

UNIVERSIDADE FEDERAL DO PARANÁ

GRAZIELLI VASSOLER RUTZ

EFEITOS DA ANISOTROPIA FÍSICA SOBRE O MÉTODO *MULTIGRID*
APLICADO À EQUAÇÃO DE DIFUSÃO

CURITIBA

2018

GRAZIELLI VASSOLER RUTZ

EFEITOS DA ANISOTROPIA FÍSICA SOBRE O MÉTODO *MULTIGRID*
APLICADO À EQUAÇÃO DE DIFUSÃO

Tese apresentada ao Curso de Pós-Graduação em Métodos Numéricos em Engenharia, Área de Concentração em Mecânica Computacional, Setores de Ciências Exatas e de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Doutora em Ciências.

Orientador: Prof. Dr. Marcio Augusto Villela Pinto.

CURITIBA

2018

R982E Rutz, Grazielli Vassoler

Efeitos da anisotropia física sobre o método multigrid aplicado à equação de difusão [recurso eletrônico] / Grazielli Vassoler Rutz , 2018.

Tese (doutorado) - Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Área de Concentração em Mecânica Computacional, Setores de Ciências Exatas e de Tecnologia, Universidade Federal do Paraná
Orientador: Prof. Dr. Marcio Augusto Villela

1. Anisotropia.. 2. Análise numérica. 3. Fourier, Analise de. I. Universidade Federal do Paraná. II. Villela, Marcio Augusto III.Título.

CDD 551.62

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em MÉTODOS NUMÉRICOS EM ENGENHARIA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **GRAZIELLI VASSOLER RUTZ** intitulada: **EFEITOS DA ANISOTROPIA FÍSICA SOBRE O MÉTODO MULTIGRID APLICADO À EQUAÇÃO DE DIFUSÃO**, após terem inquirido a aluna e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 03 de Julho de 2018.



MARCIO AUGUSTO VILLELA PINTO
Presidente da Banca Examinadora



LUCIANO KIYOSHI ARAKI
Avaliador Interno



MARCIO ANDRÉ MARTINS
Avaliador Externo



ELIAS ALFREDO GUDIÑO ROJAS
Avaliador Externo



ROBERTO DALLEDONE MACHADO
Avaliador Interno



FABIANE DE OLIVEIRA
Avaliador Externo

Ao meu pai Valter(*in memorian*);
À minha mãe Loreni (*in memorian*).

AGRADECIMENTOS

Aos meus queridos pais Valter(*in memorian*) e Loreni(*in memorian*), por terem me deixado a maior e melhor herança, que é a oportunidade e incentivo ao estudo.

Ao meu esposo Carlos,irmão Robsone toda a grande famíliapelo amor, carinho, atenção, apoio e incentivo.

Ao meu orientadorProf. Dr. Marcio Augusto Villela Pinto, por todos os ensinamentos,pela dedicação e paciência.

Aos amigos e colegas, especialmente ao Sebastião,ao Reverton e à Michely, pela amizade, apoio e troca de conhecimento.

Ao Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE) da Universidade Federal do Paraná (UFPR) pela oportunidade de cursar o doutorado.

Agradeço aos membros da banca examinadora, Prof. Dr. Marcio André Martins,Prof^ª. Dra. Fabiane de Oliveira,Prof. Dr. Elias Alfredo Gudiño Rojas, Prof. Dr. Roberto DalledoneMachado e Prof. Dr. Luciano Kiyoshi Arakipelo tempo dispensado à leitura deste trabalho e pelas excelentes contribuições.

Ao Instituto Federal de Santa Catarina (IFSC) pela licença concedida para a dedicação aos estudos.

E agradeço a Deus por me proporcionar a vivência com essas pessoas tãoespeciais.

RESUMO

Problemas anisotrópicos podem ser observados em diversos problemas de Engenharia, como por exemplo, no caso de um material com a condutividade térmica diferente de acordo com a direção. Nestes casos, os métodos disponíveis na literatura, não são totalmente eficientes. Neste contexto, este trabalho analisa diferentes algoritmos a partir do método *multigrid* e propõe métodos robustos e eficientes para a solução de problemas de anisotropia física. O modelo considerado é de difusão pura com anisotropia alinhada ao eixo de coordenada x . Para a discretização da equação é utilizado o Método das Diferenças Finitas com malhas uniformes e esquema numérico de segunda ordem. Os problemas são resolvidos com o método *multigrid* geométrico, esquema de correção e ciclo V com razão de engrossamento padrão. Realizou-se análise de Fourier local para auxiliar no projeto de um método *multigrid* eficiente. Calculou-se o fator de suavização e o fator de convergência assintótica variando-se diferentes componentes *multigrid*. Os operadores de restrição empregados foram: injeção, meia-ponderação, ponderação completa, ponderação parcial em x , ponderação parcial em y ; os operadores de prolongação estudados foram: bilinear e 7 pontos. Utilizaram-se os seguintes *solvers*: Gauss-Seidel lexicográfico, Gauss-Seidel *red-black*, Gauss-Seidel por linhas lexicográfico em x e y , Gauss-Seidel por linhas com ordenação zebra em x e y , decomposição LU incompleta de 5 e 7 pontos. Este estudo baseia-se na escolha de um operador de restrição mais eficiente e de um *solver* adequado à cada direção da anisotropia. A partir dos componentes ótimos obtidos via análise de Fourier, foram realizados experimentos para análise de complexidade e custo computacional do algoritmo proposto. A principal conclusão é de que a combinação de restrição por ponderação parcial com a decomposição ILU produz algoritmos robustos e eficientes para resolução de problemas que apresentam fortes anisotropias.

Palavras-Chave: Anisotropia física. Equação de difusão. Análise de Fourier local. *Solver* ILU.

ABSTRACT

In this work, we propose a robust and efficient multigrid method for the solution of a physical anisotropy problem. The model considered is the diffusion problem with anisotropy aligned to the axis of coordinate x . The Finite Difference Method for the discretization of the equation with uniform grids and second order numerical scheme is used. The problems are solved with the geometric multigrid method, correction scheme and V-cycle with standard coarsening. Local Fourier analysis was performed to help the design of an efficient multigrid method. The smoothing factor and asymptotic convergence factor were calculated by varying different multigrid components. The restriction operators employed were: injection, half-weighting, full weighting, partial weighting in x , partial weighting in y ; the prolongation studied were bilinear and 7points operators. The following solvers were used: lexicographic and red-black Gauss-Seidel, lexicographic lines Gauss-Seidel in x and y , zebra Gauss-Seidel in x and y , incomplete LU decomposition of 5 and 7 points. This study is based on the choice of a more efficient restriction operator and an incomplete LU decomposition suitable for each direction of the anisotropy. From the optimal components obtained through Fourier analysis, experiments were performed to analyze the complexity and computational cost of the proposed algorithm. The main conclusion is that the proposed methodology is robust and efficient for solving problems that present strong anisotropies.

Keywords:Physical anisotropy. Diffusion equation. Local Fourier Analysis. Solver ILU.

LISTA DE SÍMBOLOS

a_{ij}	Coeficientes da matriz A
A	Matriz dos coeficientes
b	Vetor de termos independentes
e	Erro numérico
E_h^θ	Espaço dos harmônicos
f	Vetor de termos independentes
h_x	Tamanho da malha na direção x
h_y	Tamanho da malha na direção y
I_h	Operador identidade
I_h^{2h}	Operador de restrição
I_{2h}^h	Operador de prolongação
K_h^{2h}	Operador de correção da malha grossa
\hat{K}_h^{2h}	Matriz 4x4 representante de K_h^{2h}
l_{ij}	Coeficientes da matriz L
l_1	Norma 1
l_∞	Norma infinito
L	Matriz triangular inferior
L	Número de níveis
L^0	Norma l_1 na estimativa inicial
L^m	Norma l_1 na iteração m
L_h	Operador Laplaciano discreto
\hat{L}_h	Notação estêncil para uma matriz triangular inferior
\tilde{L}_h	Autovalor de L_h
M_h^{2h}	Operador de duas malhas
\hat{M}_h^{2h}	Matriz 4x4 representante de M_h^{2h}

N	Número total de pontos
N_x	Número de pontos na direção x
N_y	Número de pontos na direção y
p	Ordem de complexidade
p_E	Ordem efetiva
p_L	Ordem assintótica
p_U	Ordem aparente
r	Resíduo
RA	Razão de aspecto
R	Matriz dos resíduos
S	Termo fonte
S_h	Operador de suavização
\hat{S}_h	Matriz 4x4 representante de S_h
\tilde{S}_h	Autovalor de S_h
t_{CPU}	Tempo de CPU
tol	Tolerância
T	Temperatura
T_a	Temperatura analítica
T^{low}	Região de baixas frequências
T^{high}	Região de altas frequências
T_m	Temperatura média analítica
T_{mn}	Temperatura média numérica
T_x	Derivada primeira de Tem relação a x
T_y	Derivada primeira de Tem relação a y
T_{xx}	Derivada segunda de Tem relação a x
T_{yy}	Derivada segunda de T em relação a y
u	Vetor de incógnitas

\bar{u}_0^h	Estimativa inicial na malha h
\bar{u}	Solução aproximada
u_{ij}	Coefficientes da matriz U
u_{xx}	Derivada segunda de u em relação a x
u_{yy}	Derivada segunda de u em relação a y
U	Matriz triangular superior
\hat{U}_h	Notação estêncil para uma matriz triangular superior
v_j	Modos de Fourier

Letras Gregas

ε	Coefficiente de anisotropia
σ	Parâmetro do MILU
ρ_{loc}	Fator de convergência assintóticacalculado via LFA
ρ_h	Fator de convergência calculado empiricamente
μ_{loc}	Fator de suavização calculado via LFA
Ω^h	Malha fina
Ω^{2h}	Malha grossa com engrossamento padrão
φ_h	Funções de malha
θ	Frequência
ν	Número de iterações internas do <i>solver</i>
ν_1	Número de pré-suavização
ν_2	Número de pós-suavização

Subíndices

E	ponto localizado à direita do ponto geral nodal
h	malha fina
i	posição do ponto na direção x
j	posição do nó na direção y

N	ponto localizado acima do ponto geral nodal
S	ponto localizado abaixo do ponto geral nodal
W	ponto localizado à esquerda do ponto geral nodal
2h	malha imediatamente mais grossa

Abreviaturas e Siglas

AMG	<i>Algebraic Multigrid Method</i>
CDS	<i>Central Differencing Scheme</i>
CFD	<i>Computational Fluid Dynamics</i>
CPU	<i>Central Processing Unit</i>
CS	<i>Correction Scheme</i>
DDS	<i>Downstream Differencing Scheme</i>
FAS	<i>Full Approximation Scheme</i>
FMG	<i>Full Multigrid</i>
FW	<i>Full Weighting</i>
HW	<i>Half Weighting</i>
ILU	Decomposição LU Incompleta
LFA	<i>Local Fourier Analysis</i>
MDF	Método das Diferenças Finitas
MEF	Método dos Elementos Finitos
MILU	Decomposição LU incompleta modificada
MSI	<i>Modified Strongly Implicit Method</i>
MVF	Método dos Volumes Finitos
PW	<i>Partial Weighting</i>
TDMA	<i>TriDiagonal Matrix Algorithm</i>
UDS	<i>Upwind Differencing Scheme</i>

LISTA DE FIGURAS

Figura 1: (a) Região contínua; (b) região discreta (malha).	32
Figura 2: Representação do ponto i sobre a malha unidimensional.	33
Figura 3: Pontos de uma malha bidimensional uniforme: (a) notação indicial; (b) notação geográfica.....	38
Figura 4: Ordenação lexicográfica.	40
Figura 5: Ordenação <i>red-black</i>	40
Figura 6: Ordenação por blocos: (a) linha na direção x ; (b) linha na direção y	42
Figura 7: Ordenação por blocos: (a) zebra na direção x ; (b) zebra na direção y	42
Figura 8: Estrutura da matriz A	43
Figura 9: Estrutura das matrizes L e U com a decomposição LU	43
Figura 10: Esquema $A = LU - R$ para $ILU(0)$	44
Figura 11: Malha bidimensional uniforme com 7 pontos: (a) notação geográfica; (b) notação indicial.....	48
Figura 12: Modos de Fourier.	56
Figura 13: Componentes de erro na malha fina (Ω^h) e grossa (Ω^{2h}).	57
Figura 14: Restrição por injeção.....	60
Figura 15: Operador de prolongação.	63
Figura 16: Ciclos V e W	64
Figura 17: (a) Malha fina original; (b) malha grossa com semiengrossamento na direção x	68
Figura 18: Regiões de baixas (região branca interior) e altas frequências (região hachurada).	74
Figura 19: Ordenações equivalentes do <i>solver</i> ILU	93
Figura 20: Solução analítica apresentada na Eq. (149).	95
Figura 21: Comportamento da norma l_∞ em função do número de ciclos V para uma malha 1025x1025.	97
Figura 22: Erro numérico para a norma l_∞	101
Figura 23: Ordem efetiva e aparente para a norma l_∞ do erro, erro na temperatura no ponto central e erro na temperatura média com o <i>solver</i> GS-Lex.	102

Figura 24: Erro numérico para a norma l_∞ para SG e MG.....	103
Figura 25: Erro numérico para a norma l_∞ e T(0,5;0,5).....	104
Figura 26: Ordem efetiva e aparente para a norma l_∞ do erro e temperatura no ponto central empregando o <i>solver</i> ILU.	105
Figura 27: μ_{loc} versus ε para <i>solvers</i> ponto com ordenação lexicográfica e <i>red-black</i>	107
Figura 28: μ_{loc} versus ε para x-linha-GS, y-linha-GS e alt-linha-GS.....	108
Figura 29: μ_{loc} versus ε para x-zebra-GS, y-zebra-GS e alt-zebra-GS.....	109
Figura 30: μ_{loc} versus ε para os <i>solvers</i> linha com ordenação lexicográfica e zebra com $\nu=2$	111
Figura 31: μ_{loc} para o <i>solver</i> ILU com diferentes ordenações.....	112
Figura 32: μ_{loc} versus ε para os <i>solvers</i> zebra-GS e ILU com $\nu=2$	115
Figura 33: ρ_{loc} versus ε com os <i>solvers</i> (a) GS-Lex; (b) GS-RB.	116
Figura 34: ρ_{loc} versus ε utilizando os <i>solvers</i> linha-GS com diferentes operadores de prolongação.	118
Figura 35: ρ_{loc} versus ε utilizando os <i>solvers</i> linha-GS com diferentes operadores de	119
Figura 36: ρ_{loc} versus ε para os <i>solvers</i> zebra-GS e diferentes operadores de prolongação.	120
Figura 37: ρ_{loc} versus ε utilizando os <i>solvers</i> zebra-GS e diferentes operadores de restrição.	121
Figura 38: ρ_{loc} versus coeficiente de anisotropia ε com os <i>solvers</i> tipo GS.....	123
Figura 39: ρ_{loc} versus coeficiente de anisotropia ε com o <i>solver</i> ILU.....	125
Figura 40: ρ_{loc} versus coeficiente de anisotropia ε comparando os melhores <i>solvers</i> tipo GS e os melhores <i>solvers</i> tipo ILU.....	127
Figura 41: ρ_{loc} versus ε com diferentes operadores de interpolação.	129
Figura 42: ρ_{loc} versus ε com diferentes operadores de restrição.	130
Figura 43: ρ_{loc} e ρ_h versus coeficiente de anisotropia ε	132
Figura 44: ρ_{loc} e ρ_h versus ε para malhas com diferentes números de pontos.	133

Figura 45: t_{CPU} <i>versus</i> ε para $L=9$ e ILU.	134
Figura 46: Tempo de CPU <i>versus</i> número de pontos N	135
Figura 47: Tempo de CPU <i>versus</i> número de pontos N	137
Figura 48: ρ_{loc} <i>versus</i> σ para diferentes valores de ε	139
Figura 49: ρ_h <i>versus</i> σ para diferentes valores de ε	140
Figura 50: ρ_{loc} e ρ_h <i>versus</i> ε para $\sigma=0,2$	141
Figura 51: t_{CPU} <i>versus</i> ε para $N=512 \times 512$ e MILU.....	143
Figura 52: t_{CPU} <i>versus</i> número de incógnitas N	144

LISTA DE TABELAS

Tabela 1: Métodos para solução de sistemas de equações.	36
Tabela 2: Autovalores dos operadores de restrição usados na LFA.	80
Tabela 3: Autovalores dos operadores de prolongação usados na LFA.	81
Tabela 4: Autovalores dos operadores de suavização usados na LFA.	89
Tabela 5: Autovalores dos operadores de suavização do tipo ILU usados na LFA.	92
Tabela 6: Fator de suavização utilizando GS-RB obtidos de Wienands e Joppich (2005), pg. 65, $N=128$, $\nu = 1$	107
Tabela 7: Fator de suavização utilizando x-zebra-GS e y-zebra-GS obtidos de Wienands e Joppich (2005), pg. 65, $N=128$, $\nu = 1$	110
Tabela 8: Fator de suavização utilizando alt-zebra-GS obtidos de Wienands e Joppich (2005), pg. 65, $N=128$, $\nu = 1$	110
Tabela 9: Fator de suavização utilizando ILU e MILU 5 pontos obtidos de Wesseling (1992), pg. 135.	112
Tabela 10: Fator de suavização utilizando ILU e MILU 7 pontos obtidos de Wesseling (1992), pg. 139.	113
Tabela 11: Fator de suavização utilizando ILU 7 pontos obtidos de Trottenberg (2001), pg. 261.	113
Tabela 12: Fator de convergência assintótica utilizando GS-RB obtidos de Trottenberg (2001), pg. 133. Restrição: FW, Prolongação: Bilinear, $N=128$, $\nu = 3$	117
Tabela 13: Fator de convergência assintótica utilizando x-zebra-GS e y-zebra-GS obtidos de Trottenberg (2001), pg. 137. Restrição: FW, Prolongação: Bilinear, $N=128$, $\nu = 2$	122
Tabela 14: Fator de convergência assintótica utilizando alt-zebra-GS obtidos de Trottenberg (2002), pg. 138. Restrição: FW, Prolongação: Bilinear, $N=128$, $\nu = 2$	122
Tabela 15: Fator de convergência assintótica utilizando ILU 7 pontos na direção EN obtidos de Trottenberg (2001), pg. 261. Restrição: FW, Prolongação: Bilinear, $N=128$, $\nu = 1$	125
Tabela 16: ρ_{loc} e ρ_h da Figura 43.	132

Tabela 17: Ordem de complexidade (p) para diferentes coeficientes de anisotropia (ε).	136
Tabela 18: Razão entre o número de flops de um ciclo V e N_{10} .	137
Tabela 19: Razão entre o número de flops da restrição e N_{10} .	138
Tabela 20: ρ_{loc} e ρ_h para $\sigma = 0,2$.	142
Tabela 21: Ordem de complexidade (p) para diferentes coeficientes de anisotropia (ε) e MILU.	145
Tabela 22: Razão entre o número de flops de um ciclo V e N_9 - MILU.	146
Tabela 23: Razão entre o número de flops da restrição e o número de pontos na malha mais fina N_9 - MILU.	146

LISTA DE ALGORITMOS

Algoritmo 1: Esquema de Correção (CS) para duas malhas	64
Algoritmo 2: LFA para o fator de convergência em duas malhas.....	83
Algoritmo 3: ILU_{EN-NE}	131
Algoritmo 4: $MILU_{EN-NE}$	140

SUMÁRIO

1. INTRODUÇÃO	21
1.1 Generalidades em CFD e motivação	21
1.2 Definição do problema	23
1.3 Objetivos.....	24
1.4 Delineamento do texto.....	25
2. REVISÃO BIBLIOGRÁFICA	26
2.1 Método <i>multigrid</i>	26
2.2 Problemas anisotrópicos	27
2.3 Análise de Fourier local (LFA)	29
3. FUNDAMENTAÇÃO TEÓRICA.....	31
3.1 MDF para discretização de equações	31
3.2 Métodos de resolução para sistemas de equações lineares.....	35
3.2.1 Método de Gauss-Seidel e suas variantes.....	37
3.2.2 Decomposição LU incompleta (ILU)	42
3.2.3 Decomposição LU incompleta modificada (MILU)	51
3.3 Notação estêncil.....	52
3.3.1 Notação estêncil para ILU	54
3.4 Método <i>multigrid</i>	55
3.4.1 Equação residual.....	57
3.4.2 Filosofia do <i>multigrid</i>	59
3.4.3 Componentes do <i>multigrid</i>	59
3.4.4 Ciclos e esquemas de aproximação	63
3.5 Problemas anisotrópicos	65
4. ANÁLISE DE FOURIER LOCAL	70
4.1 Conceitos gerais.....	70
4.2 Terminologia	71
4.3 Análise de suavização.....	74
4.4 Análise de duas malhas.....	76

4.5 Análise de Fourier local para os suavizadores.....	83
4.5.1 Análise de Fourier local para os <i>solvers</i> do tipo GS.....	83
4.5.2 Análise de Fourier local para os <i>solvers</i> do tipo ILU	89
5. MODELOS MATEMÁTICO E NUMÉRICO	94
5.1 Equação de difusão anisotrópica	94
5.2 Modelo numérico e detalhes computacionais.....	95
5.3 Definição da tolerância a ser utilizada.....	97
6. VERIFICAÇÃO NUMÉRICA DO CÓDIGO COMPUTACIONAL.....	99
6.1 <i>Solver</i> GS-Lex	100
6.2 <i>Solver</i> ILU	103
7. RESULTADOS PRELIMINARES	106
7.1 Fator de suavização (μ_{loc})	106
7.1.1 Fator de suavização para os <i>solvers</i> do tipo Gauss-Seidel	106
7.1.2 Fator de suavização dos <i>solvers</i> tipo ILU.....	111
7.2 Fator de convergência assintótica (ρ_{loc}).....	115
7.2.1 Fator de convergência assintótica utilizando os <i>solvers</i> do tipo Gauss-Seidel ...	116
7.2.2 Fator de convergência assintótica utilizando os <i>solvers</i> tipo ILU	124
8. RESULTADOS	128
8.1 Análise de convergência utilizando ILU	129
8.2 Análise de complexidade utilizando ILU	134
8.3 Análise de convergência utilizando MILU.....	138
8.4 Análise de complexidade utilizando MILU.....	142
9. CONCLUSÕES.....	148
9.1 Contribuições.....	150
9.2 Trabalhos futuros.....	150
REFERÊNCIAS	152

1. INTRODUÇÃO

Nesta tese, buscam-se métodos robustos e eficientes para a solução de sistemas de equações algébricas cujas matrizes de coeficientes são esparsas e de grande porte. Essas matrizes são obtidas da discretização de Equações Diferenciais Parciais (EDPs) com coeficientes anisotrópicos que modelam matematicamente muitos problemas encontrados em diversas áreas, de aplicações industriais, econômica, de biomedicina, entre outros.

Neste capítulo introduzem-se os conceitos gerais sobre a Dinâmica dos Fluidos Computacional (em inglês, *Computational Fluid Dynamics*, CFD), a motivação e os objetivos deste trabalho.

1.1 Generalidades em CFD e motivação

Um problema de Engenharia pode ser resolvido de três maneiras diferentes, são elas: experimental, analítica e numérica. Todas apresentam suas vantagens e desvantagens e a escolha do melhor método a ser utilizado depende do problema a ser resolvido e dos recursos disponíveis (MALISKA, 2004).

A partir de um fenômeno real, é necessário decidir se serão usados métodos experimentais ou métodos teóricos (analíticos e numéricos).

Os métodos experimentais trabalham com o fenômeno real, sendo cada problema resolvido com base em um experimento físico realizado em laboratório utilizando modelos em escala reduzida ou em campo, utilizando modelos em escala real. Esses métodos apresentam erros experimentais (MALISKA, 2004).

Os métodos teóricos são divididos em analíticos e numéricos. Os métodos analíticos resolvem uma classe muito restrita de problemas, pois se aplicam a problemas com equações, geometrias e condições de contorno e iniciais muito simples. Esses métodos apresentam erros de modelagem (MALISKA, 2004).

Nos métodos numéricos, as soluções numéricas são obtidas com o emprego de computadores. Aplicam-se a uma gama maior de problemas do que aqueles resolvidos

através de métodos analíticos. Apresentam erros de modelagem e erros numéricos. (MALISKA, 2004).

CFD e também a Transferência de Calor Computacional são áreas da computação científica que estuda métodos numéricos para simulação de problemas que envolvem fluidos em movimento com ou sem troca de calor. Neste trabalho, ambas as áreas serão denotadas com a sigla CFD. Sabe-se que esses métodos numéricos, muitas vezes, exigem um alto custo computacional. Esse fato ocorre porque, em geral, os problemas a serem resolvidos recaem na resolução de sistemas de equações algébricas cujas matrizes de coeficientes são esparsas e de grande porte (THEKALE et al., 2010).

A obtenção dos sistemas lineares é feita mediante a discretização do modelo matemático, que consiste em aproximar através de equações algébricas, cada termo do modelo matemático para cada ponto ou nó da malha.

