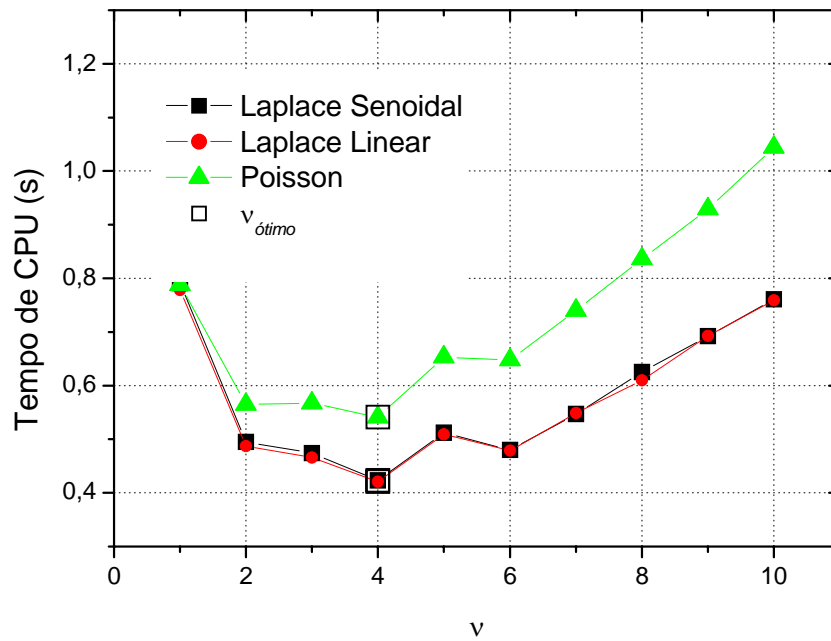


Figura 5.13: Tempo de CPU *versus*  $\nu$  para  $Q = 64$ ,  $E = 262.144$ , SE-EP.

## 5.2.2 Estudo do número de níveis ( $L$ )

Com o  $\nu_{\text{ótimo}}$  obtido para os problemas com 65.536 e 262.144 elementos e razões de aspecto 1/64, 1/16, 1/4, 4, 16, 64 verificou-se o número ótimo de níveis para a equação de Laplace senoidal.

Aqui  $L_{\text{máximo}}$  representa o número máximo possível de níveis. Por exemplo, uma malha com  $E = 1.024$  elementos e usando  $r = 2$  possui  $L = 10$ ,  $L_{\text{máximo}} = L$ , 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  $\Omega_h$ ), pode-se usar no máximo mais 9 níveis adicionais, além de  $E = 1.024$ , que são: 512 (que é o nível 2, o nível da malha imediatamente mais grossa, ou seja,  $\Omega_{2h}$ ), 256 (nível 3, ou seja,  $\Omega_{4h}$ ), 128 (nível 4, ou seja,  $\Omega_{8h}$ ), 64 (nível 5, ou seja,  $\Omega_{16h}$ ), 32 (nível 6,  $\Omega_{32h}$ ), 16 (nível 7,  $\Omega_{64h}$ ), 8 (nível 8,  $\Omega_{128h}$ ), 4 (nível 9,  $\Omega_{256h}$ ) e 2 (que é o nível 10, o nível da malha mais grossa que são dois elementos em cada direção com um ponto interno, ou seja,  $\Omega_{512h}$ ).

Observou-se, que o número de níveis utilizado em diferentes direções podem mudar o algoritmo de uma categoria para outra categoria. Por exemplo, para um problema de 2048x32 elementos e algoritmo EP-SE os números máximos de níveis são  $L_x = 11$  e  $L_y = 5$ . Ao diminuirmos até o  $L_{\text{máximo}} - 4$  teremos 7 níveis na direção  $x$  e somente um nível na direção  $y$ , ou seja, o algoritmo neste caso é o SE e não mais o EP-SE. O objetivo destas simulações é determinar o número ótimo de níveis para cada um dos algoritmos serão considerados somente os número de níveis que não mudem de uma categoria para outra. A Tab. 5.8 apresenta as malhas utilizadas para a análise do  $L_{\text{ótimo}}$ . Observou-se também que o  $L_{\text{máximo}}$  varia em função do algoritmo. Por exemplo, em uma malha com 262.144 elementos e razão de aspecto 1/64 tem-se para o algoritmo de EP,  $L_{\text{máximo}} = 12$ , tanto na direção  $x$  como na direção  $y$ . Para estes mesmos dados o número máximo de níveis no algoritmo SE é de 7 níveis na direção  $x$  e 1 um nível na direção  $y$ , pois o algoritmo SE engrossa em apenas uma direção e somente até a malha torna-se isotrópica. Já para o algoritmo SEC,  $L_{\text{máximo}} = 12$  na direção  $x$  e 1 na direção  $y$ . Os algoritmos SE-EP e EP-SE têm no máximo 12 níveis na direção  $x$  e 6 na direção  $y$ .

Tabela 5.8: Malhas utilizadas para determinar  $L_{\text{ótimo}}$ .

$Q$	Malhas	
1/64	2049x33	4097x65
1/16	1025x65	2049x129
1/4	513x129	1025x257
1	257x257	513x513
4	129x513	257x1025
16	65x1025	129x2049
64	33x2049	65x4097

As Figs. 5.14 a 5.18 mostram a influência do número de níveis para a equação de Laplace senoidal com 262.144 elementos, razões de aspecto ( $Q = 1/64, 1/16, 1/4, 1, 4, 16$  e 64) e algoritmos (EP, SE, SEC, EP-SE e SE-EP). A razão de aspecto  $Q = 1$  só é apresentada na Fig. 5.14, pois para esta razão de aspecto utilizou-se somente o algoritmo EP. Em malhas isotrópicas, ou seja, com  $Q = 1$  não é necessário utilizar semi-engrossamento, já que o objetivo do SE é tornar a malha isotrópica.

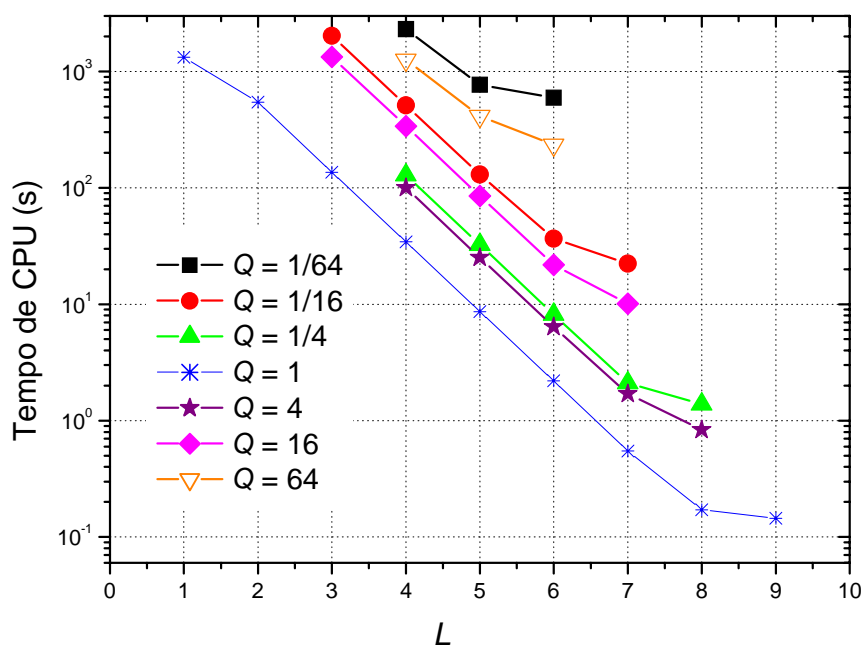


Figura 5.14: Tempo de CPU *versus* número de níveis ( $L$ ) para  $E = 262.144$ , EP e equação de Laplace senoidal.



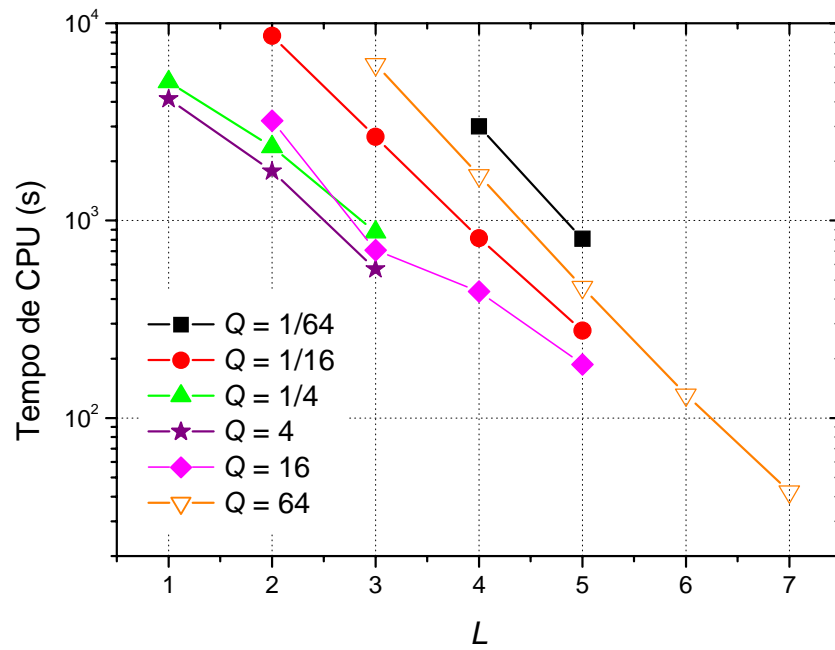


Figura 5.15: Tempo de CPU *versus* número de níveis para  $E = 262.144$ , SE e equação de Laplace senoidal.

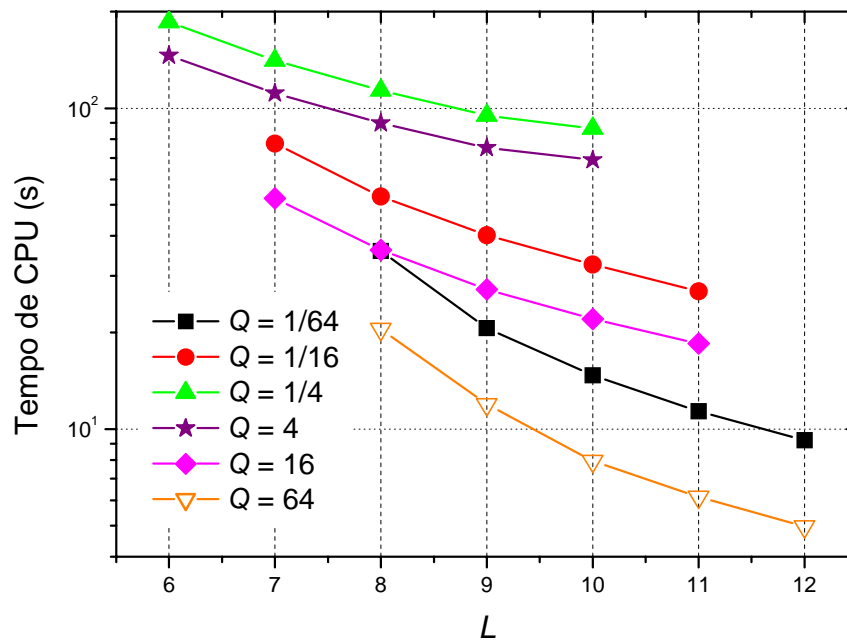


Figura 5.16: Tempo de CPU *versus* número de níveis para  $E = 262.144$ , SEC e equação de Laplace senoidal.

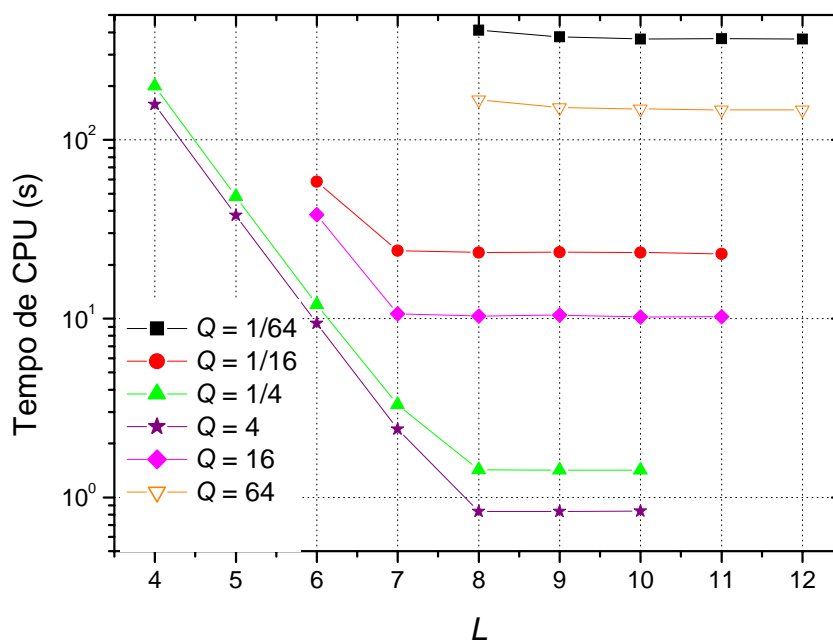


Figura 5.17: Tempo de CPU *versus* número de níveis para  $E = 262.144$ , EP-SE e equação de Laplace senoidal.

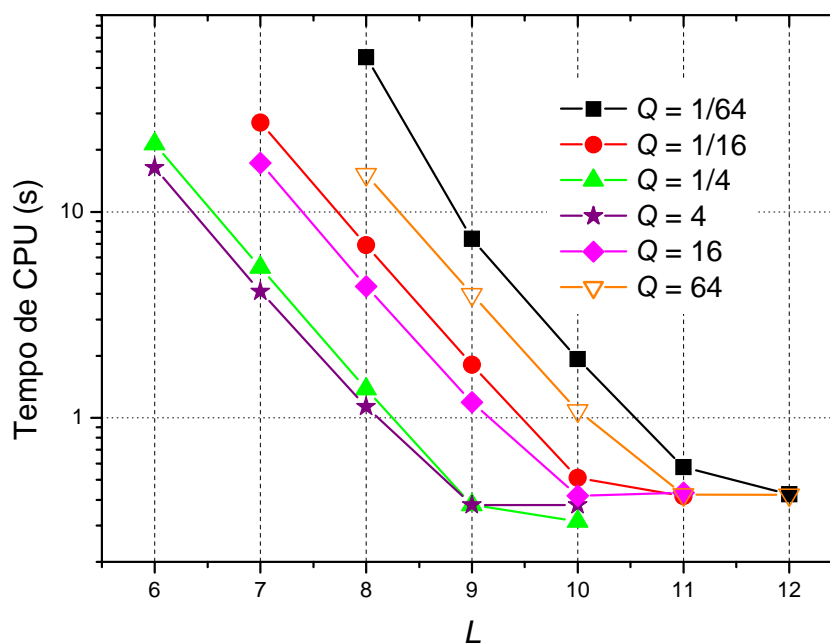


Figura 5.18: Tempo de CPU *versus* número de níveis para  $E = 262.144$ , algoritmo SE-EP e equação de Laplace senoidal.

Verifica-se nas Figs. 5.14 à 5.18 que os menores tempos de CPU ocorrem ao se utilizar os níveis:  $L_{\text{máximo}}$ ,  $L_{\text{máximo}} - 1$  e  $L_{\text{máximo}} - 2$ . Observa-se também que tempo de CPU aumenta à medida que se diminui o número de níveis. Pode-se verificar que quanto maior a

anisotropia, maior a influência do número de níveis sobre o tempo de CPU. Esta análise também já foi realizada para outras equações nos trabalhos de Pinto et al. (2005); Pinto e Marchi (2006) e Oliveira et al. (2006). Ambos concluíram que utilizar o número máximo de níveis reduz o tempo de CPU. Portanto, para as equações de Laplace linear e Poisson optou-se por utilizar o número máximo de níveis.

### 5.2.3 Comparação entre os algoritmos de engrossamento

As Figs. 5.19 a 5.21 apresentam uma comparação entre os diversos algoritmos para as razões de aspecto  $Q = 1/64, 1/16, 1/4, 1, 4, 16$  e  $64$  para as equações de Laplace senoidal, Laplace linear e Poisson em uma malha com 262.144 elementos. Verifica-se nas figuras que, para problemas anisotrópicos o algoritmo SE-EP de Zhang (2002) apresenta o menor tempo de CPU entre os cinco algoritmos e para as razões de aspecto testadas. Pode-se notar que os algoritmos EP e EP-SE ficam mais lentos à medida que se aumenta a razão de aspecto. Os algoritmos SE e SEC são mais rápidos à medida que aumenta a razão de aspecto, mas nunca melhores que o SE-EP. Em resumo, para os mesmos parâmetros (número de iterações internas, razão de aspecto, número de níveis e tamanho do problema), dentre os cinco algoritmos estudados tem-se que o tempo de CPU do algoritmo SE-EP é menor que o tempo de CPU dos demais algoritmos (SE, SEC, EP-SE), exceto para problemas isotrópicos onde o algoritmo EP apresenta bons resultados.

Zhang (2002) já havia constatado que o algoritmo *partial semicoarsening* (aqui denominado de SE-EP) é mais rápido, se comparado ao EP para a equação de Poisson bidimensional envolvendo razões de aspecto menores que a unidade ( $1/2, 1/4, 1/8$  e  $1/16$ ) e problemas com no máximo 1.048.576 elementos. Para o algoritmo SE-EP, também pode-se verificar que a razão de aspecto não influencia significativamente o tempo de CPU, ao contrário dos demais algoritmos. Pinto (2006) também fez esta mesma análise para a equação de Laplace bidimensional envolvendo as razões de aspecto  $1/1.024, 1, 2, 16, 128, 1024$  e  $8192$ .

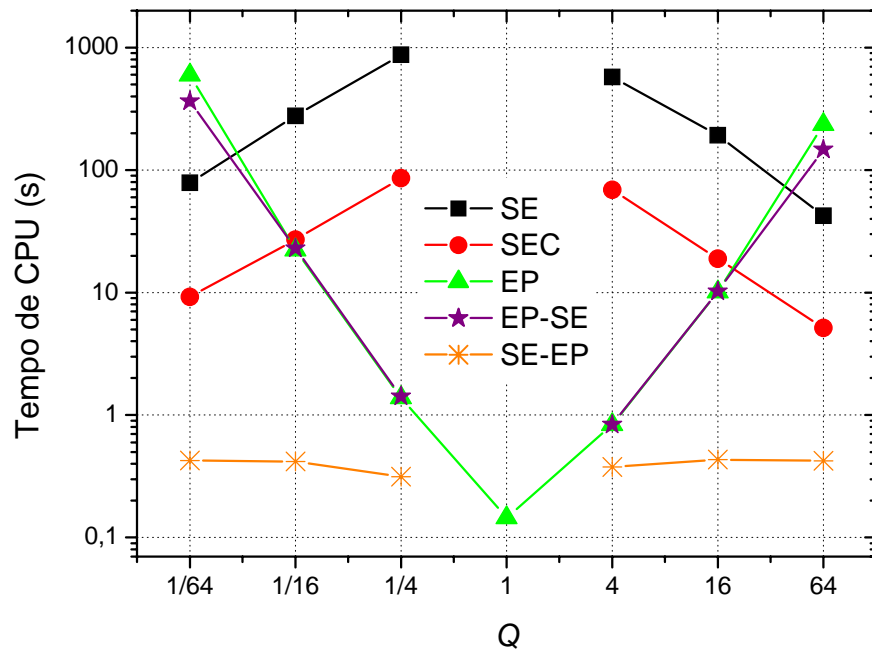


Figura 5.19: Tempo de CPU *versus* razão de aspecto para diversas razões de aspecto,  $E = 262.144$  e equação de Laplace senoidal.

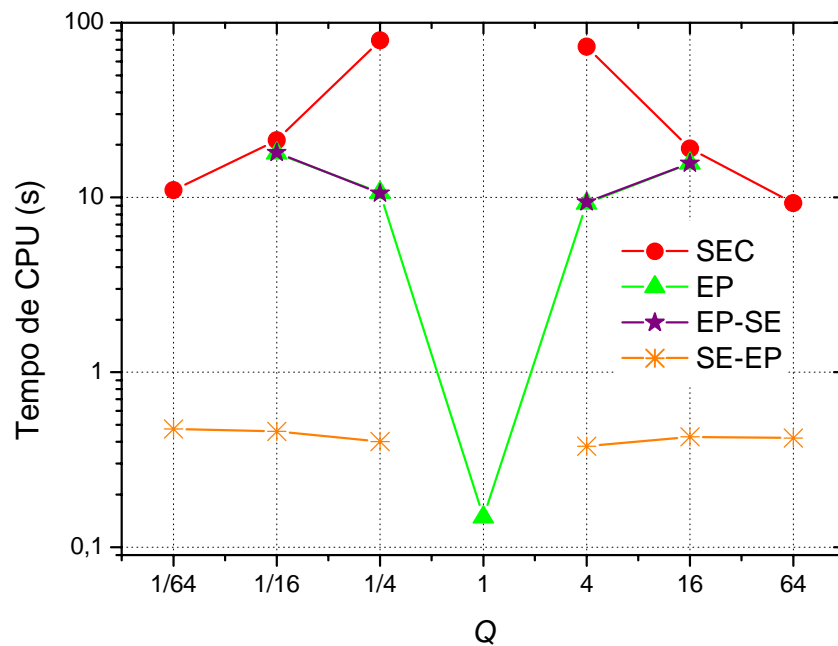


Figura 5.20: Tempo de CPU *versus* razão de aspecto para diversas razões de aspecto,  $E = 262.144$  e equação de Laplace linear.

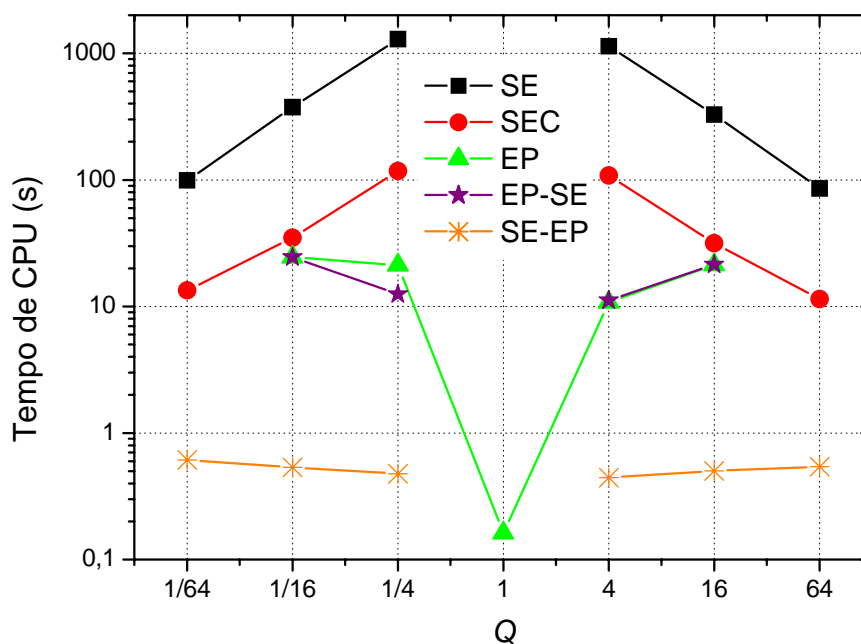


Figura 5.21: Tempo de CPU versus razão de aspecto para diversas razões de aspecto,  $E = 262.144$  e equação de Poisson.

#### 5.2.4 Influência do número de elementos ( $E$ )

Esta seção faz uma análise da influência do número de elementos sob o tempo de CPU para a equação de Laplace senoidal. A Tab. 5.9 apresenta as malhas utilizadas para este estudo.

Tabela 5.9: Malhas utilizadas para análise de  $E$ .

$Q$	Malhas				
1/64	513x9	2049x33	4097x65	8193x129	
1/16	257x17	1025x65	2049x129	4097x257	
1/4	129x33	513x129	1025x257	2049x513	
1	65x65	257x257	513x513	1025x1025	
4	33x129	129x513	257x1025	513x2049	
16	17x257	65x1025	129x2049	257x4097	
64	9x513	33x2049	65x4097	129x8193	

As Figs. 5.22 e 5.23 apresentam o tempo de CPU em função da razão de aspecto para os algoritmos EP, SE, SEC, SE-EP e EP-SE e a equação de Laplace senoidal para as razões de

aspecto  $Q = 64$  e  $Q = 1/64$ , respectivamente. Verifica-se que para todos os algoritmos e razões de aspecto em estudo o tempo de CPU aumenta à medida que aumenta o tamanho do problema. Nota-se também que para razões de aspectos distintas da unidade o algoritmo SE-EP teve o melhor desempenho para todos os valores  $E$ . Isto já era esperado, pois a ideia do algoritmo SE-EP é primeiramente tornar a malha isotrópica onde o *multigrid* com EP apresenta bons resultados. Os resultados obtidos para as equações de Laplace linear e Poisson foram similares e encontram-se no Apêndice D.

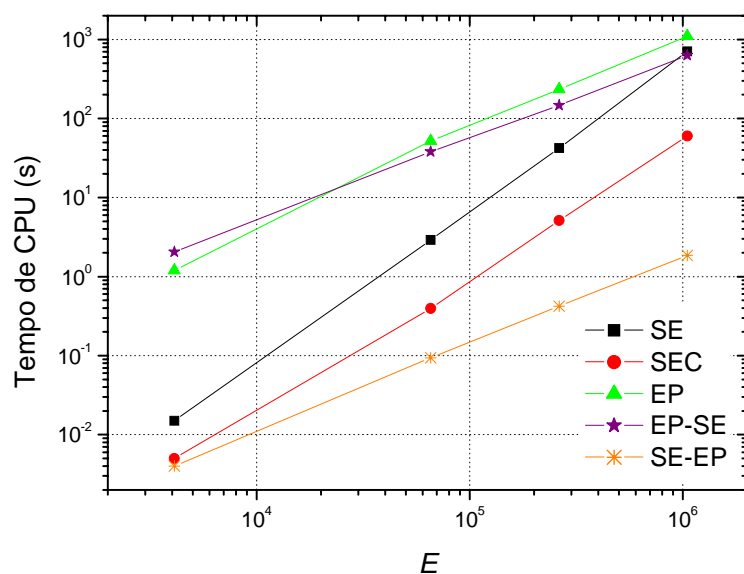


Figura 5.22: Tempo de CPU *versus*  $E$  para  $Q = 64$  e equação de Laplace senoidal.

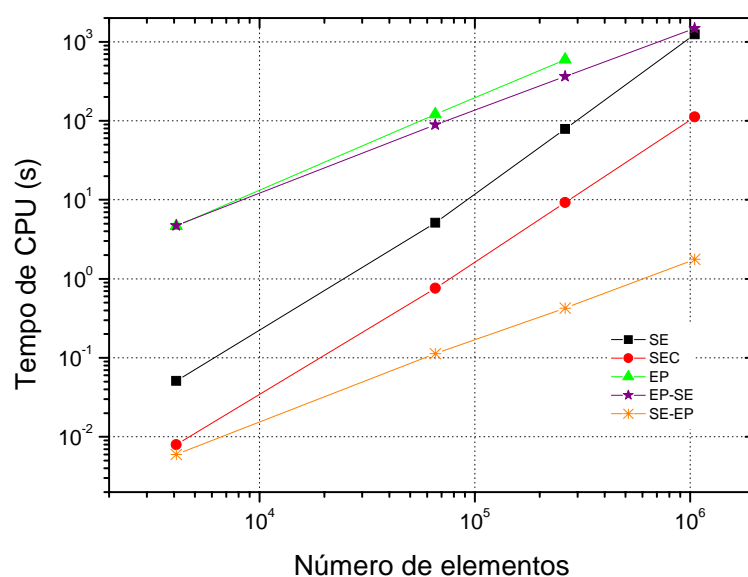


Figura 5.23: Tempo de CPU *versus*  $E$ , para  $Q = 1/64$  e equação de Laplace senoidal.

### 5.2.5 Análise de complexidade para a equação de Laplace senoidal

Para as malhas e razões de aspecto definidas na Tab. 5.9 fez-se uma análise da complexidade dos algoritmos EP, SE, SEC, SE-EP e EP. Calculou-se o expoente  $p$ , obtido pelo método dos mínimos quadrados, para a função dada por

$$t_{CPU}(N) = c N^p \quad (5.1)$$

onde  $p$  representa a ordem do *solver* associado ao método empregado e  $c$  é um coeficiente que depende de cada método e cada *solver*.  $N$  é o número de incógnitas do sistema e  $t_{CPU}$  o tempo de CPU.

Para o método *multigrid* ideal,  $p = 1$ , significando que o esforço computacional cresce linearmente com o tamanho da malha (BRANDT, 1977; HIRSCH, 1988 e TROTTENBERG et al., 2001). Assim, para um dado *hardware* e compilador, quanto menor for  $p$ , mais eficiente é o algoritmo. As Tab. 5.10 e 5.11 apresentam ordens de  $p$  os coeficientes  $c$  dos ajustes de curvas, respectivamente obtidos para os algoritmos em estudo. As malhas utilizadas na construção das Tabs. 5.10 e 5.11 estão apresentadas na Tab. 5.9. Os espaços não preenchidos nas Tabs. 5.10 e 5.11 referem-se às simulações cujo tempo de CPU foi muito alto e por este motivo não foram realizadas.

**Tabela 5.10: Valor da ordem ( $p$ ) da Eq. (5.1) para os algoritmos de engrossamento.**

<b>Algoritmos</b>						
	<b>SE</b>	<b>SEC</b>	<b>EP</b>	<b>EP-SE</b>	<b>SE-EP</b>	
<b><math>Q</math></b>						
1/64	1,55	1,14	-	-	1,04	
1/16	1,83	1,36	1,04	1,04	1,04	
¼	1,97	1,61	1,03	1,03	1,05	
1	-	-	1,04	-	-	
4	1,97	1,61	1,04	1,04	1,05	
16	1,92	1,36	1,05	1,05	1,06	
64	1,70	1,19	-	-	1,06	

Na Tab. 5.10 observa-se que os menores valores de  $p$  ocorrem para os algoritmos EP, EP-SE e SE-EP. Estes algoritmos são mais rápidos em relação ao SE e SEC por utilizarem um número maior de níveis. Na Tab. 5.11 pode-se constatar que entre os algoritmos EP, EP-SE e SE-EP, os coeficientes  $c$  são menores no algoritmo SE-EP. Portanto ele apresenta os menores

tempos de CPU conforme foi verificado nas Figs. 5.17 e 5.18. Para as equações de Laplace linear e Poisson os resultados foram similares e podem ser encontrados no Apêndice D.

Na Tab. 5.10 observa-se também que os menores valores de  $p$  para os algoritmos SE e SEC estão próximos de 1,5. Para o SE isto ocorre devido ao uso de poucos níveis. Por exemplo, para uma malha de 2049x513, o SE engrossa somente a malha mais refinada. Para a direção  $x$  tem-se 3 níveis (2049, 1025, 513). Na direção  $y$  tem-se um único nível com 513 variáveis. Portanto, na direção  $y$  aplicou-se *singlegrid* em uma malha bem refinada, logo a taxa de convergência diminui ficando próxima à do *singlegrid*.

**Tabela 5.11: Valor da ordem ( $c$ ) da Eq. (5.1) para os algoritmos de engrossamento.**

$Q$	Algoritmos				
	SE	SEC	EP	EP-SE	SE-EP
1/64	2,48E-07	9,50E-06	-	-	1,07E-06
1/16	2,91E-08	1,24E-06	3,89E-05	3,79E-05	1,07E-06
1/4	1,66E-08	1,61E-07	2,72E-05	2,71E-05	8,16E-07
1	-	-	3,24E-07	-	-
4	1,60E-08	1,45E-07	2,23E-05	2,27E-05	7,97E-07
16	7,53E-09	1,08E-06	2,96E-05	2,89E-05	7,63E-07
64	3,05E-08	4,57E-6	-	-	7,54E-07

As Tabs. 5.12 à 5.14 apresentam o *speed-up* do algoritmo SE-EP em relação aos algoritmos EP, SE, SEC e EP-SE para as equações de Laplace senoidal, Laplace linear e Poisson em malha com 262.144 elementos. As malhas utilizadas nas Tabs. 5.12 à 5.14 estão apresentadas na Tab. 5.9.

**Definição 5.1:** *Speed-up* é uma medida utilizada para determinar o aumento de velocidade obtido durante a execução de um programa utilizando um algoritmo “A” em relação a sua execução utilizando um algoritmo “B” (GALANTE, 2006). O *speed-up* é dado pela fórmula:

$$S_p = \frac{t_{CPU}(\text{algoritmo A})}{t_{CPU}(\text{algoritmo B})} \quad (5.2)$$

Por exemplo, para calcular o *speed-up* do algoritmo SE-EP em relação ao EP divide-se o tempo de CPU do algoritmo A, neste caso EP pelo tempo de CPU do algoritmo (B) SE-EP. Para a razão 1/64 tem-se  $S_p = \frac{t_{CPU}(\text{EP})}{t_{CPU}(\text{SE} - \text{EP})} = \frac{78,719}{0,426} = 184,79$ . Isto significa que o



algoritmo SE-EP é cerca de 180 vezes mais rápido que o EP para a malha e razão de aspecto especificados.

**Tabela 5.12: *Speed-up* do SE-EP em relação aos algoritmos: EP, SE, SEC, EP-SE para a equação de Laplace senoidal.**

$Q$	SE	SEC	EP	EP-SE
1/64	184,79	21,69	1397,41	858,57
1/16	663,33	64,75	53,43	55,19
1/4	2786,57	273,70	4,41	4,53
4	1524,91	182,76	2,22	2,22
16	445,57	43,61	23,27	23,67
64	100,33	12,13	555,08	348,04

**Tabela 5.13: *Speed-up* do SE-EP em relação aos algoritmos: EP, SE, SEC, EP-SE para a equação de Laplace linear.**

$Q$	SE	SEC	EP	EP-SE
1/64	113,17	23,21	-	-
1/16	481,35	46,30	38,94	39,32
1/4	2022,64	197,98	26,54	26,34
4	1904,43	193,34	24,51	24,95
16	452,36	44,46	36,58	36,62
64	111,05	22,03	-	-

**Tabela 5.14: *Speed-up* do SE-EP em relação aos algoritmos: EP, SE, SEC, EP-SE para a equação de Poisson.**

$Q$	SE	SEC	EP	EP-SE
1/64	162,50	21,91	-	-
1/16	703,33	65,51	46,00	46,08
1/4	2725,26	247,33	44,61	26,28
4	2572,11	244,37	24,35	25,23
16	656,31	63,09	42,60	42,79
64	157,87	21,20	-	-

Nas Tabs. 5.12 a 5.14 foram comparados o algoritmo SE-EP com os algoritmos EP, SE, SEC e EP-SE. Verificou-se que para todas as razões de aspecto em estudo o algoritmo SE-EP foi muito mais rápido comparado aos demais. Esta melhora no tempo de CPU foi entre 2 a 2.786 vezes, o que comprova a vantagem do uso deste algoritmo para problemas anisotrópicos e para as razões de aspecto estudadas. Isto já era esperado, pois a idéia do algoritmo SE-EP é transformar a malha anisotrópica em uma malha isotrópica, onde o método *multigrid* converge bem. Os demais algoritmos mantêm a anisotropia, o que faz com que a convergência do método *multigrid* se deteriore. Também observou-se que o *speed-up* varia de

acordo com a equação, isto ocorre devido o fato do tempo de CPU não ser o mesmo para todas as equações.

A Tab. 5.15 apresenta o *speed-up* do algoritmo SEC em relação ao SE para as equações de Laplace senoidal, Laplace linear e Poisson para uma malha anisotrópica com 262.144 elementos. O *speed-up* foi calculado através da Eq. (5.2). Neste caso o algoritmo “A” é o SE e o algoritmo “B” SEC. Verifica-se que o algoritmo SEC é cerca de 10 vezes mais rápido que o SE para as razões de aspecto em estudo para os casos estudados. O algoritmo de SEC aplica o semi-engrossamento até a malha mais grossa possível, utilizando assim um maior número de níveis do que o algoritmo de SE. Como já foi visto na seção 5.2.2, quanto maior o número de níveis, menor o tempo de CPU. Portanto o algoritmo SEC apresenta um menor tempo de CPU em relação ao SE.

**Tabela 5.15: *Speed-up* do SEC em relação ao SE para as equações de Laplace senoidal, Laplace linear e Poisson.**

$Q$	Laplace senoidal	Laplace linear	Poisson
1/64	8,52	4,88	7,42
1/16	10,24	10,40	10,74
1/4	10,18	10,22	11,02
4	8,34	9,85	10,53
16	10,22	10,17	10,40
64	8,27	5,04	7,45

### 5.3 CONCLUSÃO DO CAPÍTULO

Neste capítulo foram resolvidos numericamente três problemas bidimensionais lineares de condução de calor, governados pelas equações de Laplace e Poisson, com condições de contorno de Dirichlet. Utilizou-se o esquema de aproximação CDS e o esquema de correção CS do método *multigrid* geométrico. Foram apresentados cinco algoritmos utilizados em problemas anisotrópicos: Engrossamento padrão (EP), semi-engrossamento (SE), semi-engrossamento completo (SEC), engrossamento padrão seguido de semi-engrossamento (EP-SE) e semi-engrossamento seguido de engrossamento padrão (SE-EP). Com base nos resultados obtidos neste capítulo, verificou-se que, para as equações e razões de aspecto estudadas:

- 1) Para os mesmos parâmetros ( $v$ ,  $Q$ ,  $L$  e  $E$ ) o algoritmo SE-EP apresentou um melhor desempenho, isto é, um menor tempo de CPU entre os cinco algoritmos analisados para os problemas anisotrópicos. A melhora no tempo de CPU ao utilizar-se SE-EP está entre 2 a 2.786 vezes.
- 2) A variação da razão de aspecto resulta em grande variação do  $v_{\text{ótimo}}$  para os algoritmos EP, SE, SEC e EP-SE. Para o algoritmo SE-EP o  $v_{\text{ótimo}}$  permanece constante em todas as razões de aspecto.
- 3) A variação da razão de aspecto afeta significativamente o tempo de CPU para os algoritmos EP, SE, SEC e EP-SE. Para o algoritmo SE-EP, a razão de aspecto não influencia significativamente o tempo de CPU, desde que seja utilizado o mesmo número de elementos.
- 4) Para todos os algoritmos de engrossamento analisados,  $L_{\text{máximo}}$  apresenta um melhor desempenho médio para qualquer razão de aspecto e qualquer tamanho de problema. Quanto maior a anisotropia, maior a influência do número de níveis no tempo de CPU.
- 5) O  $v_{\text{ótimo}}$  para cada equação varia de acordo com a oscilação da função. As equações de Laplace linear e Poisson possuem a mesma oscilação nas direções  $x$  e  $y$ , portanto o  $v_{\text{ótimo}}$  é igual para todas as razões de aspecto. Para a equação de Laplace linear  $v_{\text{ótimo}} = 4$  e para Poisson  $v_{\text{ótimo}} = 3$ . A equação de Laplace senoidal é mais oscilatória na direção  $y$ , portanto o  $v_{\text{ótimo}}$  varia em função da razão de aspecto, sendo  $v_{\text{ótimo}} = 2$  para razões de aspecto menores que a unidade e  $v_{\text{ótimo}} = 4$  para razões de aspecto maiores que a unidade.
- 6) O comportamento qualitativo dos parâmetros estudados, com exceção do  $v_{\text{ótimo}}$ , é o mesmo para as equações de Laplace senoidal, Laplace linear e Poisson.
- 7) Para os mesmos parâmetros ( $v$ ,  $Q$ ,  $L$  e  $E$ ) o algoritmo SEC apresentou um melhor desempenho, isto é, um menor tempo de CPU em relação ao SE para as malhas e razões de aspecto estudadas. A melhora no tempo de CPU ao utilizar-se SEC está entre 5 a 11 vezes.

## 5.4 PARÂMETROS COM MELHOR DESEMPENHO MÉDIO

Com base nos resultados obtidos neste capítulo obteve-se os seguintes parâmetros com melhor desempenho médio que podem ser utilizados para problemas iguais ou similares aos estudados.

Problemas bidimensionais lineares isotrópicos:

- Algoritmo: EP.
- Número de iterações internas:  $\nu = 2$ .
- Número de níveis:  $L = L_{máximo}$ .

Problemas bidimensionais lineares anisotrópicos:

- Algoritmo: SE-EP.
- Número de iterações internas: Para a equação de Laplace senoidal  $\nu = 2$  para problemas com razões de aspecto menores que a unidade e  $\nu = 4$  para razões de aspecto maiores que a unidade. Para a equação de Laplace linear  $\nu = 4$  e para Poisson  $\nu = 3$ , independente da razão de aspecto.
- Número de níveis:  $L = L_{máximo}$ .

## 6 SEMI-ENGROSSAMENTO SEGUIDO DE ENGROSSAMENTO PADRÃO

O capítulo 5 apresenta um estudo de cinco algoritmos de engrossamento para problemas com anisotropia geométrica: engrossamento padrão (EP), semi-engrossamento, semi-engrossamento completo (SEC), engrossamento padrão seguido de semi-engrossamento (EP-SE) e semi-engrossamento seguido de engrossamento padrão (SE-EP). Dentre eles, o algoritmo SE-EP foi o que apresentou melhor desempenho. O objetivo deste capítulo é fazer um estudo mais detalhado deste algoritmo através da análise de outras malhas e razões de aspecto. Também são analisados alguns parâmetros do método *multigrid* como *solvers*, tipos de restrição, número de níveis e número de iterações internas visando uma otimização do algoritmo. São utilizadas para isto, as equações de Laplace senoidal, Laplace linear e Poisson com condições de contorno de Dirichlet dadas pelas Eqs. (3.1) descritas detalhadamente na seção 3.1.

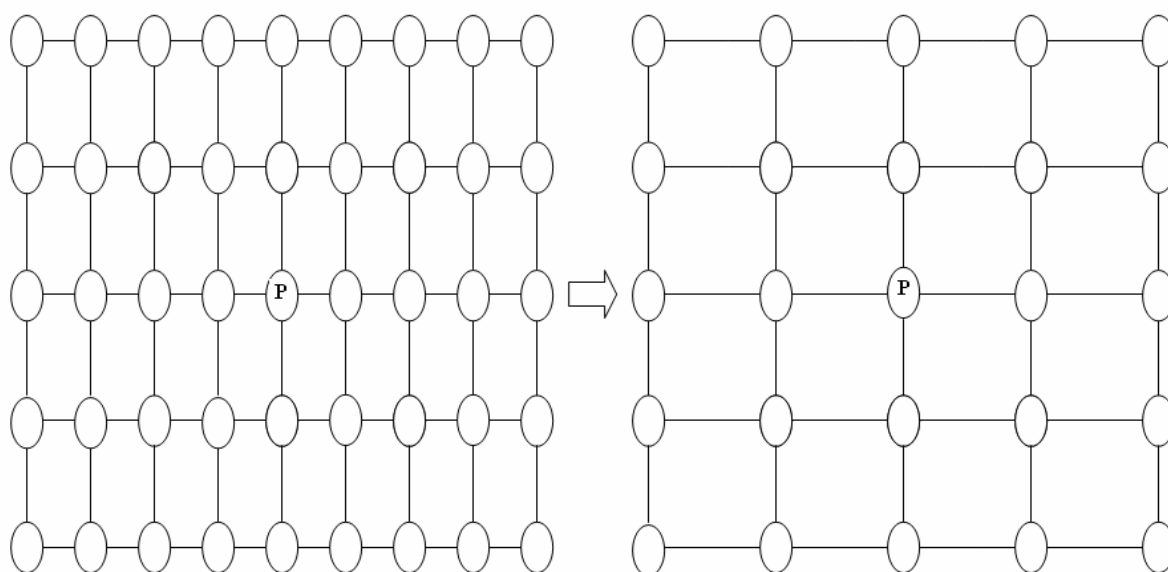
Este capítulo está dividido da seguinte forma: a primeira seção apresenta os tipos de restrição utilizados. A segunda seção apresenta os resultados numéricos obtidos: razão de aspecto, número de elementos, análise de complexidade e uma análise dos parâmetros ótimos (*solvers*, restrição e iterações internas). As últimas seções apresentam a conclusão do capítulo e os parâmetros com melhor desempenho médio.

### 6.1 OPERADORES DE RESTRIÇÃO

Neste trabalho foram utilizados seis operadores de restrição: injeção (I), meia ponderação (HW), ponderação completa (FW), meia ponderação geométrica (GHW), ponderação geométrica completa (GFW) e ponderação parcial (PW). As restrições por injeção, meia ponderação e ponderação completa podem ser encontradas na seção 2.3.3 do livro TROTTENBERG et al., (2001). As restrições por ponderação geométrica, ponderação geométrica completa e ponderação parcial são propostas neste trabalho para serem utilizadas em problemas anisotrópicos. Apresenta-se a seguir a definição e a notação estêncil (*stencil*) para cada tipo de restrição.

### 6.1.1 Injeção (I)

Entre os operadores de restrição conhecidos, um dos mais utilizados é o operador de restrição por injeção (BRIGGS et al., 2000; TROTTEBERG et al., 2001; WESSELING, 1992). A restrição por injeção restringe apenas o ponto central (P) para a malha mais grossa. A Fig. 6.1 apresenta uma malha correspondente a este processo. O ponto central (P) que será injetado tem peso um. Sua notação estêncil é dada pela Eq. (6.1).



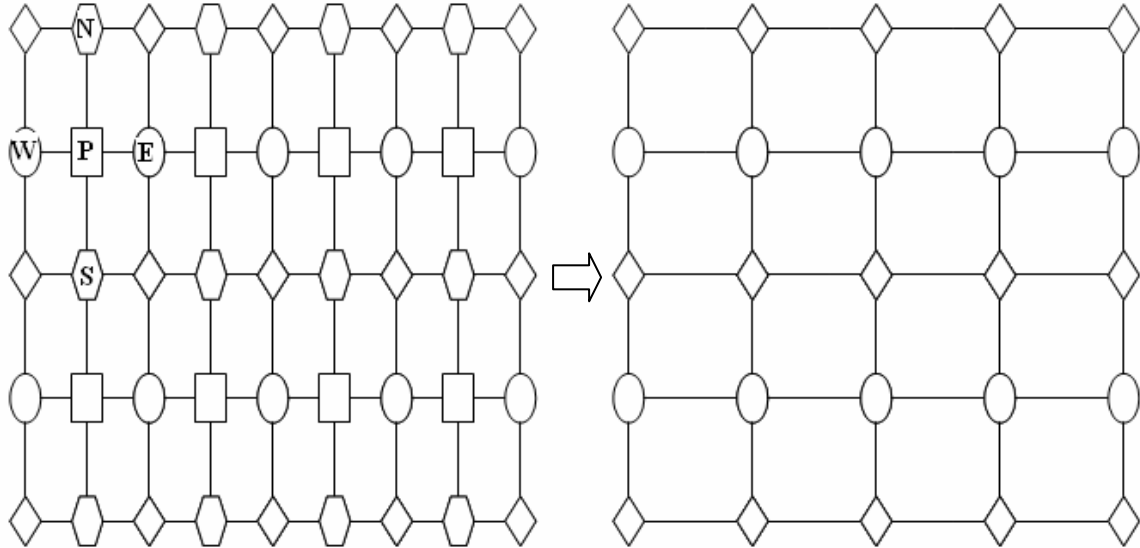
**Figura 6.1: Nó utilizado para a restrição por injeção.**

$$I_h^H = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{matrix} H \\ h \end{matrix} \quad (6.1)$$

### 6.1.2 Meia ponderação (HW)

A restrição por meia ponderação também é encontrada na literatura (BRIGGS et al., 2000; TROTTEBERG et al., 2001; WESSELING, 1992). A restrição por meia ponderação utiliza o ponto central (P) e também os quatro pontos vizinhos N, W, E e S. A Fig. 6.2

apresenta a malha correspondente a este processo. O ponto central (P) que será restrito tem um peso maior em relação aos demais. Sua notação estêncil é dada pela Eq. (6.2).



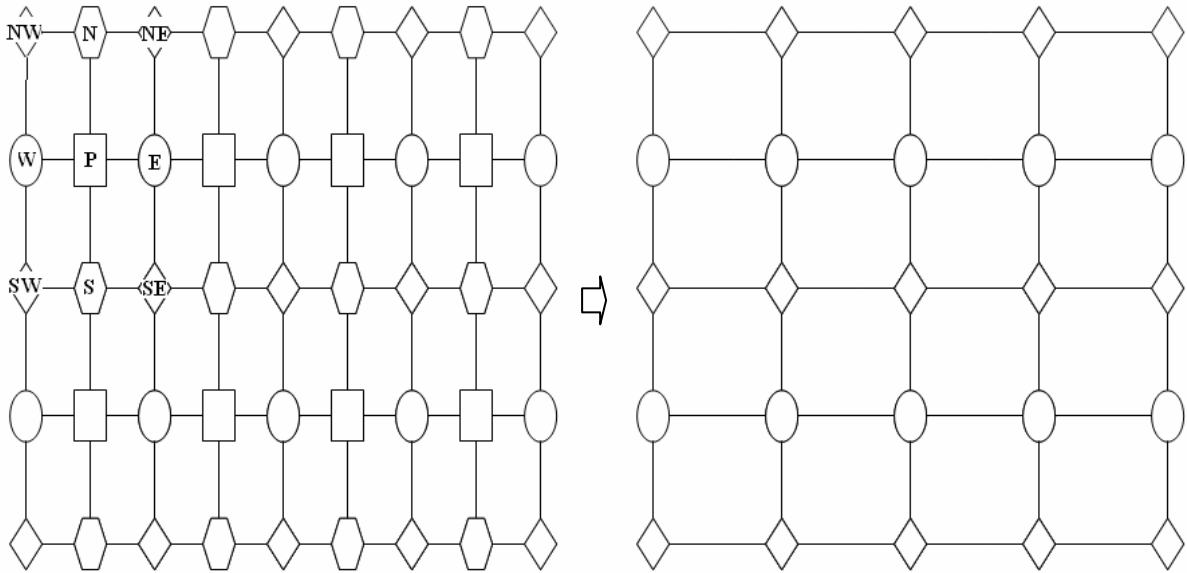
**Figura 6.2: Nós utilizados para a restrição por meia ponderação.**

$$I_h^H = \frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (6.2)$$

### 6.1.3 Ponderação completa (FW)

A restrição por ponderação completa pode ser encontrada em Briggs et al. (2000); Trottenberg et al. (2001) e Wesseling (1992). A restrição por ponderação completa é feita através de uma média ponderada entre o ponto central (P) e todos os pontos vizinhos (N, W, E, S, NW, NE, SW e SE). Os pontos N, S, E e W recebem um peso maior em relação aos demais, com exceção de P) conforme pode ser visto na Eq. (6.3). A Fig. 6.3 apresenta a malha correspondente a este processo e a notação estêncil é dada pela Eq. (6.3).

$$I_h^H = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^H \quad (6.3)$$



**Figura 6.3: Nós utilizados para a restrição por ponderação completa.**

#### 6.1.4 Meia ponderação geométrica (GHW)

É uma versão modificada da restrição por meia ponderação utilizada para problemas anisotrópicos. No estêncil apresentado na Eq. (6.2) verifica-se que os pesos correspondentes aos pontos N, S, L e W são iguais. Para problemas isotrópicos isto se justifica pelo fato de que as distâncias entre cada ponto e o ponto central são iguais. Isto já não ocorre em problemas anisotrópicos (Fig. 6.2). Logo, a idéia da meia ponderação geométrica é atribuir pesos de acordo com a distância entre cada ponto e o ponto central, considerando a anisotropia da malha dada pelo fator  $Q$ . Seu estêncil é dado por:

$$I_h^H = \frac{1}{4Q+4} \begin{bmatrix} 0 & Q & 0 \\ 1 & 2Q+2 & 1 \\ 0 & Q & 0 \end{bmatrix}_h^H \quad (6.4)$$



onde  $Q = \frac{h_x}{h_y}$ , com  $h_x$  e  $h_y$  indicando o tamanho da malha nas direções  $x$  e  $y$  respectivamente.

Note que para o problema isotrópico, ou seja,  $Q = 1$ , tem-se a restrição por HW.

### 6.1.5 Ponderação geométrica completa (GFW)

É uma versão modificada da restrição por ponderação completa utilizada para problemas anisotrópicos. Considera-se o peso de acordo com a anisotropia como em GHW. Seu estêncil está apresentado pela Eq. (6.5). Note que para o problema isotrópico, ou seja,  $Q = 1$ , tem-se a restrição por FW.

$$I_h^H = \frac{1}{2(6Q+2)} \begin{bmatrix} Q & 2Q & Q \\ 2 & 4Q & 2 \\ Q & 2Q & Q \end{bmatrix}_h^H \quad (6.5)$$

### 6.1.6 Ponderação parcial (PW)

A ponderação parcial utiliza pesos somente na direção em que é realizado o engrossamento. Apresenta-se a seguir os estêncis para o engrossamento na direção  $x$  e na direção  $y$ , respectivamente:

$$I_h^H = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}_h^H \quad \text{e} \quad I_h^H = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix}_h^H \quad (6.6)$$

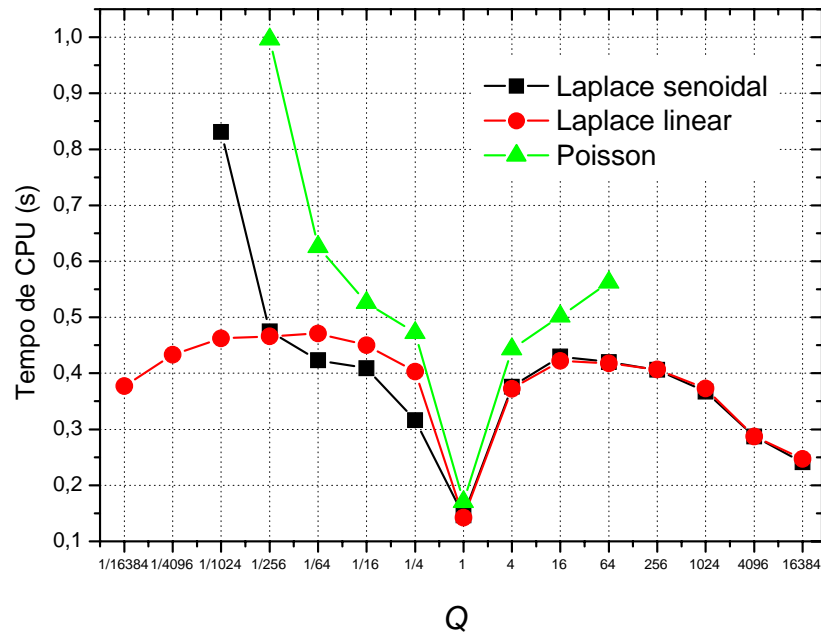
## 6.2 RESULTADOS NUMÉRICOS

Os dados de implementação foram os mesmos definidos na seção 3.3. Cerca de 1.000 simulações foram realizadas com as seguintes variantes: algoritmo SE-EP; equações: (Laplace senoidal, Laplace linear e Poisson); tipos de restrição: (injeção, meia ponderação, ponderação completa, meia ponderação geométrica, ponderação geométrica completa e ponderação parcial); número de elementos  $E = (4, 16, 64, 256, 1.024, 4.096, 16.384, 65.536, 262.144, 1.048.576, 4.194.304, 16.777.216 \text{ e } 67.108.864)$ ; razões de aspecto  $Q = (1/16.384, 1/4.096, 1/1.024, 1/256, 1/64, 1/16, 1/4, 1, 4, 16, 64, 256, 1.024, 4.096 \text{ e } 16.384)$ . A tabela com as malhas utilizadas encontra-se no apêndice E.

Nesta seção, adotou-se o número máximo possível de níveis. São apresentados a seguir os resultados mais representativos do estudo realizado.

### 6.2.1 Razão de aspecto ( $Q$ )

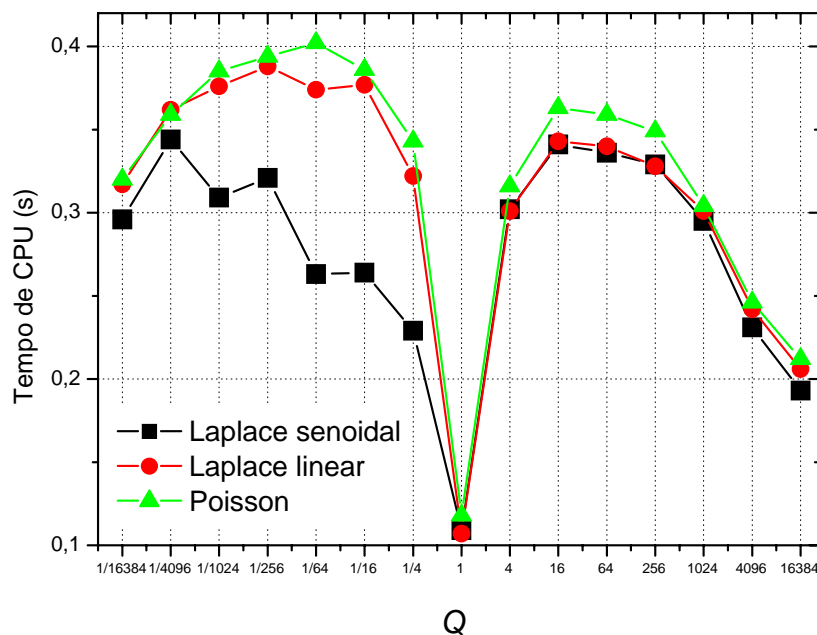
A Fig. 6.4 apresenta o tempo de CPU para o algoritmo SE-EP, diversas razões de aspecto e equações de Laplace senoidal, Laplace linear e Poisson para problemas com 262.144 elementos. As simulações foram realizadas para todas as razões de aspectos definidas na seção 6.2. Observa-se que algumas delas não estão representadas graficamente. Isto ocorreu devido o fato de que para algumas razões de aspecto não foi possível atingir a tolerância estabelecida. Verifica-se que o menor tempo de CPU é obtido em problemas isotrópicos. Verifica-se também que o tempo de CPU varia de acordo com a razão de aspecto. Para a equação de Laplace senoidal e razões de aspecto menores que a unidade o tempo de CPU aumenta à medida que aumenta a anisotropia. Para as razões de aspecto maiores que a unidade o tempo de CPU aumenta até  $Q \approx 16$  e depois começa a diminuir, porém sempre é maior que o isotrópico. Na equação de Laplace linear e razões de aspecto menores que a unidade o tempo de CPU aumenta até  $Q \approx 1/64$ . Para razões de aspecto maiores que a unidade o tempo de CPU aumenta até  $Q \approx 16$  e depois começa a diminuir. Para a equação de Poisson o tempo de CPU aumenta à medida que aumenta a anisotropia, independentemente da razão de aspecto ser maior ou menor que a unidade.



**Figura 6.4:** Tempo de CPU *versus* razão de aspecto para o algoritmo SE-EP,  $E = 262.144$  para as equações de Laplace senoidal, Laplace linear e Poisson com  $\varepsilon = 10^{-10}$ .

Para a equação de Laplace linear esperava-se que o tempo de CPU fosse o mesmo para as razões de aspecto simétricas, por exemplo,  $t_{CPU}(Q = 16) = t_{CPU}(Q = 1/16)$ . A diferença entre o tempo de CPU para as razões de aspecto simétricas ocorreu devido a imprecisão da função CPU\_time. Observou-se nas simulações que o número de ciclos (iterações externas) realizado para as razões de aspecto simétricas é sempre o mesmo.

Verificou-se que com um  $\varepsilon = 10^{-10}$ , não foi possível resolver o problema para todas as razões de aspecto nas equações de Laplace linear e Poisson. Isto aconteceu devido o fato da norma  $l_2$  começar a oscilar antes que esta tolerância fosse atingida. Utilizando uma tolerância um pouco maior,  $\varepsilon = 10^{-7}$ , foi possível fazer uma análise mais geral, envolvendo todas as razões de aspecto em estudo. A Fig. 6.5 apresenta o tempo de CPU obtido com esta tolerância. Os demais parâmetros utilizados são os mesmos da Fig. 6.4.



**Figura 6.5:** Tempo de CPU *versus* razão de aspecto o algoritmo SE-EP,  $E = 262.144$  para as equações de Laplace senoidal, Laplace linear e Poisson com  $\varepsilon = 10^{-7}$ .

Na Fig. 6.5 pode-se verificar que o menor tempo de CPU é obtido em problemas isotrópicos. Verifica-se também que o tempo de CPU varia de acordo com a razão de aspecto. Para todas as equações em estudo verifica-se que o tempo de CPU aumenta até certo ponto e começa a decrescer para problemas muito anisotrópicos. De uma forma geral, pode-se concluir que o algoritmo SE-EP apresenta os menores tempo de CPU para razões de aspecto bem próximas da unidade ou muito distintas da unidade. Para as razões de aspecto intermediárias o tempo de CPU é maior. Isto pode ser explicado pelo fato de que para problemas anisotrópicos com malhas levemente alongadas ( $0 < Q \approx 1$ ), o algoritmo SE-EP comporta-se de forma muito semelhante ao problema isotrópico, onde o método *multigrid* apresenta bons resultados. Por outro lado, para problemas anisotrópicos com malhas fortemente alongadas ( $Q \gg 1$ ), o algoritmo SE-EP comporta-se de forma semelhante aos problemas unidimensionais, devido à predominância do semi-engrossamento; e neste caso o método *multigrid* também apresenta bons resultados.

As conclusões obtidas através da Fig. 6.5 são mais conclusivas, pois a tolerância foi atingida para todas as razões de aspecto. Desta forma o comportamento do tempo de CPU em função da razão de aspecto pode ser observado como um todo.

### 6.2.2 Número de elementos ( $E$ )

As Figs. 6.6 e 6.7 mostram a influência do número de elementos e da razão de aspecto no tempo de CPU para o algoritmo de SE-EP para a equação de Laplace senoidal. A Fig.6.6 apresenta as razões de aspecto da forma  $Q \leq 1$  e a Fig. 6.7 as razões  $Q \geq 1$ .

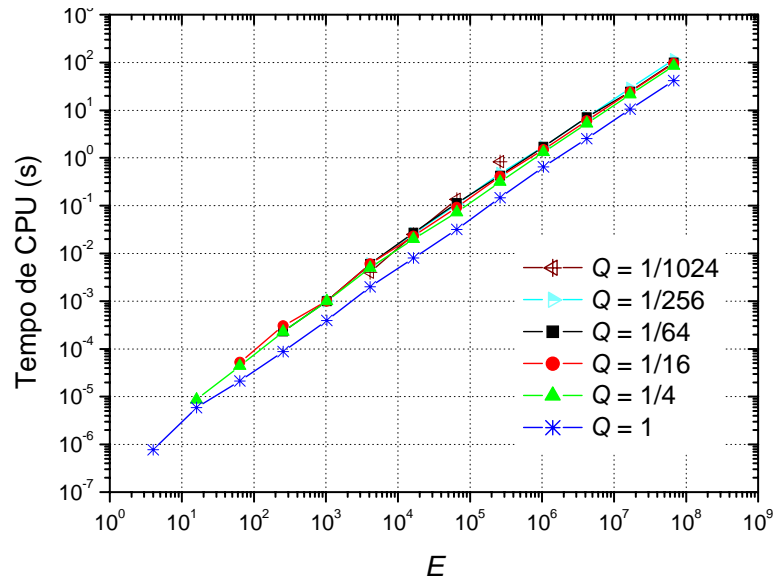


Figura 6.6: Tempo de CPU *versus* número de elementos para  $Q \leq 1$ , algoritmo SE-EP e equação de Laplace senoidal com  $\varepsilon = 10^{-10}$ .

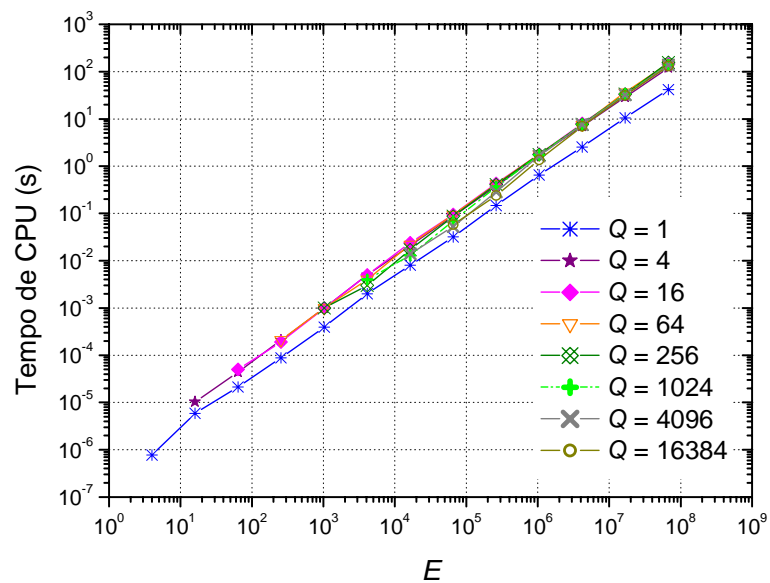


Figura 6.7: Tempo de CPU *versus* número de elementos para  $Q \geq 1$ , algoritmo SE-EP e equação de Laplace senoidal com  $\varepsilon = 10^{-10}$ .

Nas Figs. 6.6 e 6.7 verifica-se que a razão de aspecto não influencia significativamente no tempo de CPU. Observa-se também que o tempo de CPU aumenta em função do tamanho do problema. Os resultados obtidos para as equações de Laplace linear e Laplace senoidal foram similares e estão apresentados nas Figs. 6.8 à 6.11. Para os gráficos referentes as equações de Laplace senoidal e Poisson utilizou-se um número menor de razões de aspecto. Isto foi necessário devido ao fato da tolerância não ser atingida para todas as razões de aspecto em estudo. A partir de um certo número de iterações a norma  $l_2$  começa a oscilar antes que a tolerância seja atingida.

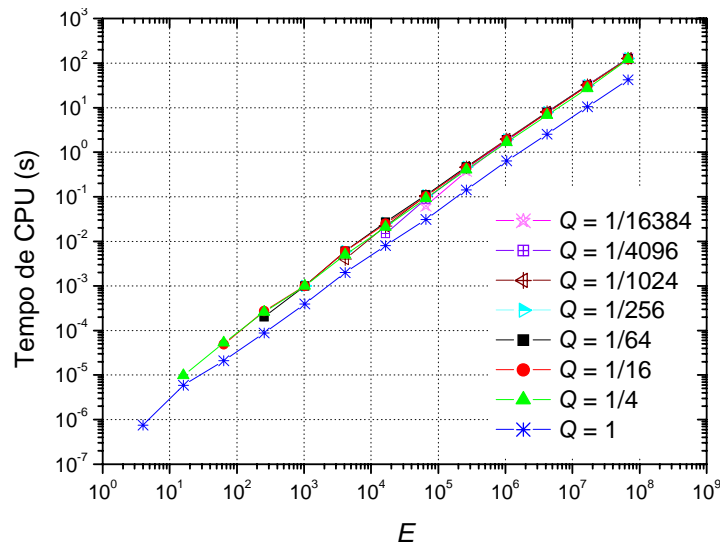


Figura 6.8: Tempo de CPU *versus* número de elementos para  $Q \leq 1$ , algoritmo SE-EP e equação de Laplace linear com  $\varepsilon = 10^{-10}$ .

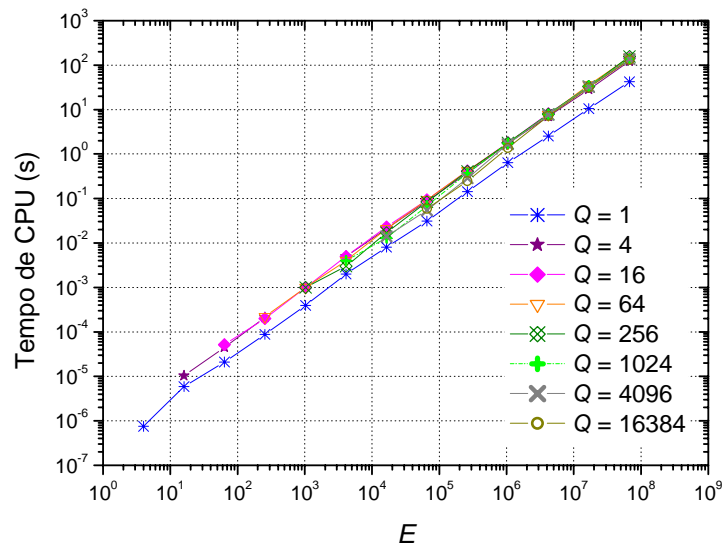


Figura 6.9: Tempo de CPU *versus* número de elementos para  $Q \geq 1$ , algoritmo SE-EP e equação de Laplace linear com  $\varepsilon = 10^{-10}$ .

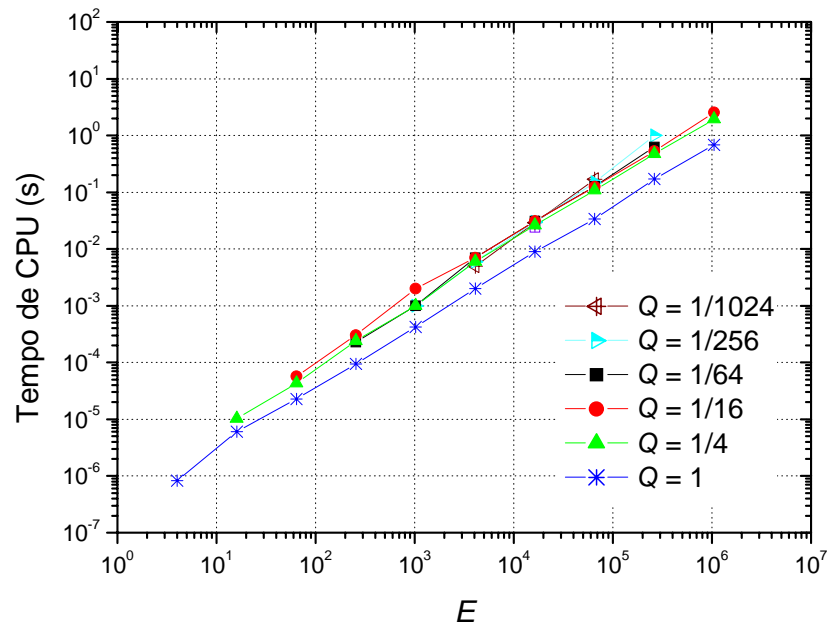


Figura 6.10: Tempo de CPU *versus* número de elementos para  $Q \leq 1$ , algoritmo SE-EP e equação de Poisson com  $\varepsilon = 10^{-10}$ .

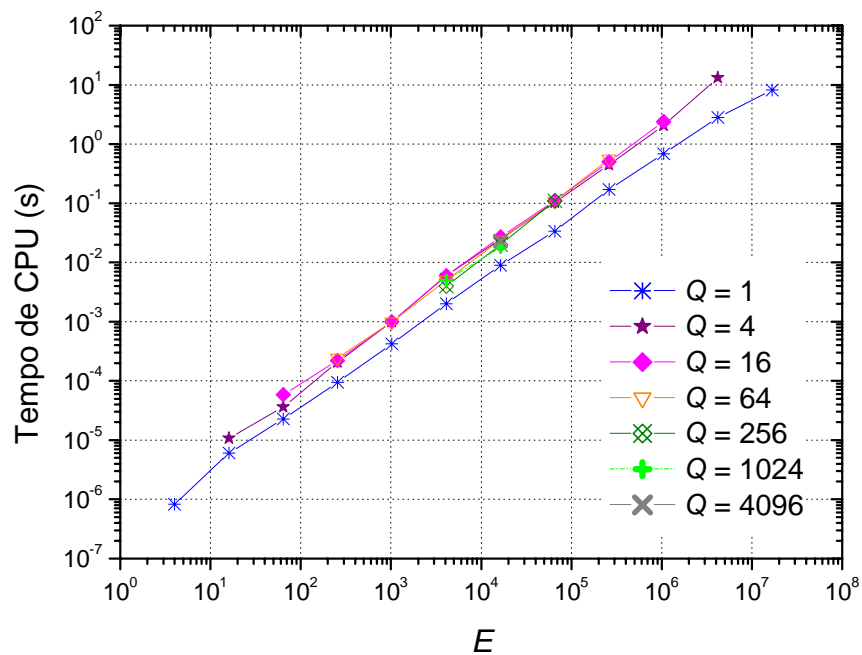


Figura 6.11: Tempo de CPU *versus* número de elementos para  $Q \geq 1$ , algoritmo SE-EP e equação de Poisson com  $\varepsilon = 10^{-10}$ .

### 6.2.3 Análise de complexidade para o algoritmo SE-EP

Esta subseção faz uma análise da complexidade do algoritmo SE-EP para as equações de Laplace senoidal, Laplace linear e Poisson. As malhas utilizadas nesta análise são apresentadas no Apêndice E. Calculou-se o coeficiente  $c$  e o expoente  $p$ , obtido pelo método dos mínimos quadrados, para a função dada pela Eq. (5.1) apresentada na seção 5.2.5.

A Tab. 6.1 apresenta os coeficientes  $c$  e as ordens  $p$  dos ajustes de curvas obtidos para o algoritmo SE-EP e as três equações em estudo. Verifica-se que todos os coeficientes  $p$  são próximos de um, para todas as razões de aspecto, tanto em problemas isotrópicos como em problemas anisotrópicos. Isto indica que o algoritmo SE-EP está bem próximo do método *multigrid* ideal. Os espaços da tabela que não foram preenchidos referem-se a problemas onde a tolerância estabelecida não foi atingida.