A discretização do problema envolve a discretização do domínio de cálculo (gerando a malha) e a discretização da equação diferencial (gerando o sistema de equações). Esse processo conduz a um sistema de equações algébricas da forma

$$Au = f. \quad (1)$$

onde $A \in R^{N \times N}$ é a matriz dos coeficientes, $u \in R^N$ é a variável de interesse, $f \in R^N$ é o vetor independente e N é o número de incógnitas do sistema de equações.

Em CFD, os métodos tradicionalmente empregados nesse processo são: Métodos das Diferenças Finitas (MDF) (PLETCHER et al., 2013; GOLUB; ORTEGA, 1992; FERZIGER; PERIC, 2002; SAAD, 2003), Método dos Elementos Finitos (HUGHES, 2000; SAAD, 2003), Método dos Volumes Finitos (GOLUB; ORTEGA, 1992; MALISKA, 2004).

O sistema de equações algébricas dado pela Eq. (1) pode ser resolvido empregando-se métodos diretos ou iterativos. Neste trabalho foram utilizados os métodos iterativos devido às características dos sistemas lineares em CFD, mencionadas acima (matrizes dos coeficientes esparsas e de grande porte) para os quais os métodos iterativos são recomendados (BURDEN e FAIRES, 2009). O método *multigrid* é um dos métodos mais efetivos para acelerar a convergência de métodos iterativos na resolução

de sistemas lineares ou não lineares, problemas isotrópicos ou anisotrópicos, dentre outros (BRIGGS et al., 2000; TROTTENBERG, 2001; WESSELING et al., 2001).

Problemas anisotrópicos são muito comuns na Engenharia e surgem em muitos fenômenos, como no caso de um material com a condutividade térmica diferente de acordo com a direção. Nesse caso, os coeficientes das equações diferenciais são distintos entre si e geram o que é chamado de anisotropia física (ou anisotropia dos coeficientes).

Anisotropias também podem surgir a partir de discretizações de malhas com espaçamento diferente em cada direção, por exemplo, em problemas de camada limite, que neste caso conduz a uma anisotropia geométrica. Nos problemas com fortes anisotropias, tanto física quanto geométrica, a eficiência do método *multigrid* não tem sido totalmente alcançada (WESSELING e OOSTERLEE, 2001; LARSSON et al., 2005).

A Análise de Fourier local (em inglês, *Local Fourier Analysis*, LFA) é uma ferramenta muito utilizada para prever o desempenho do *multigrid*, pois permite calcular o fator de suavização de um método iterativo e o fator de convergência assintótica do *multigrid* variando seus componentes.

1.2 Definição do problema

CFD trata da resolução de problemas que envolvem fluidos em movimentos ou transferência de calor utilizando métodos numéricos. Sabe-se que esses métodos, muitas vezes, exigem um alto custo computacional. Esse fato ocorre, pois geralmente os problemas que se deseja resolver envolvem muitas variáveis e as matrizes associadas aos coeficientes são esparsas e de grande porte (BURDEN e FAIRES, 2009).

No presente trabalho propõe-se um método robusto e eficiente na resolução de problemas com anisotropia física utilizando LFA. O modelo matemático considerado é o de difusão bidimensional, em que a anisotropia física aparece nos coeficientes e será denominado por anisotropia difusiva. Para a discretização das equações foi utilizado o MDF com esquema central de segunda ordem (em inglês, *Central Differencing Scheme*, CDS).

Serão estudados diferentes *solvers* com o intuito de analisar seus desempenhos para problemas anisotrópicos, tais como: Gauss-Seidel por pontos com ordenação lexicográfica (GS-Lex); Gauss-Seidel por pontos com ordenação *red-black* (GS-RB); Gauss-Seidel por linhas com ordenação lexicográfica (*x*-linha-GS, *y*-linha-GS e *alt*-linha-GS); Gauss-Seidel por linhas com ordenação zebra (*x*-zebra-GS, *y*-zebra-GS e *alt*-zebra-GS) e decomposição LU incompleta de 7 pontos (aqui será designada ILU). (TROTTEBERG et al., 2001; BRIGGS et al., 2000; SAAD, 2003). Diferentes operadores de restrição/prolongação serão utilizados.

O problema-modelo considerado é o de difusão bidimensional, com coeficientes anisotrópicos e será apresentado no capítulo 5.

1.3 Objetivos

O objetivo geral deste trabalho é propor um método *multigrid* robusto e eficiente para resolver problemas difusão que apresentam anisotropia nos coeficientes (anisotropia física).

E os objetivos específicos são:

- Realizar um estudo do desempenho de diferentes métodos de resolução de sistemas de equações algébricas (*solvers*) para problemas anisotrópicos utilizando o *multigrid*;
- Analisar o desempenho de diferentes operadores de restrição e prolongação para problemas anisotrópicos;
- Propor uma metodologia para resolução de problemas com anisotropia física;
- Realizar Análise de Fourier Local (em inglês, *Local Fourier Analysis*, LFA) para prever e verificar a eficiência da metodologia proposta.
- Analisar a influência da anisotropia física sobre o custo computacional do método *multigrid*;

1.4 Delineamento do texto

Este texto apresenta nove capítulos além das referências bibliográficas. No segundo capítulo encontra-se a revisão bibliográfica do método *multigrid*, problemas anisotrópicos e análise de Fourier local (em inglês, *Local Fourier Analysis*, LFA). No terceiro capítulo apresenta-se a fundamentação teórica, como: Método das Diferenças Finitas (MDF), LFA, o método *multigrid* e anisotropia física. No capítulo quatro é detalhada a LFA. Os modelos matemático e numérico são apresentados no quinto capítulo. No sexto capítulo, apresenta-se a verificação numérica do código computacional empregado. Os resultados preliminares e os resultados principais serão apresentados nos capítulos 7 e 8, respectivamente. E, finalmente, no capítulo 9 serão apresentadas as conclusões, contribuições e sugestões para trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

Neste capítulo será descrita a revisão bibliográfica usada nos estudos do método *multigrid*, problemas anisotrópicos e LFA.

2.1 Método *multigrid*

Fedorenko (1964) foi um dos pioneiros no estudo do método *multigrid* investigando a convergência de problemas de valor de fronteira de segunda ordem utilizando a equação de Poisson. Bakhvalov (1966) realizou um estudo da convergência para as equações elípticas através da equação de advecção-difusão.

Somente na década de 70 o método *multigrid* teve seu potencial reconhecido através do estudo desenvolvido por Brandt (1977). Este trabalhou com diversos problemas de transferência de calor e escoamento de fluidos unidimensionais e bidimensionais, lineares e não lineares. Brandt é conhecido como o pai dos métodos *multigrid* e muitos consideram seu estudo como sendo a origem do *multigrid* moderno (CRAIG, 1996).

O método *multigrid* pode ser dividido em *multigrid* geométrico e *multigrid* algébrico (*Algebraic Multigrid Method*, AMG) (BRIGGS et al., 2000; WESSELING, 1992). O *multigrid* geométrico é utilizado quando malhas estruturadas são consideradas na discretização do problema. O AMG pode ser usado para vários tipos de problemas, em que a aplicação do *multigrid* geométrico é difícil ou impossível. Um exemplo que pode ser resolvido pelo AMG é quando a discretização na malha mais fina não permite engrossamento uniforme para todos os pontos (RUGE e STÜBEN, 1986).

O método *multigrid* com Esquema de Correção (em inglês, *Correction Scheme*, CS) é indicado para a resolução de problemas lineares. Para problemas não lineares é indicado o Esquema de Aproximação Completa (em inglês, *Full Approximation Scheme*, FAS) (BRIGGS et al., 2000).

No *multigrid* geométrico pode-se considerar diversas formas de percorrer a malha, chamados de ciclos. Os mais utilizados são o Ciclo V e o Ciclo W. Uma forma de acelerar a convergência do método *multigrid* é iniciá-lo na malha mais grossa a fim de

obter-se boas estimativas iniciais. Esse processo é denominado *multigrid* completo (em inglês, *Full Multigrid*, FMG) (BRIGGS et al., 2000; WESSELING, 1992; TROTTEMBERG et al., 2001).

Métodos *multigrids* são métodos eficientes para a resolução de sistemas lineares dado pela Eq. (1) associados com equações diferenciais parciais elípticas como a de Poisson (TROTTEMBERG et al., 2001). Ele foi desenvolvido inicialmente para equações elípticas, mas tem sido aplicado com grande sucesso a uma variedade de problemas como equações de Euler e Navier-Stokes discretizadas (GHIA et al., 1982).

O método consiste em usar um conjunto de malhas e executar, ora iterações em cada nível de malha, ora soluções aproximadas dessa equação em malhas mais grossas (BRIGGS et al., 2000; WESSELING, 1992; TROTTEMBERG et al., 2001).

2.2 Problemas anisotrópicos

Problemas de anisotropia física foram investigados por Hutchinson e Raithby (1986). Estes desenvolveram um método *multigrid* baseado no método de correção aditiva de Settari e Aziz (1973) e mostram como esse método pode ser usado para melhorar a razão de convergência em problemas bidimensionais e tridimensionais quando os coeficientes das equações algébricas se tornam fortemente anisotrópicos.

A ideia básica do método *multigrid* é que um processo de suavização é executado na malha mais grossa, até que as componentes de erro se tornem suaves. Para problemas isotrópicos, suavizadores básicos como Gauss-Seidel produzem erros suaves em todas as direções coordenadas, o mesmo não ocorre para problemas anisotrópicos (BRIGGS et al., 2000).

Rabi e Lemos (2001) discretizaram a equação de advecção-difusão bidimensional, com coeficientes constantes. Para a discretização das equações utilizaram o Método de Volumes Finitos. Empregaram o método *multigrid* com o esquema CS, *solver* Gauss-Seidel, utilizando ciclos V e W. Os autores apresentaram um estudo para diferentes campos de velocidades, número de malhas, número de interações internas em cada nível.

Zhang (2002) abordou problemas anisotrópicos utilizando semiengrossamento seguido de engrossamento padrão, para diferentes razões de aspecto e diferentes *solvers* e constatou que este algoritmo é eficiente para a anisotropia geométrica estudada.

Karaa e Zhang (2002) apresentaram um estudo da anisotropia física utilizando a equação de advecção-difusão bidimensional com coeficientes constantes e variáveis. Eles concluíram que o método *multigrid* utilizando um esquema compacto das Diferenças finitas de 4ª ordem degenera para os casos em que o número de Reynolds é alto (forte anisotropia).

Segundo Cordazzo (2006), para problemas anisotrópicos, métodos iterativos básicos, tais como, Gauss-Seil e Jacobi, suavizam os erros apenas em uma das direções coordenadas e também reduzem com eficiência apenas os erros cujos comprimentos de onda são equivalentes ao tamanho da malha, afetando a convergência do método.

Fischer e Huckle (2006) desenvolveram o método *multigrid* para sistemas de equações algébricas, em que a anisotropia não ocorre ao longo dos eixos coordenados, mas sim em direções arbitrárias. Fischer e Huckle (2008) utilizaram o método *multigrid* desenvolvido em Fischer e Huckle (2006) e testaram diferentes *solvers* para a resolução do sistema de equações algébricas. Utilizaram ainda algoritmos de engrossamento padrão e semiengrossamento. Concluíram que tais algoritmos são adequados para sistemas de equações lineares mais complexos (que envolvem anisotropia).

Pinto e Marchi (2006) estudaram problemas anisotrópicos (anisotropia geométrica) envolvendo a equação de Laplace bidimensional. Utilizaram diferentes tipos de algoritmos de engrossamento. Os autores compararam essas técnicas para diferentes razões de aspecto e constataram que o algoritmo “semiengrossamento seguido de engrossamento padrão” com suavizador Gauss-Seidel lexicográfico é eficiente para as anisotropias estudadas.

Suero et al. (2008) apresentaram um estudo do efeito da anisotropia física sobre o método *multigrid* na solução da equação de advecção-difusão bidimensional com coeficientes anisotrópicos nos termos advectivos. Utilizaram o método *multigrid* geométrico com esquema de correção FAS, ciclo V e *solver* implícito MSI (em inglês, *Modified Strongly Implicit*, MSI) para resolução dos sistemas de equações algébricas. Foram considerados diferentes campos de velocidades (nulo, constantes e variáveis) e

estudaram a influência do número de iterações internas do *solver*, número de malhas e número de incógnitas. A principal conclusão é de que a anisotropia física, neste tipo de problema, afeta pouco o desempenho do *multigrid*, pois quanto maior o número de nós da malha, mais próximo do isotrópico se tornamos problemas anisotrópicos.

Oliveira et al.(2012) analisaram a anisotropia geométrica paravários tipos de engrossamento e razões de aspecto. Também analisaram alguns parâmetros do método *multigrid* tais como: *solvers*, tipos de restrição, número de níveis e número de iterações internas. Concluíram que o engrossamento por ponderação parcial (em inglês, *Partial Weighting*, PW) teve um bom desempenho.

Vinogradova e Krukier (2013)resolveram um problema de advecção-difusão tridimensional com coeficientes de anisotropia intermediários, utilizando MDF para a discretização das equações e ILU para a resolução dos sistemas de equações lineares.Concluíram que a metodologia proposta é eficaz, mas tem a desvantagem de apresentar limitações nos coeficientes de derivadas mistas, o que não tem significado físico.

Dedner et al. (2014) apresentaram a eficiência do método *multigrid* geométrico preconditionado para problemas anisotrópicos em modelos de Geofísica. Foram obtidos bons resultados para resolver a correção da pressão, encontrado na equação diferencial.

2.3Análise de Fourier local (LFA)

Diversos trabalhos utilizando o método *multigrids*são encontrados na literatura e demonstram que a escolha dos componentes *multigrid* têm um papel fundamental na convergência ou não deste método. Trottenberg et al.(2001) afirmam que tais escolhas podem ser difíceis e que pequenas alterações podem melhorar sensivelmente a convergência.

Neste sentido, aLFA pode auxiliar nestas escolhas uma vez que permite prever o desempenho do método *multigrid*, pois fornece estimativas de taxas de convergência mediante a variação de seus componentes.

Wienands e Joppich (2005) apresentaram um estudo bem aprofundado sobre a LFA e suas aplicações em diversos problemas, entre eles, os problemas anisotrópicos.

Eles calcularam o fator de convergência do método *multigrid* para a equação de difusão anisotrópica, variando *solvers* e operadores de restrição e prolongação.

Johannsen (2005) resolveu um problema de difusão anisotrópica, utilizando o Método dos Elementos Finitos para a discretização das equações e o método de decomposição LU incompleta (ILU) de 9 pontos para a resolução dos sistema de equações lineares. Empregou LFA para mostrar as excelentes propriedades de suavização do ILU.

Pinto et al. (2016) resolveram o problema de difusão anisotrópica utilizando ILU em malhas triangulares. Utilizaram LFA para mostrar as boas propriedades de suavização do *solver* e convergência assintótica do *multigrid*.

Franco et al.(2018) realizaram LFA em problemas transientes e obtiveram o valor crítico do parâmetro que representa o grau de anisotropia espaço/tempo para a equação de Fourier 1D e 2D.

Neste capítulo foi apresentada a revisão bibliográfica sobre o método *multigrid*, problemas anisotrópicos e LFA com as principais contribuições de diversos autores sobre estes tópicos, os quais são os mais importantes desta pesquisa. Em seguida será apresentada a fundamentação teórica.

3.FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será descrito o referencial teórico usado nesta tese: discretização de equações diferenciais via Método das Diferenças Finitas (MDF), métodos de resolução de sistemas lineares, método *multigrid*, LFA e problemas anisotrópicos.

3.1MDF para discretização de equações

Os problemas em CFD podem ser modelados matematicamente por meio de equações diferenciais. Para a resolução numérica desses problemas é necessário, inicialmente, expressar de forma adequada o domínio de cálculo, ou seja, a solução numérica dessas equações requer a discretização do domínio de cálculo, de modo a gerar uma malha na qual esta solução será obtida.

Os termos da equação diferencial são aproximados, substituindo-se as derivadas existentes por expressões algébricas, resultando em um sistema de equações algébricas que envolvem a função incógnita (MARCHI, 2001). Os métodos mais empregados na discretização de equações diferenciais são Método das Diferenças Finitas (MDF) (FERZIGER e PERIC, 2002), Método de Elementos Finitos (MEF) (HUGHES, 2000; REDDYE e GARTLING, 1994) e Métodos de Volumes Finitos (MVF) (MALISKA, 2004).

O Método das Diferenças Finitas é o precursor dos métodos numéricos para simulação de problemas de Engenharia e tem como princípio fundamental aproximar, através de expressões algébricas, cada termo do modelo matemático para cada ponto ou nó da malha (FERZIGER e PERIC, 2002).

Na Figura 1 pode-se observar a representação de uma região contínua de um domínio e à direita o conjunto de pontos discretos (malha), utilizados para a obtenção da solução numérica. Considerando os nós da malha, localizados na interseção das linhas horizontais com as verticais, considera-se que esses estão separados entre si por uma distância h_x e h_y , respectivamente, não necessariamente iguais. Os índices i e j identificam um ponto na i -ésima coluna e j -ésima linha, respectivamente.

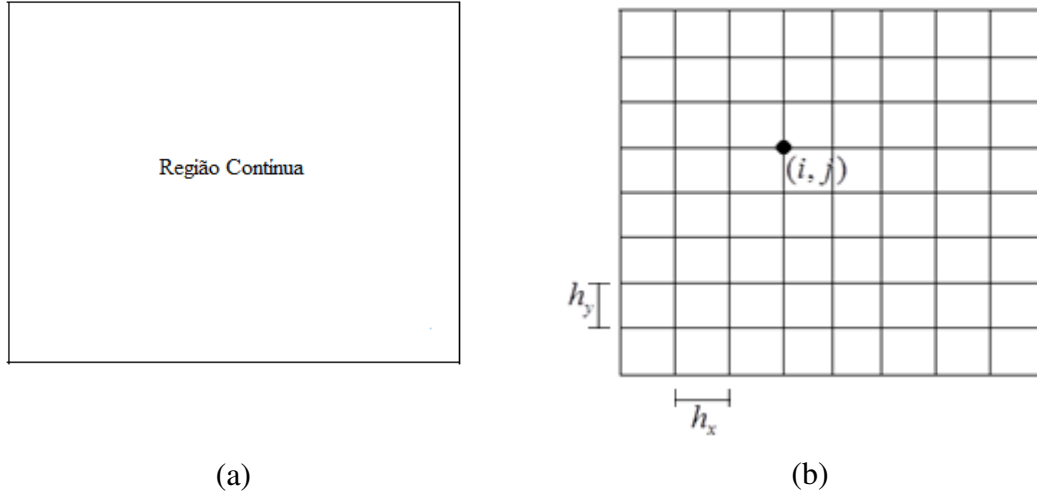


Figura 1: (a) Região contínua; (b) região discreta (malha).

Para o caso unidimensional pode-se obter a derivada de uma função contínua $f(x)$ como

$$\frac{df}{dx} = \lim_{h_x \rightarrow 0} \frac{f(x + h_x) - f(x)}{h_x}. \quad (2)$$

As aproximações das Diferenças Finitas podem ser obtidas de várias formas. A mais comum é a expansão por série de Taylor, a qual permite estimar o valor da função $f(x_1)$ conhecendo o valor de $f(x_0)$. Seja $f(x)$ uma função de interesse, contínua no intervalo $[a, b]$ e que possua derivadas de ordem n também contínuas nesse intervalo. O teorema de Taylor permite escrever, para todo $x \in [a, b]$,

$$f(x) = f(x_0) + h_x \left. \frac{df}{dx} \right|_{x_0} + \frac{(h_x)^2}{2!} \left. \frac{d^2 f}{dx^2} \right|_{x_0} + \frac{(h_x)^3}{3!} \left. \frac{d^3 f}{dx^3} \right|_{x_0} + \dots + R_n, \quad (3)$$

onde $h_x = x - x_0$ denota o espaçamento e R_n é o resto, definido como

$$R_n = \frac{(h_x)^{n+1}}{(n+1)!} \cdot \frac{d^{n+1} f}{dx^{n+1}} \cdot f(\xi), \quad \xi \in [a, b]. \quad (4)$$

A Figura 2 mostra alguns pontos de uma malha unidimensional. Neste caso os pontos são uniformemente espaçados, com $h_x = x_{i+1} - x_i$ e $x_i = a + ih_x, i = 0, 1, \dots, n + 1$.

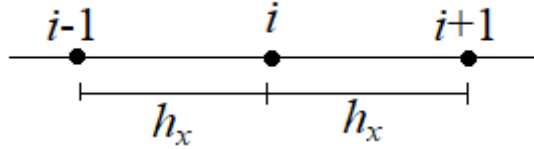


Figura 2: Representação do ponto i sobre a malha unidimensional.

A primeira derivada de uma função $f(x)$ no ponto $x_i = ih_x$ pode ser obtida expandindo $f(x_i + h_x)$ em série de Taylor em torno do ponto x_i , ou seja,

$$f(x_i + h_x) = f(x_i) + h_x \left. \frac{df}{dx} \right|_{x_i} + \frac{(h_x)^2}{2!} \left. \frac{d^2 f}{dx^2} \right|_{x_i} + \frac{(h_x)^3}{3!} \left. \frac{d^3 f}{dx^3} \right|_{x_i} + \dots \quad (5)$$

As expressões para a primeira derivada podem ser obtidas por meio da Eq. (5) e define o tipo de esquema e a ordem do erro de truncamento cometido (ordem dos erros a serem descartados no processo de discretização). A seguir serão apresentados alguns dos esquemas mais comuns e que serão usados no decorrer deste texto. Para outras aproximações, veja por exemplo Burden e Faires (2016) e Ferziger e Peric (2002). Isolando-se a primeira derivada da Eq. (5), obtém-se

$$\left. \frac{df}{dx} \right|_{x_i} = \frac{f(x_i + h_x) - f(x_i)}{h_x} - \frac{(h_x)}{2!} \left. \frac{d^2 f}{dx^2} \right|_{x_i} + \frac{(h_x)^2}{3!} \left. \frac{d^3 f}{dx^3} \right|_{x_i} - \dots \quad (6)$$

Pode-se simplificar a notação para a Eq. (6) considerando a variável dependente $u = u(x)$ e independente x , desprezando-se os termos de ordem superior, escrevendo

$$\left(\frac{du}{dx} \right)_i \approx (u'_{DDS})_i = \frac{u_{i+1} - u_i}{h_x}. \quad (7)$$

A aproximação apresentada na Eq. (7) é um método de 1ª ordem, pois pela Eq. (6) pode-se notar que o primeiro termo descartado tem a potência 1 em h_x . Tal aproximação é denominada diferença adiantada (em inglês, *Downstream Differencing Scheme*, DDS).

A Eq. (8) apresenta a aproximação do método de 1ª ordem, denominada diferença atrasada (em inglês, *Upwind Differencing Scheme*, UDS).

$$\left(\frac{du}{dx}\right)_i \approx (u'_{UDS})_i = \frac{u_i - u_{i-1}}{h_x}. \quad (8)$$

A Eq. (9) apresenta a aproximação de um método de 2ª ordem, denominada diferença central de dois pontos (em inglês, *Central Differencing Scheme*, CDS).

$$\left(\frac{du}{dx}\right)_i \approx (u'_{CDS})_i = \frac{u_{i+1} - u_{i-1}}{2h_x}. \quad (9)$$

Utilizando-se manipulações algébricas, é possível obter aproximações para a derivada segunda,

$$\left(\frac{d^2u}{dx^2}\right)_i \approx (u''_{CDS})_i = \frac{u_{i-1} - 2u_i + u_{i+1}}{(h_x)^2}. \quad (10)$$

A aproximação apresentada na Eq. (10) é um método de 2ª ordem no qual é utilizada a diferença central de 3 pontos para a aproximação da derivada e denominada CDS.

Para o caso bidimensional, a aproximação para a derivada de segunda ordem em relação a x é dada por

$$\left(\frac{\partial^2 u}{\partial x^2}\right) \approx \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(h_x)^2}. \quad (11)$$

A aproximação para a derivada de segunda ordem em relação a y é dada por:

$$\left(\frac{\partial^2 u}{\partial y^2}\right) \approx \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{(h_y)^2}, \quad (12)$$

onde h_y é o espaçamento entre os pontos na direção de y .

Considera-se, então, a equação de Poisson dada por

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), \quad (13)$$

onde u é a variável de interesse dependente de x e y , e f é o termo fonte.

Uma aproximação de segunda ordem para a Eq. (13) pode ser obtida pela substituição dos esquemas CDS, dados pelas Eqs.(11) e (12) considerando $h = h_x = h_y$.

Com isso tem-se

$$\left(\frac{1}{h^2}\right) \cdot (u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1}) = f_{i,j}. \quad (14)$$

Uma coleção de equações como a Eq.(14) gera um sistema de equações lineares que pode ser resolvido utilizando-se métodos diretos ou iterativos, que serão descritos na próxima seção.

3.2 Métodos de resolução para sistemas de equações lineares

A discretização de um modelo matemático em CFD geralmente conduz a um sistema de equações algébricas dado pela Eq.(1). Vários são os métodos que podem ser

usados para a resolução desses sistemas e podem ser classificados em diretos e iterativos. Os métodos diretos fornecem a solução exata do problema (a menos de erros de arredondamento); já os métodos iterativos determinam uma solução aproximada que atende a certo critério de parada previamente estabelecido, que pode ser baseado no erro, no resíduo ou no número de iterações.