**Tabela 6.1: Valores de  $c$  e  $p$  para Eq. (5.1) e algoritmo SE-EP.**

$Q$	Laplace senoidal		Laplace linear		Poisson	
	$c$	$p$	$c$	$p$	$c$	$p$
1/16384	-	-	4,24E-07	1,09	-	-
1/4096	-	-	5,41E-07	1,08	-	-
1/1024	9,93E-08	1,28	7,33E-07	1,06	1,29E-07	1,27
1/256	1,03E-06	1,03	9,12E-07	1,05	2,04E-07	1,23
1/64	9,54E-07	1,03	7,47E-07	1,06	4,12E-07	1,15
1/16	9,23E-07	1,03	7,48E-07	1,06	7,97E-07	1,08
1/4	6,43E-07	1,05	6,50E-07	1,06	4,93E-07	1,11
1	2,56E-07	1,06	2,51E-07	1,06	2,73E-07	1,06
4	5,82E-07	1,07	5,83E-07	1,07	4,26E-07	1,12
16	5,90E-07	1,08	5,99E-07	1,07	5,41E-07	1,10
64	5,52E-07	1,08	5,56E-07	1,08	4,53E-07	1,12
256	4,34E-07	1,09	4,29E-07	1,10	1,89E-07	1,20
1024	3,69E-07	1,10	3,66E-07	1,10	2,30E-06	0,92
4096	2,39E-07	1,13	2,42E-07	1,12	-	-
16384	2,05E-07	1,13	2,13E-07	1,13	-	-

### 6.2.4 Análise dos parâmetros ótimos para o algoritmo dois estágios

A idéia do algoritmo SE-EP é aplicar o semi-engrossamento até a malha tornar-se isotrópica e em seguida aplicar o engrossamento padrão. Como são utilizados dois sub-



algoritmos diferentes (algoritmo dois estágios) dentro do algoritmo principal, os parâmetros ótimos podem diferir para cada um deles se o estudo for feito separadamente. Nesta etapa do trabalho fez-se o estudo dos seguintes parâmetros: *solvers* (Gauss-Seidel lexicográfico e *red-black*) e tipo de restrição (injeção, meia ponderação, ponderação completa, meia ponderação geométrica, ponderação geométrica completa e ponderação parcial), para cada algoritmo em particular. Primeiramente foram fixados os parâmetros referentes ao EP: *solver* Gauss-Seidel lexicográfico e restrição por injeção. Foi realizada uma seqüência de simulações com variações do *solver* e do tipo de restrição para o SE. Desta forma, verificou-se para o *solver* Gauss-Seidel lexicográfico qual tipo de restrição apresenta o menor tempo de CPU. O mesmo processo foi repetido para o *solver* Gauss-Seidel *red-black*. Os parâmetros ótimos determinados foram fixados neste bloco. Em seguida foi feita a mesma análise para o algoritmo EP, determinado também qual o melhor *solver* e tipo de restrição para este bloco. Tendo os parâmetros ótimos para cada bloco, estes foram fixados e na seqüência foi realizada uma análise do número de iterações internas, número de níveis e complexidade do algoritmo ótimo.

### **Estudo dos *solvers* e tipos de restrição**

O estudo de *solvers* e tipos de restrição foi realizado para as equações de Laplace senoidal, Laplace linear e Poisson. Para cada equação foram fixados os parâmetros referentes ao EP: *solver* Gauss-Seidel lexicográfico e restrição por injeção. Foi realizada uma seqüência de simulações variando o *solver* e o tipo de restrição para o bloco do SE. As Tab. 6.2 e 6.3 apresentam o tempo de CPU em segundos (s) obtido para as malha 2049x129 e 129x2049, respectivamente. Para o *solver* GS-RB não utilizou-se restrição por injeção pois este *solver* não converge quando associado à restrição por injeção devido ao raio espectral não ser limitado (TROTTEMBERG et al. (2001).

Verifica-se nas Tabs. 6.2. e 6.3, para todas as equações, que o menor tempo de CPU foi obtido para o *solver* GS-RB e restrição por ponderação parcial. As restrições meia ponderação geométrica e ponderação geométrica completa apresentam menor tempo de CPU em relação à meia ponderação e a ponderação completa. Por exemplo, para a razão de aspecto 1/16 a meia ponderação geométrica é cerca de quatro vezes mais rápida que a meia ponderação e a meia ponderação geométrica completa é cerca de três vezes mais rápida que a

ponderação completa, para esta mesma razão. No entanto elas perdem para a ponderação parcial, sendo esta mais indicada para problemas anisotrópicos.

A seguir foi feita uma análise do algoritmo EP fixando-se para o SE os parâmetros ótimos obtidos no passo anterior (GS-RB e PW). As Tabs. 6.4 e 6.5 apresentam o tempo de CPU em segundos (s) obtido para as malha 2049x129 e 129x2049, respectivamente. Vale lembrar que ao aplicar o EP a malha já é isotrópica, portanto são utilizados somente as restrições I, HW e FW. Verifica-se nas Tabs. 6.4. e 6.5 que o menor tempo de CPU foi obtido para o *solver* GS-RB. O tipo de restrição não influencia significativamente o tempo de CPU.

**Tabela 6.2: Tempo de CPU (s) para a análise do bloco SE, malha 2049x129 e  $Q = 1/16$ .**

Restrição	Laplace Senoidal		Laplace Linear		Poisson	
	<i>Solver</i>		<i>Solver</i>		<i>Solver</i>	
	GS-LEX	GS-RB	GS-LEX	GS-RB	GS-LEX	GS-RB
I	0,411	-	0,476	-	0,424	-
HW	2,820	2,844	3,086	2,840	3,238	3,090
FW	16,328	10,766	9,469	5,711	7,109	4,438
PW	0,309	0,265	0,461	0,278	0,354	0,305
GHW	0,667	0,563	0,844	0,618	0,857	0,671
GFW	5,836	1,781	5,156	1,906	5,930	2,025

**Tabela 6.3: Tempo de CPU (s) para a análise do bloco SE, malha 129x2049 e  $Q = 16$ .**

Restrição	Laplace Senoidal		Laplace Linear		Poisson	
	<i>Solver</i>		<i>Solver</i>		<i>Solver</i>	
	GS-LEX	GS-RB	GS-LEX	GS-RB	GS-LEX	GS-RB
I	0,431	-	0,432	-	0,470	-
HW	0,871	1,701	2,828	3,055	0,802	1,507
FW	3,258	6,797	8,805	6,078	3,396	4,854
PW	0,433	0,295	0,432	0,297	0,403	0,237
GHW	0,437	0,428	0,792	0,671	0,412	0,389
GFW	1,513	1,549	4,823	2,050	1,522	1,643

**Tabela 6.4: Tempo de CPU (s) para a análise do bloco EP, malha 2049x129 e  $Q = 1/16$ .**

Restrição	Laplace Senoidal		Laplace Linear		Poisson	
	<i>Solver</i>		<i>Solver</i>		<i>Solver</i>	
	GS-LEX	GS-RB	GS-LEX	GS-RB	GS-LEX	GS-RB
I	0,263	0,260	0,279	-	0,316	-
HW	0,267	0,260	0,280	0,274	0,313	0,301
FW	0,265	0,261	0,335	0,277	0,301	0,304

**Tabela 6.5: Tempo de CPU (s) para a análise do bloco EP, malha 129x2049 e  $Q = 16$ .**

Restrição	Laplace Senoidal		Laplace Linear		Poisson	
	Solver		Solver		Solver	
	GS-LEX	GS-RB	GS-LEX	GS-RB	GS-LEX	GS-RB
I	0,295	0,292	0,297	-	0,234	-
HW	0,294	0,291	0,300	0,293	0,237	0,229
FW	0,297	0,292	0,359	0,294	0,282	0,233

Fazendo uma análise dos dois blocos do algoritmo SE e EP concluiu-se que o *solver* GS-RB associado à restrição por ponderação parcial apresenta o menor tempo de CPU para ambos os blocos. Em relação ao tipo de restrição para o SE o menor tempo de CPU obtido ao utilizar restrição por ponderação parcial. Para o bloco EP o tipo de restrição não afeta significativamente o tempo de CPU.

A análise também foi realizada para problemas isotrópicos. O algoritmo utilizado neste caso é o EP. A Tab. 6.6 apresenta o tempo de CPU obtido para uma malha 513x513. Verifica-se que o menor tempo de CPU é obtido com o uso do *solver* GS-RB e que o tipo de restrição não afeta significativamente o tempo de CPU.

**Tabela 6.6: Tempo de CPU (s) para a análise do bloco EP, malha 513x513 e  $Q = 1$ .**

Restrição	Laplace Senoidal		Laplace Linear		Poisson	
	Solver		Solver		Solver	
	GS-LEX	GS-RB	GS-LEX	GS-RB	GS-LEX	GS-RB
I	0,149	0,079	0,148	-	0,147	-
HW	0,154	0,079	0,151	0,078	0,144	0,077
FW	0,172	0,078	0,173	0,080	0,171	0,078

Os parâmetros ótimos obtidos para as equações de Laplace senoidal, Laplace linear e Poisson, algoritmo EP e SE-EP, para problemas isotrópicos e anisotrópicos, respectivamente, foram:

- Para problemas isotrópicos, ou seja,  $Q = 1$ , o *solver* que apresentou o menor tempo de CPU foi o Gauss-Seidel *red-black*. O tipo de restrição não afeta significativamente o tempo de CPU.
- Para problemas anisotrópicos, ou seja,  $Q \neq 1$ , o *solver* Gauss-Seidel *red-black* apresentou o menor tempo de CPU para os dois blocos do algoritmo (SE e EP). Para o bloco do SE o melhor tipo de restrição foi a ponderação parcial. Para o bloco do EP o tipo de restrição não afetou significativamente o tempo de CPU.

Verifica-se que o tempo de CPU utilizando os parâmetros ótimos é muito próximo para as três equações em estudo. Neste trabalho denomina-se algoritmo ótimo o algoritmo que utiliza o *solver* e o operador de restrição que obtiveram o menor tempo de CPU. Para problemas isotrópicos ele é denominado de EP ótimo e para problemas anisotrópicos de SE-EP ótimo. Os algoritmos EP e SE-EP com os parâmetros utilizados no capítulo 5 são denotados por EP padrão e SE-EP padrão. A Tab. 6.7 apresenta os parâmetros (*solvers*, restrição e  $\nu$ ) utilizados nos algoritmos SE-EP padrão (utilizado como referência) e SE-EP ótimo. A Tab. 6.8 apresenta os parâmetros (*solvers*, restrição e  $\nu$ ) utilizados nos algoritmos EP e EP ótimo.

**Tabela 6.7: Parâmetros para os algoritmos SE-EP padrão e SE-EP ótimo.**

	<b>SE-EP padrão</b>	<b>SE-EP ótimo</b>
<i>Solver</i>	GS-LEX	GS-RB
Restrição	Injeção	PW para o SE FW para o EP
Número de iterações internas	LS: $\nu = 2$ para $Q < 1$ $\nu = 4$ para $Q > 1$ LL: $\nu = 4$ para $\forall Q$ Po: $\nu = 3$ para $\forall Q$	1

**Tabela 6.8: Parâmetros para os algoritmos EP e EP ótimo.**

	<b>EP</b>	<b>EP ótimo</b>
<i>Solver</i>	GS-LEX	GS-RB
Restrição	Injeção	PW
Número de iterações internas	2	2

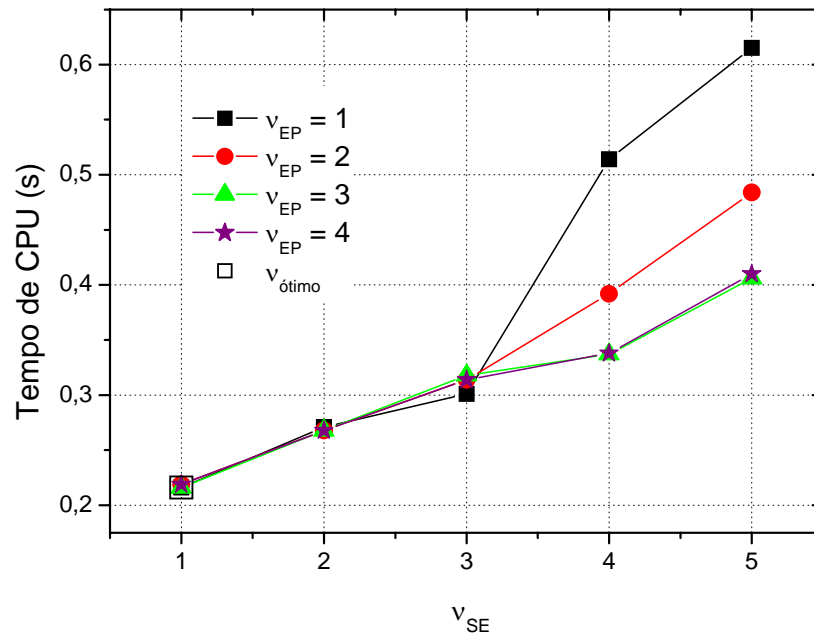
Os parâmetros referentes ao *solver* e tipo de restrição apresentados nas Tabs. 6.7 e 6.8 são os valores ótimos obtidos para os algoritmos SE-EP ótimo e EP ótimo. O número de iterações internas utilizado foi o valor ótimo obtido para o algoritmo SE-EP. Na seqüência é apresentada a análise do número de iterações internas para o algoritmo SE-EP.

### Estudo do número de iterações internas ( $v$ )

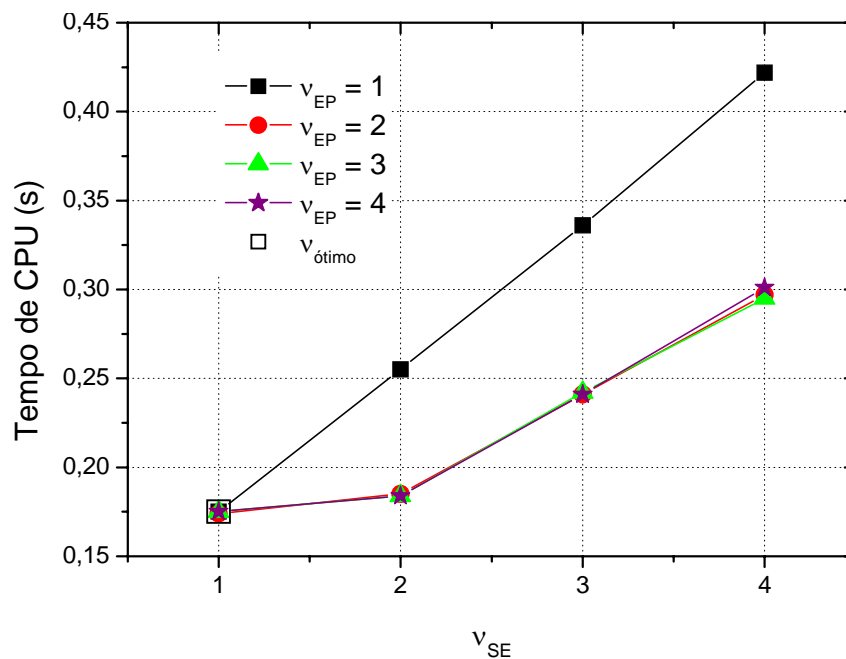
Nesta subsecção é feita uma análise do  $v_{\text{ótimo}}$  para os algoritmos SE-EP ótimo e EP ótimo. Utilizam-se os parâmetros ótimos obtidos acima. Para o EP, optou-se pela restrição por ponderação completa. Denota-se por  $v_{EP}$  o número de iterações internas para o EP e  $v_{SE}$  o número de iterações internas para o SE. O  $v_{\text{ótimo}}$  é determinado através de uma combinação de  $v_{EP}$  e  $v_{SE}$ . Verifica-se que para  $Q < 1$  tem-se  $v_{EP} = v_{SE} = 1$  e para  $Q > 1$  tem-se  $v_{EP} = 2$  e  $v_{SE} = 1$ . Apesar dos valores de  $v$  serem diferentes para  $Q > 1$  também pode-se utilizar  $v_{EP} = v_{SE} = 1$ , visto que o acréscimo no tempo de CPU é pequeno. Por exemplo, em uma malha de 262.144 elementos este acréscimo é de 0,14; 0,02 e 0,001 segundo para  $Q = 4, 16$  e  $64$ , respectivamente.

As Fig. 6.12 e 6.13 apresentam o estudo do  $v$  para o algoritmo SE-EP ótimo para a equação de Laplace senoidal. Considera-se um problema com 262.144 elementos e razões  $Q = 1/64$  (Fig. 6.11) e  $Q = 64$  (Fig. 6.12). Para as duas razões de aspecto, verifica-se que o menor tempo de CPU é obtido para  $v_{EP} = v_{SE} = 1$ . O estudo do  $v$  também foi realizado para as razões de aspecto  $Q = 1/16, 1/4, 4$  e  $16$ . Os resultados obtidos foram similares.

Observou-se que para o algoritmo SE-EP ótimo, o  $v_{\text{ótimo}}$  é o mesmo para todas as equações em estudo e para qualquer  $Q \neq 1$ , ou seja, a oscilação da função devido as condições de contorno não interfere em  $v_{\text{ótimo}}$ . Vale lembrar que para o algoritmo SE-EP padrão, o  $v_{\text{ótimo}}$  varia de acordo com a equação. Também em relação ao algoritmo SE-EP quando a oscilação da equação não for igual nas duas direções obtém-se um  $v_{\text{ótimo}}$  diferente para razões de aspecto menores que a unidade e maiores que a unidade. O algoritmo proposto nesta tese, SE-EP ótimo é mais estável possuindo um único  $v_{\text{ótimo}}$  para todas as equações e razões de aspecto em estudo. Para o algoritmo EP ótimo também obteve-se o mesmo  $v_{\text{ótimo}}$  para todas as equações em estudo.

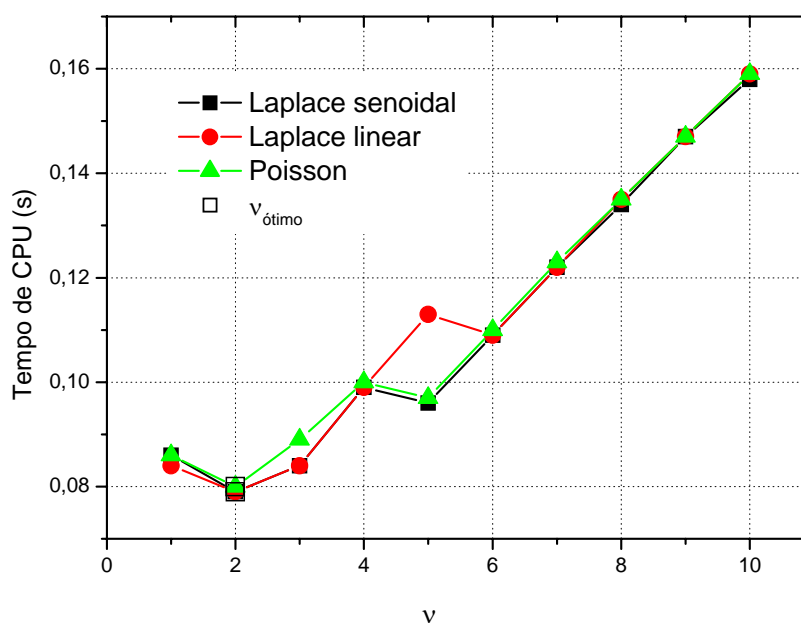


**Figura 6.12:** Tempo de CPU *versus*  $v_{EP}$  para  $Q = 1/64$ ,  $E = 262.144$ , SE-EP ótimo e equação de Laplace senoidal.



**Figura 6.13:** Tempo de CPU *versus*  $v_{EP}$  para  $Q = 64$ ,  $E = 262.144$ , SE-EP ótimo e equação de Laplace senoidal.

A Fig. 6.14 apresenta o estudo do  $v$  para o algoritmo EP ótimo para as equações de Laplace senoidal, Laplace linear e Poisson. Considera-se um problema com 262.144 elementos e uma malha isotrópica ( $Q = 1$ ). Verifica-se que para todas as equações em estudo o menor tempo de CPU é obtido para  $v = 2$ .



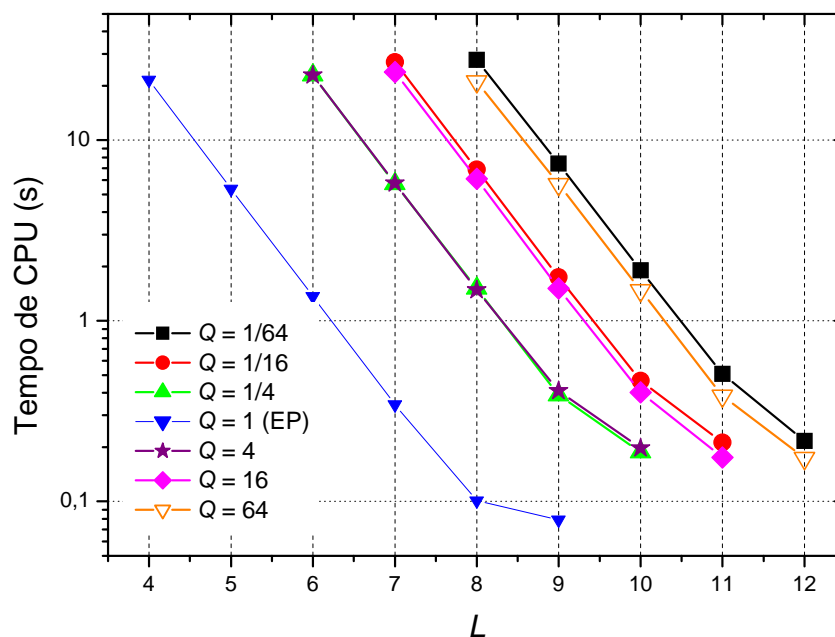
**Figura 6.14:** Tempo de CPU *versus*  $v$  para  $Q = 1$ ,  $E = 262.144$ , EP ótimo para as equações de Laplace senoidal, Laplace linear e Poisson.

### Estudo do número de níveis ( $L$ )

Nesta seção é feita uma análise do número de níveis para a equação de Laplace senoidal. Utiliza-se as mesmas malhas e razões de aspecto utilizadas no estudo do  $v_{\text{ótimo}}$ .

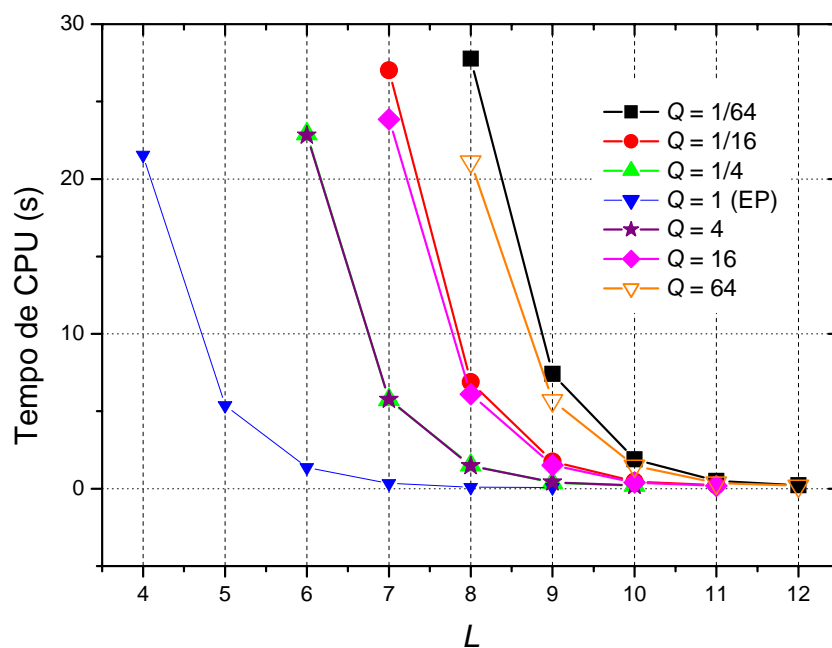
O algoritmo SE-EP possui os seguintes parâmetros ótimos (*solver* GS-RB para ambos os blocos; restrição por ponderação parcial para o bloco do SE e restrição por ponderação completa para o bloco do EP).

A Fig. 6.15 apresenta o tempo de CPU *versus* número de níveis para a equação de Laplace senoidal, algoritmo SE-EP ótimo e um problema com 262.144 elementos. Verifica-se que os menores tempos de CPU ocorrem ao se utilizar um número maior de níveis:  $L_{\text{máximo}}, L_{\text{máximo}} - 1$  ou  $L_{\text{máximo}} - 2$ . Utilizando estes números de níveis, a razão de aspecto não afeta significativamente o tempo de CPU. Verifica-se que ao utilizar um número menor de níveis a razão de aspecto interfere no tempo de CPU. Para  $Q = 1$ , utiliza-se o algoritmo EP, mais adequado para problemas isotrópicos.



**Figura 6.15:** Tempo de CPU *versus* número de níveis ( $L$ ) para  $E = 262.144$ , algoritmo SE-EP ótimo e equação de Laplace senoidal.

A Fig. 6.16 apresenta o gráfico da Fig. 6.15 em uma escala linear. Desta forma pode-se verificar que quanto mais anisotrópico for o problema ( $0 < Q \ll 1$  e  $Q \gg 1$ ), maior a influência do número de níveis sobre o tempo de CPU.



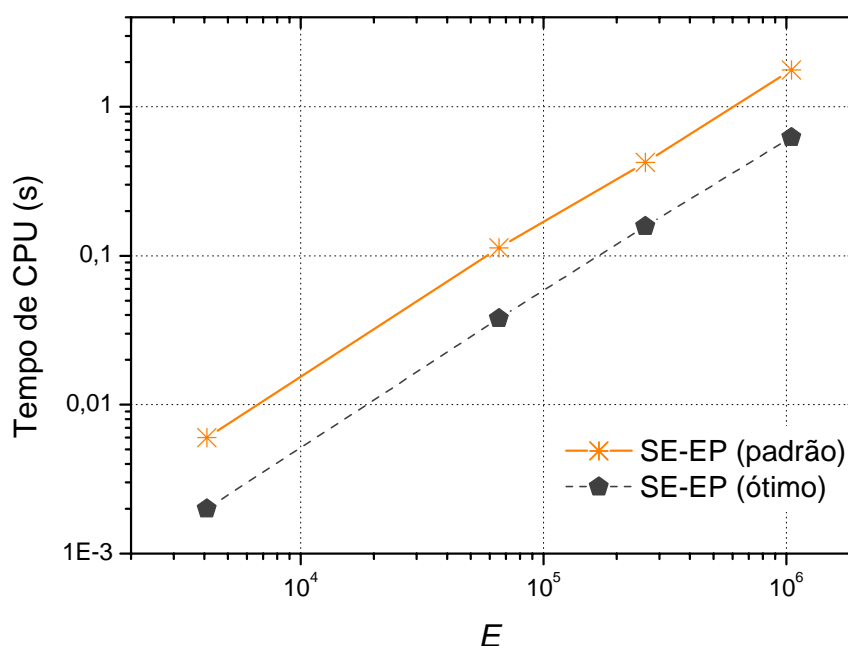
**Figura 6.16:** Tempo de CPU *versus* número de níveis ( $L$ ) para  $E = 262.144$ , algoritmo SE-EP ótimo e equação de Laplace senoidal.



## Comparação entre os algoritmos

Nas Figs. 6.17 e 6.18 são feitas comparações entre os algoritmos SE-EP padrão e SE-EP ótimo para a equação de Laplace Senoidal. A Fig. 6.17 apresenta a razão de aspecto  $Q = 1/64$  e a Fig. 6.18 a razão  $Q = 64$ . Verifica-se que o algoritmo SE-EP ótimo obteve um menor tempo de CPU em relação ao SE-EP padrão. Esta redução no tempo de CPU é em média 2,3 vezes. Os resultados foram similares para outras razões de aspecto e para as três equações em estudo. Os coeficientes angulares obtidos na linearização dos dados das Figs. 6.17 e 6.18 encontram-se na Tab. 6.11.

Na Fig. 6.19 faz-se uma comparação entre os algoritmos EP e EP ótimo para a equação de Laplace senoidal e  $Q = 1$ . Verifica-se que o tempo de CPU do algoritmo EP ótimo é menor em relação ao EP. Os resultados obtidos para as equações de Laplace linear e Poisson são similares e encontram-se no Apêndice E.



**Figura 6.17:** Tempo de CPU *versus* número de elementos ( $E$ ) para o algoritmo SE-EP,  $Q = 1/64$  e equação de Laplace senoidal.

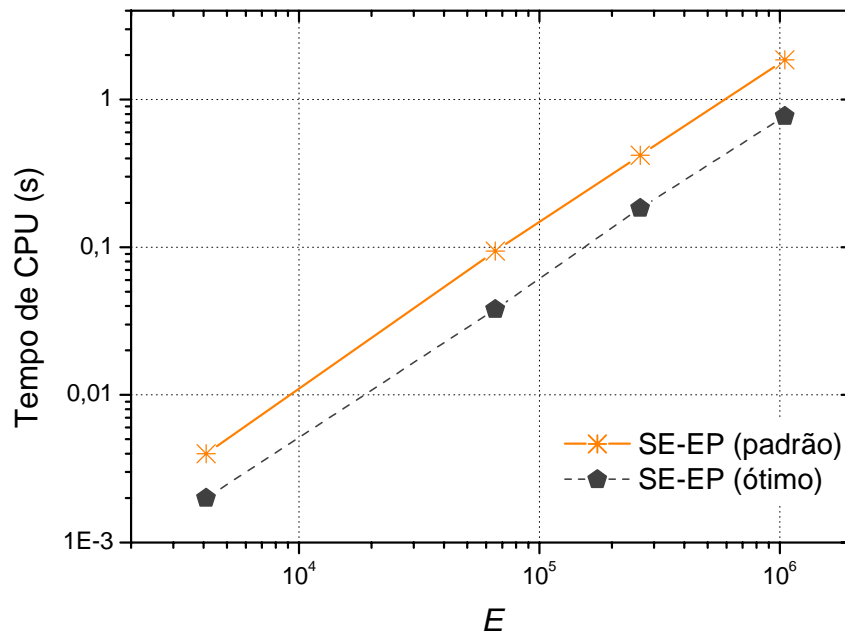


Figura 6.18: Tempo de CPU *versus* número de elementos ( $E$ ) para o algoritmo SE-EP,  $Q = 64$ , equação de Laplace senoidal.

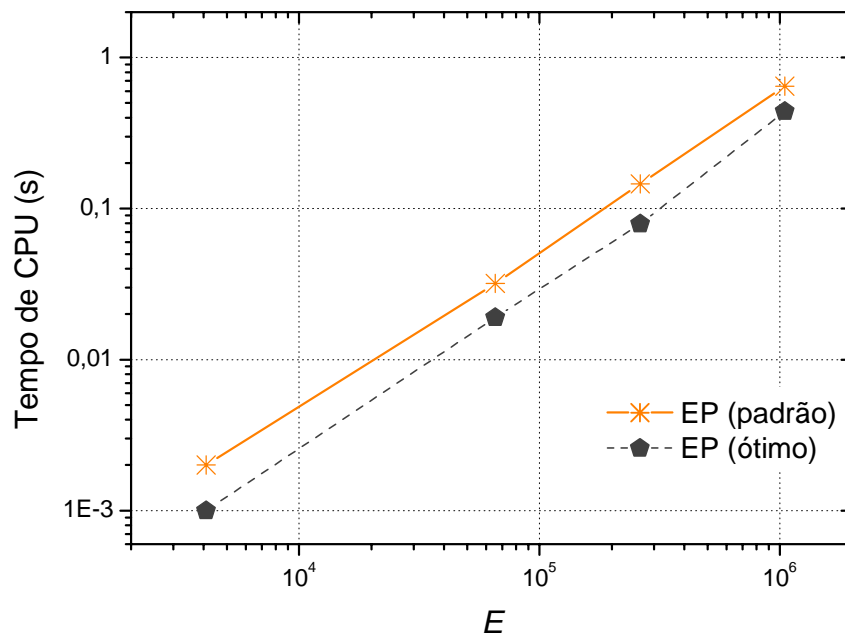


Figura 6.19: Tempo de CPU *versus* número de elementos ( $E$ ) para o algoritmo EP e equação de Laplace senoidal e  $Q = 1$ .

### Singlegrid versus multigrid

Esta seção tem por objetivo comparar o desempenho dos métodos *singlegrid* (SG) e *multigrid* (MG) em problemas isotrópicos e anisotrópicos. O estudo foi realizado para a equação de Laplace senoidal. Primeiramente é feita a descrição dos parâmetros utilizados em problemas isotrópicos. Para ambos os métodos, *singlegrid* e *multigrid* foram consideradas as malhas  $N = 3 \times 3, 5 \times 5, 9 \times 9, 17 \times 17, 33 \times 33, 65 \times 65, 129 \times 129, 257 \times 257$  e  $513 \times 513$ . Para o método *singlegrid* foram utilizados os solvers GS-LEX e GS-RB. Para o método *multigrid* foram consideradas duas variações: *multigrid* com algoritmo EP e EP ótimo. Os parâmetros para o MG são os mesmos utilizados no capítulo 5 (GS-LEX e restrição por injeção) e os valores ótimos obtidos:  $L_{máximo}$  e  $\nu = 2$ . Para o EP ótimo foram utilizados os parâmetros GS-RB, restrição por ponderação completa,  $L_{máximo}$  e  $\nu = 2$ .

A Fig. 6.20 mostra os resultados obtidos para a equação de Laplace senoidal considerando-se malhas isotrópicas. Verifica-se que o método EP (*multigrid*) é significativamente mais rápido que o SG, cerca de sete mil vezes para uma malha  $513 \times 513$  (comparando-se SG (GS-LEX) e EP). Analisando o método SG, foi verificado que para uma malha  $513 \times 513$ , o SG (GS-RB) é cerca de três vezes mais rápido em relação ao SG (GS-LEX). Para o método *multigrid* verificou-se que o EP ótimo é mais rápido que o EP padrão, cerca de 2 vezes para uma malha  $513 \times 513$ . Pode-se concluir que o *multigrid* com EP é muito mais rápido que o SG para problemas isotrópicos.

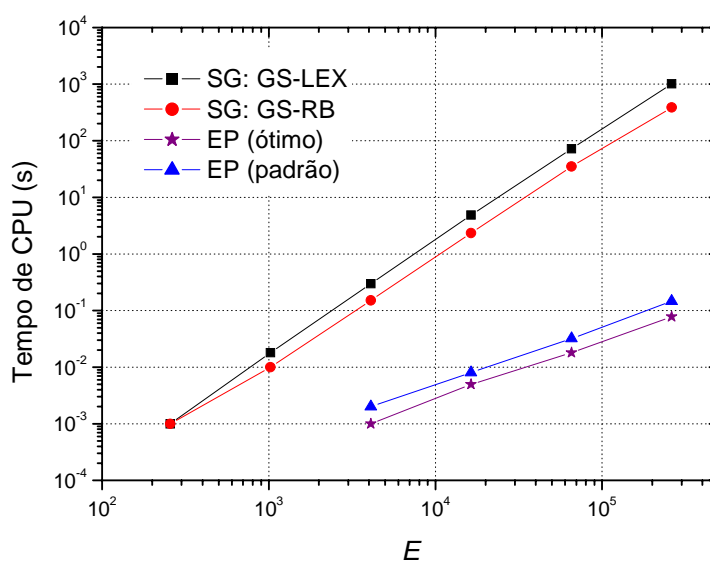
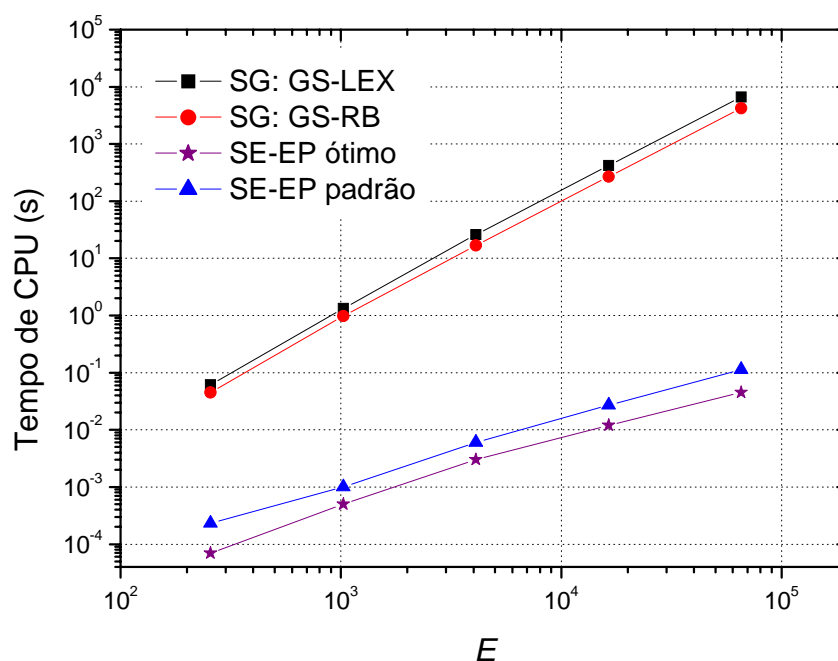


Figura 6.20: Tempo de CPU versus número de elementos ( $E$ ) para um problema isotrópico e equação de Laplace senoidal.

A seguir apresenta-se uma comparação dos métodos SG (GS-LEX e GS-RB) e MG (SE-EP padrão e SE-EP ótimo) para a equação de Laplace senoidal e malhas anisotrópicas. Foram utilizadas as razões de aspecto  $Q = 1/64, 1/16, 16$  e  $64$  e as malhas  $N = 129 \times 3, 257 \times 5, 513 \times 9, 1025 \times 17$  e  $2049 \times 33$ . Para o método SG foram considerados os mesmos parâmetros definidos para problemas isotrópicos. Para o SE-EP ótimo foram utilizados o algoritmo SE-EP, *solver* GS-RB, restrição por ponderação parcial para o SE e ponderação completa para o EP,  $L_{máximo}$  e  $\nu = 1$ . A Fig. 6.21 mostra os resultados obtidos para a razão de aspecto  $Q = 1/64$ . Verifica-se que o método SE-EP ótimo é significativamente mais rápido que o SG (GS-RB), cerca de nove mil vezes para uma malha  $2049 \times 33$ . Analisando o método SG foi verificado que para uma malha  $2049 \times 33$ , o SG (GS-RB) é cerca de duas vezes mais rápido em relação ao SG (GS-LEX). Verifica-se também que o algoritmo SE-EP ótimo é cerca de três vezes mais rápido que o algoritmo SE-EP padrão para uma malha  $2049 \times 33$ . Pode-se concluir que o método *multigrid* com SE-EP é muito mais rápido que o SG também para problemas anisotrópicos.



**Figura 6.21:** Tempo de CPU *versus* número de elementos ( $E$ ) para um problema anisotrópico com  $Q = 1/64$ , e equação de Laplace senoidal.

A eficiência do algoritmo SE-EP ótimo em relação ao SG: GS-RB pode ser verificada através do *speed-up*. O *speed-up* foi calculado através da Eq. (5.2). Neste caso o algoritmo

“A” é o SG: GS-RB e o algoritmo “B” SE-EP ótimo, ou seja,  $S_p = \frac{t_{CPU}(SG : GS - RB)}{t_{CPU}(SE - EP \text{ ótimo})}$ . A

Tab. 6.9 apresenta o *speed-up* do método *singlegrid* com *solver* Gauss-Seidel *red-black* (SG: GS-RB) em relação ao método *multigrid* (SE-EP ótimo) para problemas isotrópicos e anisotrópicos e malhas com 256, 1.024, 4.096 e 16.384 elementos. Pode-se verificar que o SE-EP ótimo é mais rápido em relação ao SG: GS-RB para todas as malhas em estudo. Tanto para problemas isotrópicos como para problemas anisotrópicos. Verifica-se também que ganho no tempo de CPU é maior para problemas anisotrópicos do que para problemas isotrópicos. Os valores não preenchidos na Tab. 6.9 referem-se ao tempo de CPU muito próximo de zero, onde não foi possível calcular o *speed-up*.

**Tabela 6.9: *Speed-up* do SG: GS-RB em relação SE-EP ótimo para a equação de Laplace senoidal.**

$E$	Problema isotrópico ( $Q = 1$ )	Problema anisotrópico ( $Q = 1/64$ )
256	-	645,62
1.024	-	1946,00
4.096	152	5578,00
16.384	470	22397,17
65.536	1952,23	93204,51

### Análise de complexidade

Para as malhas e razões de aspecto definidas na Tab. 5.9 fez-se uma análise da complexidade do algoritmo SE-EP ótimo e as equações de Laplace senoidal, Laplace linear e Poisson. Os espaços não preenchidos nas Tab.s 6.10 e 6.11 são referentes as simulações que não atingiram a tolerância estabelecida. Calculou-se o coeficiente  $c$  e o expoente  $p$ , obtido pelo método dos mínimos quadrados, para a Eq. 5.1. A Tab. 6.10 apresenta os coeficientes  $c$  e as ordens  $p$  dos ajustes de curvas obtidos para os algoritmos em estudo. Verifica-se que todas as ordens  $p$  são próximas de 1, para todas as razões de aspecto, o que indica que o algoritmo SE-EP ótimo está bem próximo do método *multigrid* ideal.

Tabela 6.10: Valores de  $c$  e  $p$  para Eq. (5.1) e algoritmo SE-EP ótimo.

$Q$	Laplace senoidal		Laplace linear		Poisson	
	$c$	$p$	$C$	$p$	$c$	$p$
1/64	5,75E-07	1,02	-	-	-	-
1/16	5,47E-07	1,03	2,76E-07	1,07	5,35E-07	1,03
1	1,14E-07	1,09	9,70E-04	1,09	<b>1,16E-07</b>	<b>1,09</b>
16	2,38E-07	1,08	5,28E-07	1,03	<b>2,35E-07</b>	<b>1,08</b>
64	2,61E-07	1,07	-	-	-	-

A Tab. 6.11 faz uma comparação entre o coeficiente  $c$  e a ordem  $p$  para a equação de Laplace senoidal obtidos com os algoritmos SE-EP padrão e SE-EP ótimo. Verifica-se que todas as ordens  $p$  são próximas de 1, para todas as razões de aspecto, o que indica que os dois algoritmos (SE-EP padrão e SE-EP ótimo) estão bem próximos do método *multigrid* ideal.

Tabela 6.11: Valores de  $c$  e  $p$  para SE-EP padrão e SE-EP ótimo e equação de Laplace senoidal.

$Q$	$C$		$p$	
	SE-EP padrão	SE-EP ótimo	SE-EP padrão	SE-EP ótimo
1/64	1,24E-06	5,75E-07	1,02	1,02
1/16	1,51E-06	5,47E-07	1,00	1,03
1	3,32E-07	1,14E-07	1,04	1,09
16	7,67E-07	2,38E-07	1,06	1,08
64	4,07E-07	2,61E-07	1,11	1,07

A eficiência do algoritmo SE-EP ótimo em relação ao SE-EP padrão pode ser verificada através do *speed-up*. O *speed-up* foi calculado através da Eq. (5.2). Neste caso o algoritmo “A” é o SE-EP padrão e o algoritmo “B” SE-EP ótimo (com o número ótimo de iterações internas), ou seja,  $S_p = \frac{t_{CPU}(\text{SE-EP padrão})}{t_{CPU}(\text{SE-EP ótimo})}$ . A Tab. 6.12 apresenta o *speed-up* para uma malha com 262.144 elementos para algumas malhas definidas na Tab. 5.9. O algoritmo SE-EP ótimo reduz o tempo de CPU em uma média 2,3 vezes em relação ao algoritmo SE-EP padrão.

Tabela 6.12: *Speed-up* do SE-EP padrão em relação SE-EP ótimo.

$Q$	Laplace senoidal	Laplace linear	Poisson
1/16	2,66	2,68	2,51
1	1,87	1,87	2,03
16	2,28	2,18	2,88

Através dos resultados obtidos nas simulações conclui-se que para problemas isotrópicos ( $Q = 1$ ), o menor tempo de CPU é obtido com o uso do algoritmo EP, *solver* Gauss-Seidel *red-black* e  $\nu = 2$ . O tipo de restrição não afetou significativamente o tempo de CPU. Para problemas anisotrópicos ( $Q \neq 1$ ), o menor tempo de CPU é obtido com o uso do algoritmo SE-EP, *solver* Gauss-Seidel *red-black*. Para o SE restrição por ponderação parcial e para o EP restrição por ponderação completa e  $\nu = 1$ , tanto para o SE como para o EP, ou seja, o algoritmo SE-EP ótimo.

### 6.3 COMPARAÇÃO COM A LITERATURA

A literatura apresenta vários estudos envolvendo problemas anisotrópicos e algoritmos de engrossamento. Zhang (2002) aborda problemas anisotrópicos acoplando-se a técnica de semi-engrossamento seguida de engrossamento padrão (engrossamento em ambas as direções) intitulado como “*partial semicoarsening*”. Foram feitas comparações para a equação de Poisson bidimensional com a técnica denominada de engrossamento padrão (*full coarsening*). Zhang (2002) comparou estas técnicas para as razões de aspecto  $Q = 1, 1/2, 1/4, 1/8$  e  $1/16$  (razões de aspecto modestas) e as malhas com  $E = 4.096, 8.192, 16.384, 32.768, 65.536, 131.072, 262.144, 524.288$  e  $1.048.576$ . Zhang constatou que o algoritmo *partial semicoarsening* com os suavizadores *red-black* e *four-color* Gauss-Seidel são eficientes para as anisotropias estudadas.

Este trabalho confirma os resultados obtidos por Zhang para as equações de Laplace senoidal, Laplace linear e Poisson, porém utilizando-se as razões de aspecto  $Q = 1/16384, 1/4096, 1/1024, 1/256, 1/64, 1/16, 1/4, 1, 4, 16, 64, 256, 1.024, 4.096, 16.384$  e as malhas  $E = 4, 16, 64, 256, 1.024, 4.096, 16.384, 65.536, 262.144, 1.048.576, 4.194.304, 16.777.216$  e  $67.108.864$ . Também foram propostos outros tipos de restrição (meia ponderação geométrica, ponderação geométrica completa e ponderação parcial) além de serem utilizados os tipos de restrição já existentes na literatura (injeção, meia ponderação e ponderação completa). Verificou-se também o *solver*,  $\nu$  e tipo de restrição mais apropriados para cada bloco do algoritmo SE-EP.

## 6.4 CONCLUSÃO DO CAPÍTULO

Neste capítulo foram resolvidos numericamente três problemas bidimensionais lineares de condução de calor, governados pelas equações de Laplace e Poisson, com condições de contorno de Dirichlet. Utilizou-se o esquema de aproximação CDS e o esquema de correção CS do método *multigrid* geométrico. Foram apresentados seis tipos de restrição em problemas anisotrópicos: injeção (I), meia ponderação (HW), ponderação completa (FW), meia ponderação geométrica (GHW), ponderação geométrica completa (GFW) e ponderação parcial (PW). Com base nos resultados obtidos neste capítulo, verificou-se que, para as equações estudadas:

- 1) Para os mesmos parâmetros ( $\nu$ ,  $Q$ ,  $L$  e  $E$ ), o algoritmo SE-EP ótimo apresentou um menor tempo de CPU comparado ao SE-EP padrão. A melhora no tempo de CPU ao utilizar-se SE-EP ótimo está entre duas e três vezes.
- 2) Para todos os algoritmos de engrossamento analisados o  $L_{máximo}$  resulta praticamente no menor tempo de CPU para qualquer razão de aspecto e qualquer tamanho de problema. Quanto maior a anisotropia, maior a influência do número de níveis sobre o tempo de CPU.
- 3) Para problemas isotrópicos ( $Q = 1$ ) o tipo de restrição não afeta significativamente o tempo de CPU. As restrições meia ponderação geométrica e ponderação geométrica completa convergem mais rapidamente em relação à meia ponderação e a ponderação completa para problemas anisotrópicos. A restrição por ponderação parcial apresenta entre todos os operadores de restrição, o menor tempo de CPU para os problemas anisotrópicos.
- 4) Para o algoritmo SE-EP ótimo obteve-se um único  $\nu_{ótimo}$  para todas as equações em estudo e para qualquer  $Q \neq 1$ . O menor tempo de CPU foi obtido para  $\nu_{SE} = \nu_{EP} = 1$ .
- 5) Para problemas isotrópicos o menor tempo de CPU é obtido com  $\nu = 2$  e algoritmo EP.
- 6) Para os métodos *singlegrid* e *multigrid*, problemas isotrópicos e anisotrópicos o *solver* Gauss-Seidel *red-black* é significativamente mais rápido que o Gauss-Seidel lexicográfico.
- 7) O método *multigrid* com algoritmo SE-EP ótimo é significativamente mais rápido que o *singlegrid* para malhas isotrópicas e anisotrópicas. Para uma malha isotrópica



513x513 é cerca de sete mil vezes mais rápido e para uma malha anisotrópica 2049x33, cerca de seis mil vezes mais rápido.

- 8) O comportamento qualitativo dos parâmetros estudados é o mesmo para as equações de Laplace senoidal, Laplace linear e Poisson.

## 6.5 PARÂMETROS COM O MELHOR DESEMPENHO MÉDIO

Com base nos resultados obtidos neste capítulo obteve-se os seguintes parâmetros com melhor desempenho médio que podem ser utilizados para problemas iguais ou similares aos estudados.

Problemas bidimensionais lineares isotrópicos:

- Algoritmo: EP.
- *Solver*: GS-RB.
- O tipo de restrição não afeta significativamente o tempo de CPU.
- Número de iterações internas:  $\nu_{EP} = 2$ .
- Número de níveis:  $L_{máximo}$ .

Problemas bidimensionais lineares anisotrópicos:

- Algoritmo: SE-EP.
- *Solver*: GS-RB.
- Restrição para o SE: ponderação parcial.
- Restrição para o EP: ponderação completa.
- Número de iterações internas:  $\nu_{SE} = \nu_{EP} = 1$ .
- Número de níveis:  $L_{máximo}$ .

## 7 CONCLUSÃO

A conclusão deste trabalho está dividida em cinco seções: conclusão geral, contribuições, parâmetros com desempenho ótimo, extrapolações e trabalhos futuros; descritos a seguir.

### 7.1 CONCLUSÃO GERAL

Neste trabalho foram resolvidos numericamente três problemas bidimensionais lineares de condução de calor, governados pelas equações de Laplace (senoidal e linear) e Poisson, todas com condições de contorno de Dirichlet. Utilizou-se o esquema de aproximação CDS e o esquema de correção CS do método *multigrid* geométrico. Foram usados cinco algoritmos para problemas anisotrópicos: engrossamento padrão (EP), semi-engrossamento (SE), semi-engrossamento completo (SEC), engrossamento padrão seguido de semi-engrossamento (EP-SE) e semi-engrossamento seguido de engrossamento padrão (SE-EP). Com base nos resultados obtidos neste trabalho, verificou-se que:

- 1) O algoritmo SE-EP ótimo obteve uma redução no tempo de CPU em média 2,3 vezes em relação ao algoritmo SE-EP padrão.
- 2) Para os mesmos parâmetros ( $\nu$ ,  $Q$ ,  $L$  e  $E$ ), o algoritmo SE-EP padrão apresentou o menor tempo de CPU entre os cinco algoritmos analisados para os problemas anisotrópicos. A melhora no tempo de CPU ao utilizar-se SE-EP está entre 2 a 2786 vezes em comparação com o engrossamento padrão.
- 3) A variação da razão de aspecto resulta em grande variação do  $\nu_{\text{ótimo}}$  para os algoritmos EP, SE, SEC e EP-SE. Para o algoritmo SE-EP, o  $\nu_{\text{ótimo}}$  permanece constante em todas as razões de aspecto. Para o algoritmo SE-EP ótimo os parâmetros que apresentam o menor tempo de CPU são:  $\nu_{SE} = \nu_{EP} = 1$  para  $Q \neq 1$ . Para  $Q = 1$  (problema isotrópico) o algoritmo EP com  $\nu = 2$  apresenta o menor tempo de CPU.

- 4) Para todos os algoritmos de engrossamento analisados o  $L_{máximo}$  resulta praticamente no menor tempo de CPU para qualquer razão de aspecto e qualquer tamanho de problema. Quanto maior a anisotropia, maior a influência do número de níveis sobre o tempo de CPU.
- 5) Para problemas isotrópicos o tipo de restrição não afeta significativamente o tempo de CPU.
- 6) A restrição por ponderação parcial apresenta o menor tempo de CPU para os problemas anisotrópicos.
- 7) As restrições por meia ponderação geométrica e ponderação geométrica completa convergem mais rapidamente para problemas anisotrópicos em relação às suas similares meia ponderação e ponderação completa.
- 8) Para os métodos *singlegrid* e *multigrid*, problemas isotrópicos e anisotrópicos o *solver* Gauss-Seidel *red-black* é significativamente mais rápido que o Gauss-Seidel lexicográfico.
- 9) O método *multigrid* otimizado é significativamente mais rápido que o *singlegrid* para malhas isotrópicas e anisotrópicas. Para uma malha isotrópica 513x513 é cerca de 6.904 vezes mais rápido e para uma malha anisotrópica 2049x33, cerca de 5.524 vezes mais rápido.
- 10) O comportamento qualitativo dos parâmetros estudados é o mesmo para as equações de Laplace senoidal, Laplace linear e Poisson.

## 7.2 CONTRIBUIÇÕES

Este trabalho contribui com a literatura no sentido de que:

- 1) Obteve-se uma redução no tempo de CPU necessário para resolver as equações de Laplace bidimensional (senoidal e linear) e Poisson bidimensional em malhas estruturadas uniformes e uniformes por direção com alta razão de aspecto.
- 2) Foram apresentados novos operadores de restrição para problemas anisotrópicos (meia ponderação geométrica, ponderação geométrica completa e ponderação parcial), os

quais apresentaram melhores resultados em comparação com os operadores já existentes.

- 3) Confirmou-se a afirmação obtida por Zhang (2002) de que o algoritmo SE-EP é o algoritmo mais adequado para problemas anisotrópicos com  $0 < Q \ll 1$ . Esta conclusão foi estendida para razões de aspecto  $Q \gg 1$  e para  $Q$  muito maiores que os estudados por Zhang (2002). Problemas de interesse da Engenharia.
- 4) Apresentou-se um algoritmo de engrossamento SE-EP ótimo, no qual os parâmetros ótimos são os mesmos para todas as equações em estudo e para qualquer razão de aspecto. Este algoritmo também apresenta menor tempo de CPU em comparação com os demais algoritmos em estudo.
- 5) Verificou-se que o método *multigrid* é significativamente mais rápido que o *singlegrid* para problemas anisotrópicos.

### 7.3 PARÂMETROS COM MELHOR DESEMPENHO MÉDIO

Com base nos resultados obtidos nesta tese obteve-se os seguintes parâmetros com melhor desempenho médio que podem ser utilizados para problemas iguais ou similares aos estudados.

Problemas bidimensionais lineares isotrópicos:

- Algoritmo: EP.
- *Solver*: GS-RB.
- O tipo de restrição não afeta significativamente o tempo de CPU.
- Número de iterações internas:  $\nu = 2$ .
- Número de níveis:  $L_{máximo}$ .

Problemas bidimensionais lineares anisotrópicos:

- Algoritmo: SE-EP.
- *Solver*: GS-RB.
- Restrição para o SE: ponderação parcial.
- Restrição para o EP: ponderação completa.
- Número de iterações internas:  $\nu_{SE} = \nu_{EP} = 1$ .
- Número de níveis:  $L_{máximo}$ .

## 7.4 EXTRAPOLAÇÕES

Os problemas abordados nesta tese são problemas modelados por equações simples, onde todos apresentam soluções analíticas conhecidas. Espera-se que as conclusões possam ser estendidas também para equações mais gerais e para problemas que não possuam soluções analíticas conhecidas. Apresentam-se a seguir algumas possibilidades de extrapolação dos resultados obtidos nesta 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 visto nos capítulo 6;
- 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, tanto na fase térmica do processo de resolução, como na fase elástica;

## 7.5 TRABALHOS FUTUROS

Apresentam-se a seguir algumas questões que abrem caminho para novas pesquisas:

- **Critério de parada:** utilizou-se um critério de parada fixo, especificando o número de iterações internas em cada ciclo. Propõe-se o estudo de um critério de parada dinâmico que consiste em monitorar a taxa de convergência da solução numérica.
- **Ciclos:** utilizou-se o ciclo V. Propõe-se o estudo de outros tipos de ciclos como dente-de-serra e Hortmann e algumas variações nestes ciclos já existentes.
- **Full multigrid:** utilizou-se o método *multigrid* com ciclo V. Propõe-se o estudo do *full multigrid* para problemas anisotrópicos.
- **Operadores de prolongação:** utilizou-se o operador de prolongação por interpolação bilinear. Propõe-se o estudo de outros operadores, como por exemplo, o operador de

interpolação corrigida utilizado por Zhang (2002). Propõe-se também a análise de novos operadores de prolongação mais adequados para problemas anisotrópicos.

- **Solvers:** utilizou-se o *solver* Gauss-Seidel lexicográfico e *red-black*. Propõe-se o estudo do *solver* Gauss-Seidel quatro cores e do gradiente conjugado.
- **Outras anisotropias:** 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).
- **Problemas 3D:** estudou-se problemas anisotrópicos bidimensionais. Propõe-se o estudo de problemas anisotrópicos tridimensionais.
- **Multigrid paralelo:** utilizar técnicas de programação paralela com o objetivo de melhorar o desempenho do método *multigrid*.
- **Problemas transientes:** considerou-se o regime permanente. Propõe-se o estudo de problemas de transferência de calor e massa em regime transiente.
- **Análise de erros:** verificou-se qual o algoritmo de engrossamento resulta no menor tempo de CPU para uma dada razão de aspecto. Propõe-se uma análise das razões de aspecto de forma a verificar qual razão de aspecto implicará o menor erro de discretização.

## REFERÊNCIAS

- ARAKI, K.L. **Malha de 100 elementos (101 nós) à esquerda e 25 elementos (26 nós) à direita**. Ilustração, 2007.
- BAKHVALOV, N. S. On the Convergence of Relaxation Method with Natural Constraints on the Elliptic operator. **URSS Comp. Math. Phys**, v.1, n.5, 1966.
- BRANDT, A., **Multi-Level Adaptive Solutions to Boundary-Value Problems, Mathematics of Computation**, v. 31, pp. 333-390, 1977.
- BREBBIA, C. A., TELLES, J.C., WROBEL, L.C., **Boundary Element Techniques**, Springer-Verlag, 1984.
- BRIGGS, W. L.; HENSON, V.E.; MCCORMICK, S.F. **A Multigrid Tutorial**, 2 ed., SIAM, 2000.
- BURDEN, R. L.; FAIRES, J. D. **Análise Numérica**, Tradutor: Ricardo Lenzi Tombi. São Paulo: Pioneira Thomson Learning, 2008.
- CHAPRA, S. C.; CANALE, R. P. **Introduction to Computing for Engineers**. 2 ed. New York: McGraw-Hill, 1994.
- CHISHOLM, T. **Multigrid Acceleration of an Approximately-Factored Algorithm for Steady Aerodynamic Flows**. Dissertação de mestrado. University of Toronto, 1997.
- CORDAZZO, J. **Simulação de Reservatórios de Petróleo utilizando o método EbFVM e Multigrid Algébrico**. Tese de doutorado em Engenharia Mecânica. UFSC. Florianópolis, SC, 2006.
- CRAIG C. D. **Multigrid Methods in Science and Engineering**. IBM T.F. Watson Research Center and Yale University, IEE, 1996.
- CUNHA, C. **Métodos Numéricos**, 2 ed. rev. Campinas, SP: Editora da UNICAMP, 2003.
- 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, pp. 19-25, 1989.
- DE ZEEW, P. M. Matrix-Dependent Prolongations and Restrictions in a Blackbox Multigrid Solver. **Journal of Computational Applied Mathematics**, v.3, pp.1-27, 1990.
- ELIAS, S. R.; STUBLEY, G. D; RAITHBY, G. D. An Adaptive Agglomeration Method for Additive Correction Multigrid. **International Journal for Numerical Methods in Engineering**, v. 40, pp. 887-903, 1997.
- FEDORENKO, R. P. On the Speed of Convergence of an Iteration Process. **USSR. Comput. Math. and Math. Phys**, v. 4, n.3, 1964.

FERZIGER, J. H.; PERIC, M., **Computational Methods for Fluid Dynamics**, 3 ed., Berlin: Springer, 2002.

FISCHER, R; HUCKLE, T. Multigrid Methods for Anisotropic BTTB Systems. **Linear Algebra and its Applications**. n.417, pp. 314-333, 2006.

FISCHER, R; HUCKLE, T. Multigrid Solution Techniques for Anisotropic Structered Linear Systems. **Applied Numerical Mathematic**. n.58, pp. 407-421, 2008.

FLETCHER, C.A.J. **Computational Methods for Fluid Dynamics**. v.1, 2 ed. Berlin: Springer. 1991.

FORTUNA, A, O. **Técnicas Computacionais para Dinâmica dos Fluidos**, São Paulo: Edusp, 2000.

GALANTE, G. **Métodos Multigrid Paralelos em Malhas Não-Estruturadas Aplicados à Simulação de Problemas de Dinâmica de Fluidos Computacional e Transferência de Calor**. Dissertação de Mestrado. UFRGS, Porto Alegre, RS, 2006.

GEE, W.M.; HU J.; TUMINARO S.R. A New Smoothed Aggregation Multigrid for Anisotropic Problems. **Numerical Linear Algebra Appl**, v.16, pp.19-37, 2009.

GEROLYMOS, G.A.; VALLET I. Mean-Flow-Multigrid for Implicit Reynolds-Stress-Model Computations. **AIAA Journal**. v. 43, n. 9, September, 2005.

GHIA, U.; GHIA, K. 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, pp.387-411, 1982.

GOLUB, G. H.; ORTEGA, J. M. **Scientific Computing and Differential Equations: an Introduction to Numerical Methods**, Academic Press, Inc., 1992.

GOLUB, G. H.; VAN LOAN, C. **Matrix Computations**, 2 ed., Johns Hopkins Press, Baltimore, 1989.

GREENBERG, M. D. **Advanced Engineering Mathematics**, 2 ed., Prentice Hall, Inc., New Jersey, 1998.

HIRSCH, C. **Numerical Computational of Internal and External Flows**, v.1, Wiley, 1988.

HORTMANN, M.; PERIC. M.; SCHEUERER, G. Finite Volume Multigrid Prediction of Laminar Natural Convection: Bench-Mark Solutions. **International Journal for Numerical methods in Fluid**, v.11, pp. 189-207, 1990.

HUGHES, T. J. R. **The Finite Element Method – Linear Static and Dynamic Finite Element Analysis**, Mineola (USA): Dover Publications, Inc., 2000.

HUTCHINSON, B. R., RAITHBY. A Multigrid Method based on the Additive Correction Strategy. **Numerical Heat Transfer**, v. 9, pp. 511-537, 1986.



INCROPERA, F. P.; DEWITT, D. P. **Fundamentos de Transferência de Calor e de Massa**, 4 ed. Rio de Janeiro: LTC Editora, 1998.

KELLER, S. C. **O Método Multigrid de Correções Aditivas para a Solução Numérica Acoplada das Equações de Navier-Stokes com Malhas Não-Estruturadas**. Tese de doutorado em Engenharia Mecânica. Universidade Federal de Santa Catarina. Florianópolis, SC, 2007.

LARSSON, J.; LIEN, F. S.; YEE, E. Conditional Semicoarsening Multigrid Algorithm for the Poisson Equation on Anisotropic Grids. **Journal of Computational Physics**, v. 208, pp. 368-383, 2005.

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

MANZANO, L. **Implementation of Multigrid for Aerodynamic Computations on Multi-Block Grids**, Dissertação de mestrado. Department of aerospace science and engineering, University of Toronto, 1999.

MARCHI, C. H. **Verificação de Soluções Numéricas Unidimensionais em Dinâmica dos Fluidos**. Tese de doutorado em Engenharia Mecânica. Universidade Federal de Santa Catarina. Florianópolis, SC, 2001.

MARCHI, C. H.; SCHNEIDER, F. A. **Introdução à Mecânica Computacional**. Curitiba, UFPR, 2004.

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

MEHTA, U. B. Guide to Credible Computer Simulations of Fluid Flows. **Journal of Propulsion and Power**, v. 12, n. 5, pp. 940 – 948, 1996.

MESQUITA, M. S. **Solução Numérica de Escoamentos bidimensionais Não-isotérmicos usando o Método Multigrid**. Dissertação de mestrado em Engenharia Aeronáutica e Mecânica. Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, SP, 2000.

MESQUITA, M. S.; DE LEMOS, M. J. S. Optimal Multigrid Solutions of Two-dimensional Convection-Conduction Problems. **Applied Mathematics and Computation**, v. 152, pp. 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, pp. 412-432, 2001.

MULDER, W.A. A New *Multigrid* Approach to Convection Problems. **Journal of Computational Physics**, v. 83, pp. 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, pp. 215-229, 1993.

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

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, pp. 128-151, 1993.

PARTER, S.V. Estimates for Multigrid Methods Based on Red-Black Gauss-Seidel Smoothing. **Numerical Mathematics**, n. 52, pp. 701-723, 1988.

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

PINTO, M. A. V. **Comportamento do Método Multigrid Geométrico em Problemas de Transferência de Calor**. Tese de doutorado. UFPR, Curitiba, PR, 2006.

PINTO, M. A. V.; MARCHI, C. H. **Efeito dos Parâmetros do Método Multigrid CS e FAS sobre o tempo de CPU para a Equação de Laplace Bidimensional**, Proceedings of ENCIT 2006.

PINTO, M. A. V.; MARCHI, C. H. **Optimum Parameters of a Geometric Multigrid for the Two-Dimensional Laplace's Equation**. Proceedings of COBEM, 2007.

RABI, J. R. **Aplicação do Método Multigrid na Solução de Problemas 2-D Simples de Mecânica dos Fluidos e Transferência de Calor**. Tese de doutorado. ITA, São José dos Campos, SP, 1998.

RABI, J. A.; DE LEMOS, M. J. S. The Effects of Peclet Number and Cycling Strategy on Multigrid Numerical Solutions of Convective-Advective Problems. 7 th, AIAA/ASME Int Thermcs & HT Conf, Paper AIAA-98-2584, Albuquerque, New Mexico, USA, June 15-18, 1998.

REDDYE, J. N.; GARTLING, D. K. **The Finit Element Method in Heat Transfer and Fluid Dynamics**. Boca Raton (USA): CRC Press, 1994.

RADESPIEL, R.; SWANSON, R. C. Progress with Multigrid Schemes for Hyperbolic Flow Problems. **Journal of Computational Physics**, v. 116, pp. 103-122, 1995.

ROACHE, P. J. **Fundamentals of Computational Fluid Dynamics**, Albuquerque, USA: Hermosa Publishers, 1998.

SANTIAGO, C. D.; MARCHI, C. H. **Parâmetros Ótimos do Método Multigrid Geométrico CS e FAS para problemas 2D com 2 equações**, Proceedings of the XXIX Iberian Latin American Congress on the Computational Methods in Engineering (CILAMCE), Maceió, Brazil, 2008.

SETTARI, A.; AZIZ, K. A. Generalization of Additive Correction Methods for the Iterative. Solution of Matrix Equations. **SIAM Journal of Numerical Analysis**, v.10, pp.506-521, 1973.

SPIEGEL, M. R. **Estatística**. Tradução, revisão e adaptação Carlos Augusto Crusius. – 3ed. São Paulo: McGraw\_Hill do Brasil, 1994.

STÜBEN, K. **A Review of Algebraic Multigrid**, Journal of Computation and Applied Mathematics, v. 128, pp. 281-309, 2001.

SZABÓ, B.; BABUSKA, I. **Finite Element Analysis**. New York: Wiley, 1991.

TANNEHILL, J. C.; ANDERSON, D. A; PLETCHER, R. H. **Computational Fluid Mechanics and Heat Transfer**. 2 ed. Washington: Taylor & Francis, 1997.

TROTTEBERG, U.; OOSTERLEE, C.; SCHÜLLER, A. **Multigrid**, St Augustin, Germany: Academic Press, 2001.

VERSTEEG, H. K.; MALALASEKERA, W. **An Introduction to Computational Fluid Dynamic, The Finite Volume Method**, England: Longman, 2 ed., 2007.

WESSELING, P. **An Introduction to Multigrid Methods**, New-York: 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, pp. 311-334, 2001.

YAN, J; THIELE, F. A Modified Full Multigrid Algorithm for the Navier-Stokes Equations. **Numerical heat transfer**. Part B, n. 34, pp.323-338, 1998.

YAN, J; THIELE, F.; XUE L. Performance and Accuracy of an Modified Full Multigrid Aloritm for Fluid Flow and Heat Trasfer. **Computer & Fluids**, n.36, pp. 445-454, 2007.

ZEEUW, P. M. Development of Semi-Coarsening Techniques. **Applied Numerical Mathematics**. pp. 433-465, 1996.

ZEEUW, P. M. **The Multigrid Image Transform**. Report PNA – E0506 september 2005.

ZHANG, J. Acceleration of Five-Point Red-Black Gauss-Seidel in Multigrid for Poisson Equation. **Applied Mathematics and Computation**, v.80, pp. 73-93, 1996.

ZHANG, J. **Multigrid Acceleration Techniques and Applications to the Numerical Solution of Partial Differential Equations**. Dissertação de mestrado. Chongqing Univesity. China, 1996.

ZHANG, J. Multigrid Method and Fourth-Order Compact Scheme for 2D Poisson Equation with Unequal Mesh-Size Discretization. **Journal of Computational Physics**, v. 179, pp. 170-179, 2002.

## APÊNDICE A: SOLUÇÕES ANALÍTICAS

Este apêndice descreve as soluções analíticas obtidas para as equações de Laplace senoidal, Laplace linear e Poisson. Aqui denomina-se por equação de Laplace senoidal a equação de Laplace com condição de contorno com uma senóide ao norte e equação de Laplace linear com condições de contorno lineares conforme Tab. 3.1.

Com a finalidade de buscar-se uma solução analítica da equação de Laplace senoidal, definida pela Eq. (3.1) com condições de contorno apresentadas na Tab. 3.1, usa-se a técnica de solução por separação de variáveis para as equações diferenciais parciais elípticas (GREENBERG, 1998). Primeiramente busca-se uma solução para Eq. (3.1) da forma:

$$T(x, y) = X(x)Y(y). \quad (\text{A.1})$$

Substituindo-se Eq. (A.1) em Eq. (3.1) e manipulando-se algebricamente tem-se:

$$\frac{d^2 X / dx^2}{X} = -\frac{d^2 Y / dy^2}{Y} = \rho^2 \quad (\text{A.2})$$

onde  $\rho$  é uma constante.

A seguir, resolve-se o sistema de equações diferenciais ordinárias, dado por Eq. (A.2), onde cada uma das equações é uma equação homogênea de segunda ordem com coeficientes constantes.

$$\begin{cases} \frac{d^2 X}{dx^2} - \rho^2 X = 0 \\ \frac{d^2 Y}{dy^2} - \rho^2 Y = 0 \end{cases} \quad (\text{A.3})$$

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

$$T(x, y) = (A_1 + A_2 x)(A_3 + A_4 y) + [A_5 \cosh(\rho x) + A_6 \sinh(\rho x)][A_7 \cosh(\rho y) + A_8 \sinh(\rho y)] \quad (\text{A.4})$$

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 Tab. 3.1, estas constantes acima assumem seus valores e este problema apresenta a solução analítica dada por:

$$T(x, y) = \text{sen}(\pi x) \frac{\sinh(\pi y)}{\sinh(\pi)} \quad (\text{A.5})$$

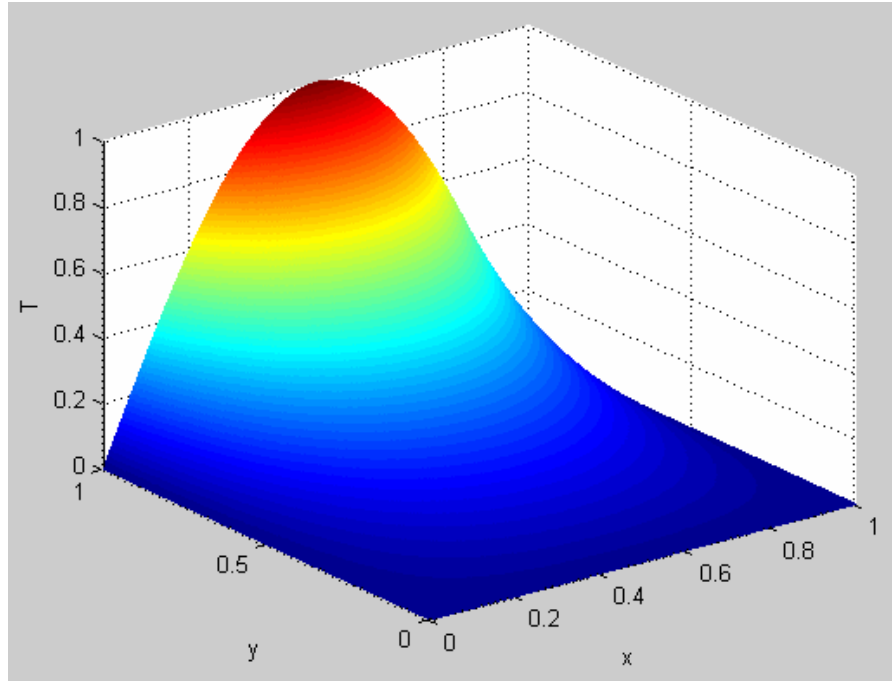
Para a equação de Laplace linear a solução analítica pode ser obtida de uma forma análoga e é dada por:

$$T(x, y) = xy \quad (\text{A.6})$$

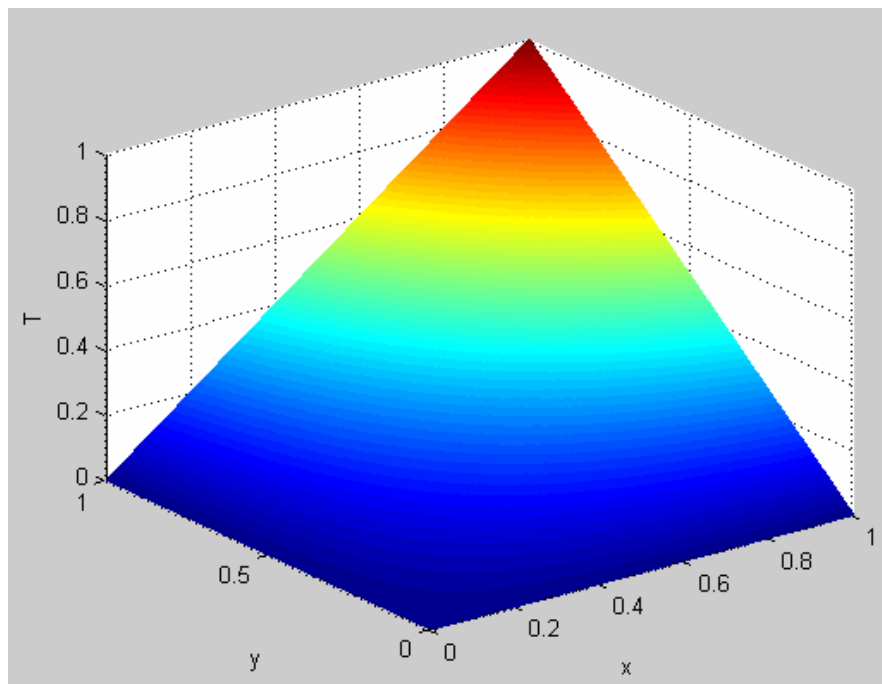
A solução analítica da equação de Poisson, com o termo fonte definido na Tab. 3.2 foi apresentada por Briggs et al., (2000) e é dada por:

$$T(x, y) = (x^2 - x^4)(y^4 - y^2) \quad (\text{A.7})$$

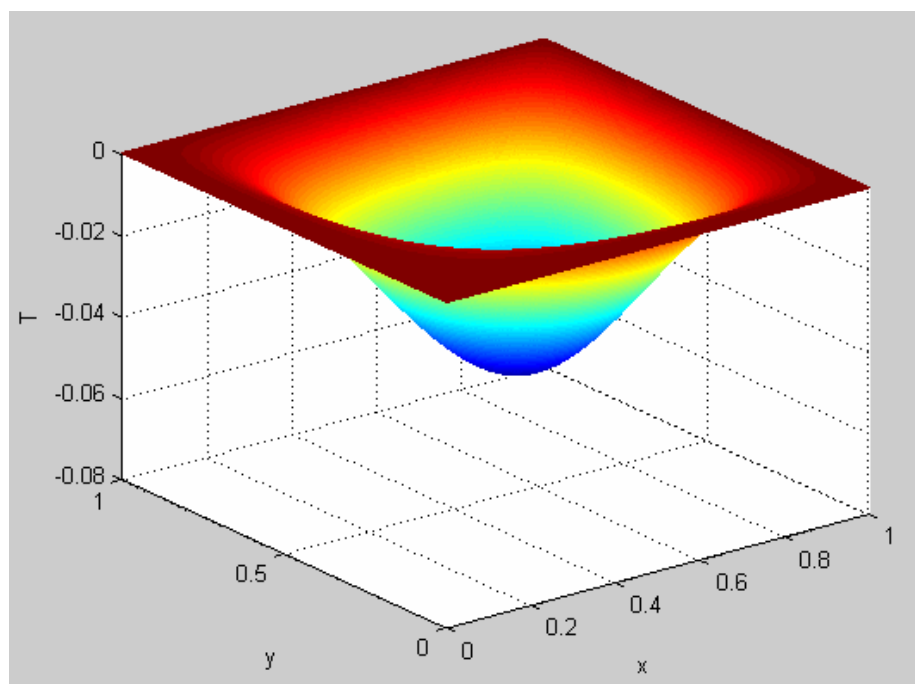
As Fig. A.1, A.2 e A.3 apresentam graficamente as soluções analíticas para as equações de Laplace senoidal, Laplace linear e Poisson, respectivamente. Verifica-se que na Fig. A.1, referente à equação de Laplace senoidal que a função é mais oscilatória na direção  $x$ . Nas Figs. A.2 e A.3, para as equações de Laplace linear e Poisson pode-se observar que a oscilação é a mesma em ambas as direções.