Na Tabela 1, apresentam-se as principais características de cada um desses métodos (diretos e iterativos), supondo que o sistema a ser resolvido tenha solução.

Tabela 1: Métodos para solução de sistemas de equações.

MÉTODOS DIRETOS	MÉTODOS ITERATIVOS
<ul style="list-style-type: none"> - São mais indicados para resolução de sistemas de pequeno porte; - Provocam o preenchimento de uma matriz esparsa; - São sensíveis a erros de arredondamento; - Elevado tempo de processamento e esforço computacional de ordem $O(N^3)$. 	<ul style="list-style-type: none"> - São adequados para resolver sistemas de grande porte; - Não provocam o preenchimento de matrizes esparsas; - São sensíveis a erros de iteração com esforço computacional de ordem $O(N^2/2)$.

Pode-se citar como exemplos de métodos diretos: eliminação de Gauss, decomposição LU, fatoração de Cholesky, TDMA (em inglês, *Tridiagonal Matrix Algorithm*), entre outros. Este último apresenta esforço computacional de ordem $O(N)$.

Pode-se citar como exemplos de métodos iterativos: Jacobi, Gauss-Seidel, decomposição LU incompleta (ILU), MSI, Gradiente Conjugado, entre outros.

A estrutura da matriz A , obtida da discretização de equações que modelam fenômenos em CFD com MDF, são matrizes de banda pentadiagonais no caso de modelos matemáticos bidimensionais. Então, devido a essas características apresentadas pela matriz A , os métodos iterativos são mais adequados para obtenção da solução (TENNEHILL et al., 1997; BURDEN e FAIRES, 2009).

Nos últimos anos, vários trabalhos têm se dedicado à busca de um método iterativo eficiente e preciso para a resolução de sistemas de equações algébricas.

Na sequência são descritos alguns métodos iterativos para a resolução de sistemas de equações lineares. Na Eq.(1), pode-se escrever a matriz A da seguinte forma

$$A = M - N, \quad (15)$$

onde M é uma matriz não-singular e N é uma matriz associada ao método. Assim, uma solução para a Eq.(1), pode ser obtida por

$$M u^{m+1} = N u^m + f, \quad (16)$$

onde m e $m+1$ são os passos de iteração.

Considerando $S = M^{-1}N$ e $K = M^{-1}f$, pode-se reescrever a Eq.(16) como

$$u^{m+1} = S u^m + K. \quad (17)$$

Vários são os métodos iterativos que podem ser escritos dessa forma. Eles são chamados de métodos iterativos básicos, pois tomam a matriz A escrita na forma da Eq.(15). Dentre esses métodos pode-se citar: Gauss-Seidel, Jacobi, ILU e SOR (em inglês, *Successive Over Relaxation*, SOR).

Nas próximas subseções serão tratados alguns métodos iterativos de interesse nesta tese.

3.2.1 Método de Gauss-Seidel e suas variantes

O método de Gauss-Seidel (BURDEN e FAIRES, 2009) é um método iterativo usado para resolver um sistema de equações dado pela Eq.(1). A discretização de uma equação diferencial conduz à resolução de uma equação algébrica para cada ponto de uma malha discretizada.

A Figura 3(a) mostra o recorte de uma malha bidimensional em que são utilizados 5 pontos para resolver cada equação do sistema de equações algébricas.

Aqui se usa a notação indicial (i, j) . A Figura 3(b) apresenta uma outra notação para os pontos da malha da Figura 3(a), “notação geográfica”. Os pontos P (central), W (oeste), E (leste), N (norte), S (sul) da Figura 3(b) correspondem aos pontos (i, j) , $(i-1, j)$, $(i+1, j)$, $(i, j+1)$, $(i, j-1)$ na Figura 3(a), respectivamente. Ao longo desta tese, cada uma dessas notações será utilizada onde for mais adequado.

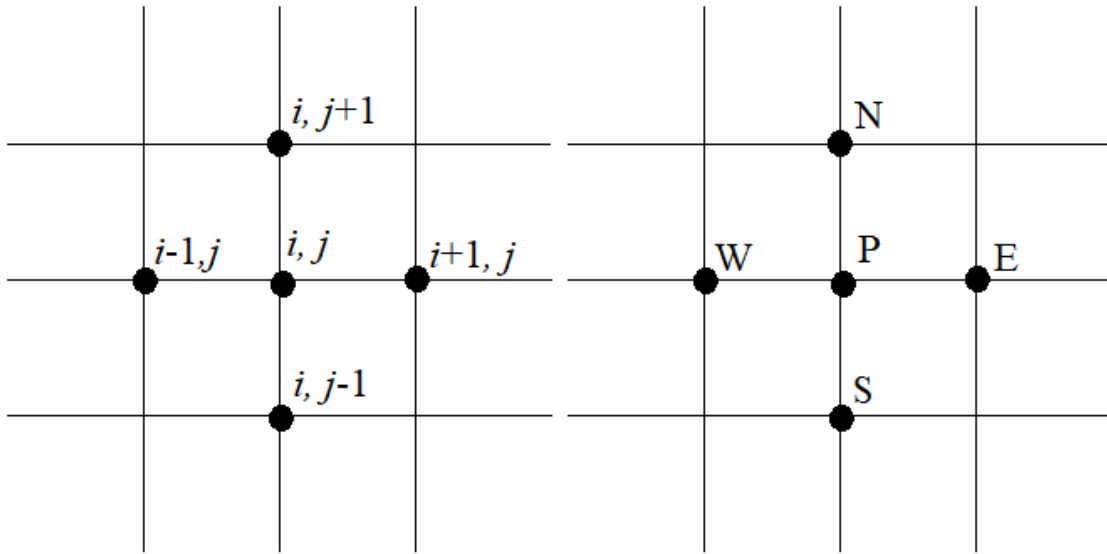


Figura 3: Pontos de uma malha bidimensional uniforme: (a) notação indicial; (b) notação geográfica

É denotado método de Gauss-Seidel por pontos se a matriz M da Eq.(15) é obtida substituindo $a_{ij} = 0$ para $j > i$. Para um esquema de 5 pontos, um método para a equação Eq. (1) para o ponto (i, j) tem a forma

$$a_{i,j}^p u_{i,j} = a_{i,j}^s u_{i,j-1} + a_{i,j}^w u_{i-1,j} + a_{i,j}^e u_{i+1,j} + a_{i,j}^n u_{i,j+1} + f_{i,j}. \quad (18)$$

Uma coleção de equações na forma da Eq.(18) escrita para cada par (i, j) do domínio produz um sistema de equações, o qual pode ser resolvido iterativamente pelo método de Gauss-Seidel. Note que a matriz A é pentadiagonal.

Esse método resolve cada uma das equações do sistema e usa em um mesmo ciclo os valores das variáveis já calculadas no ciclo.

Pode-se reescrever a Eq. (18) na forma

$$a_p u_p^{m+1} = a_n u_n^m + a_s u_s^{m+1} + a_e u_e^m + a_w u_w^{m+1} + f_p, \quad (19)$$

onde o superíndice m representa a m -ésima iteração e o subíndice, a posição do ponto na malha computacional. Pode-se observar que na iteração $(m+1)$, utilizando-se a ordem lexicográfica, são conhecidas as variáveis u_w e u_s . Em seguida será detalhada a questão da ordenação.

A eficiência dos métodos iterativos (*solver*) está fortemente ligada à ordem em que as incógnitas das equações de um sistema são determinadas. As equações e incógnitas estão associadas com os pontos em uma malha computacional, portanto, a ordenação do *solver* está associada à ordenação dos pontos em uma malha computacional.

Na ordenação lexicográfica, os pontos da malha são numerados da seguinte forma

$$k = i + (j-1)N_x, \quad (20)$$

onde N_x é o número de pontos da malha na direção x .

A

Figura 4 apresenta a numeração dos pontos da malha, utilizando a ordenação lexicográfica.

16	17	18	19	20
11	12	13	14	15
6	7	8	9	10
1	2	3	4	5

Figura 4: Ordenação lexicográfica.

Como já foi mencionado, para a notação apresentada na Eq. (19) com o método de Gauss-Seidel, utilizou-se a ordenação lexicográfica, cuja notação será GS-Lex.

É possível utilizar também a ordenação *red-black*, cuja numeração dos pontos da malha é apresentada na Figura 5. Neste caso, uma aplicação de Gauss-Seidel com essa ordenação terá a notação GS-RB.

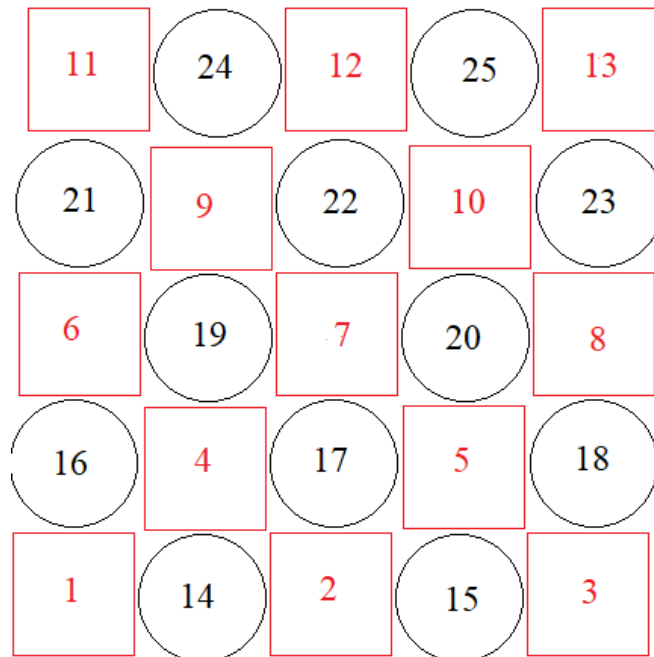


Figura 5: Ordenação *red-black*.

Denota-se o método de Gauss-Seidel por blocos se $a_{ij} = 0$ para $j > i + 1$. Neste, as incógnitas correspondentes às linhas da malha são atualizadas simultaneamente, ou seja, todos os pontos de uma mesma linha são utilizados em uma iteração.

A

5	5	5	5	5
4	4	4	4	4
3	3	3	3	3
2	2	2	2	2
1	1	1	1	1

Figura 6(a) apresenta a ordenação dos pontos da malha por linhas na direção x . Ao se utilizar o método de Gauss-Seidel com essa ordenação, tem-se o método de Gauss-Seidel linha em x com notação x -linha-

GS.A

5	5	5	5	5
4	4	4	4	4
3	3	3	3	3
2	2	2	2	2
1	1	1	1	1

Figura 6(b) mostra a ordenação dos pontos da malha por linhas na direção y . Ao se utilizar o método de Gauss-Seidel com essa ordenação, tem-se o método de Gauss-Seidel linha em y com notação y -linha-GS.

Quando se realiza uma iteração na direção x e em seguida realiza-se uma iteração na direção y , este é o conhecido método de Gauss-Seidel linha alternado, aqui denotado alt-linha-GS.

5	5	5	5	5	5	4	3	2	1
4	4	4	4	4	5	4	3	2	1
3	3	3	3	3	5	4	3	2	1
2	2	2	2	2	5	4	3	2	1
1	1	1	1	1	5	4	3	2	1

Figura 6: Ordenação por blocos: (a) linha na direção x ; (b) linha na direção y .

A Figura 7(a) apresenta a ordenação zebra na direção x . Quando se realiza uma iteração de Gauss-Seidel com essa ordenação, tem-se o método de Gauss-Seidel zebra na direção x , com notação x -zebra-GS. A Figura 7(b) apresenta a ordenação zebra na direção y gerando o método de Gauss-Seidel zebra na direção y , com notação y -zebra-GS.

Quando se realiza uma iteração de Gauss-Seidel zebra na direção x e em seguida realiza-se uma iteração na direção y , este é o conhecido método de Gauss-Seidel zebra alternado, aqui denotado alt-zebra-GS.

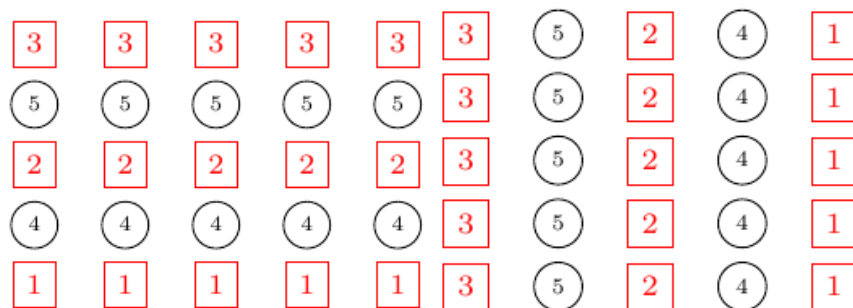


Figura 7: Ordenação por blocos: (a) zebra na direção x ; (b) zebra na direção y .

3.2.2 Decomposição LU incompleta (ILU)

Ao se aplicar a decomposição LU convencional a matrizes esparsas tem-se duas matrizes triangulares L e U , também esparsas. Porém, alguns dos elementos que anteriormente estavam zerados em A , agora se tornam diferentes de zero em L e U . Como por exemplo, pode ser observado nas Figs. Figura 8 e Figura 9, uma decomposição LU em uma matriz A pentadiagonal.

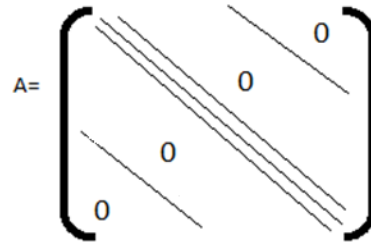


Figura 8: Estrutura da matriz A.

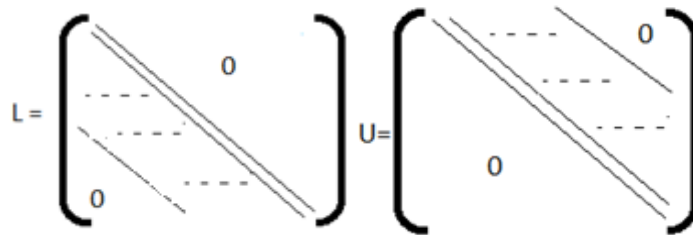


Figura 9: Estrutura das matrizes L e U com a decomposição LU.

Note que houve um preenchimento em algumas posições das matrizes L e U, fazendo com que elas deixem de ter a mesma esparsidade da matriz A.

Como alternativa para minimizar esse dano, e assim poder desfrutar de matrizes L e U com boa esparsidade, tem-se a técnica de decomposição LU incompleta (ILU) (WESSELING, 1992; TROTTERBERG et al., 2001; SAAD, 2003). Esta técnica trata-se de uma adaptação da decomposição LU, tornando-se este método, que era direto, em um método iterativo.

A decomposição LU incompleta (ILU) é obtida se a matriz M da Eq. (15) é substituída pelo produto

$$M = LU. \quad (21)$$

Uma decomposição alternativa para M ainda pode ser

$$M = LD^{-1}U. \quad (22)$$

onde D é uma matriz diagonal. Assim, as Eqs. (21) e (22) são equivalentes. Pode-se descrever a decomposição ILU considerando D uma matriz diagonal (SAAD, 2003), ou ainda, inserir a diagonal de D na matriz U .

Portanto, ao reescrever a Eq. (15) utilizando-se anotação $A = LU - R$, onde N será denominada R , ou seja, a matriz dos resíduos.

A decomposição ILU 5 pontos de uma matriz pentadiagonal, que o caso específico aqui abordado, torna as matrizes L e U como sendo esparsas também, mas agora, com os elementos correspondentes que estavam zerados em A , ou seja, as matrizes L e U mantêm a mesma esparsidade de A . Este caso é denominado de ILU(0) por não permitir nenhum preenchimento nas matrizes L e U . Entretanto é introduzida também a matriz dos resíduos R conforme esquema apresentado na Figura 10.

$$\begin{bmatrix} & & & & \\ & & & & 0 \\ & & & & \\ & & & & \\ 0 & & & & \end{bmatrix} = \begin{bmatrix} & & & & \\ & & & & 0 \\ & & & & \\ & & & & \\ 0 & & & & \end{bmatrix} \begin{bmatrix} & & & & \\ & & & & \\ & & & & 0 \\ & & & & \\ & & & & \end{bmatrix} - \begin{bmatrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ 0 & & & & \end{bmatrix}$$

Figura 10: Esquema $A = LU - R$ para ILU(0).

É possível observar, na Figura 10, que as matrizes L e U mantêm a mesma esparsidade da matriz A . É possível observar também que a matriz R dos resíduos contém duas diagonais adicionais (linhas não-tracejadas). Neste caso, as “linhas tracejadas” em R representam as posições das diagonais não nulas de A .

A grosso modo, na decomposição ILU 5 pontos, para o cálculo dos elementos da matriz L (l_{ij}) e dos elementos da matriz U (u_{ij}), procede-se como na decomposição LU convencional, mas se substitui por zero, em cada passo, todos os elementos de L e de U correspondente a uma diagonal zero da matriz A . Os termos não nulos, que foram zerados em L e U , são alocados em suas respectivas posições na matriz R dos resíduos.

Para a malha apresentada na Figura 3, a decomposição ILU será apresentada considerando a malha $\Omega^h = \{(i, j), i = 1, \dots, N_x, j = 1, \dots, N_y\}$, onde N_x é o número de

A seguir será dada a estrutura das matrizes L, D e U. Os elementos não nulos dessas matrizes têm a mesma localização dos elementos não nulos de A. A estrutura da matriz L é dada por

$$L = \begin{bmatrix} 1 & & & & & \\ \gamma_2 & \ddots & & & & \\ & \ddots & \ddots & & & \\ \alpha_{N_x+1} & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & & & \alpha_N & & \\ & & & & \gamma_N & \\ & & & & & 1 \end{bmatrix}. \quad (25)$$

A estrutura da matriz D é dada por

$$D = \begin{bmatrix} \delta_1 & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & & & \ddots \\ & & & & & & \delta_N \end{bmatrix}. \quad (26)$$

A estrutura da matriz U é dada por

$$U = \begin{bmatrix} 1 & \mu_1 & & \eta_1 & & & \\ & \ddots & \ddots & & \ddots & & \\ & & \ddots & \ddots & & \ddots & \\ & & & \ddots & \ddots & & \ddots \\ & & & & \ddots & \ddots & \\ & & 0 & & & \ddots & \eta_{N-N_x} \\ & & & & & & \mu_{N-1} \\ & & & & & & 1 \end{bmatrix}. \quad (27)$$

Segundo Saad (2003), Wesseling (1992) e Trottenberg et al. (2001), para problemas anisotrópicos modelados por matrizes pentadiagonais (que é o caso específico deste estudo), a decomposição ILU 7 pontos é mais eficiente do que a decomposição de 5 pontos (ILU(0)) apresentada anteriormente.

A decomposição ILU 7 pontos de uma matriz pentadiagonal torna as matrizes L e U como sendo esparsas também, mas não com a mesma esparsidade da matriz A original. Agora, tal decomposição permite que haja uma diagonal adicional tanto em L como em U pegando, portanto, dois pontos adicionais para a discretização, Por isto o nome ILU 7 pontos. Este caso é denominado, de uma forma geral, como ILU(1) por permitir apenas uma diagonal adicional em cada uma das matrizes L e U. Note que neste caso, a matriz dos resíduos contém menor erro, por isto o melhor desempenho da decomposição ILU.

A Figura 11 apresenta ordenação dos pontos em uma malha em que são considerados 7 pontos, incluindo nordeste (NE) e sudoeste (SW) ao que foi apresentado na Figura 3.

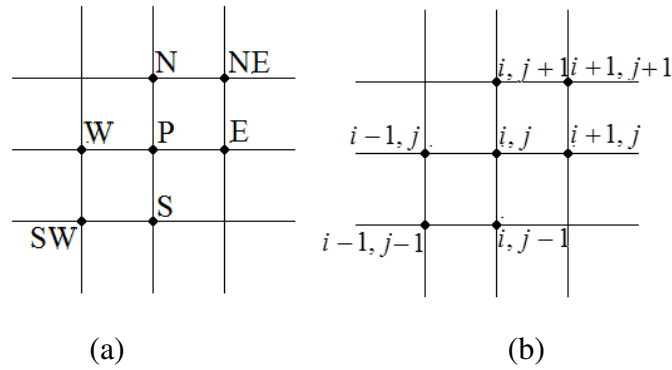


Figura 11: Malha bidimensional uniforme com 7 pontos: (a) notação geográfica; (b) notação indicial.

Note que é possível a decomposição ILU 9 pontos, que no caso de matriz pentadiagonal representa ILU(2), e assim por diante, até ILU(p), onde $0 \leq p \leq N_x$. Se $p=0$ as matrizes L e U apresentam-se sem nenhum preenchimento (esparsidade igual à esparsidade da matriz A) e $p=N_x$ representa o preenchimento completo das diagonais intermediárias entre a posição Sul $(i, j-1)$ e a posição Central (i, j) . Neste caso ter-se-ia a decomposição LU padrão.

Entretanto, segundo Saad (2003) e Wesseling (1992), para problemas anisotrópicos modelados por matrizes pentadiagonais, a decomposição ILU 9 pontos tem eficiência equivalente à decomposição de 7, porém com custo computacional superior. Por isto, neste trabalho optou-se por trabalhar com decomposição até 7 pontos.

Para uma decomposição ILU 7 pontos (ou ILU(1)), a estrutura da matriz L é dada por

$$\begin{aligned}\mu_k + \beta_k \delta_{k-N_x+1}^{-1} g_{k-N_x+1} &= q_k, \\ \varsigma_k + \gamma_k \delta_{k-1}^{-1} g_{k-1} &= 0, \\ \eta_k &= g_k.\end{aligned}$$

Da Eq. (30) observa-se que os elementos da matriz dos resíduos foram descartados. Nas próximas seções, será apresentada uma modificação para ILU, em que são considerados elementos não nulos na matriz R (este método será denominado MILU, em inglês, *ModifiedIncomplete LU factorization*). A partir da Eq. (30) pode-se descrever a decomposição ILU 7 pontos na sua forma recursiva como

$$\begin{aligned}\alpha_k &= a_k, \\ \beta_k &= -a_k \delta_{k-N_x}^{-1} \mu_{k-N_x}, \\ \gamma_k &= c_k - a_k \delta_{k-N_x}^{-1} \varsigma_{k-N_x}, \\ \delta_k &= d_k - a_k \delta_{k-N_x}^{-1} g_{k-N_x} - \beta_k \delta_{k-N_x+1}^{-1} \varsigma_{k-N_x+1} - \gamma_k \delta_{k-1}^{-1} \mu_{k-1}, \\ \mu_k &= q_k - \beta_k \delta_{k-N_x+1}^{-1} g_{k-N_x+1}, \\ \varsigma_k &= -\gamma_k \delta_{k-1}^{-1} g_{k-1}, \\ \eta_k &= g_k.\end{aligned}\tag{31}$$

Dado o sistema $Au = f$ e a decomposição ILU $A = LU - R$, tem-se:

$$\begin{aligned}(\text{LU} - R)u &= f, \\ \text{LU} u^{m+1} &= f + Ru^m, \\ \text{LU} u^{m+1} &= f + (\text{LU} - A)u^m, \\ \text{LU} u^{m+1} &= f + \text{LU} u^m - Au^m, \\ \text{LU} (u^{m+1} - u^m) &= f - Au^m, \\ \text{LU} e^m &= r^m.\end{aligned}$$

Com isso, cria-se o processo iterativo

$$\begin{aligned} Ly^m &= r^m, \\ Ue^m &= y^m, \\ u^{m+1} &= u^m + e^m, \\ u^m &= u^{m+1} \\ r^m &= f - Au^m \end{aligned}$$

Por possuir boas propriedades de suavização (mais detalhes na seção 3.4), o *solver* ILU pode ser usado para resolver uma ampla gama de problemas e será usado como um suavizador para o método *multigrid* (WESSELING, 1992; TROTTEMBERG et al., 2001; SAAD, 2003).

3.2.3 Decomposição LU incompleta modificada (MILU)

Na decomposição ILU apresentada na seção anterior, alguns elementos foram descartados durante o processo de eliminação. Existem técnicas que tentam reduzir esse efeito compensando as entradas descartadas. Um exemplo é somar na diagonal principal de U os módulos dos elementos que foram descartados. Esta técnica é denominada decomposição ILU modificada (em inglês, *Modified ILU decomposition*, MILU) e sua formulação é apresentada a seguir.

Na decomposição ILU, a matriz R contém zeros em todas as posições que correspondem à estrutura não nula de L e U , inclusive ao longo da diagonal principal, isto é, $R_{ii} = 0$. Com a introdução de um parâmetro σ pode-se definir a decomposição MILU como $A = L^\sigma U^\sigma - R^\sigma$. Neste caso a diagonal de R^σ não será mais zero. A ideia é adicionar à diagonal de U todos os elementos descartados na decomposição.