**Figura A.1: Solução analítica da equação de Laplace senoidal.**

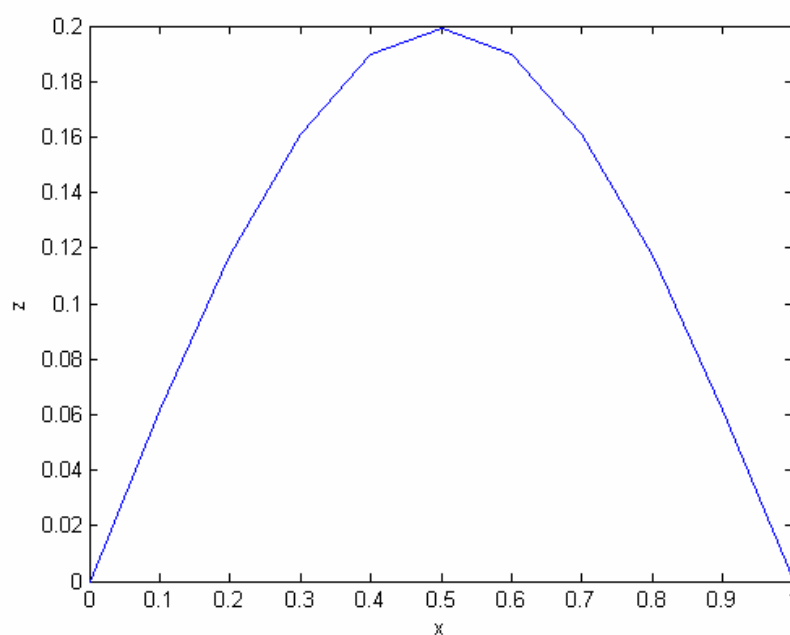


**Figura A.2: Solução analítica da equação de Laplace linear.**

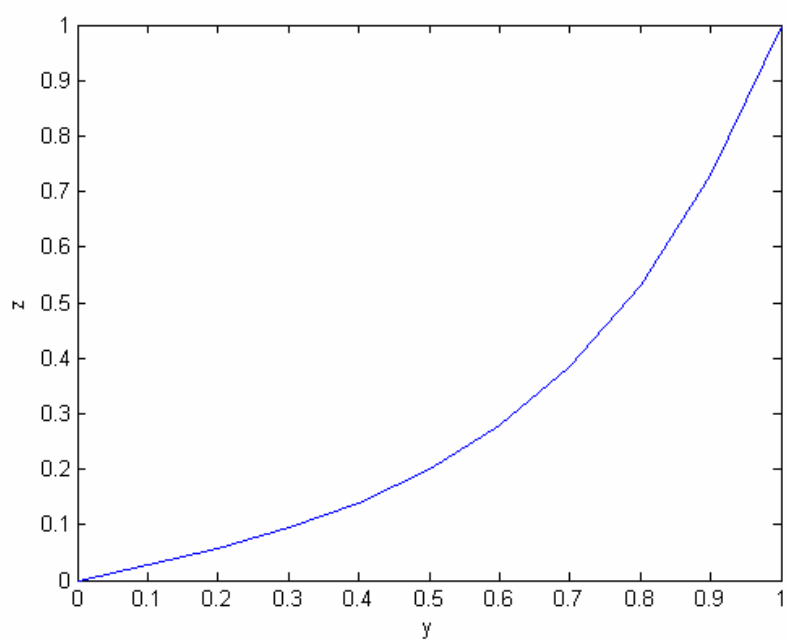


**Figura A.3: Solução analítica da equação de Poisson.**

As Figs. A4 e A5 apresentam os perfis da equação de Laplace senoidal nas direções  $x$  e  $y$ , respectivamente. Estas figuras têm por objetivo mostrar que a função é mais oscilatória na direção  $x$ . Elas são apresentadas somente para a equação de Laplace senoidal pois as equações de Laplace linear e Poisson possuem a mesma oscilação nas duas direções.



**Figura A.4: Perfil da temperatura em  $y = 1/2$ .**



**Figura A.5:** Perfil da temperatura em  $x = 1/2$ .



## APÊNDICE B: MEDIDAS DE DISPERSÃO

O grau de afastamento dos dados numéricos em torno de um valor central, chama-se variação ou dispersão de dados. Algumas medidas de dispersão muito utilizadas são: desvio padrão, variância e coeficiente de variação. Outras medidas de dispersão podem ser encontradas em Spiegel (1994).

Motivação: Considere as séries A e B, onde  $A = \{20, 20, 20\}$  e  $B = \{15, 10, 20, 25 \text{ e } 30\}$ . As duas séries apresentam a mesma média ( $\bar{x} = 20$ ). A série A não apresenta dispersão, enquanto a série B apresenta dispersão em torno da média. Logo, a média é muito mais representativa para a série A do que para a série B.

A variância e o desvio padrão são medidas estatísticas utilizadas para avaliar o grau de variabilidade, ou dispersão dos valores em torno da média. Servem para medir a representatividade da média.

O desvio padrão é a mais importante medida de variabilidade. Permite a comparação entre dois ou mais conjuntos de valores expressos na mesma unidade. Ele pode ser calculado para uma amostra (desvio padrão amostral, representado por  $s$ ), ou para a população (desvio padrão populacional, representado por  $\sigma$ ). É a medida mais utilizada na comparação de diferenças de grupos, por ser a mais precisa. O desvio padrão envolve todos os valores e não despreza os sinais de afastamento. Contorna a inconveniência da soma dos afastamentos ser nula, através dos seus quadrados. As Eqs. B.1 e B.2 apresentam o desvio padrão da amostra e da população, respectivamente:

$$s = \sqrt{\sum_{i=1}^n (x_i - \bar{X})^2 / (n-1)} \quad (\text{B.1})$$

$$\sigma = \sqrt{\sum_{i=1}^N (x_i - \bar{X})^2 / N} \quad (\text{B.2})$$

onde  $n$  é o tamanho da amostra,  $N$  o tamanho da população.

A variância é a média aritmética dos quadrados dos desvios, ou seja, o desvio padrão elevado ao quadrado.

O coeficiente de variação (*CV*) é uma medida de dispersão que mede a relação entre o desvio padrão e a média aritmética, podendo ser em termos percentuais. É indicado quando se deseja comparar distribuições diferentes, como por exemplo, quando possuem médias distintas. O coeficiente de variação pode ser calculado através das Eqs. B3 e B4 para a amostra e para a população, respectivamente.

$$CV = \frac{s}{\bar{x}} \cdot 100\% \quad (\text{B.3})$$

$$CV = \frac{\sigma}{\mu} \cdot 100\% \quad (\text{B.4})$$

onde  $\bar{x}$  é a média da amostra e  $\mu$  a média da população.

Entre dois ou mais conjuntos de valores o menos disperso é o que apresenta o menor coeficiente de variação ou a menor medida de dispersão. Nesta tese utilizou-se o coeficiente de variação para calcular a variabilidade dos dados em torno da média. Por ser utilizada uma amostra, utilizou-se o coeficiente de variação amostral dado pela Eq. (B.3).

## APÊNDICE C: TESTES DE COERÊNCIA

Este apêndice apresenta algumas tabelas referentes ao capítulo 4 que trata da verificação numérica do código computacional utilizado nesta tese.

A Tab. C.1 apresenta as soluções analíticas obtidas para a temperatura média  $\overline{TA}$  e para a temperatura analítica no nó central  $T(0,5;0,5)$  considerando um domínio  $0 \leq x, y \leq 1$  para as equações de Laplace senoidal, Laplace linear e Poisson. As soluções analíticas foram obtidas através das fórmulas apresentadas na Tab. 3.2.

**Tabela C. 1: Soluções analíticas para as equações de Laplace senoidal, Laplace linear e Poisson.**

<b>Equações</b>	<b>Variáveis</b>	$\overline{TA}$	$T(0,5;0,5)$
Laplace senoidal		1,8585392046028584E-01	1,9926840766919338E-01
Laplace linear		2,5000000000000000E-01	2,5000000000000000E-01
Poisson		-1,7777777777777777E-02	-3,5156250000000000E-02

As Tabs. C.2, C.3 e C.4 apresentam os resultados numéricos obtidos para cada uma das variáveis de interesse: temperatura numérica média ( $\overline{TN}$ ), obtida pela regra do trapézio; temperatura numérica no nó central ( $T(0,5,0,5)$ ); norma infinito  $\|TA - TN\|_\infty$  e norma  $\bar{l}_1$  ( $\|TA - TN\|_1 / N$ ). Apresentam também o erro de discretização médio obtido em cada malha. Pode-se verificar que para as equações de Laplace senoidal e Poisson o erro de discretização diminui à medida que se aumenta o número de nós da malha. Na Tab. C.3, referente à equação de Laplace linear pode-se observar que a temperatura numérica média é igual a temperatura numérica no nó central, isto é,  $\overline{TN} = T(0,5,0,5) = 2,5000000000000000E-01$ . Isto ocorre para todas as malhas. Verifica-se também que o erro de discretização, a norma  $l_\infty$  e a norma  $\bar{l}_1$  são iguais a zero, mostrando que a equação de Laplace linear não possui erro de discretização e pode ser resolvida exatamente em qualquer malha.

Tabela C. 2: Resultados numéricos para as variáveis de interesse referentes à equação de Laplace senoidal.

$N$	$Ed$ médio	$l_{\infty}$	$\bar{l}_1$	$\overline{TN}$	$T(0,5;0,5)$
3x3	5,636844E-03	5,0731592330806624E-02	5,0731592330806624E-02	1,8750000000000000E-01	2,5000000000000000E-01
5x5	3,668750E-03	1,6565446555151464E-02	1,0190971445135042E-02	1,9089866283237816E-01	2,1338834764831849E-01
9x9	1,308702E-03	4,3224982010843238E-03	2,1633639070369705E-03	1,8746727426741994E-01	2,0291522352182767E-01
17x17	3,803887E-04	1,1088416764469811E-03	4,8858820314361367E-04	1,8628065798769100E-01	2,0018802296405147E-01
33x33	1,018723E-04	2,7796145785130566E-04	2,7796145785130566E-04	1,8596209139754166E-01	1,9949881658540525E-01
65x65	2,631767E-05	6,9627163395091785E-05	2,8015156480503440E-05	1,8588105649176406E-01	1,9932604163761730E-01
129x129	6,685630E-06	1,7409801076617715E-05	6,8978587584104706E-06	1,8586071030532547E-01	1,9928281814770354E-01
257x257	1,684680E-06	4,3526384783110217E-06	1,7112098751223166E-06	1,8585561828646396E-01	1,9927201041301326E-01
513x513	4,228286E-07	1,0881713833899020E-06	4,2614491791506444E-07	1,8585434493964240E-01	1,9926930836291079E-01
1025x1025	1,059145E-07	2,7204481117815149E-07	1,0632899611055989E-07	1,8585402658155617E-01	1,9926863284310664E-01
2049x2049	2,650453E-08	6,8011253739896915E-08	2,6556344367644331E-08	1,8585394699069238E-01	1,9926846396271036E-01
4097x4097	6,629372E-09	1,7002821040001947E-08	6,6358495910259976E-09	1,8585392709288273E-01	1,9926842174257320E-01
8193x8193	1,657754E-09	4,2507420361381776E-09	1,6585638550163480E-09	1,8585392211846538E-01	1,9926841118755750E-01

Tabela C. 3: Resultados numéricos para as variáveis de interesse referentes à equação de Laplace linear.

$N$	$Ed$ médio	$l_{\infty}$	$\bar{l}_1$	$\overline{TN}$	$T(0,5;0,5)$
3x3	0,000000E+00	0,0000000000000000E+00	0,0000000000000000E+00	2,5000000000000000E-01	2,5000000000000000E-01
5x5	0,000000E+00	0,0000000000000000E+00	0,0000000000000000E+00	2,5000000000000000E-01	2,5000000000000000E-01
9x9	0,000000E+00	0,0000000000000000E+00	0,0000000000000000E+00	2,5000000000000000E-01	2,5000000000000000E-01
17x17	6,124054E-18	5,5511151231257827E-17	7,8660072282903887E-18	2,5000000000000000E-01	2,5000000000000000E-01
33x33	5,093460E-19	1,3877787807814457E-17	5,7718817007657034E-19	2,5000000000000000E-01	2,5000000000000000E-01
65x65	0,000000E+00	0,0000000000000000E+00	0,0000000000000000E+00	2,5000000000000000E-01	2,5000000000000000E-01
129x129	3,569904E-16	1,9984014443252818E-15	3,6832275183968372E-16	2,5000000000000000E-01	2,5000000000000000E-01
257x257	7,936737E-16	5,3290705182007514E-15	8,0617227587799620E-16	2,5000000000000000E-01	2,5000000000000000E-01
513x513	7,101122E-16	4,8849813083506888E-15	7,1568164464879077E-16	2,5000000000000000E-01	2,5000000000000000E-01
1025x1025	9,273049E-16	6,2172489379008766E-15	9,3093425372704094E-16	2,5000000000000000E-01	2,5000000000000000E-01
2049x2049	2,057518E-15	7,3274719625260332E-15	2,0615400586913219E-15	2,4999999999999997E-01	2,4999999999999734E-01
4097x4097	1,796768E-15	2,7977620220553945E-14	1,7985235399315068E-15	2,5000000000000000E-01	2,5000000000000078E-01
8193x8193	4,291675E-15	3,8191672047105385E-14	4,2937709637195945E-15	2,5000000000000000E-01	2,5000000000000000E-01

Tabela C. 4: Resultados numéricos para as variáveis de interesse referentes à equação de Poisson.

$N$	$Ed$ médio	$l_{\infty}$	$\bar{l}_1$	$\overline{TN}$	$T(0,5;0,5)$
3x3	1,302083E-03	1,1718750000000000E-02	1,1718750000000000E-02	-5,8593750000000000E-03	-2,3437500000000000E-02
5x5	7,653809E-04	3,0212402343750000E-03	2,1260579427083330E-03	-1,3944625854492186E-02	-3,2135009765625000E-02
9x9	2,614644E-04	7,6388269493931488E-04	4,3221667748825926E-04	-1,6761326943250256E-02	-3,4397805438322174E-02
17x17	7,506301E-05	1,9672536981162059E-04	9,6414271856017883E-05	-1,7519986387460374E-02	-3,4966502200603149E-02
33x33	5,172550E-06	1,2292229022589640E-05	5,5061792052227906E-06	-1,7761593813288172E-02	-3,5144388434512563E-02
65x65	2,003844E-05	4,9171472962766805E-05	2,2707456143685775E-05	-1,7713099542326385E-02	-3,5108805546975951E-02
129x129	1,313747E-06	3,0730169524872952E-06	1,3554507378342706E-06	-1,7773730886207921E-02	-3,5153284580626908E-02
257x257	3,310281E-07	7,6827935460710428E-07	3,3624105832670757E-07	-1,7776765998606541E-02	-3,5155508643412339E-02
513x513	8,308189E-08	1,9207253980357830E-07	8,3733509779252106E-08	-1,7777524829465697E-02	-3,5156064660738916E-02
1025x1025	2,081114E-08	4,8018161370733115E-08	2,0892592417702095E-08	-1,7777714540478946E-02	-3,5156203665176800E-02
2049x2049	5,207873E-09	1,2004561539269432E-08	5,2180543949414778E-09	-1,7777761968438034E-02	-3,5156238416293124E-02
4097x4097	1,302604E-09	3,0011417379016692E-09	1,3038770208708005E-09	-1,7777773825443249E-02	-3,5156247104072613E-02
8193x8193	3,257308E-10	7,5029384788427578E-10	3,2588984996017338E-10	-1,7777776789686422E-02	-3,5156249276012158E-02

A Tab. C.5 apresenta o cálculo de  $h$  utilizado para a discretização dos modelos matemáticos.

**Tabela C. 5: Cálculo de  $h$  para a discretização dos modelos numéricos.**

Níveis	$N$	$E$	$h$
1	3x3	4	5,00000E-01
2	5x5	16	2,50000E-01
3	9x9	64	1,25000E-01
4	17x17	256	0,62500E-01
5	33x33	1024	3,12530E-02
6	65x65	4096	1,56250E-02
7	129x129	16384	7,81250E-03
8	257x257	65536	3,90625E-03
9	513x513	262144	1,95312E-03
10	1025x1025	1048576	9,76563E-04
11	2049x2049	4194304	4,88281E-04
12	4097x4097	16777216	2,44141E-04
13	8193x8193	67108864	1,22070E-04

As Tabs. C.6 a C.9 apresentam a ordem efetiva, ordem aparente e o erro numérico obtidos para as variáveis de interesse: temperatura no nó central para o domínio  $0 \leq x, y \leq 1$ , temperatura média, norma  $l_\infty$  e norma  $\bar{l}_1$  para a equação de Laplace senoidal.

**Tabela C. 6: Ordem efetiva ( $p_E$ ), ordem aparente ( $p_U$ ) para a aproximação numérica da temperatura no nó central para a Equação de Laplace senoidal.**

$h$	Ordem efetiva ( $p_E$ )	Ordem aparente ( $p_U$ )	Erro numérico
5,00000E-01	Não se aplica!	Não se aplica!	5,07315923308066E-02
2,50000E-01	1,84515048753888E+00	Não se aplica!	1,41199399791251E-02
1,25000E-01	1,95302469884476E+00	1,80561102509819E+00	3,64681585263429E-03
0,62500E-01	1,98753498608929E+00	1,94119915674493E+00	9,19615294858091E-04
3,12530E-02	1,99683391455261E+00	1,98441284068924E+00	2,30408916211868E-04
1,56250E-02	1,99920528036784E+00	1,99604201067094E+00	5,76339684239216E-05
7,81250E-03	1,99980111881802E+00	1,99900657633050E+00	1,44104785101643E-05
3,90625E-03	1,99995026750443E+00	1,99975139687716E+00	3,60274381896596E-06
1,95312E-03	1,99998756601424E+00	1,99993783431067E+00	9,00693717410855E-07
9,76563E-04	1,99999686828470E+00	1,99998446523512E+00	2,25173918146132E-07
4,88281E-04	1,99999904317249E+00	1,99999614332073E+00	5,62935168716357E-08
2,44141E-04	1,99999798087382E+00	1,99999939727254E+00	1,40733989143434E-08
1,22070E-04	2,00000435171251E+00	1,99999585726321E+00	3,51833911593176E-09

**Tabela C. 7: Ordem efetiva ( $p_E$ ), ordem aparente ( $p_U$ ) para a aproximação numérica da temperatura média e Equação de Laplace senoidal.**

$h$	Ordem efetiva ( $p_E$ )	Ordem aparente ( $p_U$ )	Erro numérico
5,00000E-01	Não se aplica!	Não se aplica!	1,64607953971416E-03
2,50000E-01	-1,61574654532650E+00	Não se aplica!	5,04474237209232E-03
1,25000E-01	1,64471773980351E+00	Não se aplica!	1,61335380713410E-03
0,62500E-01	1,91864196280568E+00	1,53193902259091E+00	4,26737527405161E-04
3,12530E-02	1,98003605084631E+00	1,89718660458244E+00	1,08170937255822E-04
1,56250E-02	1,99503128255894E+00	1,97497958360487E+00	2,71360314782229E-05
7,81250E-03	1,99875919432797E+00	1,99378507249158E+00	6,78984503963166E-06
3,90625E-03	1,99968988466998E+00	1,99844874176737E+00	1,69782617787407E-06
1,95312E-03	1,99992246555735E+00	1,99961234381821E+00	4,24479356556328E-07
9,76563E-04	1,99998052164127E+00	1,99990311266182E+00	1,06121271911907E-07
4,88281E-04	1,99999508590583E+00	1,99997566683169E+00	2,65304083454482E-08
2,44141E-04	1,9999992857129E+00	1,99999347168029E+00	6,63260241474623E-09
1,22070E-04	1,99998226063807E+00	2,00000581793084E+00	1,65817099241300E-09

**Tabela C. 8: Ordem efetiva ( $p_E$ ), ordem aparente ( $p_U$ ) para a norma  $l_\infty$  e Equação de Laplace senoidal.**

$h$	Ordem efetiva ( $p_E$ )	Ordem aparente ( $p_U$ )	Erro numérico
5,00000E-01	Não se aplica!	Não se aplica!	5,07315923308066E-02
2,50000E-01	1,61470734886095E+00	Não se aplica!	1,65654465551515E-02
1,25000E-01	1,93823982614222E+00	1,48061647844639E+00	4,32249820108432E-03
0,62500E-01	1,96281197533112E+00	1,92966338331493E+00	1,10884167644698E-03
3,12530E-02	1,99609663035533E+00	1,95150332690875E+00	2,77961457851306E-04
1,56250E-02	1,99716269821885E+00	1,99574016534315E+00	6,96271633950918E-05
7,81250E-03	1,99975053039058E+00	1,99629885571918E+00	1,74098010766177E-05
3,90625E-03	1,99993761623702E+00	1,99968815945408E+00	4,35263847808898E-06
1,95312E-03	1,99998440340784E+00	1,99992201995141E+00	1,08817138338990E-06
9,76563E-04	1,99998955455268E+00	1,99998268633889E+00	2,72044815508021E-07
4,88281E-04	1,99999884807556E+00	1,99998645669512E+00	6,80112581807890E-08
2,44141E-04	1,99999745181569E+00	1,99999931349631E+00	1,70028445767301E-08
1,22070E-04	1,99999961376913E+00	1,99999673116357E+00	4,25071228216112E-09

**Tabela C. 9: Ordem efetiva ( $p_E$ ), ordem aparente ( $p_U$ ) para a norma  $l_1$  média para a Equação de Laplace senoidal.**

$h$	Ordem efetiva ( $p_E$ )	Ordem aparente ( $p_U$ )	Erro numérico
5,00000E-01	Não se aplica!	Não se aplica!	5,07315923308066E-02
2,50000E-01	2,31559286191127E+00	Não se aplica!	1,01909714451350E-02
1,25000E-01	2,23594331003551E+00	2,33632619248034E+00	2,16336390703697E-03
0,62500E-01	2,14658543093157E+00	2,26100219242236E+00	4,88588203143614E-04
3,12530E-02	2,08146172485873E+00	2,16615161226242E+00	1,15441128297226E-04
1,56250E-02	2,04287784963602E+00	2,09361054372237E+00	2,80151564805034E-05
7,81250E-03	2,02198705748486E+00	2,04963669600073E+00	6,89785875841047E-06
3,90625E-03	2,01113187529170E+00	2,02555061869403E+00	1,71120987484298E-06
1,95312E-03	2,00560068068281E+00	2,01296142202150E+00	4,26144917915064E-07
9,76563E-04	2,00280902880283E+00	2,00652762417148E+00	1,06328997753223E-07
4,88281E-04	2,00140651142696E+00	2,00327562504504E+00	2,65563465155177E-08
2,44141E-04	2,00070235094995E+00	2,00164100299527E+00	6,63585529182617E-09
1,22070E-04	2,00035488007435E+00	2,00081811800805E+00	1,6585579438184E-09

As Tabs. C.10 a C.13 apresentam a ordem efetiva, ordem aparente e o erro numérico obtidos para as variáveis de interesse: temperatura no nó central para o domínio  $0 \leq x, y \leq 1$ , temperatura média, norma  $l_\infty$  e norma  $\bar{l}_1$  para a equação de Poisson.

**Tabela C. 10: Equação de Poisson para a temperatura no nó central.**

$h$	Ordem efetiva ( $p_E$ )	Ordem aparente ( $p_U$ )	Erro numérico
5,00000E-01	Não se aplica!	Não se aplica!	1,17187500000000E-02
2,50000E-01	1,95560588064155E+00	Não se aplica!	3,02124023437500E-03
1,25000E-01	1,99402526874869E+00	1,94249607658411E+00	7,58444561677826E-04
0,62500E-01	1,99896057682150E+00	1,99237482129241E+00	1,89747799396851E-04
3,12530E-02	1,99977182476602E+00	1,99869000237658E+00	4,74444530240495E-05
1,56250E-02	1,99994499507917E+00	1,99971409377410E+00	1,18615654874366E-05
7,81250E-03	1,99998637716086E+00	1,99993120061448E+00	2,96541937309241E-06
3,90625E-03	1,99999660538212E+00	1,99998296772696E+00	7,41356587660724E-07
1,95312E-03	1,99999911129876E+00	1,99999577007492E+00	1,85339261084272E-07
9,76563E-04	1,99999975310713E+00	1,99999889736252E+00	4,63348232004890E-08
4,88281E-04	1,99999986604808E+00	1,99999971546014E+00	1,15837068756508E-08
2,44141E-04	1,99999966728077E+00	1,99999993230386E+00	2,89592738678124E-09
1,22070E-04	1,99998805326986E+00	2,00000353863970E+00	7,23987841899643E-10

**Tabela C. 11: Equação de Poisson para a temperatura numérica média.**

$h$	Ordem efetiva ( $p_E$ )	Ordem aparente ( $p_U$ )	Erro numérico
5,00000E-01	Não se aplica!	Não se aplica!	1,19184027777778E-02
2,50000E-01	1,63658782449176E+00	Não se aplica!	3,83315192328558E-03
1,25000E-01	1,91499074502818E+00	1,52128606417590E+00	1,01645083452751E-03
0,62500E-01	1,97926444739273E+00	1,89248214904480E+00	2,57791390317393E-04
3,12530E-02	1,99485185664487E+00	1,97400597337733E+00	6,46782354513823E-05
1,56250E-02	1,99871526031837E+00	1,99356021928940E+00	1,61839644895949E-05
7,81250E-03	1,99967895961885E+00	1,99839378873040E+00	4,04689156984598E-06
3,90625E-03	1,99991974992688E+00	1,99959868129721E+00	1,01177917122586E-06
1,95312E-03	1,99997992767996E+00	1,99989968974590E+00	2,52948312070074E-07
9,76563E-04	1,99999496259069E+00	1,99997491598489E+00	6,32372988206632E-08
4,88281E-04	1,99999862863863E+00	1,99999374057108E+00	1,58093397328098E-08
2,44141E-04	2,00000015143526E+00	1,99999812103947E+00	3,95233451833802E-09
1,22070E-04	1,99998873553926E+00	2,00000395675347E+00	9,88091344501651E-10



**Tabela C. 12: Equação de Poisson para norma  $l_\infty$ .**

$h$	Ordem efetiva ( $p_E$ )	Ordem aparente ( $p_U$ )	Erro numérico
5,00000E-01	Não se aplica!	Não se aplica!	1,17187500000000E-02
2,50000E-01	1,95560588064155E+00	Não se aplica!	3,02124023437500E-03
1,25000E-01	1,98371789071739E+00	1,94596745080756E+00	7,63882694939315E-04
0,62500E-01	1,95716808855361E+00	1,99281405448109E+00	1,96725369811621E-04
3,12530E-02	2,00028954259146E+00	1,94250696094842E+00	4,91714729627668E-05
1,56250E-02	2,00007502079623E+00	2,00036103781155E+00	1,22922290225896E-05
7,81250E-03	2,00001892107621E+00	2,00009371989117E+00	3,07301695248730E-06
3,90625E-03	1,99995283483689E+00	2,00004095011016E+00	7,68279354607104E-07
1,95312E-03	1,99997971096901E+00	1,99994387584693E+00	1,92072539803578E-07
9,76563E-04	1,9999920622157E+00	1,99997321248818E+00	4,80181613707331E-08
4,88281E-04	1,99999745261527E+00	1,99999979075791E+00	1,20045615392694E-08
2,44141E-04	1,9999934955139E+00	1,99999682030230E+00	3,00114173790167E-09
1,22070E-04	1,99998382227000E+00	2,00000452535210E+00	7,50293847884276E-10

**Tabela C. 13: Equação de Poisson para norma  $l_1$  média.**

$h$	Ordem efetiva ( $p_E$ )	Ordem aparente ( $p_U$ )	Erro numérico
5,00000E-01	Não se aplica!	Não se aplica!	1,17187500000000E-02
2,50000E-01	2,46256586936143E+00	Não se aplica!	2,12605794270833E-03
1,25000E-01	2,29835426988852E+00	2,50163706274986E+00	4,32216677488259E-04
0,62500E-01	2,16443611671523E+00	2,33461421121571E+00	9,64142718560179E-05
3,12530E-02	2,08608062540920E+00	2,18774262808263E+00	2,27074561436858E-05
1,56250E-02	2,04404262372858E+00	2,09928236483336E+00	5,50617920522279E-06
7,81250E-03	2,02227888376502E+00	2,05107920568151E+00	1,35545073783427E-06
3,90625E-03	2,01120487353817E+00	2,02591367256465E+00	3,36241058326708E-07
1,95312E-03	2,00561889892201E+00	2,01305246210718E+00	8,37335097792521E-08
9,76563E-04	2,00281358090192E+00	2,00655037034402E+00	2,08925924177021E-08
4,88281E-04	2,00140762905341E+00	2,00328131851639E+00	5,21805439494148E-09
2,44141E-04	2,00070417961151E+00	2,00164188352197E+00	1,30387702087080E-09
1,22070E-04	2,00035147954799E+00	2,00082168896632E+00	3,25889849960173E-10

## APÊNDICE D: ALGORITMOS DE ENGROSSAMENTO

Este apêndice é um complemento dos resultados apresentados no capítulo 5. Ele está dividido na seguinte forma: primeiramente é apresentado o estudo do número de iterações internas para as equações de Laplace senoidal, Laplace linear e Poisson. A seguir é feito um estudo da influência do número de elementos e a análise de complexidade.

### D1. ESTUDO DO NÚMERO DE ITERAÇÕES INTERNAS ( $\nu$ ):

A Tab. D.1 mostra as dimensões das malhas finas ( $N = N_x \times N_y$ ), as respectivas razões de aspecto ( $Q$ ) utilizadas para determinar o número ótimo de iterações internas e o número de elementos. Para cada equação e algoritmo em estudo utilizou-se um conjunto diferente de malhas de acordo com o tempo de CPU utilizado em cada simulação e a quantidade de simulações necessárias para a obtenção do  $\nu_{\text{ótimo}}$ . O  $\nu_{\text{ótimo}}$  para a razão de aspecto  $Q = 1$  foi determinada apenas para o algoritmo EP, visto que todos os algoritmos de engrossamento recaem no EP quando  $Q = 1$ .

**Tabela D.1: Malhas utilizadas para a determinação do  $\nu_{\text{ótimo}}$ .**

$Q$	Malhas				
1/64	513x9	4097x65	8193x129	16385x257	65537x1025
1/16	257x17	2049x129	4097x257	8193x513	32769x2049
1/4	129x33	1025x257	2049x513	4097x1025	16385x4097
1	65x65	513x513	1025x1025	2048x2048	81934x8193
4	33x129	257x1025	513x2049	1025x4097	4097x16385
16	17x257	129x2049	257x4097	513x8193	2049x32769
64	9x513	65x4097	129x8193	257x16385	1025x65537
$E$	4.096	262.144	1.048.576	4.194.304	67.108.864

As Figs. D.1 e D.2 apresentam o tempo de CPU em função do número de iterações internas para a equação de Laplace senoidal, algoritmo SE-EP e razões de aspecto  $Q = 1/64$ ,  $1/16$ ,  $1/4$ ,  $4$ ,  $16$  e  $64$ . Consideram-se problemas com 4.194.304 e 67.108.864 elementos.

Observa-se que para qualquer tamanho de problema  $\nu_{\text{ótimo}} = 4$ , para as razões de aspecto maiores que a unidade e  $\nu_{\text{ótimo}} = 2$ , para razões de aspecto menores que a unidade.

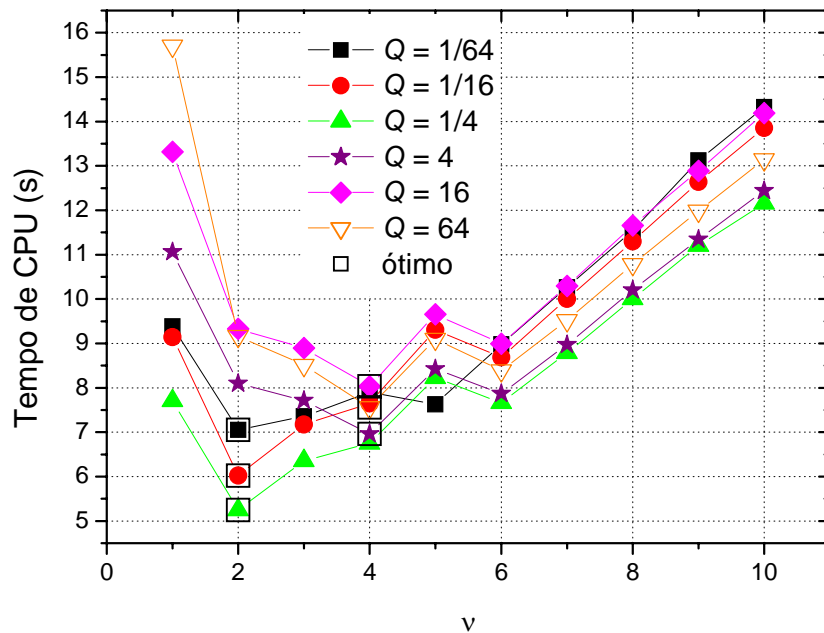


Figura D.1: Tempo de CPU *versus*  $\nu$  para  $E = 4.194.304$ , SE-EP e equação de Laplace senoidal.

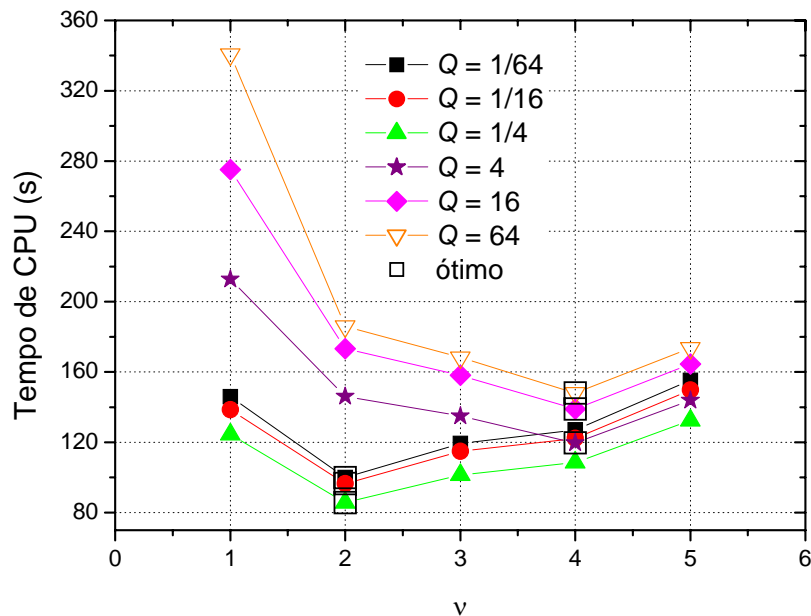


Figura D.2: Tempo de CPU *versus*  $\nu$  para  $E = 67.108.864$ , SE-EP e equação de Laplace senoidal.