Assumindo que a decomposição incompleta da matriz A é realizada a cada linha, a diagonal principal de R e os elementos u_{ii} de U podem ser modificados de tal forma que

$$R_{kk} \leftarrow \sigma \sum |R_{kk}|, \quad (32)$$

$$u_{kk} \leftarrow u_{kk} + R_{kk}.$$

3.3 Notação estêncil

Nesta seção, será introduzida a notação estêncil, a qual será adotada no decorrer do texto para facilitar a descrição da LFA. Para tal, considera-se um operador discreto L_h em uma malha cartesiana retangular Ω^h e funções de malhas do tipo (TROTTEBERG et al., 2001).

$$\omega_h : \Omega^h \rightarrow \square, \quad (33)$$

$$(x, y) \rightarrow \omega_h(x, y).$$

Um estêncil geral $[s_{k_1 k_2}]_h$ é dado por

$$[s_{k_1 k_2}]_h = \begin{bmatrix} \vdots & \vdots & \vdots \\ \cdots & s_{-1,1} & s_{0,1} & s_{1,1} & \cdots \\ \cdots & s_{-1,0} & s_{0,0} & s_{1,0} & \cdots \\ \cdots & s_{-1,-1} & s_{0,-1} & s_{1,-1} & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}_h \quad (s_{k_1 k_2} \in \square), \quad (34)$$

onde $k_1, k_2 \in \mathbf{Z}$ indicam as posições no estêncil e definem um conjunto de funções de malha dadas por

$$[s_{k_1 k_2}]_h \omega_h(x, y) = \sum_{(k_1, k_2)} s_{k_1 k_2} \omega_h(x + k_1 h_x, y + k_2 h_y), \quad (35)$$

onde h_x e h_y indicam as distâncias entre os pontos nas direções x e y , respectivamente.

Assume-se que somente um número finito de coeficientes $s_{k_1 k_2}$ não são nulos.

Os estêncis mais utilizados são os compactos de cinco pontos e de nove pontos, dados respectivamente por

$$\begin{bmatrix} & s_{0,1} & \\ s_{-1,0} & s_{0,0} & s_{1,0} \\ & s_{0,-1} & \end{bmatrix}_h \mathbf{e} \begin{bmatrix} s_{-1,1} & s_{0,1} & s_{1,1} \\ s_{-1,0} & s_{0,0} & s_{1,0} \\ s_{-1,-1} & s_{0,-1} & s_{1,-1} \end{bmatrix}_h. \quad (36)$$

Considerando o operador Laplaciano $L_u = -\Delta u = -(u_{xx} + u_{yy})$. A discretização em uma malha $\Omega^h : [0,1] \times [0,1]$ pelo MDF e a aproximação de segunda ordem CDS conduzem a

$$\begin{aligned} L_h u_h(x, y) &= -\Delta_h u_h(x, y) \\ &= \frac{1}{h^2} [4u_h(x, y) - u_h(x-h, y) - u_h(x+h, y) - u_h(x, y-h) - u_h(x, y+h)] \\ &= \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h u_h(x, y). \end{aligned} \quad (37)$$

Assim,

$$L_h = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h, \quad (38)$$

representa o estêncil de cinco pontos do operador Laplaciano discreto Δ_h .

3.3.1 Notação estêncil para ILU

Para investigar as propriedades de suavização do método ILU, e realizar LFA considera-se a sua notação estêncil, que será apresentada a seguir.

Para um operador de diferenças discreto com cinco pontos tem-se

$$L_h = \begin{bmatrix} & * & \\ * & * & * \\ & * & \end{bmatrix}_h, \quad (39)$$

onde * significa possíveis elementos não nulos.

Uma decomposição ILU com sete pontos pode ser representada na notação estêncil por

$$L_h = \hat{L}_h \hat{U}_h - R_h, \quad (40)$$

onde \hat{L}_h é uma matriz triangular inferior, \hat{U}_h é uma matriz triangular superior e R_h a matriz dos resíduos.

Dependendo da ordenação dos pontos na malha, oito diferentes decomposições ILU podem ser definidas. Dentre todos os casos, serão apresentados três possíveis exemplos:

1. Se a ordenação dos pontos na malha é primeiro na direção leste (E) e em seguida na direção norte (N), a decomposição EN é da forma

$$\hat{L}_h = \begin{bmatrix} 0 & 0 & \\ * & * & 0 \\ & * & * \end{bmatrix}_h, \quad \hat{U}_h = \begin{bmatrix} * & * & \\ 0 & * & * \\ & 0 & 0 \end{bmatrix}_h, \quad R_h = \begin{bmatrix} * & 0 & 0 \\ & 0 & 0 & 0 \\ & & 0 & 0 & * \end{bmatrix}_h. \quad (41)$$

2. Se a ordenação dos pontos na malha é primeiro na direção norte (N) e em seguida na direção leste (E), a notação estêncil correspondente a decomposição NE é da forma

$$\hat{L}_h = \begin{bmatrix} * & 0 & \\ * & * & 0 \\ & * & 0 \end{bmatrix}_h, \quad \hat{U}_h = \begin{bmatrix} 0 & * & \\ 0 & * & * \\ & 0 & * \end{bmatrix}_h, \quad R_h = \begin{bmatrix} * & & \\ 0 & 0 & \\ 0 & 0 & 0 \\ & 0 & 0 \\ & & * \end{bmatrix}_h. \quad (42)$$

3. Se a ordenação dos pontos na malha é primeiro na direção norte (N) e em seguida na direção oeste (W), a notação estêncil correspondente a decomposição NW é da forma

$$\hat{L}_h = \begin{bmatrix} 0 & 0 & \\ 0 & * & * \\ & * & * \end{bmatrix}_h, \quad \hat{U}_h = \begin{bmatrix} * & * & \\ * & * & 0 \\ & 0 & 0 \end{bmatrix}_h, \quad R_h = \begin{bmatrix} 0 & 0 & * \\ 0 & 0 & 0 \\ * & 0 & 0 \end{bmatrix}_h. \quad (43)$$

O *solver* ILU pode ser utilizado para problemas anisotrópicos bidimensionais (WESSELING, 1992; TROTTEMBERG et al., 2001; SAAD, 2003) que serão vistos com mais detalhes na seção 3.5.

3.4 Método *multigrid*

Ao se aplicar um método iterativo básico para resolução de um sistema de equações algébricas pode-se observar que o erro cai rapidamente nas primeiras iterações, à medida que o número de iterações vai aumentando, o erro passa a decair mais lentamente. Isso significa que os métodos iterativos básicos apresentam propriedades de suavização de erros locais de alta frequência (componentes oscilatórias do erro), enquanto as baixas frequências são mantidas praticamente inalteradas (WESSELING, 1992; BRIGGS et al., 2000; TROTTEMBERG et al., 2001).

A grande vantagem do método *multigrid* em relação a outros métodos iterativos é que ele acelera a taxa de convergência e leva informações do problema para diversas malhas, desde as mais finas até as mais grossas, percorrendo portanto, todas as frequências de erros, desde as mais oscilatórias até as mais suaves.

Com um suavizador executam-se iterações em cada nível de malha até atingir o critério de convergência fixado para o sistema de equações original (malha mais fina). O processo consiste em executar poucas iterações em cada malha, já que o erro decai rapidamente nas primeiras iterações. Usa-se uma sequência de malhas, um suavizador e operadores de transferência entre as malhas.

Para visualizar os efeitos da suavização do erro inicial com métodos iterativos, Briggs et al. (2000) resolveram um problema homogêneo unidimensional que tem solução exata conhecida (nula) em que o erro na aproximação é a própria solução aproximada. Os modos de Fourier são dados por

$$v_j = \text{sen}\left(\frac{jk\pi}{N}\right), \quad 0 \leq j \leq N, \quad k \leq N-1, \quad (44)$$

onde N é o número de pontos, k é o número de ondas ou modos de Fourier e j é a componente do vetor v .

Na Figura 12 podem ser visualizados os modos de Fourier para $k=1$, $k=3$ e $k=6$. Os valores de k denotam quantos “meios-senos” constituem a curva no domínio do problema. Nota-se que pequenos valores de k correspondem a ondas mais longas e suaves, enquanto grandes valores de k correspondem a ondas oscilatórias.

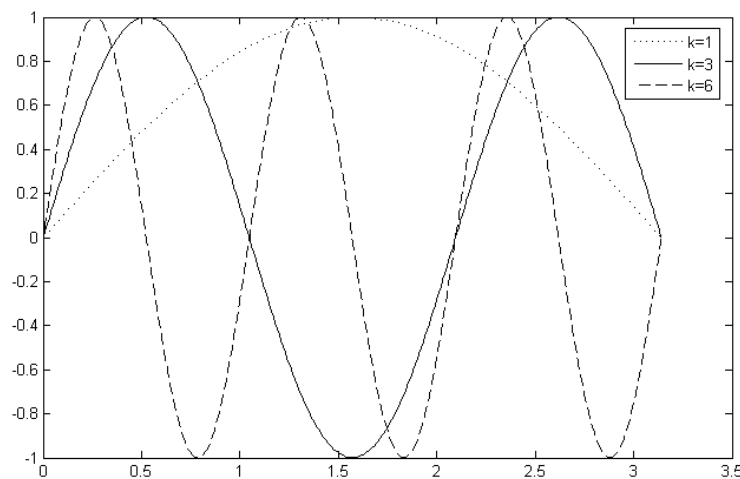


Figura 12: Modos de Fourier.

Fonte: Briggs et al. (2000)

Na Figura 13, observa-se como as componentes de erro suave apresentam-se em uma malha fina, denotada por Ω^h e em uma malha mais grossa, denotada por Ω^H . Neste trabalho utiliza-se $H = 2h$. Neste caso, o engrossamento será designado como padrão com razão de engrossamento (RE) dada por $RE = \frac{\dim(\Omega^H)}{\dim(\Omega^h)} = 2$, que é a razão entre a dimensão dos elementos da malha grossa e a dimensão dos elementos da malha fina. Observa-se que o modo de Fourier na malha Ω^h é um modo suave, enquanto que na malha Ω^{2h} torna-se um modo mais oscilatório.

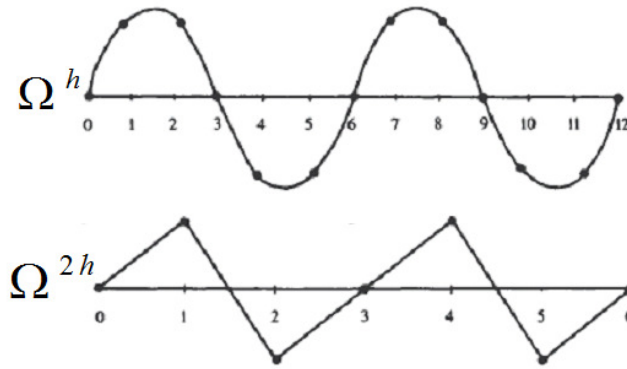


Figura 13: Componentes de erro na malha fina (Ω^h) e grossa (Ω^{2h}).

Fonte: Briggs et al. (2000)

3.4.1 Equação residual

Supondo que uma aproximação para a solução exata u da Eq. (1) seja dada por \bar{u} , então o erro algébrico pode ser calculado através de

$$e = u - \bar{u}. \quad (45)$$

A magnitude do erro definido pela Eq.(45), pode ser medida por uma norma vetorial, dentre as quais se destacam: a euclidiana ou norma-2 e a norma infinito (FORTUNA, 2000), cujas equações estão representadas a seguir.

$$\|e\|_2 = \sqrt{\sum_{i=1}^n e_i^2}, \quad (46)$$

e

$$\|e\|_\infty = \max_i |e_i|, \quad i = 1, 2, 3, \dots, N, \quad (47)$$

onde e_i é a i -ésima componente do erro.

Na prática, geralmente não se conhece a solução exata do sistema de equações algébricas e consequentemente não se pode determinar o erro. O que se pode determinar é o quanto \bar{u} está próximo da solução exata u ao longo das iterações, ou seja, calcular o resíduo, denotado r , em cada iteração. Este processo pode ser obtido utilizando-se a equação dada por

$$r = f - A\bar{u}. \quad (48)$$

Substituindo a Eq. (1) na Eq.(48), obtém-se

$$r = Au - A\bar{u}, \quad (49)$$

que para problemas lineares pode ser escrito como

$$r = A(u - \bar{u}). \quad (50)$$

Substituindo-se a Eq.(45) na Eq.(50), tem-se

$$Ae = r. \quad (51)$$

A Eq.(51) é chamada de equação residual e desempenha um papel importante para o *multigrid*, pois indica que o erro, dado pela Eq.(45), satisfaz o mesmo conjunto de equações que a solução u quando f é substituído pelo resíduo r .

Considerando-se que uma aproximação \bar{u} tenha sido obtida para o sistema linear dado pela Eq. (1), por meio de algum método iterativo, o resíduo pode ser calculado com a Eq.(48). Para melhorar essa aproximação, calcula-se o erro com a Eq.(51) e depois se corrige a solução atual obtendo uma nova aproximação, através da expressão

$$u = \bar{u} + e . \quad (52)$$

Assim, a solução é a aproximação atual acrescida do erro.

3.4.2 Filosofia do *multigrid*

Os métodos iterativos clássicos apresentados na seção 3.2 são eficientes somente na remoção de componentes do erro de altas frequências, o mesmo não ocorrendo para as baixas frequências. Para que as componentes do erro de baixas frequências possam ser eliminadas com eficiência, o método *multigrid* procura trabalhar com uma sequência de malhas cada vez mais grossas.

Como as componentes suaves do erro aparecem oscilatórias em malhas mais grossas, sugere-se a transferência do problema para uma malha com menor número de pontos que a malha original. Nessa malha mais grossa, o processo de suavização é repetido até que as componentes do erro tornem-se suaves e o problema é novamente transferido para uma malha mais grossa, com um menor número de pontos. Assim, sucessivamente, até chegar à malha mais grossa possível ou pré-estabelecida.

3.4.3 Componentes do *multigrid*

Para transferir informações da malha fina (Ω^h) para a malha grossa (Ω^{2h}) e vice-versa, são utilizados operadores de transferência de malhas. As informações da malha Ω^h são transferidas para a malha Ω^{2h} , utilizando-se um operador, chamado de operador de restrição (WESSELING, 1992; BRIGGS et al., 2000; TROTTEBERG et al., 2001) e denotado neste texto por I_h^{2h} . O operador de restrição mais utilizado é o de

injeção, o qual transfere pontos da malha fina para pontos coincidentes na malha grossa e é definido como

$$\bar{u}^{2h} = I_h^{2h} \bar{u}^h, \quad (53)$$

ou, para cada ponto da malha

$$\bar{u}_{i,j}^{2h} = I_h^{2h} \bar{u}_{2i,2j}^h, \quad 1 \leq i \leq \frac{N_x}{2} - 1, \quad 1 \leq j \leq \frac{N_y}{2} - 1. \quad (54)$$

A Figura 14 mostra como o operador de restrição por injeção transfere pontos da malha fina para a malha grossa.

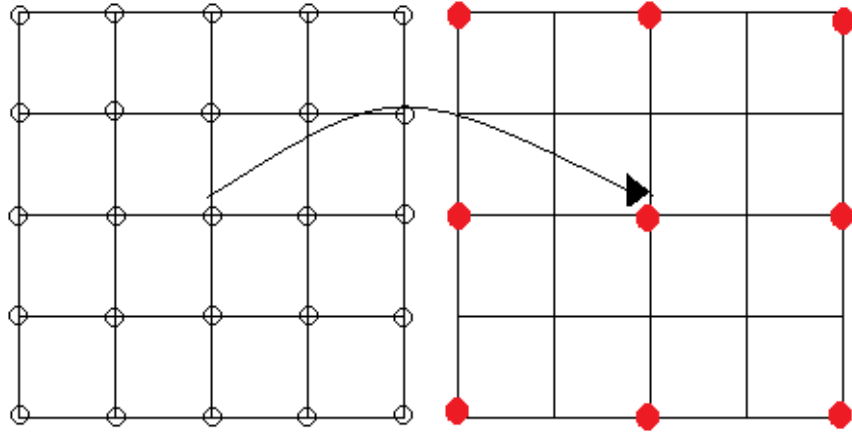


Figura 14: Restrição por injeção.

Outro operador de restrição bastante utilizado é o de ponderação completa (em inglês, *Full Weighting*, FW) que pode ser definido, para razão de engrossamento padrão (RE=2) como

$$\begin{aligned} \bar{u}_{i,j}^{2h} = \frac{1}{16} [& \bar{u}_{2i-1,2j-1}^h + \bar{u}_{2i-1,2j+1}^h + \bar{u}_{2i+1,2j-1}^h + \bar{u}_{2i+1,2j+1}^h + \\ & + 2(\bar{u}_{2i,2j-1}^h + \bar{u}_{2i,2j+1}^h + \bar{u}_{2i-1,2j}^h + \bar{u}_{2i+1,2j}^h) + 4\bar{u}_{i,j}^h] \end{aligned} \quad (55)$$

com, $1 \leq i \leq \frac{N_x}{2} - 1$, $1 \leq j \leq \frac{N_y}{2} - 1$.

Tem-se ainda o operador de meia ponderação (em inglês, *Half Weighting*, HW), definido como

$$\bar{u}_{i,j}^{2h} = \frac{1}{8} [\bar{u}_{2i,2j-1}^h + \bar{u}_{2i,2j+1}^h + \bar{u}_{2i-1,2j}^h + \bar{u}_{2i+1,2j}^h + 4\bar{u}_{i,j}^h], \quad (56)$$

com, $1 \leq i \leq \frac{N_x}{2} - 1$, $1 \leq j \leq \frac{N_y}{2} - 1$.

Pode-se escrever estes operadores em notação estêncil. Para FW, tem-se

$$I_h^{2h} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^{2h}. \quad (57)$$

Para HW, tem-se

$$I_h^{2h} = \frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}_h^{2h}. \quad (58)$$

Para injeção, tem-se

$$I_h^{2h} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_h^{2h}. \quad (59)$$

Há um outro tipo de operador de restrição que é utilizado quando se tem anisotropia geométrica. Ele é conhecido por ponderação parcial, que utiliza pesos

somente na direção em que ocorre a anisotropia, cujos estêncis (OLIVEIRA, 2010) são dados por

- Ponderação Parcial em x (PW_x):

$$I_h^{2h} = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}_h^{2h}. \quad (60)$$

- Ponderação Parcial em y (PW_y):

$$I_h^{2h} = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix}_h^{2h}. \quad (61)$$

Para transferir informações da malha grossa (Ω^{2h}) para a malha fina (Ω^h), utiliza-se um operador, chamado de operador de prolongação ou interpolação (WESSELING, 1992; BRIGGS et al., 2000; TROTTERBERG et al., 2001), denotado neste texto por I_{2h}^h . O operador de prolongação mais utilizado para problemas bidimensionais é o de interpolação bilinear, definido como

$$\bar{u}^h = I_{2h}^h \bar{u}^{2h}, \quad (62)$$

ou ainda

$$\begin{aligned} \bar{u}_{2i,2j}^h &= \bar{u}_{i,j}^{2h}, \\ \bar{u}_{2i+1,2j}^h &= \frac{1}{2} (\bar{u}_{i,j}^{2h} + \bar{u}_{i+1,j}^{2h}), \end{aligned} \quad (63)$$

$$\bar{u}_{2i,2j+1}^h = \frac{1}{2}(\bar{u}_{i,j}^{2h} + \bar{u}_{i,j+1}^{2h}),$$

$$\bar{u}_{2i+1,2j+1}^h = \frac{1}{4}(\bar{u}_{i,j}^{2h} + \bar{u}_{i+1,j}^{2h} + \bar{u}_{i,j+1}^{2h} + \bar{u}_{i+1,j+1}^{2h}),$$

$$\text{com, } 1 \leq i \leq \frac{N_x}{2} - 1, 1 \leq j \leq \frac{N_y}{2} - 1.$$

A Figura 15 mostra como o operador de interpolação bilinear transfere informações da malha grossa para a malha fina.

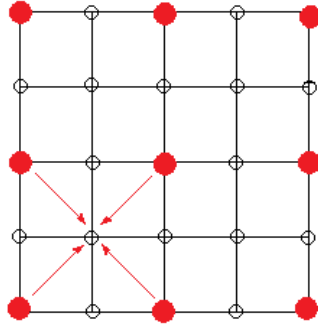


Figura 15: Operador de prolongação.

Outro operador de prolongação que será utilizado nesta tese é prolongação de 7 pontos (WIENANDS e JOPPICH (2005)), cujo estêncil é dado por

$$I_{2h}^h = \frac{1}{8} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 1 \end{bmatrix}_{2h}^h. \quad (64)$$

3.4.4 Ciclos e esquemas de aproximação

A sequência com que as diversas malhas são percorridas é denominada ciclo. Os ciclos V e W, cuja metodologia consiste em percorrer todas as malhas iniciando o processo na malha mais fina, são os mais usados. Além dos ciclos V e W, o ciclo F também aparece frequentemente na literatura.

Na Figura 16 foram considerados alguns níveis de malha para apresentar o ciclo V e o ciclo W.

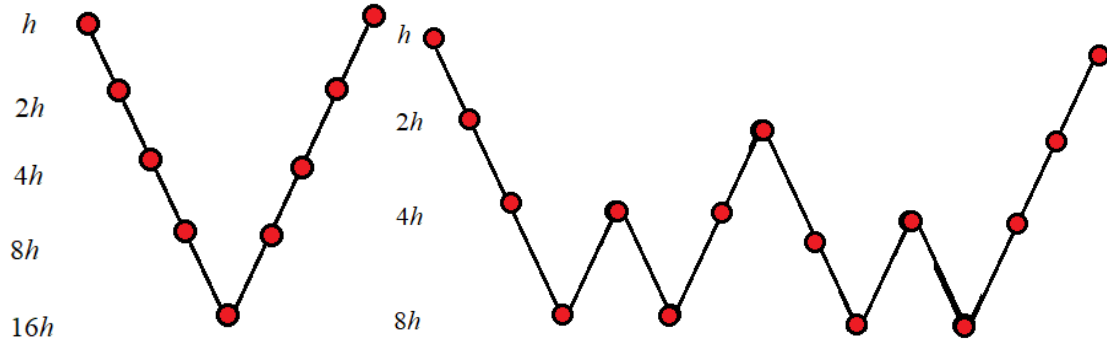


Figura 16: Ciclos V e W.

O método *multigrid* pode ser implementado utilizando-se o esquema CS ou FAS (TROTTEBERG et al., 2001).

Algoritmo 1: Esquema de Correção (CS) para duas malhas

Início: $CSTG(\bar{u}_0, \bar{u}, f, h)$

1. Suavizar $A^h u^h = f^h$, v_1 vezes na malha fina Ω^h com estimativa inicial \bar{u}_0^h ;
 2. Calcular o resíduo na malha fina, ou seja, $r^h = f^h - A^h \bar{u}^h$;
 3. Restringir r^h na malha grossa Ω^{2h} , ou seja, obter $r^{2h} = I_h^{2h} r^h$ e definir $f^{2h} = r^{2h}$.
 4. Resolver a equação residual $A^{2h} u^{2h} = f^{2h}$ com estimativa inicial $\bar{u}_0^{2h} = 0$;
 5. Definir $e^{2h} = \bar{u}^{2h}$;
 6. Interpolar e^{2h} para a malha fina, ou seja, obter $e^h = I_{2h}^h e^{2h}$;
-

7. Corrigir a aproximação da solução na malha fina, ou seja, $\bar{u}^h \leftarrow \bar{u}^h + e^h$;

8. Suavizar $A^h u^h = f^h$, v_2 vezes na malha fina Ω^h , com estimativa inicial \bar{u}^h ;

Fim de $\bar{u} \leftarrow CSTG(\bar{u}_0, \bar{u}, f, h)$.

3.5 Problemas anisotrópicos

A simulação de certo fenômenos físicos de pequena escala, em CFD, pode gerar malhas altamente distorcidas, conduzindo a uma lenta convergência dos métodos *multigrid*. Tais fenômenos são denominados anisotrópicos e podem ser classificados em físicos e geométricos.

Um modelo que exemplifica anisotropia física e geométrica para uma equação de difusão bidimensional é dado em Trottenberg et al.(2001) como

$$-\left|g^2 + \varepsilon w^2\right|u_{xx} + 2(1-\varepsilon)g w u_{xy} - \left|w^2 + \varepsilon g^2\right|u_{yy} = f, \quad (65)$$

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

Para $\alpha = 0$, diz-se que a expressão dada pela Eq.(65) está alinhada com o eixo de coordenada x e torna-se

$$-u_{xx} - \varepsilon u_{yy} = f. \quad (66)$$

Para $\alpha = \frac{\pi}{2}$, diz-se que a expressão dada pela Eq.(65) está alinhada como o eixo de coordenada y e torna-se

$$-\varepsilon u_{xx} - u_{yy} = f. \quad (67)$$

As equações Eqs.(66) e(67) apresentam anisotropia física, ou seja, anisotropia nos coeficientes. Se a equação diferencial tem coeficientes constantes para as derivadas parciais, porém distintos nas direções dos eixos coordenados, trata-se de anisotropia física ou anisotropia de coeficientes.