Para as equações de Laplace linear e Poisson utilizaram-se as razões de aspecto apresentadas na Tab. D.1 em um problema com 262.144 elementos. As Figs. D.3 e D.4 apresentam o tempo de CPU em função do número de iterações internas para as equações de Laplace linear e Poisson, algoritmo SE-EP. Para a equação de Laplace linear (Fig. D.3),

observa-se que  $\nu_{\text{ótimo}} = 4$ . Para a equação de Poisson, apresentada na Fig. D.4,  $\nu_{\text{ótimo}}$  varia entre 2, 3 e 4.

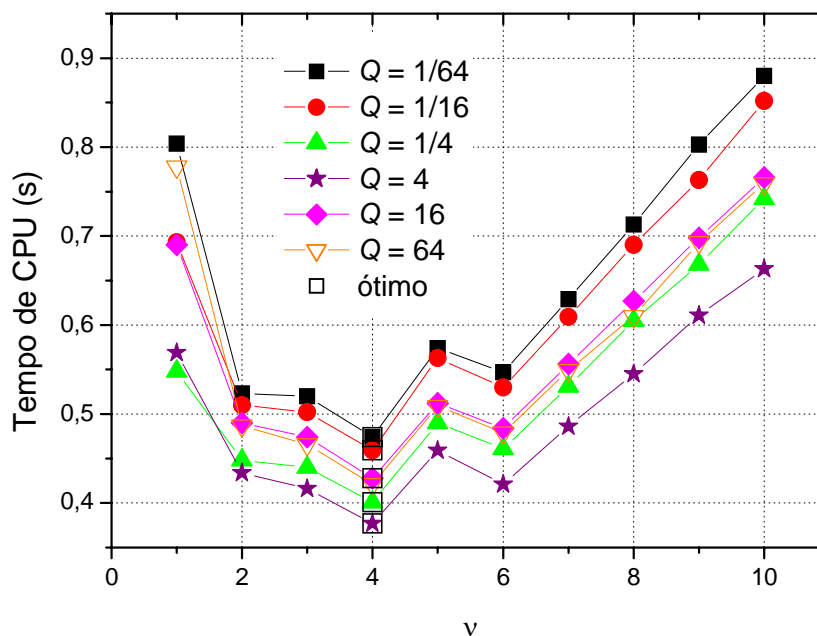


Figura D.3: Tempo de CPU *versus*  $\nu$  para  $E = 262.144$ , SE-EP e equação de Laplace linear.

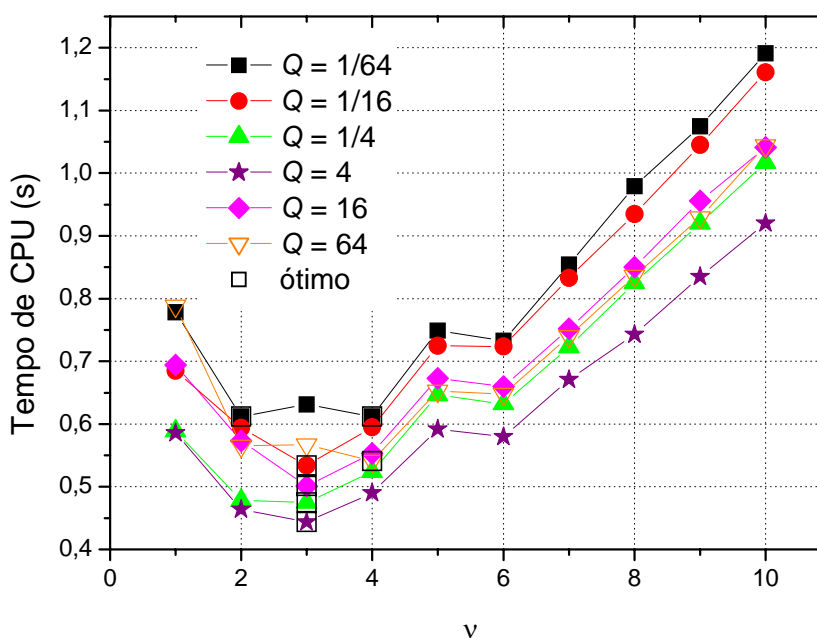


Figura D.4: Tempo de CPU *versus*  $\nu$  para  $E = 262.144$ , SE-EP e equação de Poisson.

A Tab. D.2 apresenta o  $\nu_{\text{ótimo}}$  para o algoritmo SE-EP e equação de Poisson para as razões de aspecto 1/64, 1/16, 1/4, 4, 16 e 64. Verifica-se que a razão de aspecto não interfere significativamente no  $\nu_{\text{ótimo}}$ . Ao utilizar  $\nu = 3$  o acréscimo no tempo de CPU será de 3,1% e 4,8% para as razões 1/64 e 64 respectivamente. Portanto, ao utilizar  $\nu = 3$  para todas as razões de aspecto e equação de Poisson, o acréscimo no tempo de CPU não será significativo.

**Tabela D.2:  $\nu_{\text{ótimo}}$  para o algoritmo SE-EP e equação de Poisson.**

<b>1/64</b>	<b>1/16</b>	<b>1/4</b>	<b>4</b>	<b>16</b>	<b>64</b>
2 e 4	3	3	3	3	4

Para a análise do  $\nu_{\text{ótimo}}$  nos algoritmos EP, SE, SEC e EP-SE foram utilizadas malhas com menor número de elementos. Isto foi necessário devido ao fato destes algoritmos não apresentarem boa convergência para problemas anisotrópicos, o tempo de CPU ser bem maior comparado ao SE-EP e serem necessárias mais simulações para a determinação do  $\nu_{\text{ótimo}}$ . Foram utilizadas as malhas com 65.5336, 262.144 e 4.194.304 elementos apresentadas na Tab. D.1.

As Figs. D.5 a D.7 apresentam o tempo de CPU em função do número de iterações internas para a equação de Laplace senoidal, algoritmo EP e razões de aspecto  $Q = 1/64$ , 1/16, 1/4, 1, 4, 16 e 64. Consideram-se problemas com 65.536, 262.144 e 4.194.304 elementos. Os gráficos apresentados não utilizam todos os resultados obtidos nas simulações. O número de pontos foi reduzido de forma a contemplar o intervalo que contém o  $\nu_{\text{ótimo}}$ . Verifica-se nas Figs. D.5 à D.7 que são necessárias muitas iterações internas para se obter um menor tempo de CPU. Verifica-se também que poucas iterações internas resultam em um tempo de CPU muito alto ou até mesmo a divergência da solução. Com isto pode-se concluir que o algoritmo EP não é muito estável para razões de aspecto diferentes da unidade.

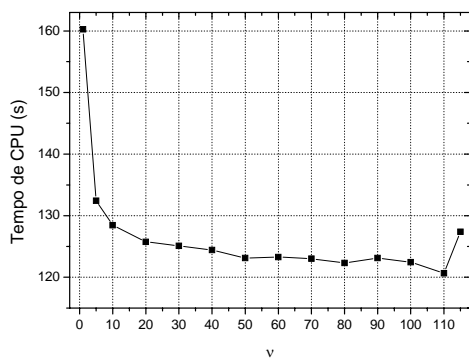
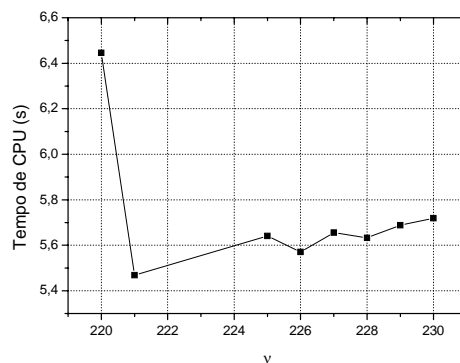
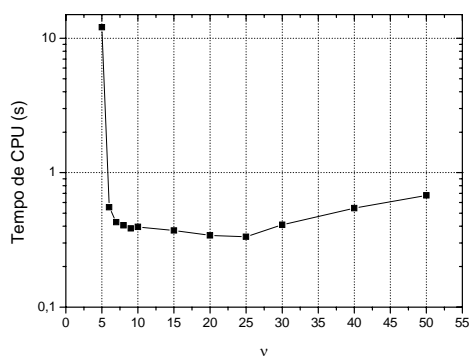
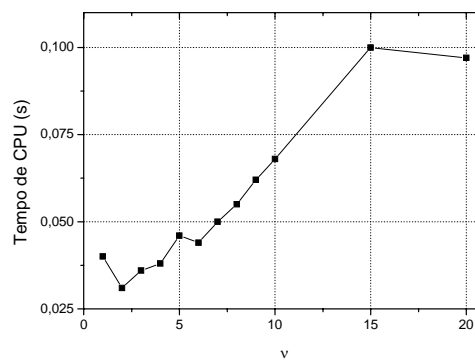
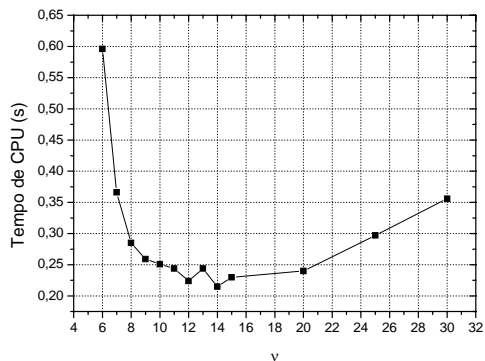
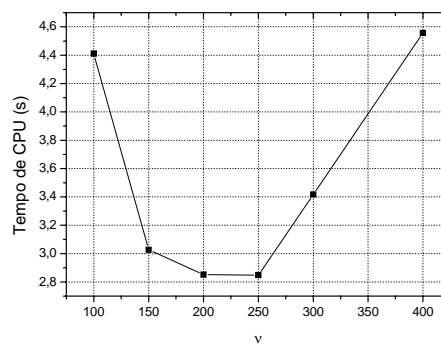
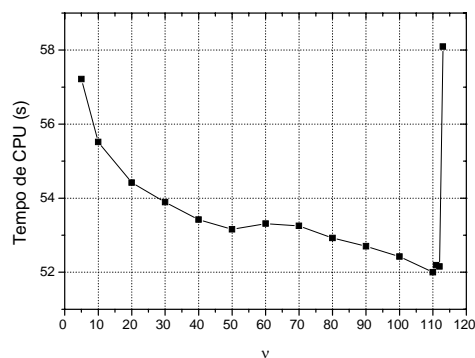
a)  $Q = 1/64$ b)  $Q = 1/16$ c)  $Q = 1/4$ d)  $Q = 1$ e)  $Q = 4$ f)  $Q = 16$ g)  $Q = 64$ 

Figura D.5: Tempo de CPU *versus*  $\nu$  para a equação de Laplace senoidal, algoritmo EP e  $E = 65.536$ .

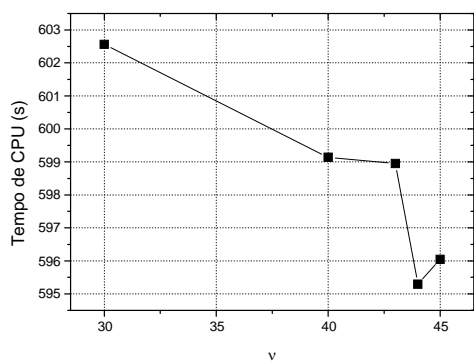
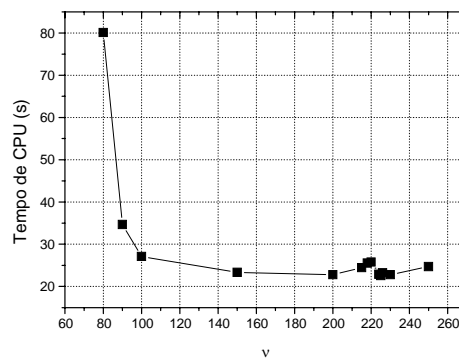
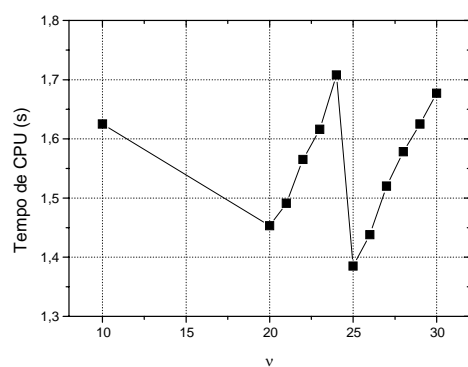
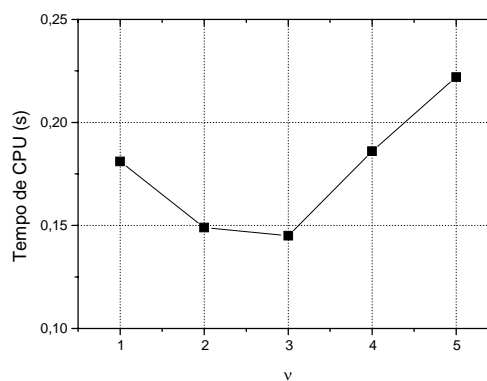
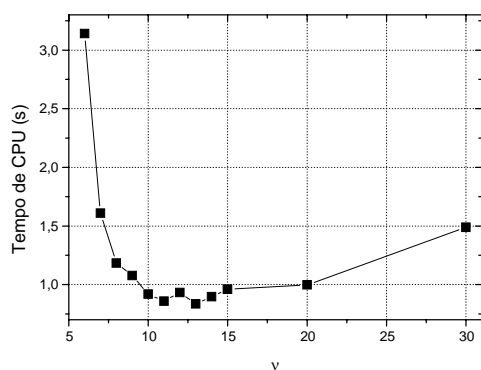
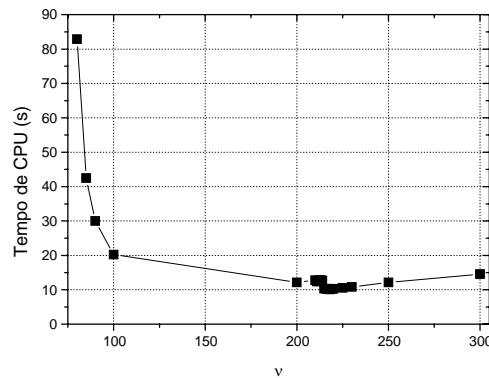
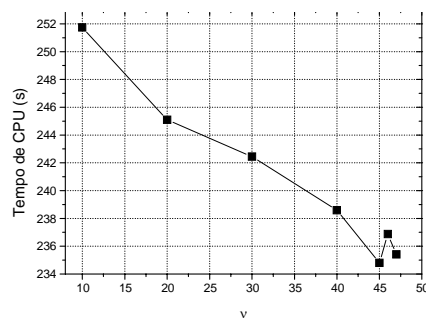
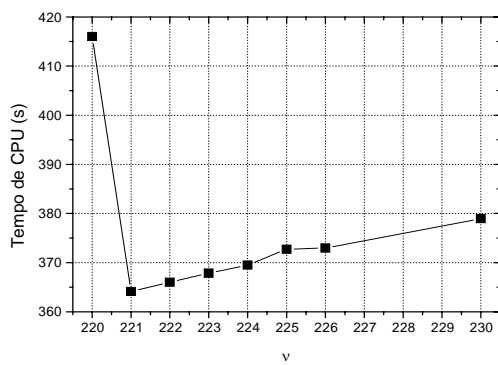
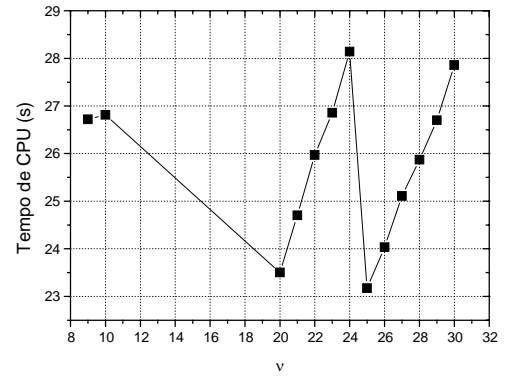
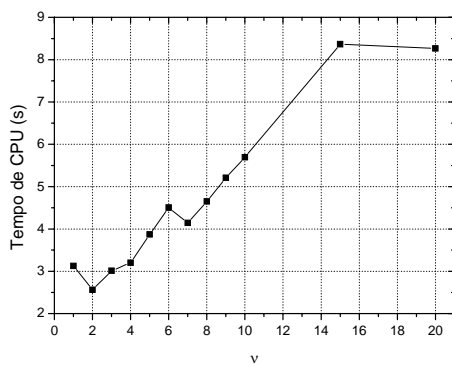
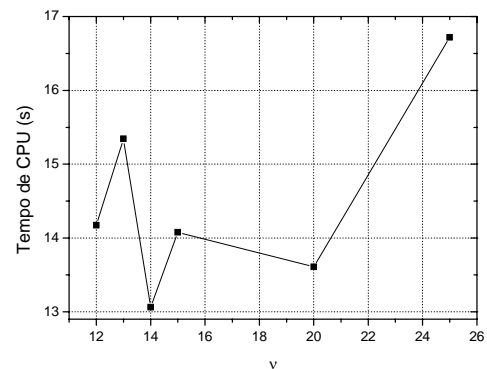
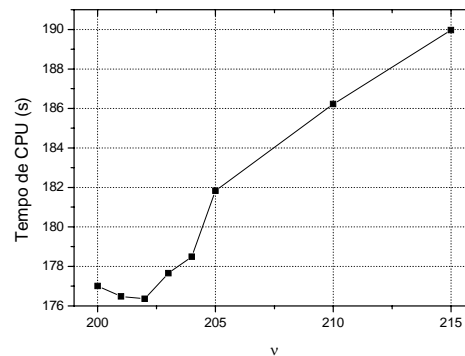
a)  $Q = 1/64$ b)  $Q = 1/16$ c)  $Q = 1/4$ d)  $Q = 1$ e)  $Q = 4$ f)  $Q = 16$ g)  $Q = 64$ 

Figura D.6: Tempo de CPU *versus*  $\nu$  para a equação de Laplace senoidal, algoritmo EP e  $E = 262.144$ .

a)  $Q = 1/16$ b)  $Q = 1/4$ c)  $Q = 1$ d)  $Q = 4$ e)  $Q = 16$ 

**Figura D.7: Tempo de CPU versus  $\nu$  para a equação de Laplace senoidal, algoritmo EP e  $E = 4.194.304$ .**

A Tab. D.3 apresenta o  $\nu_{\text{ótimo}}$  para o algoritmo EP e a equação de Laplace senoidal para cada razão de aspecto em estudo. Observa-se que nem sempre foi possível obter um único  $\nu_{\text{ótimo}}$  para todos os tamanhos de problema. Isto ocorre devido a instabilidade do algoritmo EP para problemas anisotrópicos. Verifica-se que para a razão de aspecto  $Q = 16$  não foi obtido um único  $\nu_{\text{ótimo}}$ , mas sim um intervalo contendo o  $\nu_{\text{ótimo}}$ . Através das Figs. D.5



a D.7, pode-se observar que a variação do tempo de CPU neste intervalo é muito pequena. Portanto calculou-se para cada tamanho de problema o tempo médio de CPU ( $\bar{t}_{CPU}$ ), o desvio padrão ( $s$ ) e o coeficiente de variação ( $CV$ ). A Tab. D.4 apresenta os resultados para cada um destes parâmetros. Verifica-se que os coeficientes de variação são pequenos, menores que 1%, significando que a variabilidade do tempo de CPU em relação à média é pequena, portanto pode-se utilizar qualquer  $\nu$  dentro do intervalo que o acréscimo no tempo de CPU não será significativo. Os valores não preenchidos na Tab. D.3 correspondem à simulações cujo tempo de CPU é muito alto.

**Tabela D.3:  $\nu_{\text{ótimo}}$  para cada razão de aspecto, algoritmo EP e equação de Laplace senoidal.**

$Q$	$E$	65.536	262.144	4.194.304
1/64		110	44	-
1/16		221	221	221
1/4		25	25	25
1		2	3	2
4		14	13	14
16		$200 \leq \nu \leq 250$	$200 \leq \nu \leq 250$	$200 \leq \nu \leq 204$
64		-	45	10

**Tabela D.4: Parâmetros para o intervalo recomendado de  $\nu$ ,  $Q = 16$ , EP, e equação de Laplace senoidal.**

65.536			262.144			4.194.304		
$\bar{t}_{CPU}$ (s)	$s$ (s)	$CV$	$\bar{t}_{CPU}$ (s)	$s$ (s)	$CV$	$\bar{t}_{CPU}$ (s)	$s$ (s)	$CV$
2,852	0	0	12,156	0	0	177	1,04	0,59%

As Figs. D.8 e D.9 apresentam o tempo de CPU em função do número de iterações internas para a equação de Laplace senoidal, algoritmo SE e razões de aspecto  $Q = 1/64, 1/16, 1/4, 4, 16$  e  $64$ . Consideram-se problemas com 65.536 e 262.144 elementos com as malhas já apresentadas na Tab. D.1.

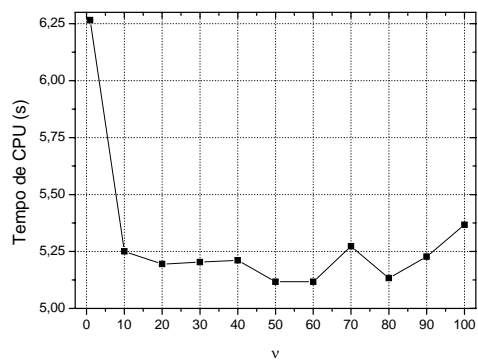
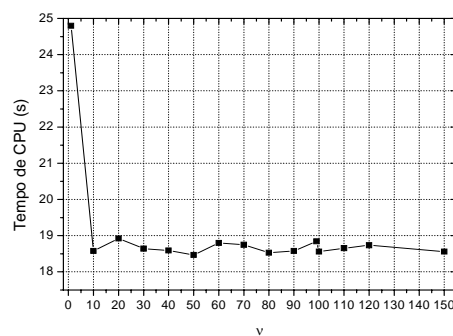
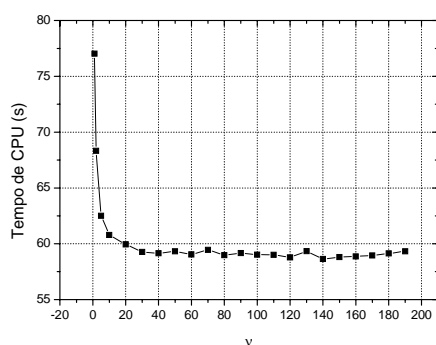
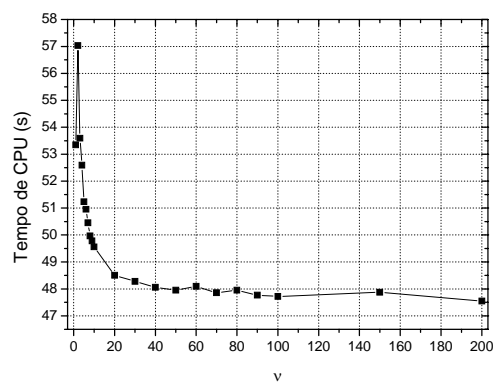
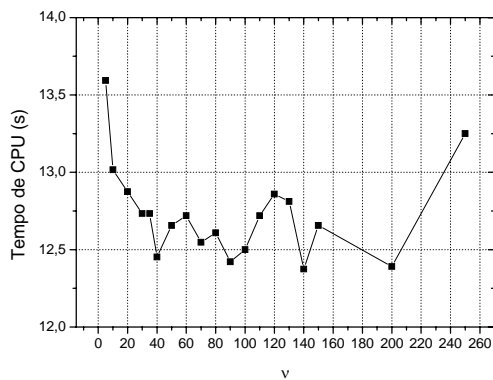
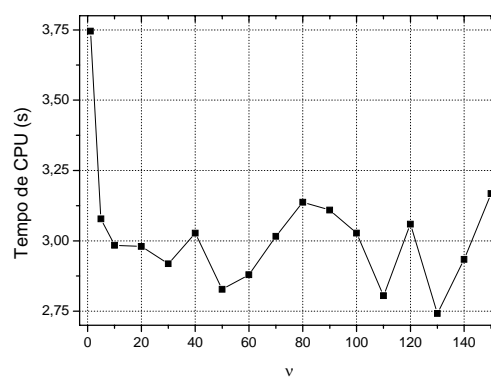
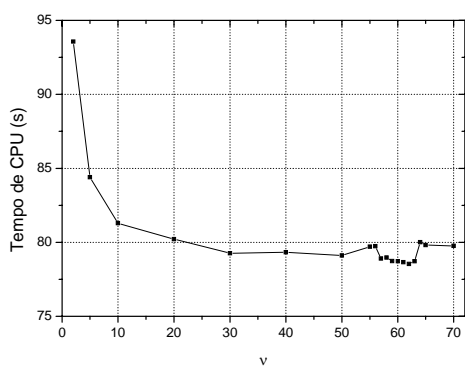
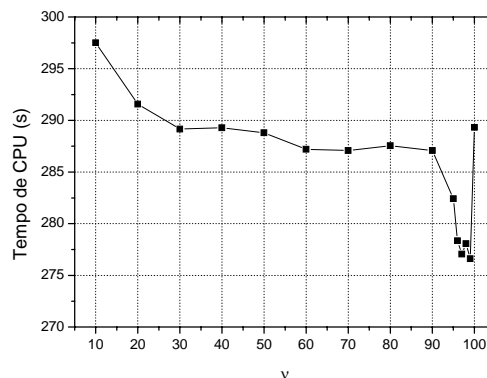
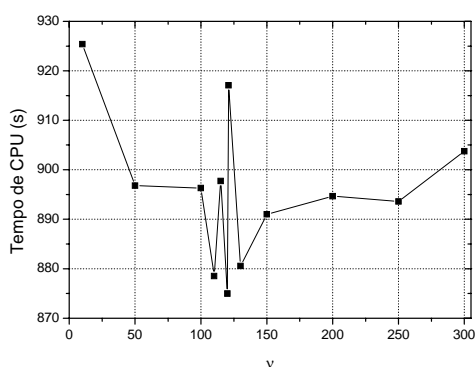
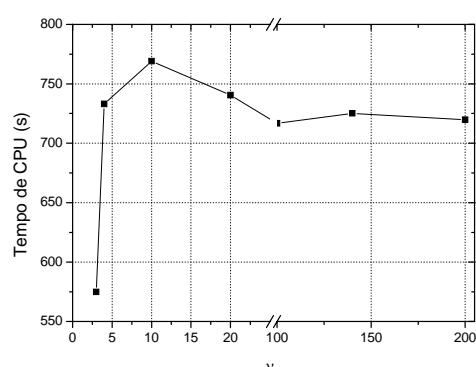
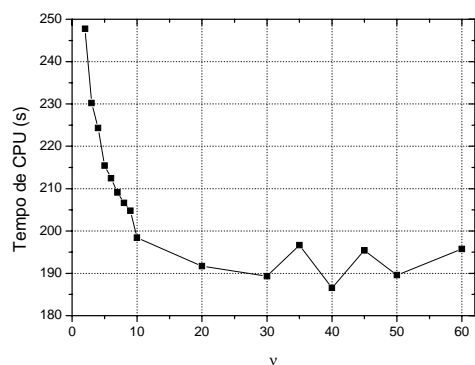
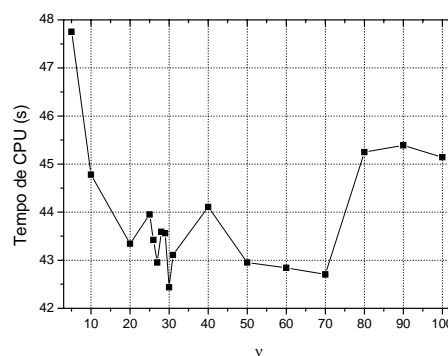
a)  $Q = 1/64$ b)  $Q = 1/16$ c)  $Q = 1/4$ d)  $Q = 4$ e)  $Q = 16$ f)  $Q = 64$ 

Figura D.8: Tempo de CPU *versus*  $\nu$  para a equação de Laplace senoidal, algoritmo SE e  $E = 65.536$ .

a)  $Q = 1/64$ b)  $Q = 1/16$ c)  $Q = 1/4$ d)  $Q = 4$ e)  $Q = 16$ f)  $Q = 64$ 

**Figura D.9: Tempo de CPU versus  $\nu$  para a equação de Laplace senoidal, algoritmo SE e  $E = 262.144$ .**

As Figs. D.10 a D.13 apresentam o tempo de CPU em função do número de iterações internas para a equação de Laplace senoidal, algoritmo SEC e razões de aspecto  $Q = 1/64$ ,  $1/16$ ,  $1/4$ ,  $4$ ,  $16$  e  $64$ . Consideram-se problemas com 4.096, 65.536, 262.144 e 1.048.576 elementos com as malhas já apresentadas na Tab. D.1.

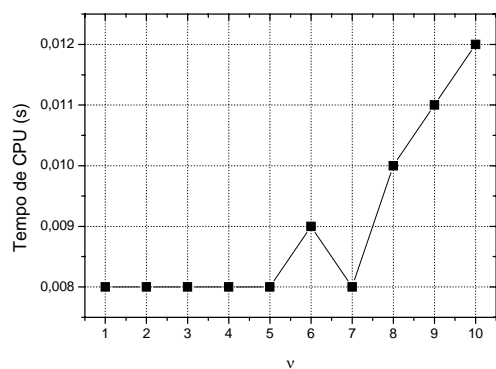
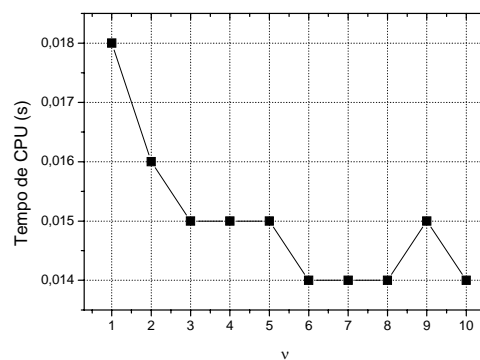
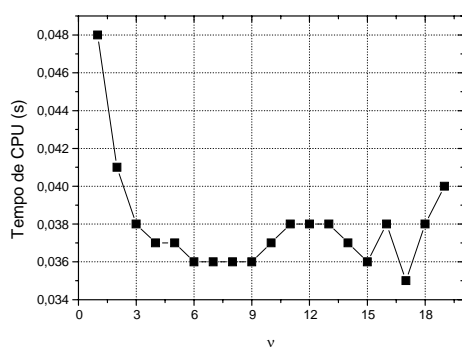
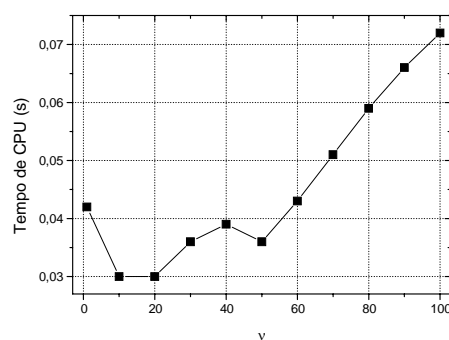
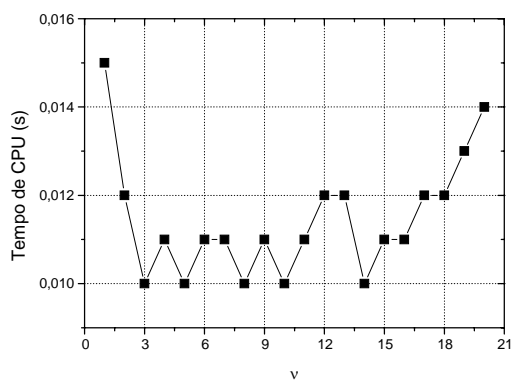
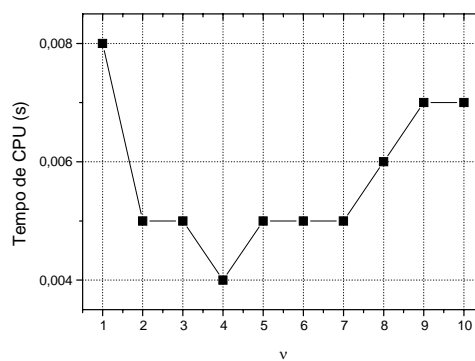
a)  $Q = 1/64$ b)  $Q = 1/16$ c)  $Q = 1/4$ d)  $Q = 4$ e)  $Q = 16$ f)  $Q = 64$ 

Figura D.10: Tempo de CPU *versus*  $\nu$  para a equação de Laplace senoidal, algoritmo SEC e  $E = 4.096$ .

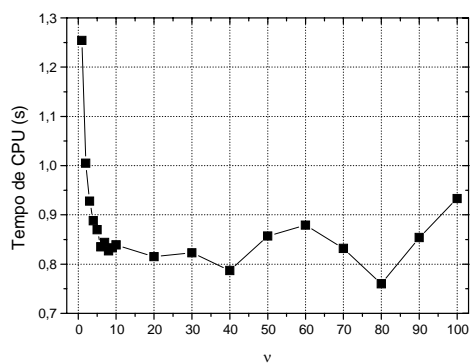
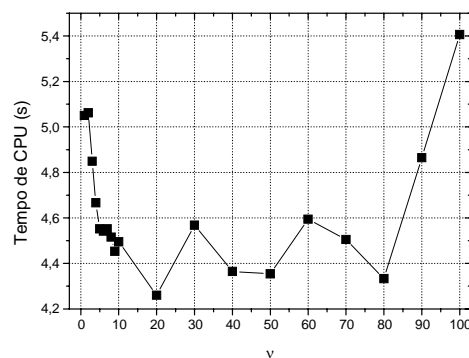
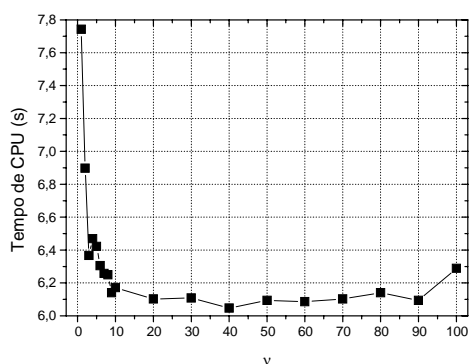
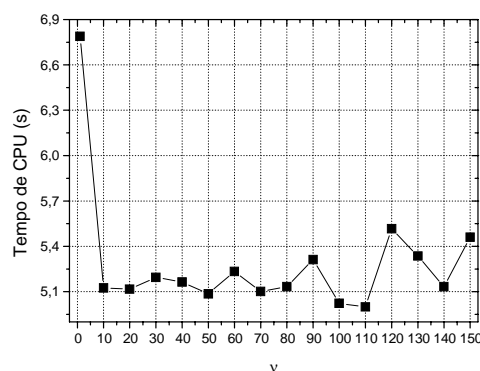
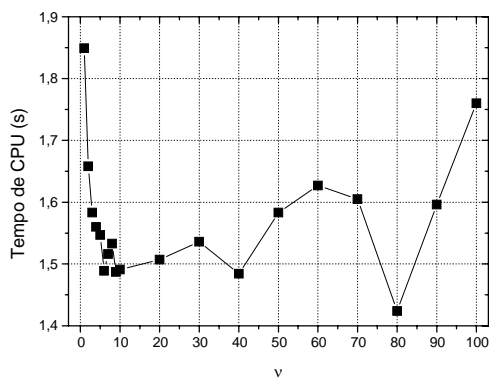
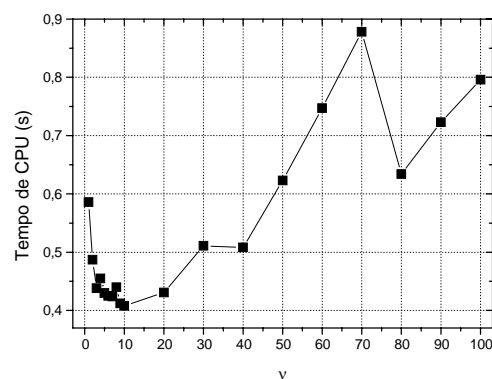
a)  $Q = 1/64$ b)  $Q = 1/16$ c)  $Q = 1/4$ d)  $Q = 4$ e)  $Q = 16$ f)  $Q = 64$ 

Figura D.11: Tempo de CPU *versus*  $\nu$  para a equação de Laplace senoidal, algoritmo SEC e  $E = 65.536$ .

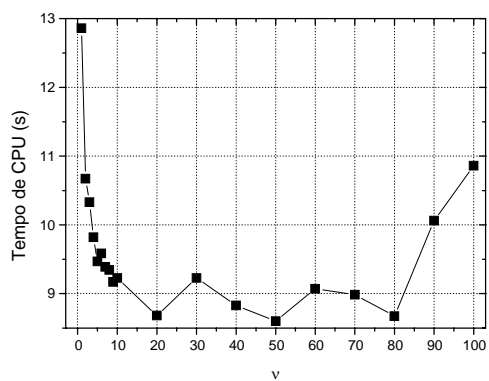
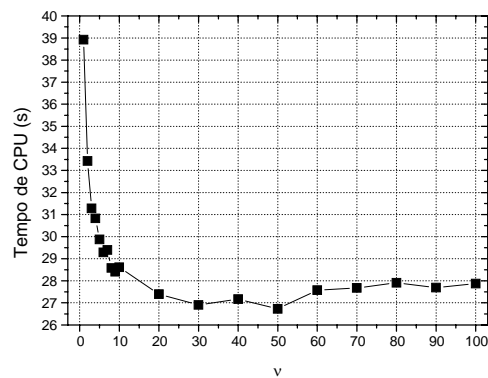
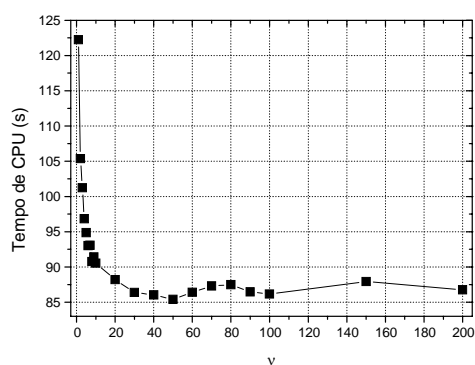
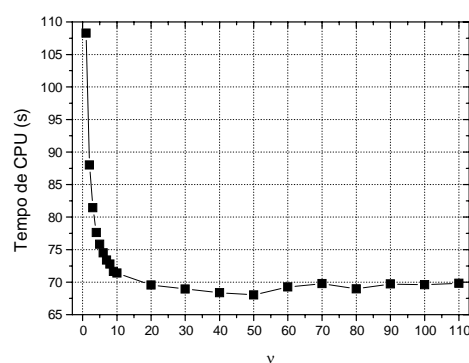
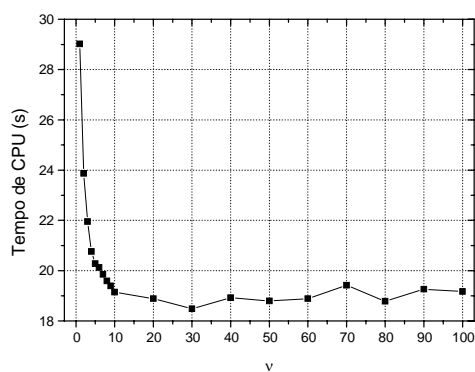
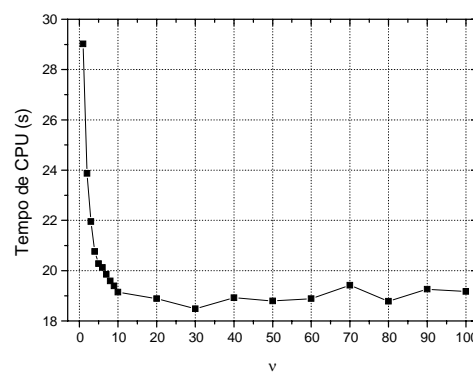
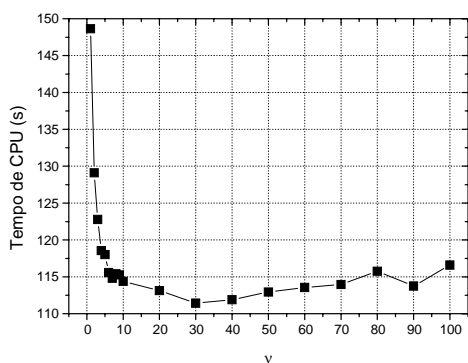
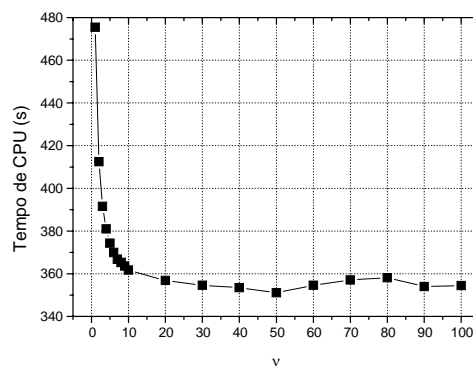
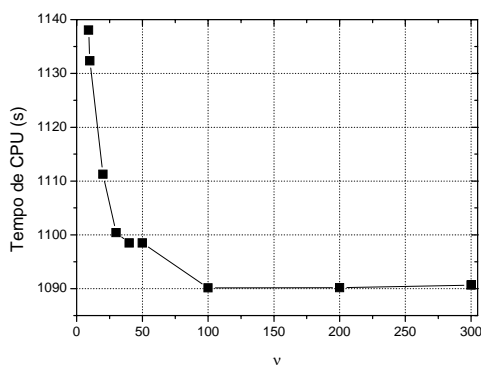
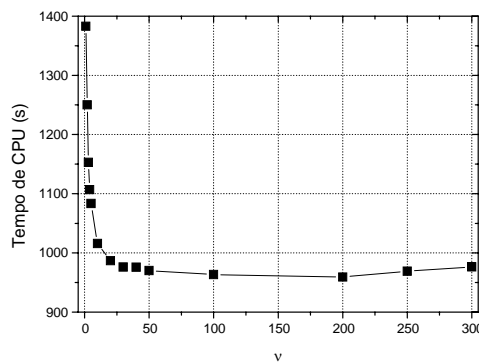
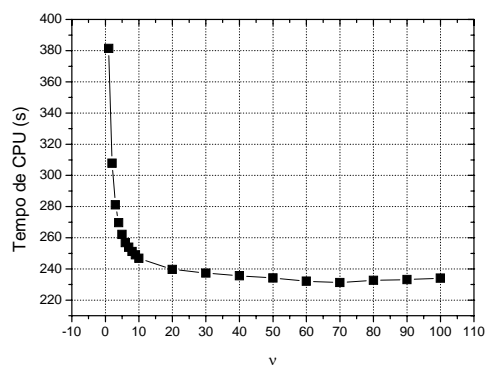
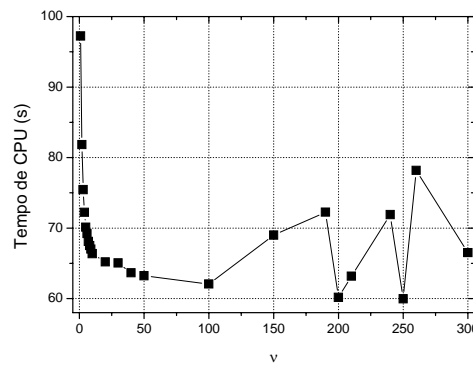
a)  $Q = 1/64$ b)  $Q = 1/16$ c)  $Q = 1/4$ d)  $Q = 4$ e)  $Q = 16$ f)  $Q = 64$ 

Figura D.12: Tempo de CPU *versus*  $\nu$  para a equação de Laplace senoidal, algoritmo SEC e  $E = 262.144$ .

a)  $Q = 1/64$ b)  $Q = 1/16$ c)  $Q = 1/4$ d)  $Q_{N,h} = 4$ e)  $Q = 16$ f)  $Q = 64$ 

**Figura D. 13: Tempo de CPU versus  $\nu$  para a equação de Laplace senoidal, algoritmo SEC e  $E = 1.048.576$**

A Tab. D.5 apresenta o intervalo ótimo de iterações internas ( $\nu_{\text{ótimo}}$ ) para a equação de Laplace senoidal, algoritmo SEC e razões de aspecto  $Q = 1/64, 1/16, 1/4, 4, 16$  e  $64$ . Verifica-se que não existe um único intervalo ótimo para todas as razões de aspectos em estudo e que este número varia de acordo com o tamanho do problema.

**Tabela D. 5:**  $\nu_{\text{ótimo}}$  para cada razão de aspecto, algoritmo SEC e equação de Laplace senoidal.

$E$	<b>4.096</b>	<b>65.536</b>	<b>262.144</b>	<b>1.048.576</b>
$Q$				
1/64	$1 \leq \nu \leq 7$	80	$70 \leq \nu \leq 90$	$30 \leq \nu \leq 50$
1/16	$6 \leq \nu \leq 10$	$10 \leq \nu \leq 50$	$30 \leq \nu \leq 60$	$30 \leq \nu \leq 60$
1/4	$10 \leq \nu \leq 50$	$20 \leq \nu \leq 70$	$30 \leq \nu \leq 60$	$100 \leq \nu \leq 300$
4	$10 \leq \nu \leq 20$	$100 \leq \nu \leq 110$	$30 \leq \nu \leq 80$	$100 \leq \nu \leq 200$
16	$2 \leq \nu \leq 18$	$70 \leq \nu \leq 90$	$20 \leq \nu \leq 80$	$60 \leq \nu \leq 80$
64	$2 \leq \nu \leq 7$	15 e 18	$20 \leq \nu \leq 50$	250

As Figs. D.14 e D.15 apresentam o tempo de CPU em função do número de iterações internas para a equação de Laplace senoidal, algoritmo EP-SE e razões de aspecto  $Q = 1/64$ ,  $1/16$ ,  $1/4$ ,  $1$ ,  $4$ ,  $16$  e  $64$ . Consideram-se problemas com 65.536 e 262.144 elementos. Os gráficos apresentados não utilizam todos os resultados obtidos nas simulações. O número de pontos foi reduzido de forma a contemplar o intervalo que contém o  $\nu_{\text{ótimo}}$ .

A Tab. D.6 apresenta o número de iterações internas que utilizam o menor tempo de CPU para a equação de Laplace senoidal. Verifica-se que a razão de aspecto influencia fortemente o número ótimo de iterações internas. O tamanho do problema também influencia o número ótimo de iterações internas.

**Tabela D.6:**  $\nu_{\text{ótimo}}$  para cada razão de aspecto, algoritmo EP-SE e equação de Laplace senoidal.

$E$	<b>65.536</b>	<b>262.144</b>
$Q$		
1/64	5000	5000
1/16	160	200
1/4	25	25
4	13	13
16	180	170
64	3500	2500



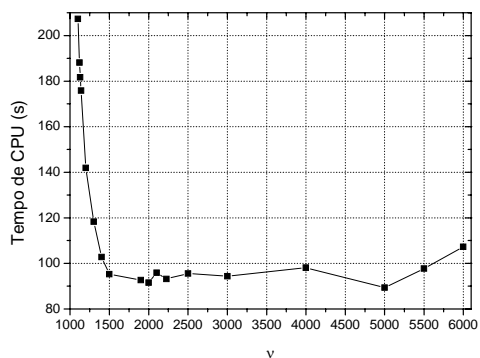
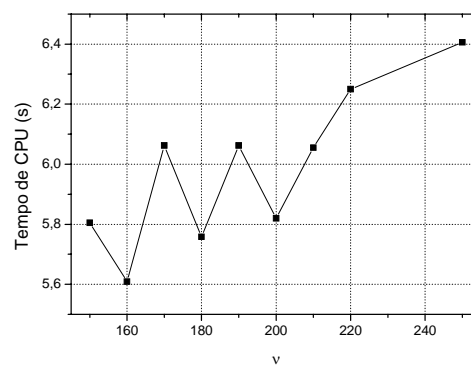
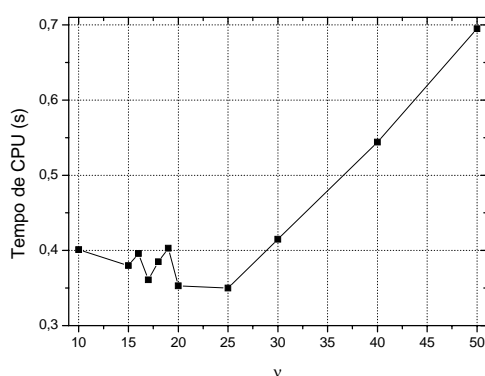
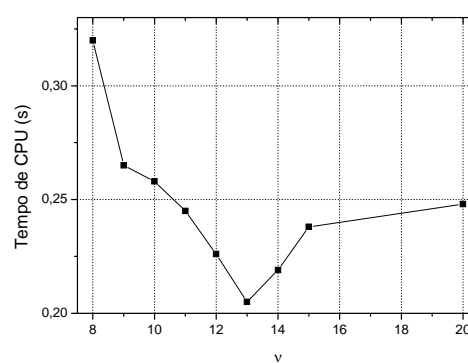
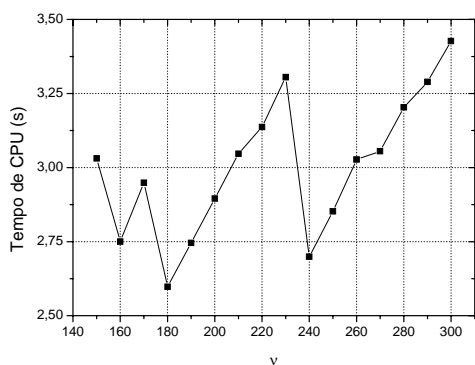
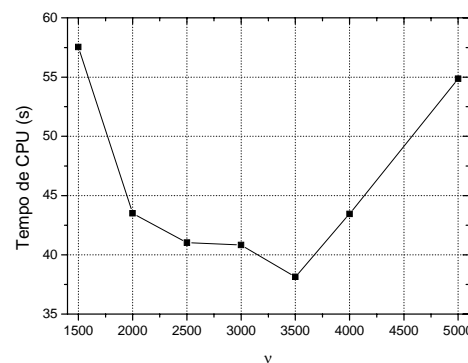
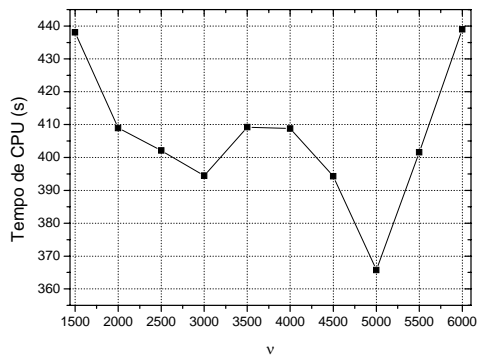
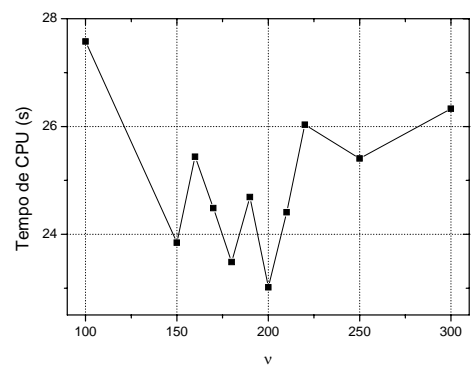
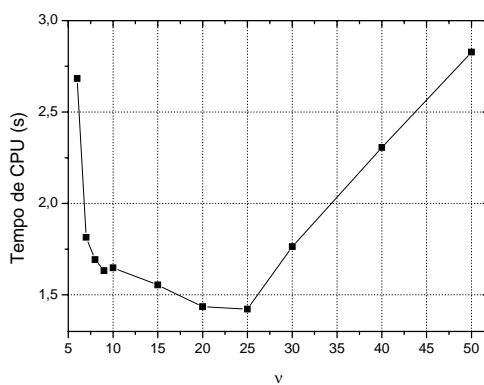
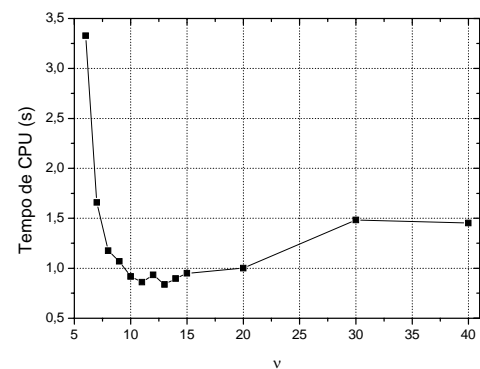
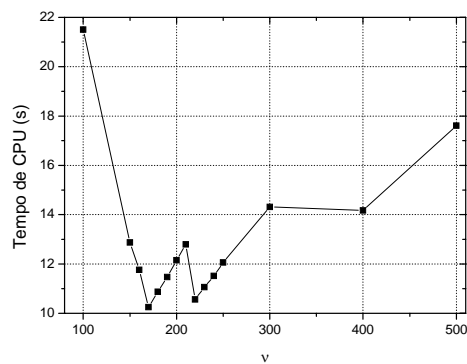
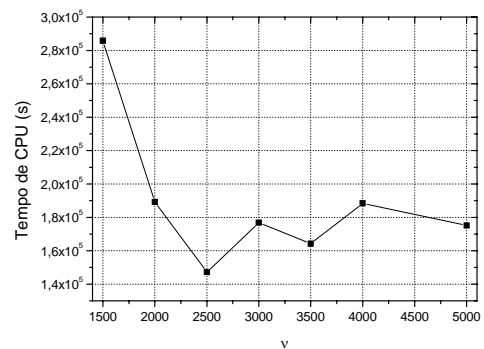
a)  $Q = 1/64$ b)  $Q = 1/16$ c)  $Q = 1/4$ d)  $Q = 4$ e)  $Q = 16$ f)  $Q = 64$ 

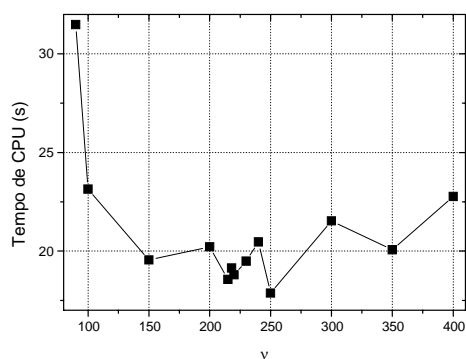
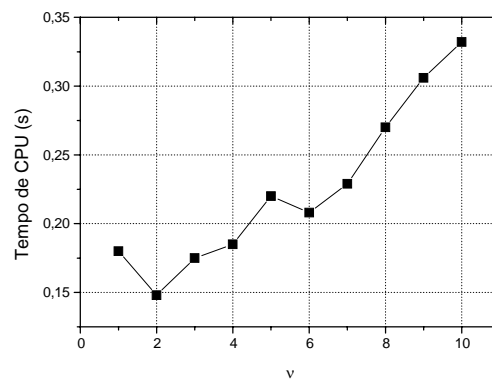
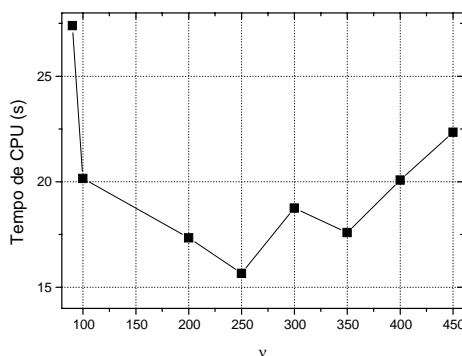
Figura D.14: Tempo de CPU versus  $\nu$  para a equação de Laplace senoidal, algoritmo EP-SE e  $E = 65.536$ .

a)  $Q = 1/64$ b)  $Q = 1/16$ c)  $Q = 1/4$ d)  $Q = 4$ e)  $Q = 16$ f)  $Q = 64$ 

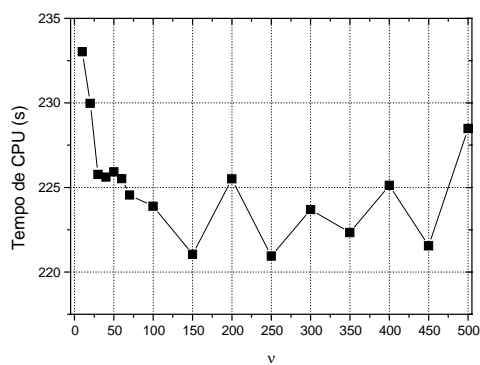
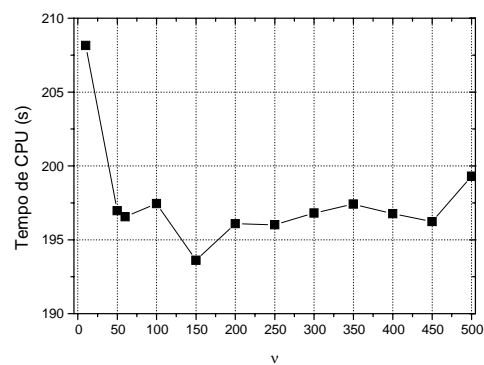
**Figura D.15: Tempo de CPU versus  $\nu$  para a equação de Laplace senoidal, algoritmo EP-SE e  $E = 262.144$ .**

As Figs. D.16 à D.19 apresentam o tempo de CPU em função do número de iterações internas para a equação de Laplace linear e algoritmos EP, SE, SEC e EP-SE, respectivamente. Consideram-se as razões de aspecto  $Q = 1/16$ , 1 e 16 e problemas com 262.144 elementos com as malhas apresentadas na Tab. D.1. Através dos resultados obtidos verificou-se que são necessárias muitas iterações internas para obter um menor tempo de

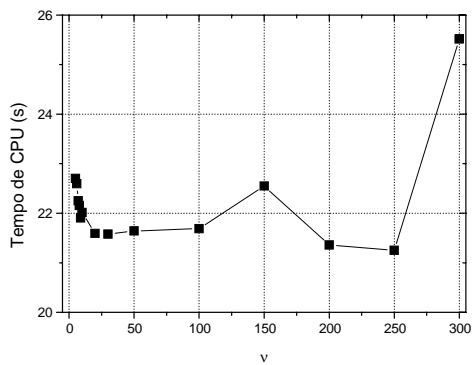
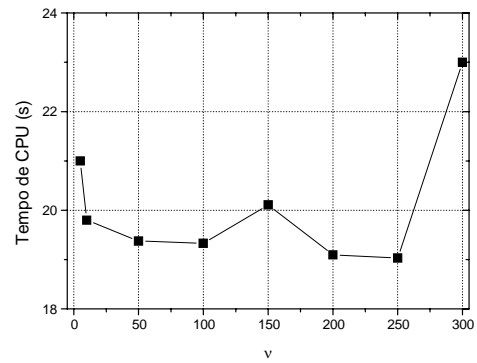
CPU. Usar poucas iterações internas pode ocasionar a divergência na solução. Verifica-se que para as razões de aspecto 1/16 e 16 obteve-se  $v_{\text{ótimo}} = 250$ , para todos os algoritmos exceto o SE. Para o caso isotrópico e o algoritmo EP, tem-se  $v_{\text{ótimo}} = 2$ .

a)  $Q = 1/16$ b)  $Q = 1$ c)  $Q = 16$ 

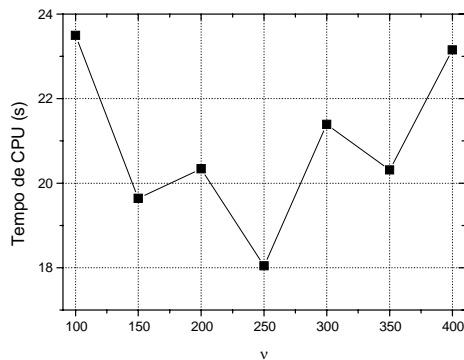
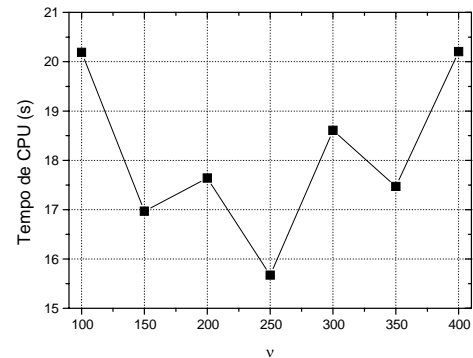
**Figura D.16: Tempo de CPU versus  $\nu$  para a equação de Laplace linear, algoritmo EP e  $E = 262.144$ .**

a)  $Q = 1/16$ b)  $Q = 16$ 

**Figura D.17: Tempo de CPU versus  $\nu$  para a equação de Laplace linear, algoritmo SE e  $E = 262.144$ .**

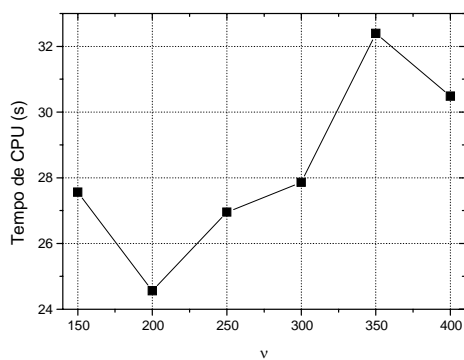
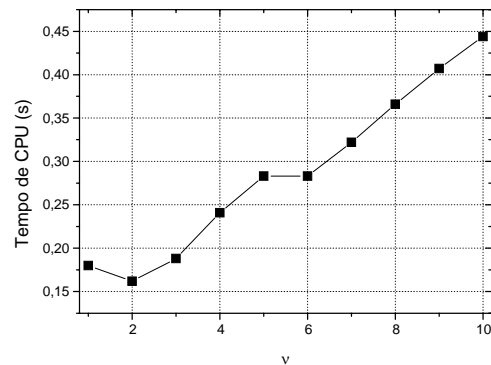
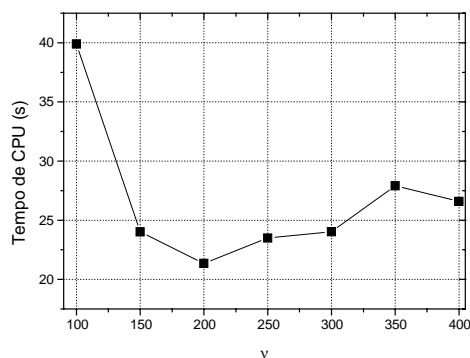
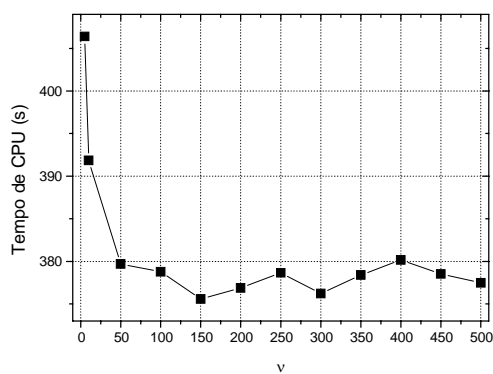
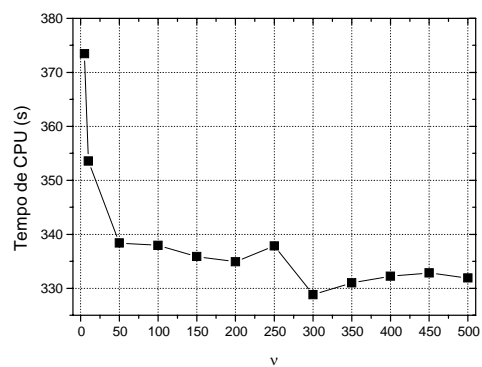
a)  $Q = 1/16$ b)  $Q = 16$ 

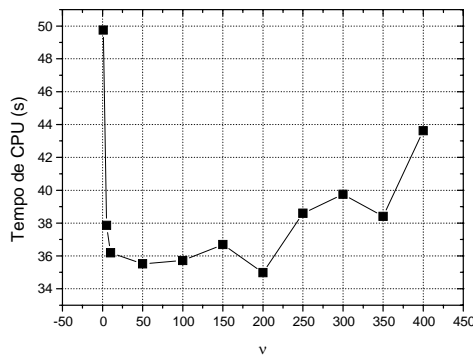
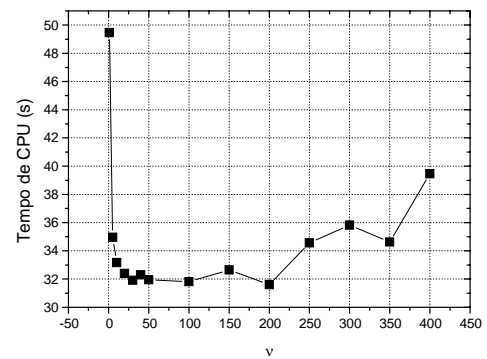
**Figura D.18: Tempo de CPU versus  $\nu$  para a equação de Laplace linear, algoritmo SEC e  $E = 262.144$ .**

a)  $Q = 1/16$ b)  $Q = 16$ 

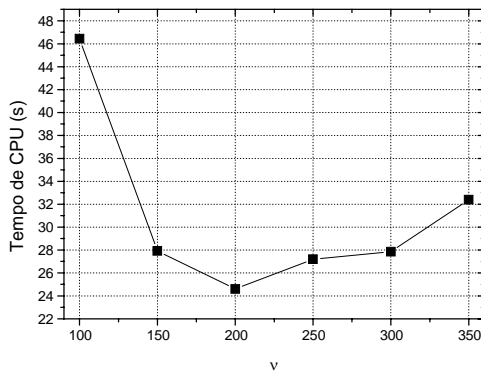
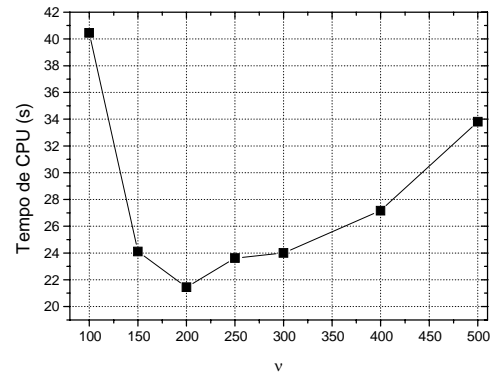
**Figura D.19: Tempo de CPU versus  $\nu$  para a equação de Laplace linear, algoritmo EP-SE e  $E = 262.144$ .**

As Figs. D.20 à D.23 apresentam o tempo de CPU em função do número de iterações internas para a equação de Poisson, razões de aspecto  $Q = 1/16$ , 1 e 16 e algoritmos EP, SE, SEC e EP-SE, respectivamente. Consideram-se problemas com  $E = 262.144$  elementos com as malhas apresentadas na Tab. D.1. Através dos resultados obtidos verificou-se que são necessárias muitas iterações internas para obter um menor tempo de CPU. Usar poucas iterações internas pode ocasionar a divergência na solução. Para o problema em estudo com  $E = 262.144$  elementos e razões de aspecto 1/16 e 16 obteve-se  $\nu_{\text{ótimo}} = 200$  para todos os algoritmos com exceção do SE. Para o algoritmo EP e o caso isotrópico tem-se  $\nu_{\text{ótimo}} = 2$ .

a)  $Q = 1/16$ b)  $Q = 1$ c)  $Q = 16$ Figura D.20: Tempo de CPU *versus*  $\nu$  para a equação de Poisson, algoritmo EP e  $E = 262.144$ .a)  $Q = 1/16$ b)  $Q = 16$ Figura D.21: Tempo de CPU *versus*  $\nu$  para a equação de Poisson, algoritmo SE e  $E = 262.144$ .

a)  $Q = 1/16$ b)  $Q = 16$ 

**Figura D.22: Tempo de CPU versus  $\nu$  para a equação de Poisson, algoritmo SEC e  $E = 262.144$ .**

a)  $Q = 1/16$ b)  $Q = 16$ 

**Figura D.23: Tempo de CPU versus  $\nu$  para a equação de Poisson, algoritmo EP-SE e  $E = 262.144$ .**

## D2. ESTUDO DA INFLUÊNCIA DO NÚMERO DE ELEMENTOS ( $E$ )

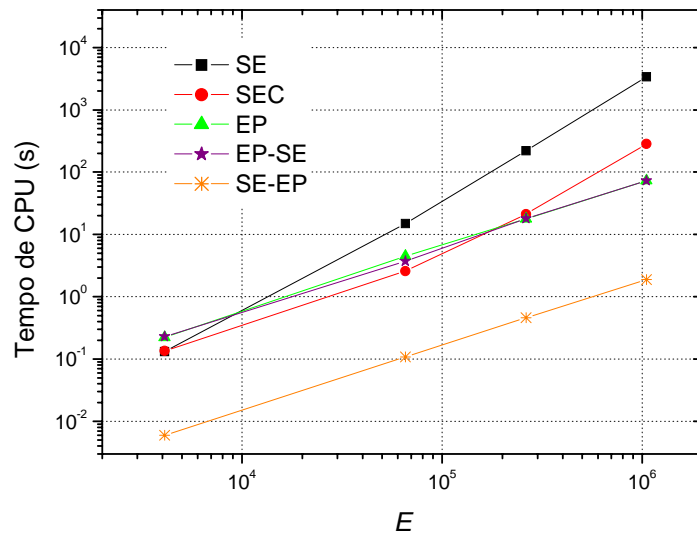
Para as equações de Laplace linear e Poisson a análise do número de elementos foi realizada utilizando-se as razões de aspecto  $Q = 1/64, 1/16, 1/4, 1, 4, 16$  e  $64$  e as malhas apresentadas na Tab. D.7.

As Figs. D.24 e D.25 apresentam o tempo de CPU em função do número de elementos para a equação de Laplace linear, algoritmos SE, SEC, EP-SE e SE-EP. A Fig. D.24 apresenta os resultados para a razão de aspecto  $Q = 1/16$  e a Fig. D.24 para a razão de aspecto  $Q = 16$ . Em ambas as figuras verifica-se que o tempo de CPU aumenta quando aumenta o número de

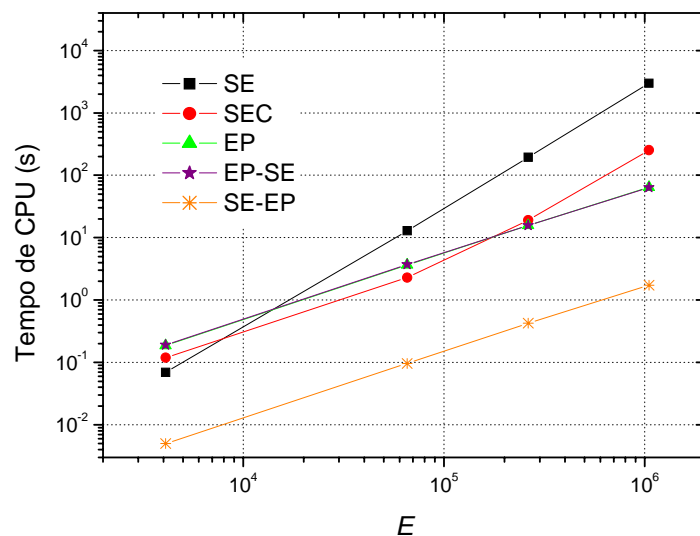
elementos. Verifica-se também que o menor tempo de CPU é obtido para o algoritmo SE-EP. Os resultados obtidos para as razões de aspecto 1/64, 1/4, 4 e 64 foram similares.

**Tabela D.7: Malhas utilizadas para análise de  $E$  para as equações de Laplace linear e Poisson.**

$Q$	Malhas				
1/64	513x9	2049x33	4097x65	8193x129	
1/16	257x17	1025x65	2049x129	4097x257	
1/4	129x33	513x129	1025x257	2049x513	
1	65x65	257x257	513x513	1025x1025	
4	33x129	129x513	257x1025	513x2049	
16	17x257	65x1025	129x2049	257x4097	
64	9x513	33x2049	65x4097	129x8193	



**Figura D.24: Tempo de CPU versus número de elementos para  $Q = 1/16$  e equação de Laplace linear.**



**Figura D.25: Tempo de CPU versus número de elementos para  $Q = 16$  e equação de Laplace linear.**

As Figs. D.26 e D.27 apresentam o tempo de CPU em função do número de elementos para a equação de Poisson, algoritmos SE, SEC, EP-SE e SE-EP. A Fig. D.26 apresenta os resultados para a razão de aspecto  $Q = 1/16$  e a Fig. D.27 para a razão de aspecto  $Q = 16$ . Em ambas as figuras verifica-se que o tempo de CPU aumenta quando aumenta o número de elementos. Verifica-se também que o menor tempo de CPU é obtido para o algoritmo SE-EP. Os resultados obtidos para as razões de aspecto 1/64, 1/4, 4 e 64 foram similares.

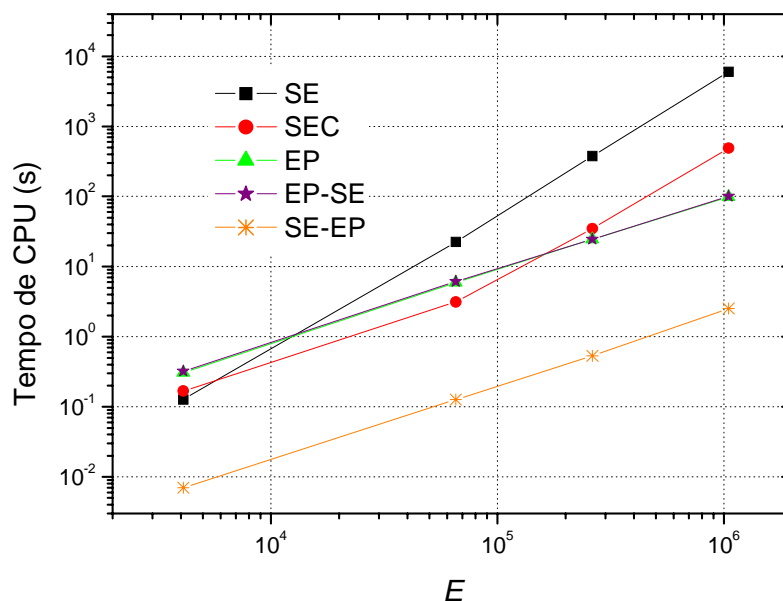


Figura D.26: Tempo de CPU *versus* número de elementos para  $Q = 1/16$  e equação de Poisson.