A partir da discretização da Eq.(67), pode-se determinar uma notação estêncil para a matriz A da Eq.(1) na forma

$$A = \frac{1}{h^2} \begin{pmatrix} & -1 & \\ -\varepsilon & 2(1+\varepsilon) & -\varepsilon \\ & -1 & \end{pmatrix}. \quad (68)$$

Se na discretização da malha obtém-se elementos com tamanho constante, porém distintos nas direções dos eixos coordenados, tem-se a chamada anisotropia geométrica ou anisotropia de malha. Um exemplo para a anisotropia geométrica, em que a discretização é baseada em malhas com razão de aspecto distinta da unidade, é dada pela Eq.(67), discretizada em uma malha com tamanho constante h_x na direção x e $h_y = \frac{h_x}{\sqrt{\varepsilon}}$ na direção y , com ε distinto da unidade. Nesse caso, a razão de aspecto RA é definida por

$$RA = \frac{h_x}{h_y} = \sqrt{\varepsilon}. \quad (69)$$

Um exemplo para a anisotropia física, onde o acoplamento entre os pontos vizinhos é muito forte em alguma das direções, é a discretização da Eq.(66) em uma malha de tamanho h em ambas as direções, isto é, $h = h_x = h_y$ com ε distinto da unidade. Neste caso a razão de aspecto RA é redefinida em função das Eqs. (65) a (67) por

$$RA = \sqrt{\varepsilon}. \quad (70)$$

Mesmo a malha sendo isotrópica $\left(\frac{h_x}{h_y} = 1\right)$, tem-se um desempenho ruim para o método *multigrid* quando o coeficiente $\varepsilon \rightarrow 0$.

A explicação para tal desempenho pode ser entendida considerando $\varepsilon = 0$ (que seria um caso limite) na Eq.(66), ou seja

$$-u_{xx} = f. \quad (71)$$

Assim a notação estêncil de A torna-se

$$A = \frac{1}{h^2} \begin{pmatrix} 0 & & \\ -1 & 2 & -1 \\ 0 & & \end{pmatrix}. \quad (72)$$

Observa-se que o problema discreto torna-se uma coleção de equações de Poisson unidimensionais na direção x , sem conexão com o eixo y . Assim, os erros na direção y , geralmente terão um padrão aleatório, longe da suavidade necessária para o engrossamento funcionar bem.

Segundo Briggs et al. (2000), existem duas abordagens diferentes para garantir uma boa convergência do *multigrid* em problemas anisotrópicos.

A primeira abordagem trata-se da suavização por pontos com engrossamento em apenas uma direção dos eixos coordenados. Essa abordagem caracteriza-se por considerar, por exemplo, o engrossamento horizontal, não considerando as linhas verticais. Trata-se de um semiengrossamento como pode ser observado na Figura 17, em que a malha grossa tem o mesmo espaçamento da malha original em y e o dobro do espaçamento em x .

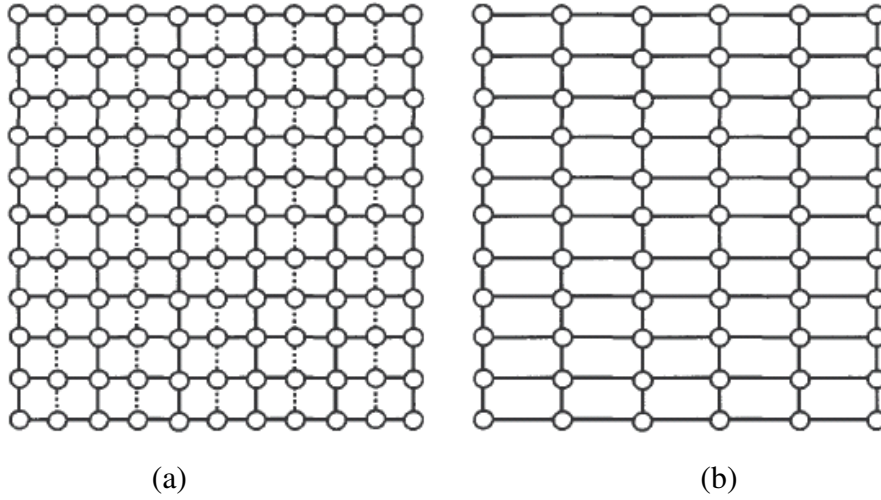


Figura 17: (a) Malha fina original; (b) malha grossa com semiengrossamento na direção x .

Fonte: Briggs et al. (2000)

A operação de transferência de informações da malha grossa para a malha fina (interpolação) pode ser realizada unidimensionalmente ao longo da linha horizontal e as equações da malha grossa podem ser corrigidas como

$$\begin{aligned}
 v_{2i,j}^h &\leftarrow v_{2i,j}^h + v_{i,j}^{2h}, \\
 v_{2i+1,j}^h &\leftarrow v_{2i+1,j}^h + \frac{v_{i,j}^{2h} + v_{i+1,j}^{2h}}{2}.
 \end{aligned} \tag{73}$$

Uma desvantagem do semiengrossamento é que ele não é tão rápido computacionalmente quanto o engrossamento completo tanto para problemas isotrópicos quanto para problemas anisotrópicos.

Na segunda abordagem, a suavização é aplicada por linhas, com engrossamento padrão nas duas direções. A relaxação por linhas ou bloco, citada na seção 3.2 apresenta um bom desempenho para resolução de problemas anisotrópicos. A relaxação pode ser desenvolvida escrevendo-se um sistema de equações na forma de blocos. No caso do problema de anisotropia física dado pela Eq.(66), a matriz A da Eq.(1) pode ser escrita na forma de blocos como

$$A = \begin{pmatrix} D & -cI & \cdots & & \\ -cI & D & -cI & \cdots & \\ \vdots & -cI & D & -cI & \\ & \vdots & \vdots & \vdots & -cI \\ & & \cdots & -cI & D \end{pmatrix}, \quad (74)$$

onde $c = \frac{\varepsilon}{h^2}$, I é a matriz identidade e D na sua forma estêncil é dada por

$$D = \frac{1}{h^2} \begin{pmatrix} -1 & 2+2\varepsilon & -1 \end{pmatrix}.$$

4. ANÁLISE DE FOURIER LOCAL

A análise de Fourier local (LFA) permite prever o desempenho do método *multigrid*, pois fornece estimativas de taxas de convergência de seus componentes. Assim, a LFA torna-se uma ferramenta poderosa para a análise quantitativa e o projeto de métodos *multigrid* eficientes.

Neste capítulo são apresentados os conceitos gerais e terminologia da LFA, análise de duas malhas para calcular o fator de convergência assintótica variando diversos componentes *multigrid*, tais como: operadores de restrição, operadores de prolongação e *solver*.

4.1 Conceitos gerais

No contexto da análise de Fourier, consideram-se funções de malha da forma

$$\varphi_h(\theta, x) = e^{i\theta x/h}, \text{ com } \theta = (\theta_1, \theta_2) \in \mathbb{R}^2, \quad (75)$$

onde x varia na malha infinita Ω_h , onde a influência das fronteiras pode ser desprezada (TROTTEBERG et al., 2001; WESSELING, 1992) e θ é um parâmetro contínuo que representa a frequência da função de malha.

Utilizando LFA é possível calcular o fator de suavização de S_h , denotado por $\mu_{loc}(S_h)$, o fator de convergência assintótica de M_h^{2h} , denotado por $\rho_{loc}(M_h^{2h})$ e fator de redução de erro para M_h^{2h} , denotado por $\gamma_{loc}(M_h^{2h})$.

O fator de suavização $\mu_{loc}(S_h)$ é o raio espectral do operador de suavização S_h e estima quanto o *solver* reduz as componentes de altas frequências do erro. O fator de convergência assintótica $\rho_{loc}(M_h^{2h})$ fornece o comportamento da convergência assintótica do método *multigrid* e $\gamma_{loc}(M_h^{2h})$ refere-se à redução do erro em um passo de iteração medido em uma norma apropriada.

4.2 Terminologia

Para a LFA serão considerados operadores discretos lineares gerais com coeficientes constantes, os quais são definidos em uma malha infinita.

Trottenberg et al. (2001) apresentam a análise de Fourier para o método *multigrid* utilizando um problema bidimensional com engrossamento padrão, os quais serão utilizados aqui para formalizar LFA.

Considerando $x = (x_1, x_2)$ uma malha de comprimento fixo

$$h = (h_1, h_2). \quad (76)$$

A malha fina é dada por

$$\Omega^h = \{x = jh := (j_1 h_1, j_2 h_2), j = (j_1, j_2) \in \mathbf{Z}^2\}, \quad (77)$$

e a malha imediatamente mais grossa é dada por,

$$\Omega^{2h} = \{x = j2h := (j_1 2h_1, j_2 2h_2), j = (j_1, j_2) \in \mathbf{Z}^2\}. \quad (78)$$

Considerando-se o operador discreto L_h correspondente ao estêncil

$$L_h \square [s_k]_h, (k = (k_1, k_2) \in \mathbf{Z}^2), \quad (79)$$

ou seja,

$$L_h \omega_h(x) = \sum_{k \in V} s_k \omega_h(x + kh), \quad (80)$$

em que k representa os índices do estêncil, e os coeficientes constantes $s_k \in \mathbb{R}$ ou \mathbb{C} são valores de L_h em relação às posições k .

As funções de malha podem ser escritas como

$$\varphi_h(\theta, x) = e^{i\theta x/h} := e^{i\theta_1 x_1/h_1} e^{i\theta_2 x_2/h_2}, \quad x \in \Omega^h. \quad (81)$$

Tomando $\theta = (\theta_1, \theta_2)$ como um parâmetro contínuo que caracteriza a frequência da função de malha, tem-se

$$\varphi_h(\theta, x) = \varphi_h(\theta', x) \text{ para } x \in \Omega^h, \quad (82)$$

se, e somente se,

$$\theta_1 \equiv \theta'_1 \pmod{2\pi} \text{ e } \theta_2 \equiv \theta'_2 \pmod{2\pi}. \quad (83)$$

Para tanto, basta considerar $\varphi_h(\theta, x)$ com $\theta \in [-\pi, \pi) \times [-\pi, \pi) = [-\pi, \pi)^2$, ou ainda $-\pi \leq \theta < \pi$.

Lema 1: Para $-\pi \leq \theta < \pi$ todas as funções de malha $\varphi_h(\theta, x)$ são autofunções de um operador discreto que pode ser descrito por um estêncil diferencial. A equação,

$$L_h \varphi_h(\theta, x) = \tilde{L}_h(\theta) \varphi_h(\theta, x) \quad (x \in \Omega^h), \quad (84)$$

é válida, e

$$L_h = [s_k]_h \text{ e } \tilde{L}_h(\theta) = \sum_k s_k e^{i\theta k}, \quad (85)$$

em que $\tilde{L}_h(\theta)$ é denominado autovalor.

A demonstração do Lema 1 pode ser encontra em Trottenberg et al. (2001).

Para análise de suavização e análise de duas malhas, deve-se distinguir componentes de alta e baixa frequência de Ω^h com relação Ω^{2h} . Para pontos na malha Ω^{2h} tem-se

$$\varphi_h(\theta, x_{2j}^h) = \varphi_{2h}(2\theta, x_j^{2h}), \quad (86)$$

ou seja, a função com número de onda θ relativo a Ω^h tem número de onda 2θ relativo a Ω^{2h} . Assim, somente

$$\varphi_h(\theta, x) \text{ com } -\frac{\pi}{2} \leq \theta < \frac{\pi}{2}, \quad (87)$$

são visíveis em Ω^h .

Para cada $\theta' \in [-\pi/2, \pi/2]^2$, três outras componentes de frequência $\varphi_h(\theta, x)$ com $\theta \in [-\pi, \pi]^2$ coincidem em Ω^{2h} com $\varphi_h(\theta', x)$ e não são distinguíveis (visíveis) em Ω^{2h} . Assim, tem-se

$$\varphi_h(\theta, x) = \varphi_h(\theta', x) \text{ para } x \in \Omega^{2h} \text{ se, e somente se, } \theta \equiv \theta' \pmod{\pi}. \quad (88)$$

Segundo Trottenberg et al. (2001) definem-se as componentes de baixas e altas frequências como se segue.

Definição 1. φ é uma componente de baixa frequência $\Leftrightarrow \theta \in T^{low} = [-\pi/2, \pi/2]^2$; φ é uma componente de alta frequência $\Leftrightarrow \theta \in T^{high} = [-\pi, \pi]^2 \setminus [-\pi/2, \pi/2]^2$.

A Figura 18 ilustra as regiões de baixas frequências (região branca interior) e altas frequências (região hachurada).

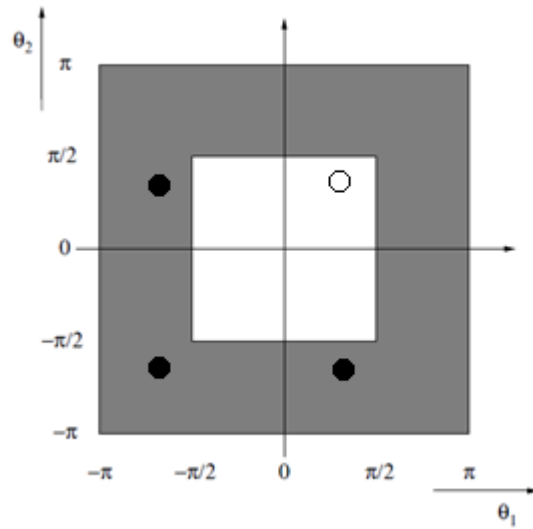


Figura 18: Regiões de baixas (região branca interior) e altas frequências (região hachurada).

Fonte: Trottenberget al.(2001)

4.3 Análise de suavização

Para descrever a análise de suavização considera-se uma equação diferencial discretizada como

$$L_h u_h = f_h, \quad (89)$$

e considera-se que um método de suavização pode ser escrito localmente como

$$L_h^+ \bar{\omega}_h + L_h^- \omega_h = f_h, \quad (90)$$

onde ω_h é uma aproximação de u_h na iteração anterior e $\bar{\omega}_h$ a aproximação na iteração atual (WESSELING, 1992; TROTTEBERG et al., 2001).

Assim, a relaxação pode ser escrita considerando a separação

$$L_h = L_h^+ + L_h^-. \quad (91)$$

Subtraindo a Eq. (90) da Eq.(89), obtém-se

$$\bar{v}_h = S_h v_h, \quad (92)$$

sendo $\bar{v}_h = u_h - \bar{\omega}_h$ e $v_h = u_h - \omega_h$ os erros e S_h o operador de suavização.

Aplicando L_h^- e L_h^+ nas autofunções $\varphi(\theta, x)$, tem-se

$$L_h^- e^{i\theta x/h} = \tilde{L}_h^-(\theta) e^{i\theta x/h} \text{ e } L_h^+ e^{i\theta x/h} = \tilde{L}_h^+(\theta) e^{i\theta x/h} \quad (93)$$

onde \tilde{L}_h^- e \tilde{L}_h^+ representam os autovalores dos operadores L_h^- e L_h^+ , respectivamente.

Lema 2: Com as hipóteses das equações Eqs. (89) e (90) todas as funções $\varphi(\theta, x)$ com $\tilde{L}_h^+ \neq 0$ são autofunções de S_h :

$$S_h \varphi(\theta, x) = \tilde{S}_h \varphi(\theta, x), \quad -\pi \leq \theta < \pi, \quad (94)$$

com fator de amplificação dado por

$$\tilde{S}_h(\theta) := -\frac{\tilde{L}_h^-(\theta)}{\tilde{L}_h^+(\theta)}. \quad (95)$$

A demonstração do Lema 2 pode ser encontrada em Trottenberg et al. (2001).

O fator de suavização $\mu(S_h)$ é definido por

$$\mu_{loc} = \mu_{loc}(S_h) := \sup \left\{ \left| \tilde{S}_h(\theta) \right| : \theta \in T^{higt} \right\}. \quad (96)$$

4.4 Análise de duas malhas

Para análise de duas malhas, ou seja, cálculo do fator de convergência assintótica $\rho_{loc}(M_h^{2h})$, utiliza-se o operador de duas malhas dado por

$$M_h^{2h} := S_h^{v_2} K_h^{2h} S_h^{v_1}, \quad (97)$$

e, o operador de correção na malha grossa dado por

$$K_h^{2h} = I_h - I_{2h}^h (L_{2h})^{-1} I_h^{2h} L_h, \quad (98)$$

onde S_h é o operador de suavização, v_1 e v_2 são, respectivamente, os números de pré e pós-suavização, I_h o estêncil do operador identidade, L_h é o operador discreto na malha Ω^h , L_{2h} é o operador discreto na malha Ω^{2h} , I_h^{2h} é o operador de restrição e I_{2h}^h é o operador de prolongação. É necessário analisar como os operadores L_h , L_{2h} , I_h^{2h} , I_{2h}^h , S_h atuam nas componentes de Fourier $\varphi(\theta, x)$.

Como observa-se na Figura 18, as quádruplas de $\varphi_h(\theta, x)$ coincidem em Ω^{2h} . Considerando-se as frequências

$$\theta^{(0,0)} := (\theta_1, \theta_2), \quad \theta^{(1,1)} := (\bar{\theta}_1, \bar{\theta}_2), \quad \theta^{(1,0)} := (\bar{\theta}_1, \theta_2), \quad \text{e} \quad \theta^{(0,1)} := (\theta_1, \bar{\theta}_2), \quad (99)$$

onde

$$\bar{\theta}_i := \begin{cases} \theta_i + \pi, & \text{se } \theta_i < 0 \\ \theta_i - \pi, & \text{se } \theta_i \geq 0 \end{cases}. \quad (100)$$

Lema 3:

1. Para qualquer baixa frequência ($\theta \in T^{low}$) tem-se

$$\varphi(\theta^{(0,0)}, x) \equiv \varphi(\theta^{(1,1)}, x) \equiv \varphi(\theta^{(1,0)}, x) \equiv \varphi(\theta^{(0,1)}, x), \text{ com } x \in \Omega^{2h}. \quad (101)$$

2. Cada uma das quatro componentes de Fourier $\varphi(\theta^\alpha, \cdot) = \varphi_h(\theta^\alpha, \cdot)$ com $\alpha \in \{(0,0), (1,1), (1,0), (0,1)\}$ coincide em Ω^{2h} com $\varphi_{2h}(2\theta^{(0,0)}, \cdot)$, ou seja,

$$\varphi_h(\theta^\alpha, x) \equiv \varphi_{2h}(2\theta^{(0,0)}, x). \quad (102)$$

A demonstração do Lema 3 pode ser encontrada em Trottenberg et al. (2001).

Todas $\varphi(\theta^\alpha, \cdot)$ são autofunções de L_h e S_h .

Definição 2. O conjunto de funções de malhas de Ω^{2h} obtido através de funções e malha em Ω^h é chamado de harmônico. O espaço gerado (*span*) por esse harmônico quadridimensional é

$$E_h^\theta := \text{span}\left\{\varphi_h(\theta^\alpha, \cdot) : \alpha = (\alpha_1, \alpha_2) \in \{(0,0), (1,1), (1,0), (0,1)\}\right\}. \quad (103)$$

Pode-se mostrar (veja Trottenberg et al. (2001)) que o espaço E_h^θ é invariante sob o operador M_h^{2h} .

Considerando $\psi \in E_h^\theta$, pode-se escrever ψ como combinação linear dos elementos de E_h^θ , ou seja,

$$\psi = A^{(0,0)}\varphi(\theta^{(0,0)}, \cdot) + A^{(1,1)}\varphi(\theta^{(1,1)}, \cdot) + A^{(1,0)}\varphi(\theta^{(1,0)}, \cdot) + A^{(0,1)}\varphi(\theta^{(0,1)}, \cdot). \quad (104)$$

Deve-se analisar como os coeficientes A^α são transformados quando os operadores dados pelas Eqs. (98) e (97) são aplicadas a ψ .

Teorema 1: O operador K_h^{2h} é representado em E_h^θ por uma matriz \hat{K}_h^{2h} de ordem 4x4, como

$$\hat{K}_h^{2h}(\theta) = \hat{I}_h - (\hat{I}_{2h}^h(\theta)) (\hat{L}_{2h}(2\theta))^{-1} (\hat{I}_{2h}^{2h}(\theta))_y \hat{L}_h(\theta), \text{ com } \theta \in T^{low}, \quad (105)$$

onde \hat{I}_h e $\hat{L}_h(\theta)$ são matrizes 4x4, $\hat{I}_{2h}^{2h}(\theta)$ é uma matriz 1x4, $\hat{I}_{2h}^h(\theta)$ é uma matriz 4x1 e $(\hat{L}_{2h}(2\theta))^{-1}$ é uma matriz 1x1.

A demonstração do Teorema 1 pode ser encontrada em Trottenberg et al. (2001) e Wienands e Joppich (2005).

Aplicando \hat{K}_h^{2h} para qualquer $\psi \in E_h^\theta$, os coeficientes correspondentes A^α , são transformados em

$$\begin{pmatrix} A^{(0,0)} \\ A^{(1,1)} \\ A^{(1,0)} \\ A^{(0,1)} \end{pmatrix} \Leftarrow \hat{K}_{2h}^h(\theta) \begin{pmatrix} A^{(0,0)} \\ A^{(1,1)} \\ A^{(1,0)} \\ A^{(0,1)} \end{pmatrix}. \quad (106)$$

Terema 2: Se $S_h : E_h^\theta \rightarrow E_h^\theta$ com $\theta \in T^{low}$, então tem-se uma representação para M_h^{2h} no espaço E_h^θ por uma matriz de ordem 4x4 da forma $\hat{M}_h^{2h}(\theta)$

$$\hat{M}_h^{2h} := (\hat{S}_h(\theta))^{v_2} \hat{K}_h^{2h}(\theta) (\hat{S}_h(\theta))^{v_1}, \quad (107)$$

onde $\hat{K}_h^{2h}(\theta)$ é dado pela Eq. (105) e $\hat{S}_h(\theta)$ representa a matriz do estêncil do suavizador S_h .

A demonstração do Teorema 2 pode ser encontrada em Trottenberg et al. (2001), Wienands e Joppich (2005).

Aplicando \hat{M}_h^{2h} para $\psi \in E_h^\theta$, tem-se

$$M_h^{2h}\psi = B^{(0,0)}\varphi(\theta^{(0,0)}, \cdot) + B^{(1,1)}\varphi(\theta^{(1,1)}, \cdot) + B^{(1,0)}\varphi(\theta^{(1,0)}, \cdot) + B^{(0,1)}\varphi(\theta^{(0,1)}, \cdot), \quad (108)$$

onde

$$\begin{pmatrix} B^{(0,0)} \\ B^{(1,1)} \\ B^{(1,0)} \\ B^{(0,1)} \end{pmatrix} \Leftarrow \hat{M}_h^{2h}(\theta) \begin{pmatrix} A^{(0,0)} \\ A^{(1,1)} \\ A^{(1,0)} \\ A^{(0,1)} \end{pmatrix}. \quad (109)$$

Para garantir a existência de M_h^{2h} , exclui-se o conjunto

$$\Lambda = \{ \theta \in T^{low} : \tilde{L}_h(\theta) = 0 \text{ ou } \tilde{L}_{2h}(\theta) = 0 \}. \quad (110)$$

A seguir, serão desenvolvidos as componentes da matriz $\hat{M}_h^{2h}(\theta)$.

1. Operador I_h : I_h é representado pela matriz identidade 4x4 em E_h^θ

2. Operador L_h : Pelo Lema 2, tem-se

$$\begin{pmatrix} A^{(0,0)} \\ A^{(1,1)} \\ A^{(1,0)} \\ A^{(0,1)} \end{pmatrix} \Leftarrow \begin{pmatrix} \tilde{L}_h(\theta^{(0,0)}) & & & \\ & \tilde{L}_h(\theta^{(1,1)}) & & \\ & & \tilde{L}_h(\theta^{(1,0)}) & \\ & & & \tilde{L}_h(\theta^{(0,1)}) \end{pmatrix} \begin{pmatrix} A^{(0,0)} \\ A^{(1,1)} \\ A^{(1,0)} \\ A^{(0,1)} \end{pmatrix}, \quad (111)$$

onde a matriz 4x4 é o operador $\hat{L}_h(\theta)$.

3. Operador de restrição I_h^{2h} : Os coeficientes A^α transformam-se como

$$A_{2h} = \hat{I}_h^{2h}(\theta) \begin{pmatrix} A^{(0,0)} \\ A^{(1,1)} \\ A^{(1,0)} \\ A^{(0,1)} \end{pmatrix}, \quad (112)$$

onde A_{2h} é o coeficiente resultante das componentes de Fourier $\varphi_{2h}(2\theta^{(0,0)}, \dots)$ e

$$\hat{I}_h^{2h}(\theta) = \begin{pmatrix} \tilde{I}_h^{2h}(\theta^{(0,0)}) & \tilde{I}_h^{2h}(\theta^{(1,1)}) & \tilde{I}_h^{2h}(\theta^{(1,0)}) & \tilde{I}_h^{2h}(\theta^{(0,1)}) \end{pmatrix}. \quad (113)$$

Os autovalores (que também são chamados de símbolos) dos operadores de restrição I_h^{2h} por injeção (INJ), ponderação completa (FW), de ponderação parcial em x (PW_x) e de ponderação parcial em y (PW_y) são mostrados na Tabela 2.

Tabela 2: Autovalores dos operadores de restrição usados na LFA.

Restrição	$\tilde{I}_h^{2h}(\theta^\alpha)$
INJ	1
FW	$\frac{1}{4}(1 + \cos \bar{\theta}_1)(1 + \cos \bar{\theta}_2)$
PW_x	$\frac{1}{2}(1 + \cos \bar{\theta}_1)$
PW_y	$\frac{1}{2}(1 + \cos \bar{\theta}_2)$

4. Solução na malha Ω^{2h} : Para $\theta = \theta^{(0,0)}$ e $\theta \in T^{low}$ tem-se:

$$\tilde{L}_{2h}(2\theta) = \sum_k s_{k,2h} e^{i2\theta k}. \quad (114)$$

5. Operador de interpolação I_{2h}^h : Para I_{2h}^h aplicado a $\varphi_{2\theta}(2\theta^{(0,0)}, \cdot)$, o operador de interpolação é dado por

$$\begin{pmatrix} A^{(0,0)} \\ A^{(1,1)} \\ A^{(1,0)} \\ A^{(0,1)} \end{pmatrix} \Leftarrow \hat{I}_{2h}^h(\theta) A_{2h}, \quad (115)$$

com

$$\hat{I}_{2h}^h(\theta) = \begin{pmatrix} \tilde{I}_{2h}^h(\theta^{(0,0)}) \\ \tilde{I}_{2h}^h(\theta^{(1,1)}) \\ \tilde{I}_{2h}^h(\theta^{(1,0)}) \\ \tilde{I}_{2h}^h(\theta^{(0,1)}) \end{pmatrix}. \quad (116)$$

Os autovalores dos operadores de prolongação I_{2h}^h , bilinear e 7 pontos são mostrados na Tabela 3.

Tabela 3: Autovalores dos operadores de prolongação usados na LFA.

Prolongação	$\tilde{I}_{2h}^h(\theta^\alpha)$
Bilinear	$(1 + \cos \bar{\theta}_1)(1 + \cos \bar{\theta}_2)$
7 pontos	$(1 + \cos \bar{\theta}_1 + \cos \bar{\theta}_2 + \cos(\bar{\theta}_1 - \bar{\theta}_2))$

6. Operador de suavização: Todas as componentes de Fourier são autofunções de S_h , onde S_h pode ser representada pela matriz diagonal de ordem 4x4

$$\begin{pmatrix} A^{(0,0)} \\ A^{(1,1)} \\ A^{(1,0)} \\ A^{(0,1)} \end{pmatrix} \Leftarrow \begin{pmatrix} \tilde{S}_h(\theta^{(0,0)}) & & & \\ & \tilde{S}_h(\theta^{(1,1)}) & & \\ & & \tilde{S}_h(\theta^{(1,0)}) & \\ & & & \tilde{S}_h(\theta^{(0,1)}) \end{pmatrix} \begin{pmatrix} A^{(0,0)} \\ A^{(1,1)} \\ A^{(1,0)} \\ A^{(0,1)} \end{pmatrix}, \quad (117)$$

onde

$$\hat{S}_h = \begin{pmatrix} \tilde{S}_h(\theta^{(0,0)}) & & & \\ & \tilde{S}_h(\theta^{(1,1)}) & & \\ & & \tilde{S}_h(\theta^{(1,0)}) & \\ & & & \tilde{S}_h(\theta^{(0,1)}) \end{pmatrix}. \quad (118)$$

Definição 3: Representando M_h^{2h} por uma matriz 4x4 \hat{M}_h^{2h} , calcula-se o fator de convergência assintótica $\rho_{loc}(M_h^{2h})$ por

$$\rho_{loc}(M_h^{2h}) = \sup \left\{ \rho_{loc}(\hat{M}_h^{2h}(\theta)) : \theta \in T^{low}, \theta \notin \Lambda \right\}, \quad (119)$$

onde $\rho_{loc}(\hat{M}_h^{2h}(\theta))$ é o raio espectral da matriz 4x4 $\hat{M}_h^{2h}(\theta)$.

O Algoritmo 2, descrito em Franco (2017), é sugerido para determinar o fator de convergência de um operador de duas malhas para o método *multigrid* com engrossamento padrão.

Algoritmo 2: LFA para o fator de convergência em duas malhas

Início: Escolher o suavizador, os operadores de restrição e prolongação

para cada um dos modos suaves, exceto (0,0) **faça**

Calcular θ^α usando a Eq. (99).

Criar a matriz diagonal de ordem 4×4 . \hat{L}_h com os autovalores de L_h .

Calcular a matriz de ordem 1×4 . \hat{I}_h^{2h} do operador de restrição I_h^{2h} .

Calcular a matriz de ordem 1×1 . \hat{L}_{2h} na malha Ω^{2h} .

Calcular a matriz de ordem 4×1 . \hat{I}_{2h}^h do operador de prolongação I_{2h}^h .

Calcular a matriz \hat{K}_h^{2h} dada pela Eq. (105).

Calcular a matriz diagonal de ordem 4×4 , \hat{S}_h com os autovalores de S_h .

Calcular os autovalores da matriz \hat{M}_h^{2h} e escolher seu valor máximo em módulo.

Guardar o autovalor máximo em cada modo de Fourier suave.

fim

Escolher o máximo dos autovalores ρ (este representa o pior fator de convergência).

4.5 Análise de Fourier local para os suavizadores

Nesta seção será detalhada LFA para os principais suavizadores. Primeiramente apresentam-se os *solvers* do tipo GS e na sequência os *solvers* do tipo ILU.

4.5.1 Análise de Fourier local para os *solvers* do tipo GS

É possível calcular o fator de suavização dos *solvers* GS-Lex e linha-GS utilizando as Eqs. (95) e (96).

Se estes *solvers* forem utilizados para resolver a equação de difusão anisotrópica dada pela Eq. (67) a sua notação estêncil dada pela Eq. (68), torna-se.

$$L_h(\varepsilon) = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -\varepsilon & 2(1+\varepsilon) & -\varepsilon \\ & -1 & \end{bmatrix}_h. \quad (120)$$

Para GS-Lex, L_h^+ e L_h^- na Eq. (95) são dados por

$$L_h^+ = \frac{1}{h^2} \begin{bmatrix} & 0 & \\ -\varepsilon & 2(1+\varepsilon) & 0 \\ & -1 & \end{bmatrix}_h \text{ e } L_h^- = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ 0 & 0 & -\varepsilon \\ & 0 & \end{bmatrix}_h, \quad (121)$$

e os autovalores (dados pelo Lema 1)

$$\tilde{L}_h(\theta) = \frac{1}{h^2} [(2+2\varepsilon) - (\varepsilon e^{i\theta_1} + e^{i\theta_2} + \varepsilon e^{-i\theta_1} + e^{-i\theta_2})]. \quad (122)$$

Os autovalores dos operadores que se seguirão, estarão na Tabela 4, que será exibida após todos os operadores serem definidos.

Para x -linha-GS, L_h^+ e L_h^- na Eq. (95) são dados por

$$L_h^+ = \frac{1}{h^2} \begin{bmatrix} & 0 & \\ -\varepsilon & 2(1+\varepsilon) & -\varepsilon \\ & -1 & \end{bmatrix}_h \text{ e } L_h^- = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ 0 & 0 & 0 \\ & 0 & \end{bmatrix}_h. \quad (123)$$

Para y -linha-GS, tem-se

$$L_h^+ = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ 0 & 2(1+\varepsilon) & -\varepsilon \\ & -1 & \end{bmatrix}_h \text{ e } L_h^- = \frac{1}{h^2} \begin{bmatrix} & 0 & \\ -\varepsilon & 0 & 0 \\ & 0 & \end{bmatrix}_h. \quad (124)$$

A análise de suavização utilizando a Eq. (118) só é possível se todas as funções de malha $\varphi(\theta, \cdot)$ são autofunções do operador S_h . Isto ocorre com o *solver*GS-Lex e linha-GS conforme apresentado nas Eqs.(121),(123) e (124).

Para os casos em que $\varphi(\theta, \cdot)$ não é autofunção de S_h , para se calcular o fator de suavização $\mu_{loc}(S_h)$, considera-se um operador de malha grossa Q_h^{2h} que anule as componentes de baixa frequência dos erros e as componentes de altas frequências permaneçam inalteradas. Nesta categoria, encontram-se os *solvers* GS-RB e zebra-GS. O operador Q_h^{2h} é definido em E_h^θ por

$$Q_h^{2h}\varphi(\theta, \cdot) = \begin{cases} 0, & \text{se } \theta = \theta^{(0,0)} \in T^{low} \\ \varphi(\theta, \cdot), & \text{se } \theta \in \{\theta^{(1,1)}, \theta^{(1,0)}, \theta^{(0,1)}\} \end{cases} \quad (125)$$

Como consequência, a matriz \hat{K}_h^{2h} dada pela Eq. (105) é substituída por

$$\hat{Q}_h^{2h}(\theta) = \begin{pmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \text{ para } \theta \in T^{low}, \quad (126)$$

para a análise de suavização. Portanto, substitui-se $M_h^{2h} = S_h^{v_2} K_h^{2h} S_h^{v_1}$ por $M_h^{2h} = S_h^{v_2} Q_h^{2h} S_h^{v_1}$ e $\rho_{loc}(M_h^{2h})$ é dado por

$$\rho_{loc}(S_h^{v_2} Q_h^{2h} S_h^{v_1}) = \sup \left\{ \rho \left(\left(\hat{S}_h(\theta) \right)^{v_2} \hat{Q}_h^{2h} \left(\hat{S}_h(\theta) \right)^{v_1} \right) : \theta \in T^{low} \right\}. \quad (127)$$

Definição4: Assumindo que S_h possui a propriedade de invariância, ou seja, se $S_h : E_h^\theta \rightarrow E_h^\theta$ com $\theta \in T^{low}$, define-se o fator de suavização $\mu_{loc}(S_h)$ de S_h considerando $\nu = \nu_1 + \nu_2$ por

$$\mu_{loc}(S_h) = \sup \left\{ \sqrt[\nu]{\rho(\hat{Q}_h^{2h}(S_h(\theta))^\nu)} : \theta \in T^{low} \right\}. \quad (128)$$

Para determinar a matriz S_h utilizando *solvers* como Gauss-Seidel por pontos com ordenação *red-black* (GS-RB) e Gauss-Seidel por linhas, com ordenação zebra (*x*-zebra-GS, *y*-zebra-GS e *alt*-zebra-GS) definem-se dois passos parciais. Para o caso de GS-RB, cada um deles é da forma Jacobi.

No primeiro passo o subconjunto $\tilde{\Omega}_h$ de Ω^h é formado pelos pontos *red* de Ω^h . No segundo passo, $\tilde{\Omega}_h$ é formado pelos pontos *black* de Ω^h e com isso tem-se o operador

$$S_h^{RB} = S_h^{BLACK} \cdot S_h^{RED}. \quad (129)$$

Em ambos os passos (*red* ou *black*) deve-se usar

$$L_h^+ = \frac{1}{h^2} \begin{bmatrix} 0 & & \\ 0 & 2(1+\varepsilon) & 0 \\ & 0 & \end{bmatrix}_h \text{ e } L_h^- = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 0 & -1 \\ & -1 & \end{bmatrix}_h. \quad (130)$$

Utilizando a Eq.(95), tem-se

$$S_h = -(L_h^+)^{-1} L_h^- = I_h - \frac{\varpi h^2}{4} L_h, \quad (131)$$

e

$$S_h^{PARCIAL} \varphi(x) = \begin{cases} 1 - \frac{\varpi h^2}{4} \tilde{L}_h(\theta) \varphi(\theta, x) & \text{com } x \in \tilde{\Omega}_h \\ \varphi(\theta, x), & \text{com } x \in \Omega^h \setminus \tilde{\Omega}_h \end{cases}. \quad (132)$$

Com isso a Eq. (104) pode ser reescrita como

$$\psi(\theta, x) = \begin{cases} \beta_1 \varphi(\theta, x) & \text{com } x \in \tilde{\Omega}_h \\ \beta_2 \varphi(\theta, x), & \text{com } x \in \Omega^h \setminus \tilde{\Omega}_h \end{cases}, \quad (133)$$

ou seja, $\psi(\theta, x)$ pode ser escrito com combinação linear de $\varphi(\theta^\alpha, \cdot)$ e $\varphi(\tilde{\theta}^\alpha, \cdot)$, onde $\tilde{\theta}^\alpha = \theta$ e $\tilde{\alpha} = (1, 1) - \alpha$.

Desta forma, tem-se para o *solver* GS-RB

$$S_h^{RED} = \begin{bmatrix} A^{(0,0)} + 1 & A^{(1,1)} - 1 & 0 & 0 \\ A^{(0,0)} - 1 & A^{(1,1)} + 1 & 0 & 0 \\ 0 & 0 & A^{(1,0)} + 1 & A^{(0,1)} - 1 \\ 0 & 0 & A^{(1,0)} - 1 & A^{(0,1)} + 1 \end{bmatrix}_h, \quad (134)$$

e

$$S_h^{BLACK} = \begin{bmatrix} A^{(0,0)} + 1 & -A^{(1,1)} + 1 & 0 & 0 \\ -A^{(0,0)} + 1 & A^{(1,1)} + 1 & 0 & 0 \\ 0 & 0 & A^{(1,0)} + 1 & -A^{(0,1)} + 1 \\ 0 & 0 & -A^{(1,0)} + 1 & A^{(0,1)} + 1 \end{bmatrix}_h. \quad (135)$$

Para x -zebra-GS, tem-se

$$S_h^{RED} = \begin{bmatrix} A^{(0,0)} + 1 & 0 & 0 & A^{(0,1)} - 1 \\ 0 & A^{(1,1)} + 1 & A^{(1,0)} - 1 & 0 \\ 0 & A^{(1,1)} - 1 & A^{(1,0)} + 1 & 0 \\ A^{(0,0)} - 1 & 0 & 0 & A^{(0,1)} + 1 \end{bmatrix}_h, \quad (136)$$

e

$$S_h^{BLACK} = \begin{bmatrix} A^{(0,0)} + 1 & 0 & 0 & -A^{(0,1)} - 1 \\ 0 & A^{(1,1)} + 1 & -A^{(1,0)} + 1 & 0 \\ 0 & -A^{(1,1)} + 1 & A^{(1,0)} + 1 & 0 \\ -A^{(0,0)} + 1 & 0 & 0 & A^{(0,1)} + 1 \end{bmatrix}_h. \quad (137)$$

Para y -zebra-GS, tem-se

$$S_h^{RED} = \begin{bmatrix} A^{(0,0)} + 1 & 0 & A^{(1,0)} - 1 & 0 \\ 0 & A^{(1,1)} + 1 & 0 & A^{(0,1)} - 1 \\ A^{(0,0)} - 1 & 0 & A^{(1,0)} + 1 & 0 \\ 0 & A^{(1,1)} - 1 & 0 & A^{(0,1)} + 1 \end{bmatrix}_h, \quad (138)$$

e

$$S_h^{BLACK} = \begin{bmatrix} A^{(0,0)} + 1 & 0 & -A^{(1,0)} + 1 & 0 \\ 0 & A^{(1,1)} + 1 & 0 & -A^{(0,1)} + 1 \\ -A^{(0,0)} + 1 & 0 & A^{(1,0)} + 1 & 0 \\ 0 & -A^{(1,1)} + 1 & 0 & A^{(0,1)} + 1 \end{bmatrix}_h. \quad (139)$$

Os autovalores dos operadores S_h apresentados nesta seção (veja Wienands e Joppich (2005)) são mostrados na Tabela 4.

Tabela 4: Autovalores dos operadores de suavização usados na LFA.

<i>Solvers</i>	$\tilde{L}_h(\theta)$
GS-Lex	$\frac{1}{h^2}[(2+2\varepsilon) - (\varepsilon e^{i\theta_1} + e^{i\theta_2} + \varepsilon e^{-i\theta_1} + e^{-i\theta_2})]$
GS-RB	$\frac{-\varepsilon + \varepsilon \cos x + \cos y}{4h^2}$
<i>x</i> -linha-GS	$\frac{e^{i\theta_2}}{h^2(-2-2\varepsilon + \varepsilon e^{-i\theta_1} + e^{-i\theta_2} + \varepsilon e^{i\theta_1})}$
<i>y</i> -linha-GS	$\frac{e^{i\theta_1}}{h^2(-2-2\varepsilon + \varepsilon e^{-i\theta_1} + e^{-i\theta_2} + \varepsilon e^{i\theta_2})}$
<i>x</i> -zebra-GS	$\frac{\cos y}{h^2(1 + \varepsilon - \varepsilon \cos x)}$
<i>y</i> -zebra-GS	$\frac{\varepsilon \cos x}{h^2(1 + \varepsilon - \cos y)}$

4.5.2 Análise de Fourier local para os *solvers* do tipo ILU

Para análise de Fourier local utilizando os *solvers* do tipo ILU, considera-se operador laplaciano com o estêncil de 5 pontos, dado pela Eq.(120).

A decomposição ILU(1) ou decomposição ILU 7 pontos desse mesmo operador será dada por

$$L_h = \begin{bmatrix} f & g & 0 \\ c & d & q \\ 0 & a & b \end{bmatrix}_h, \quad (140)$$

sendo que, neste caso, $b = f = 0$.

A decomposição ILU7 pontos, em notação estêncil, é representada por:

$$L_h = \hat{L}_h \hat{U}_h - R_h. \quad (141)$$

onde \hat{L}_h é uma matriz triangular inferior, \hat{U}_h é uma matriz triangular superior e R_h é a matriz dos resíduos.

Os autovalores de L_h , \hat{L}_h , \hat{U}_h e R_h são dadas, respectivamente, por

$$\lambda_h(\theta), \lambda_h^L(\theta), \lambda_h^U(\theta), \lambda_h^R(\theta) \text{ com } -\pi \leq \theta < \pi. \quad (142)$$

O operador de suavização S_h para o *solver*ILU, segundo Trottenberg et al.(2001), é dado por

$$S_h \varphi(\theta, x) = \tilde{S}_h \varphi(\theta, x), \quad -\pi \leq \theta < \pi, \quad (143)$$

com

$$\tilde{S}_h(\theta) := \frac{\lambda_h^L(\theta) \lambda_h^U(\theta) - \lambda_h(\theta)}{\lambda_h^L(\theta) \lambda_h^U(\theta)} = \frac{\lambda_h^R(\theta)}{\lambda_h(\theta) + \lambda_h^R(\theta)}, \quad (144)$$

onde $\lambda_h(\theta) = ae^{-i\theta_2} + be^{i(\theta_1 - \theta_2)} + ce^{-i\theta_1} + d + qe^{i\theta_1} + fe^{-i(\theta_1 - \theta_2)} + ge^{i\theta_2}$.

Dependendo da ordenação dos pontos na malha, diferentes decomposições ILU podem ser obtidas. Na sequência apresentam-se as possíveis ordenações e $\lambda_h^R(\theta)$ para essas ordenações.

Para a ordenação lexicográfica ILU_{EN}, tem-se

$$\hat{L}_h = \begin{pmatrix} 0 & 0 & 0 \\ \gamma & 1 & 0 \\ 0 & \alpha & \beta \end{pmatrix}_h, \quad \hat{U}_h = \begin{pmatrix} \zeta & \eta & 0 \\ 0 & \delta & \mu \\ 0 & 0 & 0 \end{pmatrix}_h \text{ e } R_h = \begin{pmatrix} p_2 & 0 & 0 & 0 \\ & 0 & p_3 & 0 \\ & 0 & 0 & 0 \\ & & & p_1 \end{pmatrix}_h. \quad (145)$$

Neste caso $\lambda_h^R(\theta)$ é dado por $\lambda_h^R(\theta) = p_1 e^{i(2\theta_1 - \theta_2)} + p_3 + p_2 e^{i(-2\theta_1 + \theta_2)}$, onde $p_1 = \beta \delta^{-1} \mu$, $p_2 = \gamma \delta^{-1} \varsigma$ e $p_3 = \sigma(|p_1 + p_2|)$, (p_1 , p_2 e p_3 são calculados da mesma forma para todas as ordenações de ILU), lembrando que σ é um parâmetro introduzido na decomposição. Para $\sigma = 0$ tem-se ILU padrão e para $\sigma \neq 0$ tem-se MILU, conforme definido em seções anteriores.

Os valores de $\lambda_h^R(\theta)$ para as ordenações de ILU que se seguirão, estarão na Tabela 5.

Outro exemplo de ordenação para ILU, ILU_{NE} é dada por

$$\hat{L}_h = \begin{pmatrix} \varsigma & 0 & 0 \\ \gamma & 1 & 0 \\ 0 & \alpha & 0 \end{pmatrix}_h, \hat{U}_h = \begin{pmatrix} 0 & \eta & 0 \\ 0 & \delta & \mu \\ 0 & 0 & \beta \end{pmatrix}_h \text{ e } R_h = \begin{pmatrix} p_1 & & \\ 0 & 0 & 0 \\ 0 & p_3 & 0 \\ 0 & 0 & 0 \\ & & p_2 \end{pmatrix}_h. \quad (146)$$

Com a ordenação WS, ILU_{WS} é dada por

$$\hat{L}_h = \begin{pmatrix} \varsigma & \eta & 0 \\ 0 & 1 & \mu \\ 0 & 0 & 0 \end{pmatrix}_h, \hat{U}_h = \begin{pmatrix} 0 & 0 & 0 \\ \gamma & \delta & 0 \\ 0 & \alpha & \beta \end{pmatrix}_h \text{ e } R_h = \begin{pmatrix} p_1 & 0 & 0 & 0 \\ & 0 & p_3 & 0 \\ & 0 & 0 & 0 \\ & & & p_2 \end{pmatrix}_h.$$

Com a ordenação SW, ILU_{SW} é dada por

$$\hat{L}_h = \begin{pmatrix} 0 & \eta & 0 \\ 0 & 1 & \mu \\ 0 & 0 & \beta \end{pmatrix}_h, \hat{U}_h = \begin{pmatrix} \varsigma & 0 & 0 \\ \gamma & \delta & 0 \\ 0 & \alpha & 0 \end{pmatrix}_h \text{ e } R_h = \begin{pmatrix} p_2 & & \\ 0 & 0 & 0 \\ 0 & p_3 & 0 \\ 0 & 0 & 0 \\ & & p_1 \end{pmatrix}_h.$$

Com a ordenação NW, ILU_{NW} é dada por

$$\hat{L}_h = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & \mu \\ 0 & \alpha & \beta \end{pmatrix}_h, \hat{U}_h = \begin{pmatrix} \varsigma & \eta & 0 \\ \gamma & \delta & 0 \\ 0 & 0 & 0 \end{pmatrix}_h \text{ e } R_h = \begin{pmatrix} 0 & 0 & p_1 \\ 0 & p_3 & 0 \\ p_2 & 0 & 0 \end{pmatrix}_h.$$

Com a ordenação WN, ILU_{WN} é dada por

$$\hat{L}_h = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & \mu \\ 0 & \alpha & \beta \end{pmatrix}_h, \hat{U}_h = \begin{pmatrix} \varsigma & \eta & 0 \\ \gamma & \delta & 0 \\ 0 & 0 & 0 \end{pmatrix}_h \text{ e } R_h = \begin{pmatrix} 0 & 0 & p_2 \\ 0 & p_3 & 0 \\ p_1 & 0 & 0 \end{pmatrix}_h.$$

Com a ordenação SE, ILU_{SE} é dada por

$$\hat{L}_h = \begin{pmatrix} \varsigma & \eta & 0 \\ \gamma & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}_h, \hat{U}_h = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \delta & \mu \\ 0 & \alpha & \beta \end{pmatrix}_h \text{ e } R_h = \begin{pmatrix} 0 & 0 & p_2 \\ 0 & p_3 & 0 \\ p_1 & 0 & 0 \end{pmatrix}_h.$$

Com a ordenação ES, ILU_{ES} é dada por

$$\hat{L}_h = \begin{pmatrix} \gamma & \eta & 0 \\ \mu & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}_h, \hat{U}_h = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \delta & \mu \\ 0 & \alpha & \beta \end{pmatrix}_h \text{ e } R_h = \begin{pmatrix} 0 & 0 & p_1 \\ 0 & p_3 & 0 \\ p_2 & 0 & 0 \end{pmatrix}_h.$$

Os valores de $\lambda_h^R(\theta)$ para as diferentes direções dos *solvers* do tipo ILU (ver Trottenberg et al. (2001) e Wesseling (1992)) são mostrados na Tabela 5.

Tabela 5: Autovalores dos operadores de suavização do tipo ILU usados na LFA.

<i>Solvers</i>	$\lambda_h^R(\theta)$
EN	$p_1 e^{i(2\theta_1 - \theta_2)} + p_3 + p_2 e^{i(-2\theta_1 + \theta_2)}$
NE	$p_1 e^{i(-\theta_1 + 2\theta_2)} + p_3 + p_2 e^{i(\theta_1 - 2\theta_2)}$
WS	$p_1 e^{i(-2\theta_1 + \theta_2)} + p_3 + p_2 e^{i(2\theta_1 - \theta_2)}$
SW	$p_1 e^{i(\theta_1 - 2\theta_2)} + p_3 + p_2 e^{i(-\theta_1 + 2\theta_2)}$

NW	$p_1 e^{i(\theta_1 + \theta_2)} + p_3 + p_2 e^{i(-\theta_1 - \theta_2)}$
WN	$p_1 e^{i(-\theta_1 - \theta_2)} + p_3 + p_2 e^{i(\theta_1 + \theta_2)}$
SE	$p_1 e^{i(-\theta_1 - \theta_2)} + p_3 + p_2 e^{i(\theta_1 + \theta_2)}$
ES	$p_1 e^{i(\theta_1 + \theta_2)} + p_3 + p_2 e^{i(-\theta_1 - \theta_2)}$

É possível mostrar que o fator de convergência assintótica (ρ_{loc}) e o fator de suavização (μ_{loc}) apresentam os mesmos resultados para ordenações diferentes e a Figura 19 esquematiza as ordenações equivalentes apresentadas acima, significando na prática, que apenas quatro destas direções merecem atenção. Estes resultados serão detalhados nos capítulos 7 e 8.