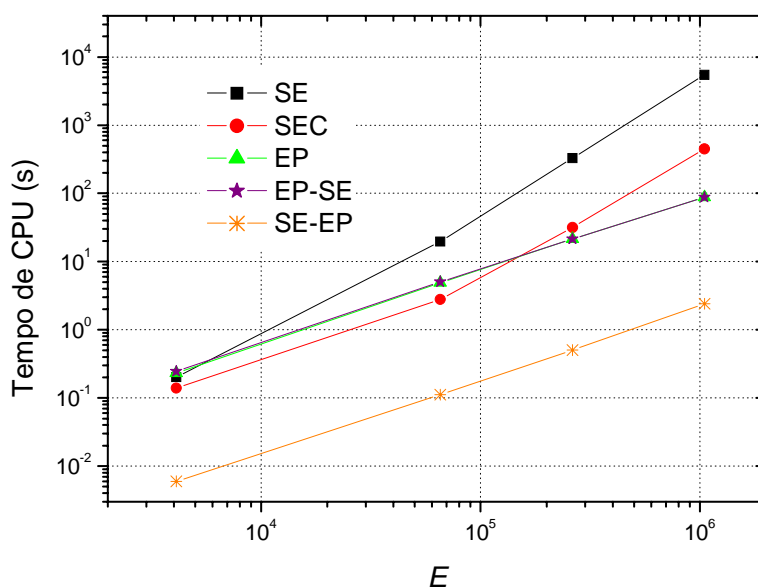


Figura D.27: Tempo de CPU *versus* número de elementos para  $Q = 16$  e equação de Poisson.



### D3. ANÁLISE DE COMPLEXIDADE

Utilizando-se as malhas e razões de aspecto definidas na Tab. D.9 fez-se uma análise da complexidade dos algoritmos EP, SE, SEC, SE-EP e EP. Calculou-se o expoente  $p$ , obtido pelo do método dos mínimos quadrados, para a função dada pela Eq. (5.1).

As Tab. D.8 e D.9 apresentam o coeficiente  $c$  e a ordem  $p$  dos ajustes de curvas obtidos para os algoritmos em estudo e equação de Laplace linear. Observa-se que os menores valores de  $p$  ocorrem para os algoritmos EP, EP-SE e SE-EP. Na Tab. D.9 pode-se constatar que entre os algoritmos EP, EP-SE e SE-EP os coeficientes  $c$  são menores no algoritmo SE-EP. Os espaços das Tabs. D.8 e D.9 que não estão preenchidos correspondem à simulações cujo tempo de CPU é muito alto.

**Tabela D.8: Valor da ordem ( $p$ ) da Eq. (5.1) para os algoritmos de engrossamento para a equação de Laplace linear.**

Algoritmos						
	SE	SEC	EP	EP-SE	SE-EP	
$Q$						
1/64	1,55	1,14	-	-	1,04	
1/16	1,83	1,36	1,04	1,04	1,04	
1/4	1,97	1,61	1,03	1,03	1,05	
1	-	-	1,04	-	-	
4	1,97	1,61	1,04	1,04	1,05	
16	1,92	1,36	1,05	1,05	1,06	
64	1,70	1,19	-	-	1,06	

**Tabela D.9: Valor do coeficiente ( $c$ ) da Eq. (5.1) para os algoritmos de engrossamento para a equação de Laplace linear.**

Algoritmos						
	SE	SEC	EP	EP-SE	SE-EP	
$Q$						
1/64	2,48E-07	9,50E-06	-	-	1,07E-06	
1/16	2,91E-08	1,24E-06	3,89E-05	3,79E-05	1,07E-06	
1/4	1,66E-08	1,61E-07	2,72E-05	2,71E-05	8,16E-07	
1	-	-	3,24E-07	-	-	
4	1,60E-08	1,45E-07	2,23E-05	2,27E-05	7,97E-07	
16	7,53E-09	1,08E-06	2,96E-05	2,89E-05	7,63E-07	
64	3,05E-08	4,57E-6	-	-	7,54E-07	

As Tab. D.10 e D.11 apresentam o coeficiente  $c$  e a ordem  $p$  dos ajustes de curvas obtidos para os algoritmos em estudo e equação de Poisson. Observa-se que os menores valores de  $p$  ocorrem para os algoritmos EP, EP-SE e SE-EP. Na Tab. D.11 pode-se constatar que entre os algoritmos EP, EP-SE e SE-EP os coeficientes  $c$  são menores no algoritmo SE-EP. Os espaços das Tabs. D.10 e D.11 que não estão preenchidos correspondem à simulações cujo tempo de CPU é muito alto.

**Tabela D.10: Valor da ordem ( $p$ ) da Eq. (5.1) para os algoritmos de engrossamento para a equação de Poisson.**

Algoritmos		SE	SEC	EP	EP-SE	SE-EP
$Q$						
1/64		1,69	1,20	-	-	1,07
1/16		1,94	1,43	1,04	1,04	1,06
1/4		1,99	1,66	1,04	1,04	1,04
1		-	-	1,06	-	-
4		2,04	1,67	1,05	1,05	1,05
16		1,84	1,44	1,07	1,06	1,08
64		1,48	1,13	-	-	1,12

**Tabela D.11: Valor do coeficiente ( $c$ ) da Eq. (5.1) para os algoritmos de engrossamento para a equação de Poisson.**

Algoritmos		SE	SEC	EP	EP-SE	SE-EP
$Q$						
1/64		7,40E-08	6,40E-06	-	-	9,31E-07
1/16		1,16E-08	7,95E-07	5,37E-05	5,96E-05	1,03E-06
1/4		1,82E-08	1,18E-07	2,82E-05	2,99E-05	1,02E-06
1		-	-	2,86E-07	-	-
4		1,08E-08	1,06E-07	2,22E-05	2,39E-05	9,52E-07
16		3,95E-08	6,08E-07	3,20E-05	3,65E-05	7,51E-07
64		6,40E-07	9,13E-06	-	-	4,31E-07

## APÊNDICE E: SEMI-ENGROSSAMENTO SEGUIDO DE ENGROSSAMENTO PADRÃO

Este apêndice possui alguns dados adicionais referentes ao capítulo 6. No capítulo 6 foi apresentado um estudo do algoritmo SE-EP envolvendo as equações de Laplace senoidal, Laplace linear e Poisson. Os parâmetros quantitativos para as três equações em estudo foram os mesmos, portanto apenas alguns resultados foram apresentados. Na maior parte, envolvendo somente a equação de Laplace senoidal. Neste apêndice são apresentados os resultados para as equações de Laplace linear e Poisson. Primeiramente são apresentadas as malhas utilizadas para a análise do número de elementos. A seguir é apresentado o estudo dos *solvers* e tipos de restrição. Também é realizado um estudo das iterações internas e uma comparação entre os algoritmos. Para problemas anisotrópicos comparam-se os algoritmos SE-EP padrão e SE-EP ótimo e para problemas isotrópicos os algoritmos EP e EP ótimo.

### E1. MALHAS UTILIZADAS PARA A ANÁLISE DO NÚMERO DE ELEMENTOS

A Tab. E.1 mostra as dimensões das malhas finas ( $N = N_x \times N_y$ ) e as respectivas razões de aspecto ( $Q$ ) utilizadas para a análise do número de elementos para as equações de Laplace senoidal, Laplace linear e Poisson para o algoritmo SE-EP.

### E2. ESTUDO DOS *SOLVERS* E TIPO DE RESTRIÇÃO

A idéia do algoritmo SE-EP é aplicar o semi-engrossamento até a malha tornar-se isotrópica e em seguida aplicar o engrossamento padrão. Como são utilizados dois algoritmos diferentes os parâmetros ótimos podem diferir para cada um deles. Nesta etapa do trabalho fez-se o estudo dos seguintes parâmetros: *solvers* (Gauss-Seidel lexicográfico e *red-black*) e tipo de restrição (injeção, meia ponderação, ponderação completa, meia ponderação geométrica, ponderação geométrica completa e ponderação parcial), para cada algoritmo em particular. Primeiramente foram fixados os parâmetros referentes ao EP: *solver* Gauss-Seidel

lexicográfico e restrição por injeção. Foram realizadas uma seqüência de simulações variando-se o *solver* e o tipo de restrição para o SE. Em seguida fez-se a análise do algoritmo EP fixando-se para o SE os parâmetros ótimos obtidos no passo anterior. Posteriormente foi feita uma análise do número de iterações internas e do número de níveis. O estudo foi realizado para as equações de Laplace senoidal, Laplace linear e Poisson.

**Tabela E.1: Malhas utilizadas para análise da influência do número de elementos ( $E$ ) no tempo de CPU.**

$Q$	Malhas								
1/16384								32769x3	65537x5
1/4096							8193x3	16385x5	32769x9
1/1024						2049x3	4097x5	8193x9	16385x17
1/256					513x3	1025x5	2049x9	4097x17	8193x33
1/64				129x3	257x5	513x9	1025x17	2049x33	4097x65
1/16			33x3	65x5	129x9	257x17	513x33	1025x65	2049x129
1/4		9x3	17x5	33x9	65x17	129x33	257x65	513x129	1025x257
1	3x3	5x5	9x9	17x17	33x33	65x65	129x129	257x257	513x513
4		3x9	5x17	9x33	17x65	33x129	65x257	129x513	257x1025
16			3x33	5x65	9x129	17x257	33x513	65x1025	129x2049
64				3x129	5x257	9x513	17x1025	33x2049	65x4097
256					3x513	5x1025	9x2049	17x4097	33x8193
1024						3x2049	5x4097	9x8193	17x16385
4096							3x8193	5x16385	9x32769
16384								3x32769	5x65537
$E$	4	16	64	256	1.024	4.096	16.384	65.536	262.144

**Continuação da Tabela E1: Malhas utilizadas para a análise de  $E$ .**

$Q$	Malhas			
1/16384	131072x9	262145x17	524289x33	1048576x65
1/4096	65537x17	131073x33	262145x65	524289x129
1/1024	32769x33	65537x65	131073x129	262145x257
1/256	16385x65	32769x129	65537x257	131073x513
1/64	8193x129	16385x257	32769x513	65537x1025
1/16	4097x257	8193x513	16385x1025	32769x2049
1/4	2049x513	4097x1025	8193x2049	16385x4097
1	1025x1025	2049x2049	4097x4097	8193x8193
4	513x2049	1025x4097	2049x8193	4097x16385
16	257x4097	513x8193	1025x16385	2049x32769
64	129x8193	257x16385	513x32769	1025x65537
256	65x16385	129x32769	257x65537	513x131073
1024	33x32769	65x65537	129x131073	257x262145
4096	17x65537	33x131073	65x262145	129x524289
16384	9x131072	17x262145	33x524289	65x1048576
$E$	1.048.576	4.194.304	16.777.216	67.108.864

### E3. ESTUDO DO NÚMERO DE ITERAÇÕES INTERNAS ( $\nu$ )

A seguir foi feita a análise do  $\nu_{\text{ótimo}}$  utilizando-se os parâmetros ótimos obtidos na seção E2. Para o EP optou-se pela restrição por ponderação completa. Denota-se por  $\nu_{EP}$  o número de iterações internas para o EP e  $\nu_{SE}$  o número de iterações internas para o SE. O  $\nu_{\text{ótimo}}$  é determinado através de uma combinação de  $\nu_{EP}$  e  $\nu_{SE}$ .

As Fig. E.1 e E.2 apresentam o estudo do  $\nu$  para o algoritmo SE-EP ótimo e equação de Laplace linear. Considera-se um problema com 262.144 elementos e razões  $Q = 1/16$  (Fig. E.1) e  $Q = 16$  (Fig. E.2). Para as duas razões de aspecto verifica-se que o menor tempo de CPU é obtido para  $\nu_{EP} = \nu_{SE} = 1$ .

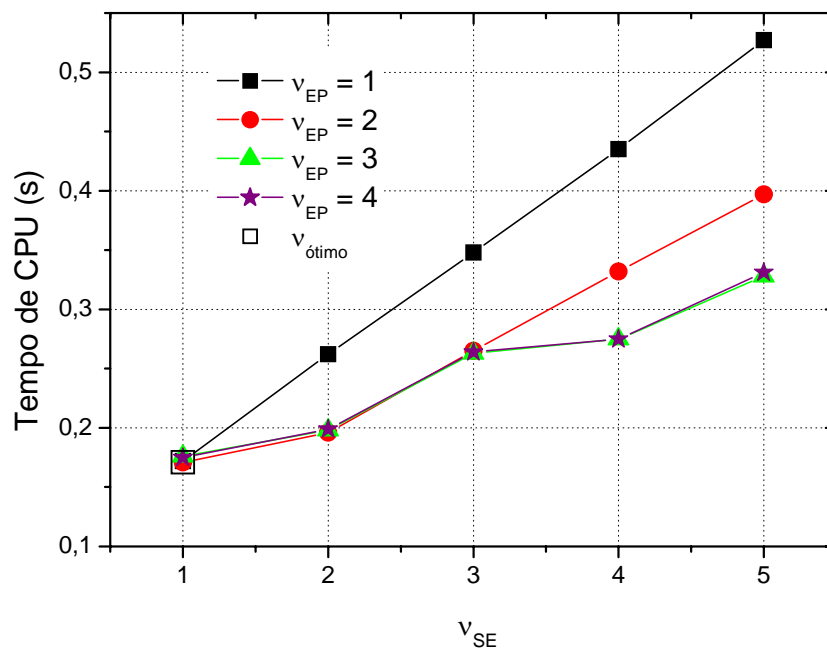


Figura E. 1: Tempo de CPU versus  $\nu_{EP}$  para  $Q = 1/6$ ,  $E = 262.144$  e SE-EP ótimo e equação de Laplace linear.

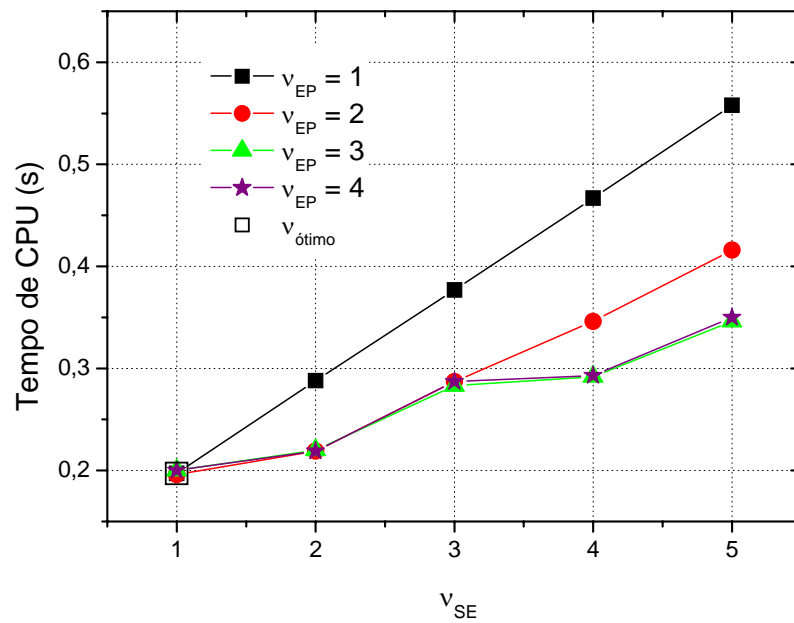


Figura E. 2: Tempo de CPU *versus*  $v_{EP}$  para  $Q = 16$ ,  $E = 262.144$  e SE-EP ótimo e equação de Laplace linear.

A Fig. E.3 apresenta o estudo do  $v$  para o algoritmo SE-EP ótimo para a equação de Poisson. Considera-se um problema com 262.144 elementos e a razão  $Q = 1/16$ . Verifica-se que o menor tempo de CPU é obtido para  $v_{EP} = v_{SE} = 1$ .

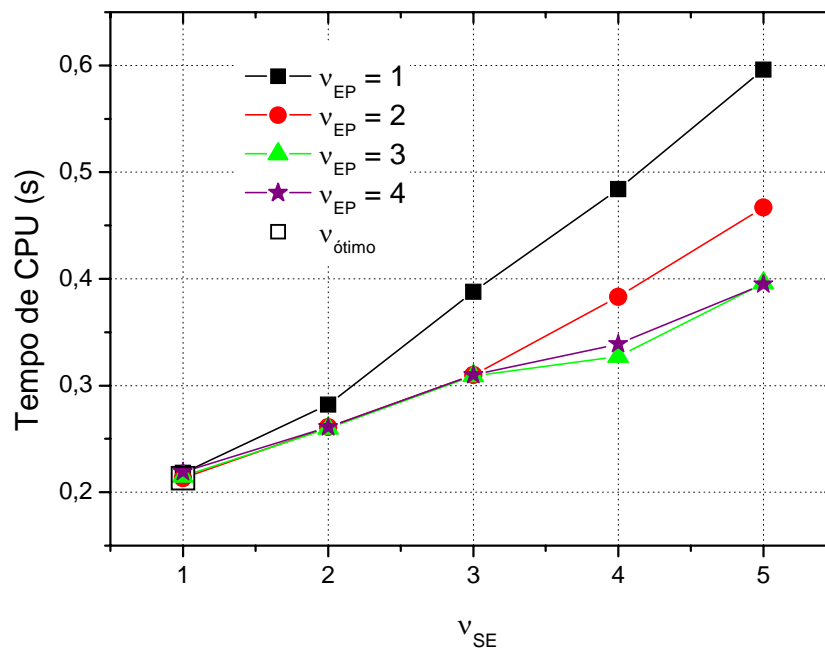
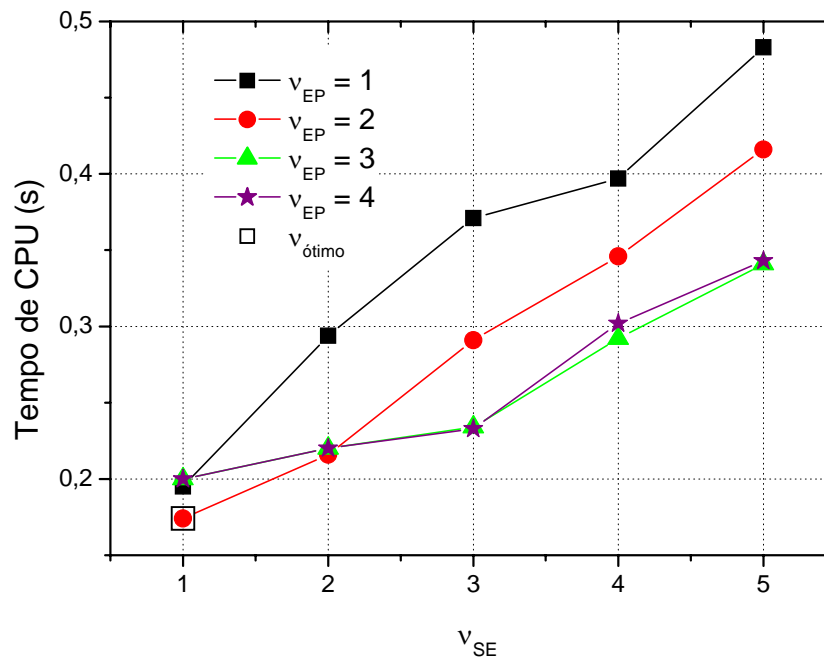


Figura E. 3: Tempo de CPU *versus*  $v_{EP}$  para  $Q = 1/16$ ,  $E = 262.144$  e SE-EP ótimo e equação de Poisson.

A Fig. E.4 apresenta o estudo do  $\nu$  para o algoritmo SE-EP ótimo para a equação de Poisson. Considera-se um problema com 262.144 elementos e a razão  $Q = 16$ . Verifica-se que o menor tempo de CPU é obtido para  $\nu_{EP} = 2$  e  $\nu_{SE} = 1$ . Observa-se também que o tempo de CPU para  $\nu_{EP} = \nu_{SE} = 1$  é bem próximo do ótimo.



**Figura E. 4: Tempo de CPU versus  $\nu_{EP}$  para  $Q = 16$ ,  $E = 262.144$  e SE-EP ótimo e equação de Poisson.**

As equações de Laplace linear e Poisson apresentaram o mesmo  $\nu_{ótimo}$  ( $\nu_{EP} = \nu_{SE} = 1$ ), para problemas anisotrópicos. Os resultados obtidos para a equação de Laplace senoidal na seção 5.2.4. foram análogos.

#### E4. COMPARAÇÃO ENTRE OS ALGORITMOS

As Figs. E.5 e E.6 fazem uma comparação entre os algoritmos SE-EP padrão e SE-EP ótimo para a equação de Laplace linear. A Fig. E.5 apresenta a razão de aspecto  $Q = 1/16$  e a Fig. E7 a razão  $Q = 16$ . Verifica-se que o algoritmo SE-EP ótimo obteve um menor tempo de CPU em relação ao SE-EP ótimo.

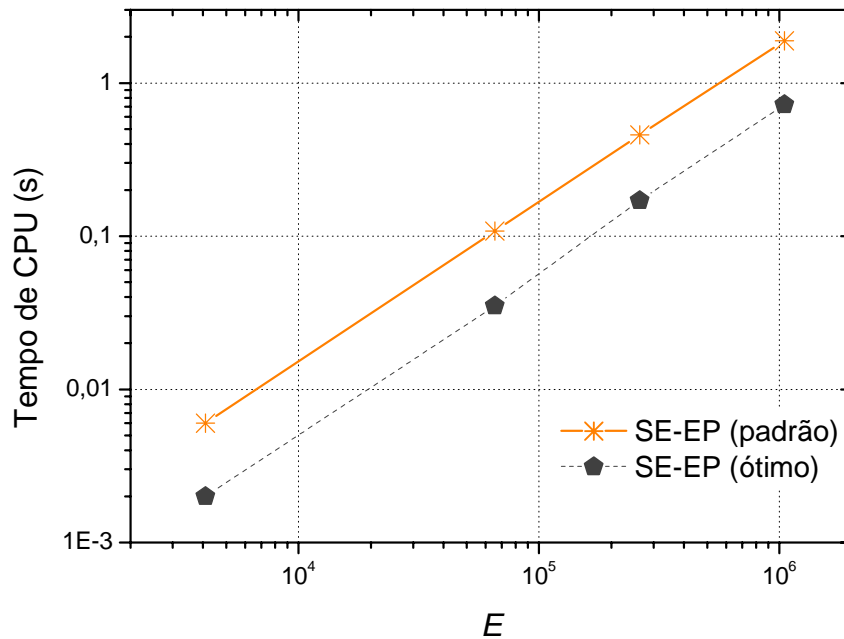


Figura E.5: Tempo de CPU *versus* número de elementos ( $E$ ) para o algoritmo SE-EP (padrão) e SE-EP (ótimo) para  $Q = 1/16$  e equação de Laplace linear.

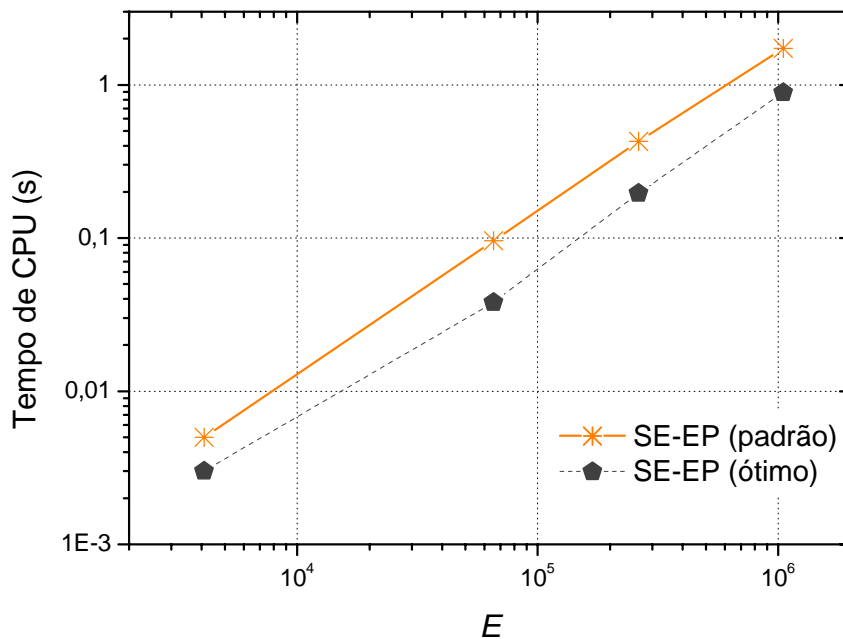
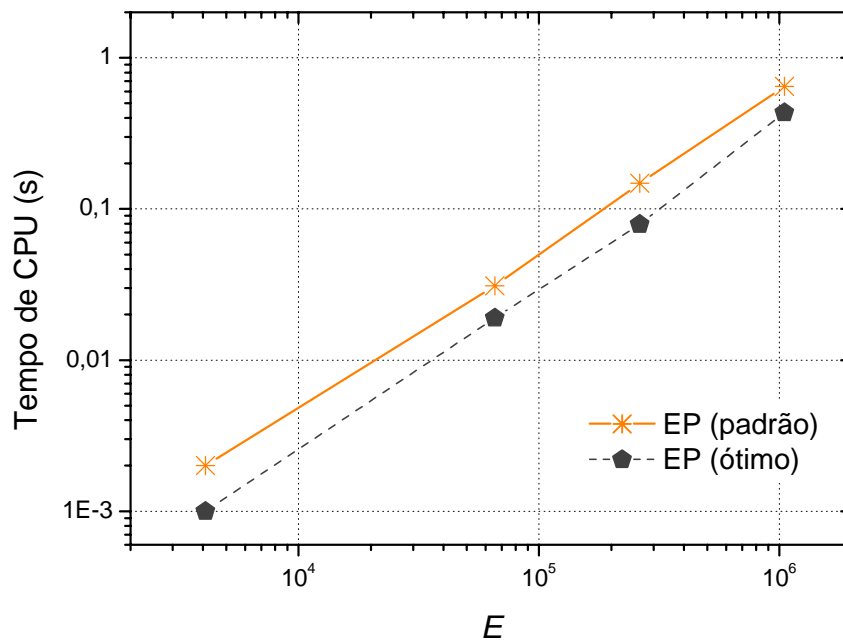


Figura E.6: Tempo de CPU *versus* número de elementos ( $E$ ) para o algoritmo SE-EP (padrão) e SE-EP (ótimo) para  $Q = 16$  e equação de Laplace linear.

A Fig. E.7 faz uma comparação entre os algoritmos EP e EP ótimo para a equação de Laplace linear e  $Q = 1$ . Verifica-se que o tempo de CPU do algoritmo EP ótimo é menor em relação ao EP.





**Figura E. 7: Tempo de CPU versus número de elementos ( $E$ ) para o algoritmo EP versus EP (ótimo) para  $Q = 1$  e equação de Laplace linear.**

Verificou-se que para todos os tamanhos de problemas e todas as razões de aspecto o tempo de CPU do SE-EP ótimo foi melhor em relação ao do SE-EP padrão. Por exemplo, para um problema com 1.048.576 elementos o algoritmo SE-EP ótimo obteve um tempo de CPU de 61,7%; 33,2% e 48,7% menor para as razões 1/16, 1 e 16 respectivamente em comparação com o SE-EP padrão. Ou seja, o algoritmo SE-EP ótimo é duas a três vezes mais rápido que o SE-EP padrão para a equação de Laplace linear.

As Figs. E.8 e E.9 fazem uma comparação entre os algoritmos SE-EP padrão e SE-EP ótimo para a equação de Poisson. A Fig. E.8 apresenta a razão de aspecto  $Q = 1/16$  e a Fig. E.9 a razão  $Q = 16$ . Verifica-se que o algoritmo SE-EP ótimo obteve um menor tempo de CPU em relação ao SE-EP padrão.

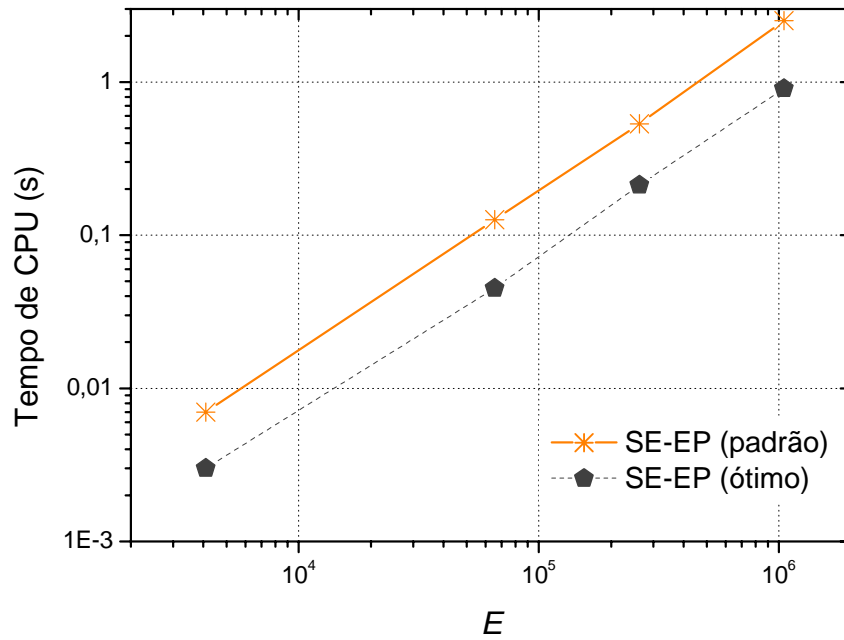


Figura E.8: Tempo de CPU *versus* número de elementos ( $E$ ) para o algoritmo SE-EP (padrão) e SE-EP (ótimo) para  $Q = 1/16$  e equação de Poisson.

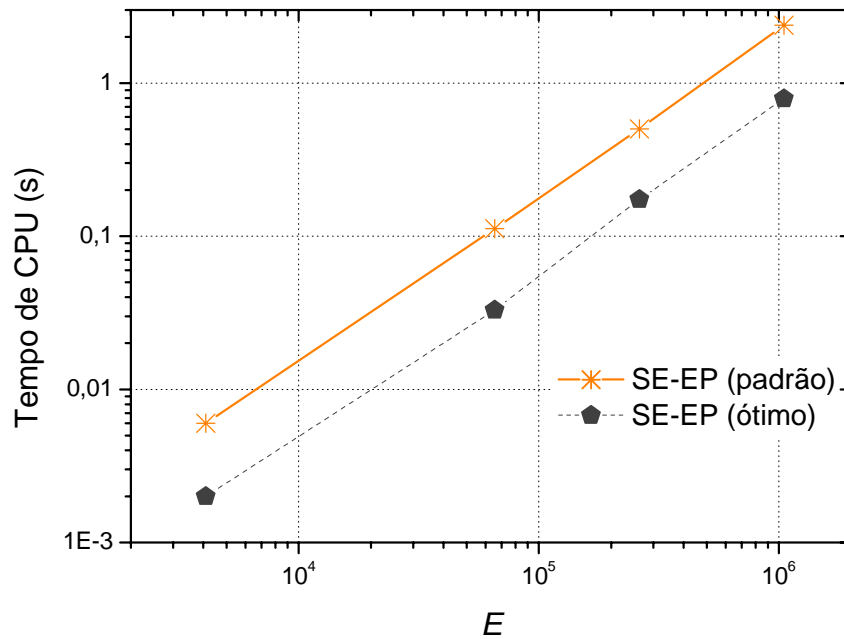
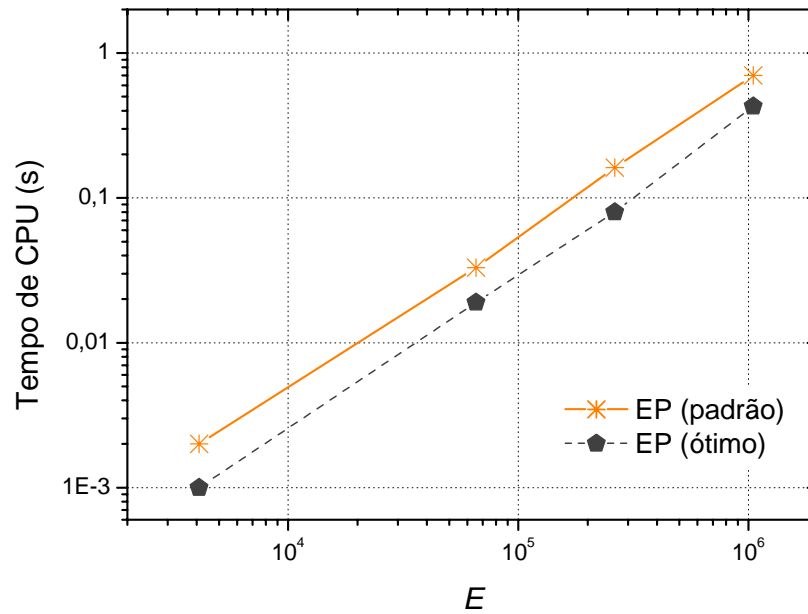


Figura E.9: Tempo de CPU *versus* número de elementos ( $E$ ) para o algoritmo SE-EP padrão *versus* SE-EP (ótimo) para  $Q = 16$  e equação de Poisson.

A Fig. E.10 faz uma comparação entre os algoritmos EP e EP ótimo para a equação de Laplace linear e  $Q = 1$ . Verifica-se que o tempo de CPU do algoritmo EP ótimo é menor em relação ao EP.



**Figura E.10: Tempo de CPU versus número de elementos ( $E$ ) para o algoritmo EP versus EP (ótimo) para  $Q = 1$  e equação de Poisson.**

Verificou-se que para todos os tamanhos de problemas e todas as razões de aspecto o tempo de CPU do SE-EP ótimo foi melhor em relação ao do SE-EP padrão. Para um problema com 1.048.576 elementos o algoritmo SE-EP ótimo obteve uma redução no tempo de CPU de 64%; 38% e 67% para as razões 1/16, 1 e 16, respectivamente, em comparação com o SE-EP padrão. Ou seja, o algoritmo SE-EP ótimo é cerca de duas a três vezes mais rápido que o SE-EP padrão. Os resultados obtidos para a equação de Laplace senoidal apresentados na seção 5.2.4 foram análogos.

## APÊNDICE F: GERAÇÃO DE MALHAS

Este apêndice descreve a geração de malhas para a anisotropia tipo III ( $h_x \neq h_y$ ,  $N_x \neq N_y$  e  $C_x = C_y$ ). A demonstração a seguir tem por objetivo encontrar uma forma de gerar malhas com mesmo número de elementos para diferentes razões de aspecto.

Algumas razões de aspecto utilizadas nesta tese foram: 1024, 4096 e 16384. Reescrevendo na base 2 ficam na forma:  $2^{10}$ ,  $2^{12}$  e  $2^{14}$ . Estas malhas, representadas na base 2, podem ser definidas por:

$$E = E_x \cdot E_y \quad (\text{F.1})$$

Consideram-se duas malhas:

$$E_1 = 2^{a_1} \cdot 2^{b_1} \quad (\text{F.2})$$

e

$$E_2 = 2^{a_2} \cdot 2^{b_2} \quad (\text{F.3})$$

onde  $a_1, b_1, a_2, b_2 \in \mathbb{Z}_*^+$ . A malha  $\Omega_1$  deve ter o mesmo número de elementos que a malha  $\Omega_2$ . Logo:

$$E_1 = E_2 \quad (\text{F.4})$$

ou seja:

$$2^{a_1} \cdot 2^{b_1} = 2^{a_2} \cdot 2^{b_2} \quad (\text{F.5})$$

ou ainda:

$$2^{a_1+b_1} = 2^{a_2+b_2} \quad (\text{F.6})$$

logo:

$$a_1 + b_1 = a_2 + b_2 \quad (\text{F.7})$$

Exemplo:

Considere  $C_x = C_y = 1$  e razão de aspecto  $Q = 2^2 = 4$ , tem-se que:

$$Q = \frac{h_x}{h_y} = \frac{\frac{1}{E_x}}{\frac{1}{E_y}} = \frac{1}{E_x} \cdot \frac{E_y}{1} \quad (\text{F.8})$$

$$Q = \frac{E_y}{E_x} = \frac{2^{b_1}}{2^{a_1}} = 2^{b_1 - a_1} \quad (\text{F.9})$$

como  $Q = 2^2$  então:

$$Q = 2^2 = 2^{b_1 - a_1} \quad (\text{F.10})$$

logo:

$$b_1 - a_1 = 2 \quad (\text{F.11})$$

Analogamente para  $Q = 2^4 = 16$  tem-se:

$$b_2 - a_2 = 4 \quad (\text{F.12})$$

Com as equações Eq. (F.11) e Eq. (F.12) obtém-se o sistema representado pela Eq. (F.13).

$$\begin{cases} b_1 - a_1 = 2 \\ b_2 - a_2 = 4 \end{cases} \quad (\text{F.13})$$

Substituindo-se as equações do sistema representado pela Eq. (F.13) em Eq. (F.7) e deixando todas as variáveis em função de  $a_1$  tem-se que:

$$a_2 = a_1 - 1, b_1 = 2 + a_1, b_2 = 4 + a_2 \quad (\text{F.14})$$

Atribuindo valores para  $a_1$  determinam-se  $a_2$ ,  $b_1$  e  $b_2$ . Substituindo estes valores na Eq. (F.14) obtém-se  $E_x$  e  $E_y$  para as razões 4 e 16. Para cada valor de  $a_1$  é determinado um valor para  $E_x$  e  $E_y$ . Este processo foi realizado para todas razões de aspecto utilizadas na tese. As malhas geradas por este processo encontram-se na Tab. E1.