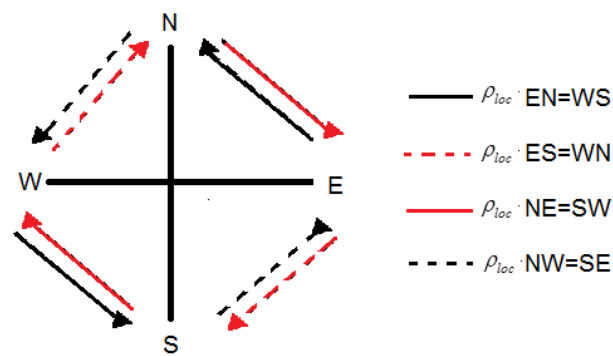


Figura 19: Ordenações equivalentes do *solver*ILU.

No próximo capítulo serão detalhados os modelos matemático e numérico empregados nesta tese.

5.MODELOS MATEMÁTICO E NUMÉRICO

Neste capítulo são apresentados os modelos matemático e numérico que foram utilizados na tese, além dos detalhes do método *multigrid* que foram empregados. A partir deste capítulo, a variável de interesse passa a ser temperatura, portanto, o sistema de equações dado na Eq. (1), torna-se

$$AT = b, \quad (147)$$

onde A é a matriz dos coeficientes, T é a variável de interesse e b é o vetor dos termos independentes.

5.1 Equação de difusão anisotrópica

Para o problema apresentado a seguir, o domínio de cálculo utilizado é dado por $0 \leq x \leq 1$, $0 \leq y \leq 1$ e a discretização das equações é feita através de malha uniforme com um número de pontos dado por $N = N_x \cdot N_y$, onde N_x e N_y são os números de pontos nas direções coordenadas x e y , respectivamente, incluindo os contornos.

O problema de anisotropia difusiva será analisado a partir da equação de difusão bidimensional dado por

$$\begin{cases} -\varepsilon T_{xx} - T_{yy} = S \\ T(0, y) = T(x, 0) = T(x, 1) = T(1, y) = 0 \end{cases} \quad (148)$$

onde T é a temperatura, T_{xx} é a derivada segunda de T em relação a x , T_{yy} é a derivada segunda em relação a y e $\varepsilon > 0$ é o fator de anisotropia.

O termo fonte S e a solução analítica são dados por

$$\begin{aligned} S &= 2[\varepsilon(1 - 6x^2)y^2(1 - y^2) + (1 - 6y^2)x^2(1 - x^2)] \text{ e} \\ T(x, y) &= (x^2 - x^4)(y^4 - y^2). \end{aligned} \quad (149)$$

A solução analítica da Eq.(148), apresentada na Eq.(149), pode ser observada na Figura 20.

A escolha desse termo fonte e solução analítica ocorre devido ao fato de estes representarem um problema que apresenta anisotropia física, portanto, representativa do problema de interesse. Para esse tipo de problema o *multigrid* não tem a eficiência esperada, ou seja, não converge ou demora a convergir. Além disso, conhecendo a sua solução analítica é possível realizar uma análise de erros.

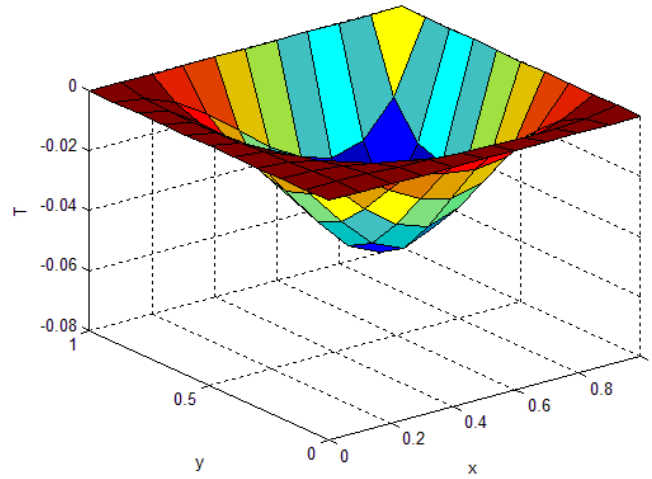


Figura 20: Solução analítica apresentada na Eq.(149).

5.2 Modelo numérico e detalhes computacionais

A discretização da Eq.(148) foi realizada mediante o uso do MDF com CDS (FERZIGER e PERIC, 2002), resultando em

$$-\varepsilon \left(\frac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{(h_x)^2} \right) - \left(\frac{T_{i,j-1} - 2T_{i,j} + T_{i,j+1}}{(h_y)^2} \right) = S_{i,j}. \quad (150)$$

Para os pontos internos, a discretização do problema é dada por

$$a_P T_P + a_W T_W + a_N T_N + a_E T_E + a_S T_S = b_P, \quad (151)$$

onde

$$a_P = \left(\frac{2\varepsilon}{h_x^2} + \frac{2}{h_y^2} \right), \quad a_W = a_E = -\frac{\varepsilon}{h_x^2}, \quad a_N = a_S = -\frac{1}{h_y^2}, \quad b_P = S_P. \quad (152)$$

Para os contornos norte (N), sul (S), leste (E) e oeste (W), conforme Figura 3(b), considerando $h_x = \frac{1}{N_x - 1}, h_y = \frac{1}{N_y - 1}$ e assumindo-se $h_x = h_y$ tem-se, $a_P = 1$,

$$a_W = a_N = a_E = a_S = 0.$$

O sistema de equações lineares dado pela Eq. (147) foi resolvido utilizando o método *multigrid* geométrico (BRIGGS et al., 2000; TROTTERBERG et al., 2001) com esquema CS, ciclo V e estimativa inicial nula.

A razão de engrossamento é dada por $RE = 2$ (engrossamento padrão). Os operadores de transferência entre as malhas empregados foram restrição por injeção (INJ), meia ponderação (HW), ponderação completa (FW), ponderação parcial em x (PW_x), ponderação parcial em y (PW_y) e prolongação por interpolação bilinear e interpolação 7 pontos.

Os sistemas de equações obtidos da discretização foram resolvidos utilizando os *solvers* ILU 7 pontos em diferentes direções (ILU_{EN} , ILU_{NE} , dentre outros).

O critério de parada utilizado para interromper o processo iterativo é baseado na norma do resíduo adimensionalizada. O resíduo do sistema de equações algébricas é definido por

$$r^m = b - AT^m, \quad (153)$$

onde T^m é a solução na iteração m .

Considerando $L^m = \|r^m\|_1$ e $L^0 = \|r^0\|_1$, se $\frac{L^m}{L^0} \leq tol$ o processo iterativo é interrompido para $tol = 10^{-10}$.

5.3 Definição da tolerância a ser utilizada

Para determinar a tolerância a ser empregada nas simulações utilizou-se a malha 1025×1025 . O processo iterativo foi executado até que o erro de máquina fosse atingido, isto é, até que o erro de iteração atingisse o mesmo nível do erro de arredondamento. A Figura 21 apresenta o comportamento da norma l_∞ do resíduo nessa situação. Observa-se que a norma começa a decrescer nos primeiros ciclos e começa a estabilizar entre 10^{-11} e 10^{-12} .

Segundo roteiro de simulações apresentado em Oliveira (2010), 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_∞ , arredondando-se para cima. Neste caso, como $l_\infty = 10^{-12}$, o valor mínimo a empregar para a tolerância seria 10^{-11} , que arredondando fica em 10^{-10} , justificando a tolerância indicada ao final da seção anterior.

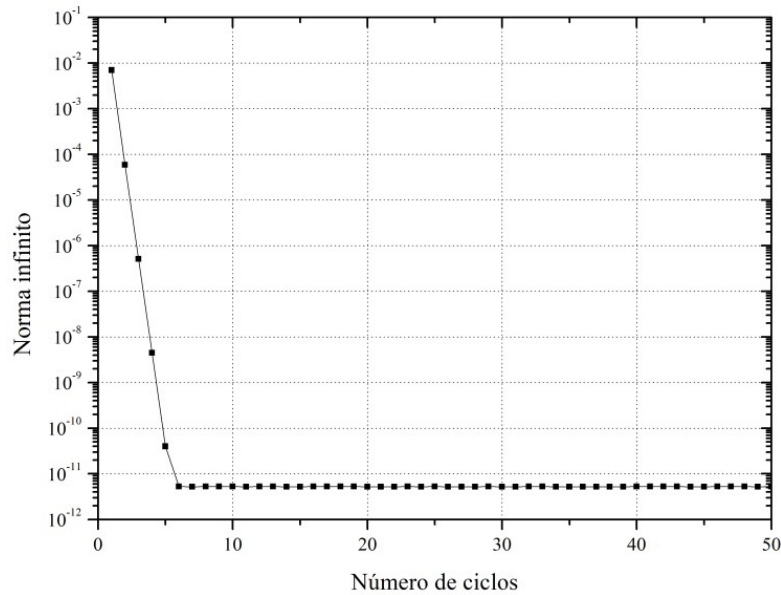


Figura 21: Comportamento da norma l_∞ em função do número de ciclos V para uma malha 1025×1025 .

Para todas as simulações utiliza-se aritmética de precisão dupla. A precisão quádrupla foi utilizada apenas para a verificação do código computacional. Os códigos numéricos foram implementados usando a linguagem Fortran 2003, com o aplicativo Intel 9.1 Visual Fortran. Todos os resultados numéricos foram obtidos em um computador com processador Intel Core i7, 2.66 GHz, 16 GB RAM e Sistema Operacional Windows 10, 64 bits.

6. VERIFICAÇÃO NUMÉRICA DO CÓDIGO COMPUTACIONAL

Para a verificação do código computacional, calculou-se o erro numérico (E), ou seja, a diferença entre a solução exata (Φ) de uma variável de interesse e a solução numérica (ϕ), isto é, (FERZIGER e PERIC, 2002)

$$E(\phi) = \Phi - \phi. \quad (154)$$

Para calcular as ordens aparente (p_U) e efetiva (p_E) do erro de discretização de cada variável de interesse, foi utilizado o procedimento recomendado por Marchi (2001).

Para analisar se a ordem aparente (p_U) se aproxima da ordem assintótica (p_L), foram obtidas três soluções numéricas (ϕ_1, ϕ_2, ϕ_3) para três malhas diferentes (h_1, h_2 e h_3 , considerando $h_x = h_y$ para estas três malhas) e calculado o valor da ordem aparente a partir da Eq.(155)

$$p_U = \frac{\log\left(\frac{\phi_2 - \phi_3}{\phi_1 - \phi_2}\right)}{\log(R_r)}, \quad (155)$$

onde $R_r = h_2/h_1$ é a razão de refino da malha, ϕ_3 é a malha mais refinada e ϕ_1 a malha mais grosseira.

Para a obtenção da ordem efetiva (p_E), calculou-se o erro numérico em duas malhas ($E(\phi_1), E(\phi_2)$). A ordem efetiva, que é função do erro numérico das variáveis de interesse, foi calculada a partir da Eq.(156)

$$p_E = \frac{\log\left(\frac{E(\phi_2)}{E(\phi_1)}\right)}{\log(R_r)}. \quad (156)$$

Segundo Marchi (2001), $p_U, p_E \rightarrow p_L$ para $h \rightarrow 0$.

Para analisar a ordem efetiva, a ordem aparente e o erro numérico, foram consideradas algumas variáveis de interesse, como por exemplo: norma infinito do erro numérico e calculada por $\|T_a - T\|_\infty$ (T_a é a temperatura analítica e T a temperatura numérica); erro na temperatura no ponto central $T(0,5; 0,5)$ e erro na temperatura média (T_m) calculada pela regra dos trapézios. São apresentados a seguir, alguns testes realizados para anisotropia difusiva. As equações utilizadas para realizar os testes são dadas pelas Eqs. (148) e (149). Nota-se que neste caso, $p_L = 2$, pois se usa o esquema CDS.

Foram escolhidos alguns dados fixos para fazer a verificação do código computacional. São eles: esquema CS, ciclo V, estimativa inicial nula, engrossamento padrão, restrição por ponderação completa e prolongação por interpolação bilinear. Utilizou-se aritmética de precisão quádrupla e o programa foi executado até o erro de máquina. Nas próximas seções são apresentados os resultados para os *solvers* GS-Lex e ILU. Os resultados para os demais *solvers* são semelhantes.

6.1 Solver GS-Lex

Os sistemas de equações obtidos da discretização foram resolvidos utilizando-se o *solver* GS-Lex. Os resultados podem ser observados a seguir.

A Figura 22 apresenta o erro numérico para a norma l_∞ do erro versus h considerando diferentes coeficientes de anisotropia (ε). Considera-se também o método *singlegrid*, quando o problema é resolvido em apenas uma malha. Observa-se que à medida que a malha torna-se mais refinada, utilizando *multigrid* ou *singlegrid*, o erro numérico tende a zero independente do coeficiente de anisotropia.

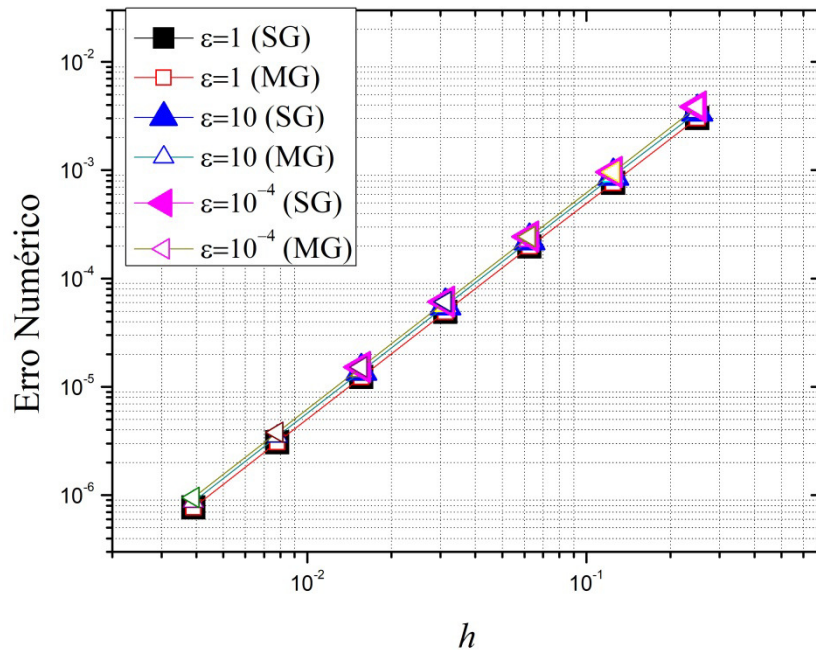


Figura 22: Erro numérico para a norma l_∞ .

A Figura 23 apresenta as ordens efetiva e aparente para a norma l_∞ do erro, erro na temperatura no ponto central e erro na temperatura média, considerando diferentes coeficientes de anisotropia.

Como o *solver* GS não apresenta bom desempenho para problemas anisotrópicos, o tempo de CPU é muito elevado, portanto, a malha mais fina considerada para as simulações foi no máximo $N=16641$ pontos. Por exemplo, para calcular a ordem aparente da norma l_∞ utilizou-se a malha mais fina com $N=4225$ pontos para $\epsilon=10$ e $N=1089$ pontos para $\epsilon=1$ e $\epsilon=0,0001$. Pode-se observar que, em todos os casos, $p_U \rightarrow p_L$ e $p_E \rightarrow p_L$.

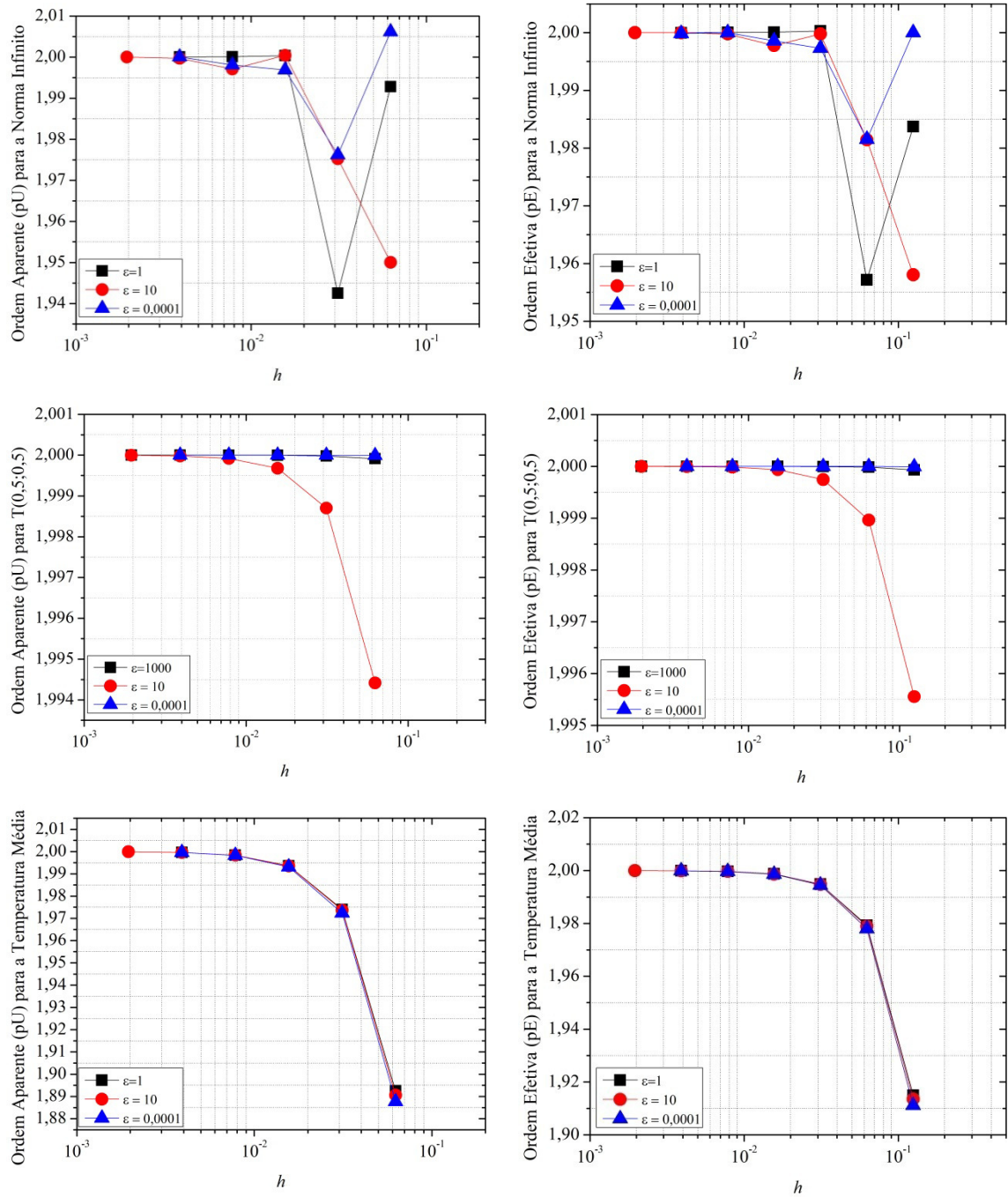


Figura 23: Ordem efetiva e aparente para a norma l_∞ do erro, erro na temperatura no ponto central e erro na temperatura média com o *solver* GS-Lex.

6.2 Solver ILU

Os sistemas de equações obtidos da discretização foram resolvidos utilizando o *solver*ILU. Os resultados podem ser observados a seguir.

A Figura 24 apresenta o erro numérico para a norma l_∞ do erro numérico *versus* h considerando diferentes coeficientes de anisotropia. Considera-se também o método *singlegrid*. Observa-se que à medida que a malha torna-se mais refinada, utilizando *multigrid* ou *singlegrid*, o erro numérico tende a zero independente do fator de anisotropia.

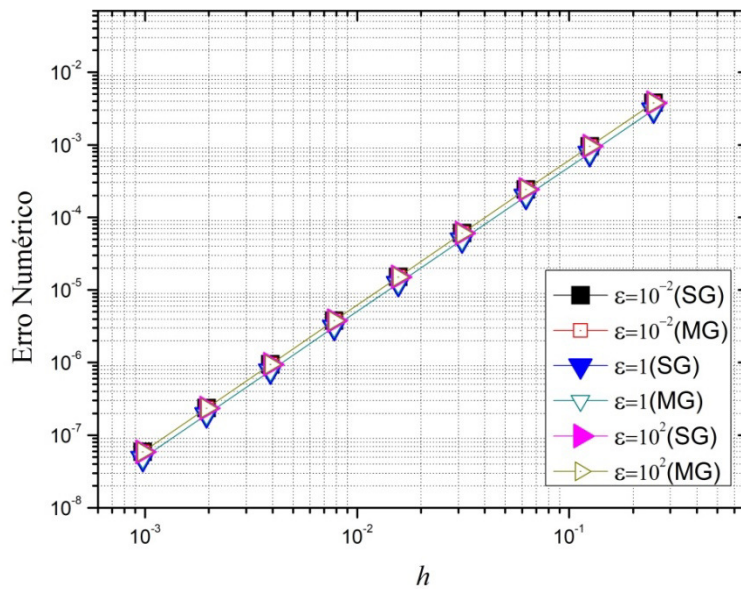


Figura 24: Erro numérico para a norma l_∞ para SG e MG.

A Figura 25 apresenta a norma l_∞ do erro numérico e erro na temperatura no ponto central *versus* h considerando diferentes coeficientes de anisotropia. Observa-se que à medida que a malha torna-se mais refinada, independente do fator de anisotropia e da variável de interesse, o erro numérico tende a zero.

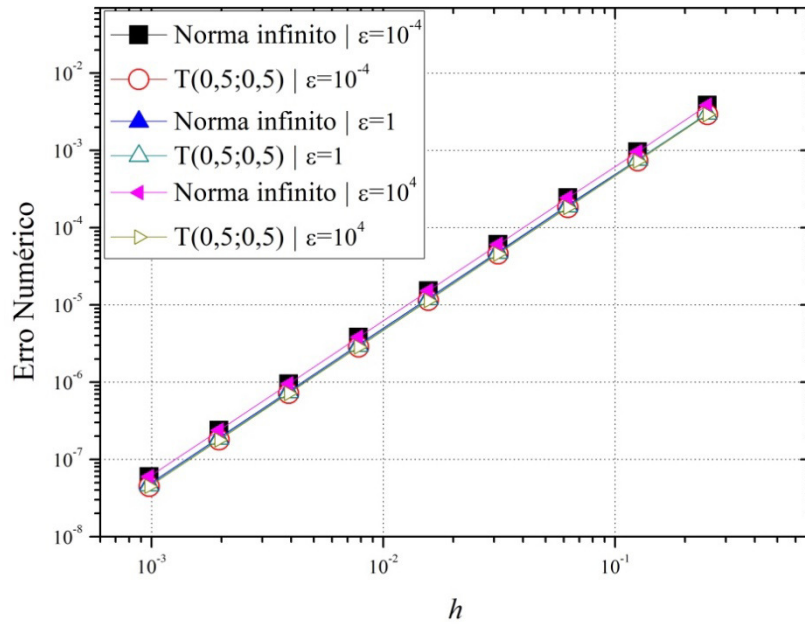


Figura 25: Erro numérico para a norma l_∞ e T(0,5;0,5).

A Figura 26 apresenta as ordens efetiva e aparente para a norma l_∞ do erro e temperatura no ponto central considerando diferentes coeficientes de anisotropia. Pode-se observar que, para todos os casos, $p_U \rightarrow p_L$ e $p_E \rightarrow p_L$.

A partir dos testes apresentados neste capítulo, pode-se concluir que os programas utilizados foram verificados e estão confiáveis. No próximo capítulo, esta conclusão será corroborada com resultados preliminares que estarão de acordo com a literatura.

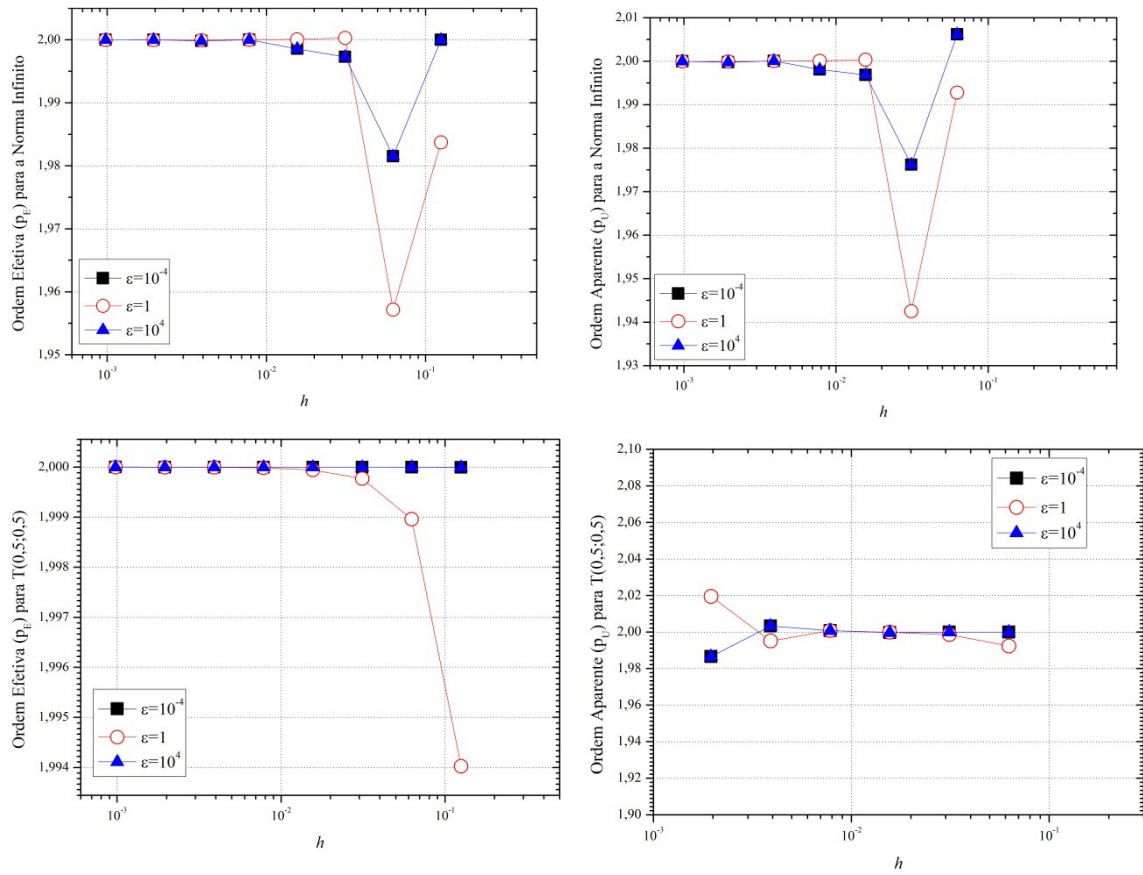


Figura 26: Ordem efetiva e aparente para a norma l_∞ do erro e temperatura no ponto central empregando o *solver* ILU.

Na Figura 26a ordem aparente para a temperatura no ponto central empregando o *solver* ILU possivelmente tem influência de erro de arredondamento mas podemos ver que a tendência para 2 é mantida, conforme as outras variáveis na mesma figura.

7.RESULTADOS PRELIMINARES

Neste capítulo são apresentados os resultados preliminares, que servirão como apoio para o desenvolvimento do capítulo 8. São apresentadas aqui análise de suavização e análise de duas malhas para o método *multigrid*. Foram utilizados os seguintes *solvers*:

- **Gauss-Seidel por pontos:** com ordenação *red-black* (GS-RB) e com ordenação lexicográfica (GS-Lex).
- **Gauss-Seidel por linhas:** ordenação lexicográfica (y-linha-GS, x-linha-GS e alt-linha-GS) e com ordenação zebra (y-zebra-GS, x-zebra-GS e alt-zebra-GS).
- **Decomposição LU incompleta:** ILU 5 pontos e ILU 7 pontos.

A Eq. (148) foi analisada para $\varepsilon = 10^\kappa$ e $\varepsilon = 10^{-\kappa}$, com $\kappa \in K = \{0, 1, 2, 3, 4, 5, 6, 7\}$. Neste trabalho $\varepsilon = 10^\kappa$ ou $\varepsilon = 10^{-\kappa}$ serão denominados de fatores de anisotropia simétricos. Por exemplo, $\varepsilon = 10^2$ é um fator de anisotropia simétrico a $\varepsilon = 10^{-2}$.

Os resultados apresentados daqui em diante considerarão $\nu = \nu_1 + \nu_2$, onde ν_1 é o número de pré-suavização e ν_2 é o número de pós-suavização conforme mostrado do capítulo 4.

7.1 Fator de suavização (μ_{loc})

Nesta seção são apresentando os resultados referentes ao fator de suavização para os *solvers* estudados. Para o cálculo do fator de suavização foi utilizada a Eq. (96). O melhor fator de suavização é o que está mais próximo de zero.

7.1.1 Fator de suavização para os *solvers* do tipo Gauss-Seidel

A Figura 27 apresenta μ_{loc} utilizando os *solvers* GS-RB e GS-Lex com $\nu = 1$.

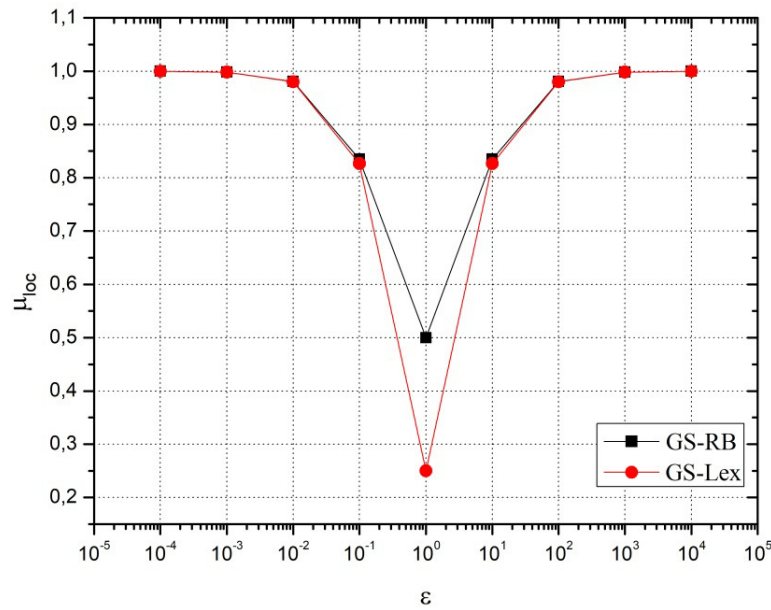


Figura 27: μ_{loc} versus ε para *solvers* ponto com ordenação lexicográfica e *red-black*.

Observa-se que para $\varepsilon=1$ (problema isotrópico), $\mu_{loc}(\text{GS-RB}) < \mu_{loc}(\text{GS-Lex})$. Observe-se também que para problemas anisotrópicos, $0 < \varepsilon \ll 1$ e $\varepsilon \gg 1$, $\mu_{loc}(\text{GS-RB}) \approx \mu_{loc}(\text{GS-Lex})$ com $\mu_{loc}(\text{GS-RB}) \rightarrow 1$ e $\mu_{loc}(\text{GS-Lex}) \rightarrow 1$ (utilizando-se a Eq. (96)), ou seja, estes *solvers* não apresentam bom fator de suavização.

A Tabela 6 apresenta uma comparação dos resultados obtidos neste trabalho para o fator de suavização do *solver* GS-RB, com os resultados apresentados por Wienands e Joppich (2005). Neste caso, tal coluna estará identificada com Wien.

Tabela 6: Fator de suavização utilizando GS-RB obtidos de Wienands e Joppich (2005), pg. 65, $N=128$, $\nu=1$.

ε	μ_{loc} (Wien)	μ_{loc}
10^{-2}	0,980	0,980296
10^{-1}	0,826	0,8264463
1	0,250	0,25

10	0,826	0,8264463
10^2	0,980	0,980296

A Figura 28 apresenta μ_{loc} utilizando os *solvers* x-linha-GS, y-linha-GS e alt-linha-GS. Para fins de comparação, considera-se: $\nu=1$ para o *solver* alt-linha-GS; $\nu=2$ para os *solvers* x-linha-GS e y-linha-GS. Isso porque, enquanto os *solvers* x-linha-GS e y-linha-GS suavizam apenas uma vez em $\nu=1$ (uma iteração interna) o *solver* alt-linha-GS suaviza duas vezes em $\nu=1$ (uma vez na direção de x e uma vez na direção de y).

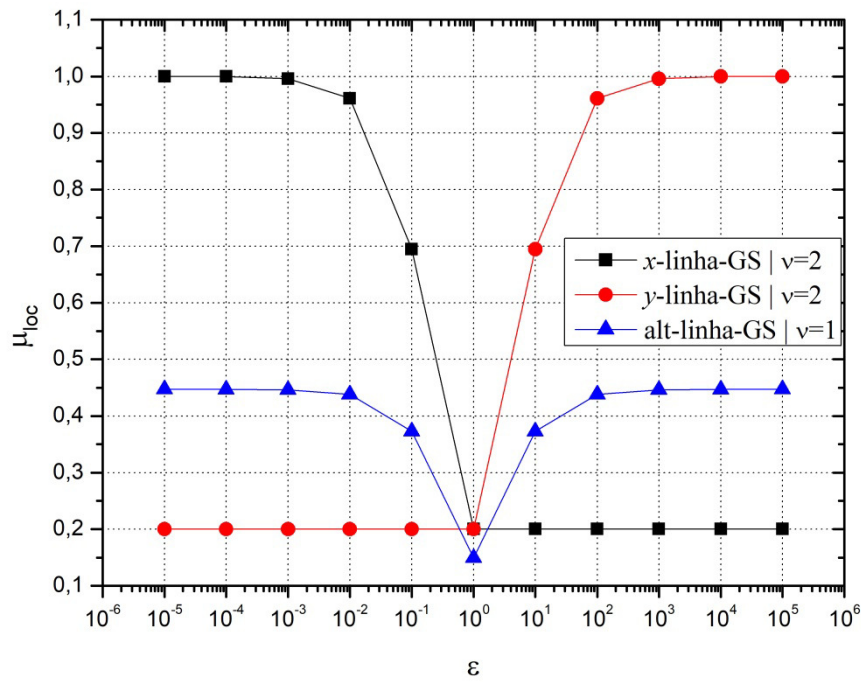


Figura 28: μ_{loc} versus ε para x-linha-GS, y-linha-GS e alt-linha-GS.

Dentre os *solvers* linha com ordenação lexicográfica analisados, observa-se que para $\varepsilon=1$, os três *solvers* apresentam fator de suavização semelhante. Para $0 < \varepsilon \ll 1$ o menor fator de suavização é μ_{loc} (y-linha-GS), no entanto para $\varepsilon \gg 1$ μ_{loc} (y-linha-GS) é muito grande. Para $\varepsilon \gg 1$ o menor fator de suavização é μ_{loc} (x-linha-GS), no entanto, para $0 < \varepsilon \ll 1$ μ_{loc} (x-linha-GS) é muito grande.

Este fato ocorre, pois para $\varepsilon \gg 1$, pela Eq.(148), tem-se forte acoplamento na direção x , portanto, *solvers* linha que privilegiam essa direção têm um melhor desempenho para acoplamentos nesta direção, conforme explicado na seção 3.5.

A Figura 29 apresenta μ_{loc} utilizando x -zebra-GS, y -zebra-GS e alt-zebra-GS. Para fins de comparação, considera-se: $\nu=1$ para o *solver* alt-zebra-GS; $\nu=2$ para os *solvers* x -zebra-GS e y -zebra-GS.

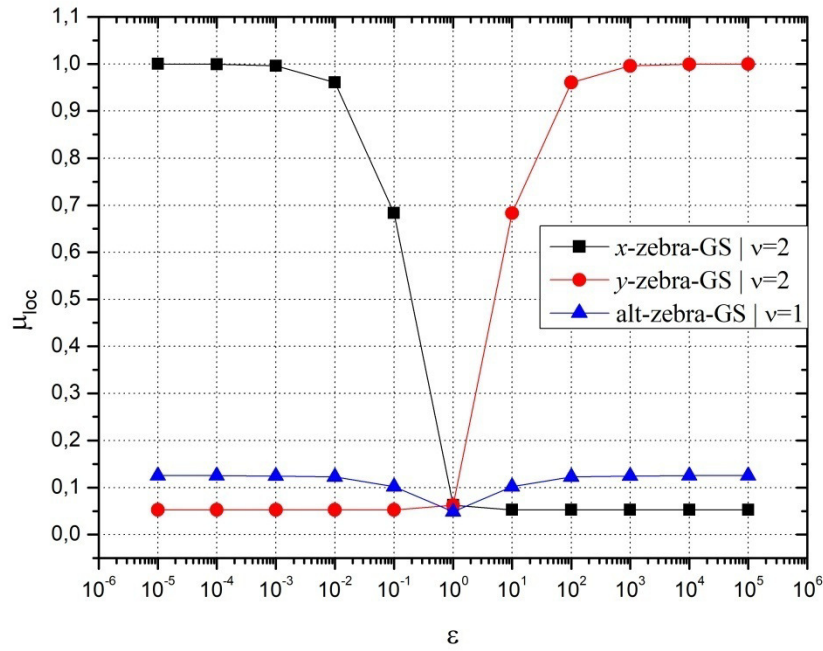


Figura 29: μ_{loc} versus ε para x -zebra-GS, y -zebra-GS e alt-zebra-GS.

Dentre os *solvers* linha com ordenação zebra analisados, observa-se que para $\varepsilon=1$, os três *solvers* apresentam fator de suavização semelhante. Para $0 < \varepsilon < 1$ o menor fator de suavização é $\mu_{loc}(y\text{-zebra-GS})$, no entanto, para $\varepsilon \gg 1$ $\mu_{loc}(y\text{-zebra-GS})$ é muito grande. Para $\varepsilon \gg 1$ o menor fator de suavização é $\mu_{loc}(x\text{-zebra-GS})$, no entanto, para $0 < \varepsilon < 1$ $\mu_{loc}(x\text{-zebra-GS})$ é muito grande.

Assim como ocorreu com os *solvers* linha com ordenação lexicográfica, para $\varepsilon \gg 1$, pela Eq.(148), tem-se forte acoplamento na direção x , portanto, *solvers* com ordenação zebra que privilegiam essa direção têm um melhor desempenho para acoplamentos nessa direção.

As Tabela 67 e 8 apresentam uma comparação dos resultados obtidos neste trabalho para o fator de suavização dos *solvers* tipo zebra-GS, com os resultados apresentados por Wienands e Joppich (2005). Neste caso, tal coluna estará identificada com Wien.

Tabela 7: Fator de suavização utilizando x-zebra-GS e y-zebra-GS obtidos de Wienands e Joppich (2005), pg. 65, $N=128, \nu=1$.

ε	x-zebra-GS	x-zebra-GS	y-zebra-GS	y-zebra-GS
	μ_{loc} (Wien)	μ_{loc}	μ_{loc} (Wien)	μ_{loc}
10^{-2}	0,980	0,980296	0,125	0,125
10^{-1}	0,826	0,8264463	0,125	0,125
1	0,250	0,25	0,250	0,25
10	0,125	0,125	0,826	0,8264463
10^2	0,125	0,125	0,980	0,980296

Tabela 8: Fator de suavização utilizando alt-zebra-GS obtidos de Wienands e Joppich (2005), pg. 65, $N=128, \nu=1$.

ε	μ_{loc}	μ_{loc}
	(Wien)	
10^{-2}	0,122	0,1222431
10^{-1}	0,102	0,1018308
1	0,048	0,048268
10	0,102	0,1018308
10^2	0,122	0,1222431

A Figura 30 apresenta μ_{loc} utilizando os *solvers* que apresentaram os melhores fatores de suavização (para pelo menos uma das direções), dados anteriormente. Lembrando que melhor aqui, significa mais próximos de zero.

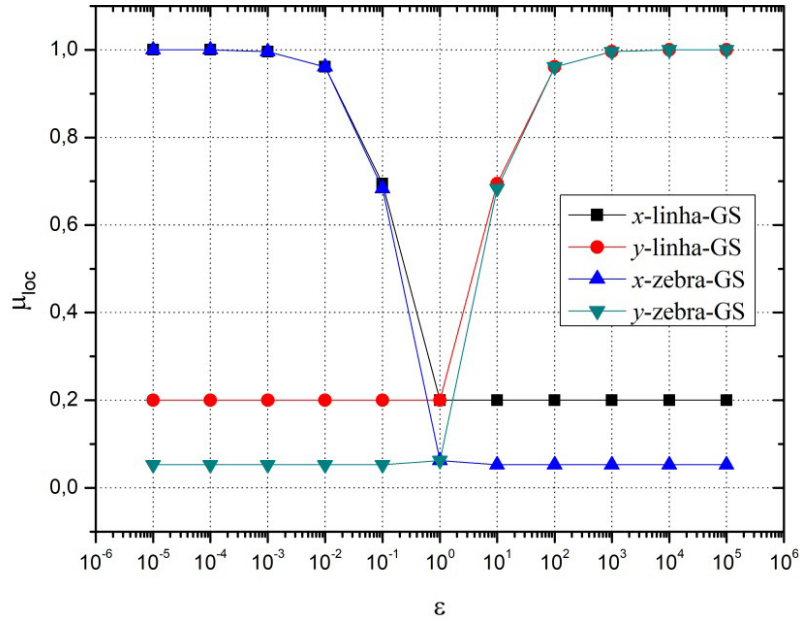


Figura 30: μ_{loc} versus ε para os *solvers* linha com ordenação lexicográfica e zebra com $\nu=2$.

Observa-se que para $\varepsilon \gg 1$, *x-zebra-GS* e *x-linha-GS* obtiveram os melhores resultados, sendo $\mu_{loc}(x\text{-zebra-GS}) < \mu_{loc}(x\text{-linha-GS})$ e para $0 < \varepsilon \ll 1$, *y-zebra-GS* e *y-linha-GS* obtiveram os melhores resultados, sendo $\mu_{loc}(y\text{-zebra-GS}) < \mu_{loc}(y\text{-linha-GS})$.

Nesta seção foram apresentando os resultados para o fator de suavização dos *solvers* do tipo GS. Pode-se notar que zebra-GS apresentam os menores valores para μ_{loc} . Todos os resultados apresentados estão de acordo com a literatura corroborando com o capítulo 5 de verificação do código computacional.

7.1.2 Fator de suavização do *solver* tipo ILU

Nesta seção serão realizados testes com o ILU e MILU 5 pontos e 7 pontos. No caso do ILU, tem-se $\sigma=0$, ou seja, diagonal D sem contribuição dos valores

descartados (veja seções 3.2.2 e 3.2.3). No caso do MILU, será usado $\sigma = 0,5$. Testes para outros valores de σ serão realizados no capítulo 8.

A Figura 31 apresenta o fator de suavização (μ_{loc}) utilizando o *solver* ILU com as seguintes ordenações: EN=WS, NE=SW, ES=WN, SE=NW; $\varepsilon = 10^{\kappa}$ e $\varepsilon = 10^{-\kappa}$, com $\kappa \in \{0, 1, 2, 3, 4, 5\}$.

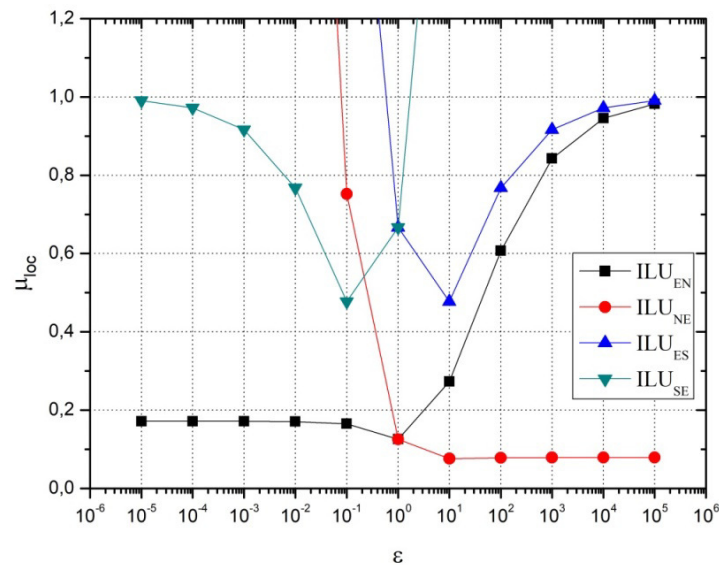


Figura 31: μ_{loc} para o *solver* ILU com diferentes ordenações.

Observa-se que, para $0 < \varepsilon \ll 1$ a melhor ordenação para o *solver* ILU 7 pontos é EN=WS. Para $\varepsilon \gg 1$, a melhor ordenação para o *solver* ILU é NE=SW.

Da Tabela 9 à Tabela 11 apresenta-se uma comparação dos resultados obtidos neste trabalho para o fator de suavização dos *solvers* tipo ILU, com os resultados apresentados por Trottenberg et al. (2001) ou Wesseling (1992). Neste caso, as colunas estarão identificadas com Trott ou Wess, respectivamente.

Tabela 9: Fator de suavização utilizando ILU e MILU 5 pontos obtidos de Wesseling (1992), pg. 135.

$\sigma = 0$	$\sigma = 0,5$
--------------	----------------

ε	μ_{loc} (Wess)	μ_{loc}	μ_{loc} (Wess)	μ_{loc}
1	0,20	0,203517	0,20	0,200214
10^{-1}	0,48	0,477458	0,26	0,261964
10^{-2}	0,77	0,767552	0,30	0,300974
10^{-3}	0,92	0,916218	0,32	0,321384
10^{-5}	0,99	0,991115	0,33	0,332118

Tabela 10: Fator de suavização utilizando ILU e MILU 7 pontos obtidos de Wesseling (1992), pg. 139.

ε	$\sigma = 0$		$\sigma = 0,5$	
	μ_{loc} (Wess)	μ_{loc}	μ_{loc} (Wess)	μ_{loc}
1	0,13	0,125886	0,11	0,109968
10^{-1}	0,17	0,165101	0,089	0,088466
10^{-2}	0,17	0,170896	0,091	0,091011
10^{-3}	0,17	0,171505	0,091	0,091277
10^{-5}	0,17	0,17156	0,086	0,091298

Tabela 11: Fator de suavização utilizando ILU 7 pontos obtidos de Trottenberg (2001), pg. 261.

ε	μ_{loc} (Trott)	μ_{loc}
10^{-4}	0,17	0,1715649
10^{-3}	0,17	0,1715048
10^{-2}	0,17	0,1708961

10^{-1}	0,17	0,1651007
1	0,13	0,1258859
10	0,27	0,2733908
10^2	0,61	0,6070012
10^3	0,84	0,8425066
10^4	0,95	0,9457425

Pode-se notar que os fatores de suavização para os *solvers* tipo ILU estão em acordo com os valores disponíveis na literatura corroborando com o capítulo 5 de verificação do código computacional.

A seguir, será realizada uma comparação entre os fatores de suavização dos *solvers* que obtiveram o melhor desempenho, dentre GS e ILU.

A Figura 32 apresenta uma comparação entre μ_{loc} usando os *solvers* zebra-GS e os *solvers* ILU.

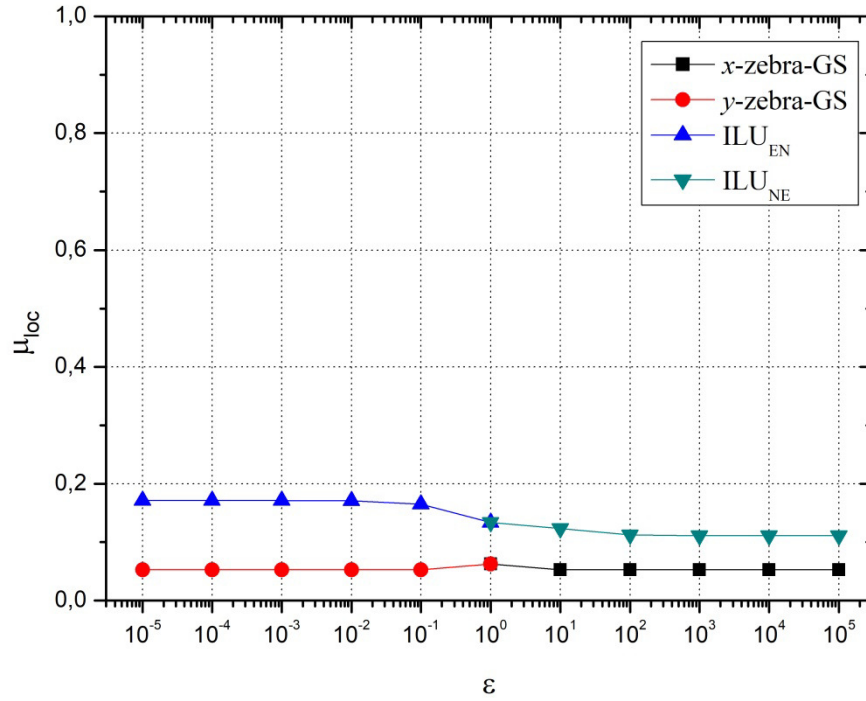


Figura 32: μ_{loc} versus ε para os solvers zebra-GS e ILU com $\nu=2$.

Observa-se que os solvers zebra-GS apresentam fator de suavização melhor, em comparação com os solvers ILU, no entanto, estes solvers quando usados como suavizadores no *multigrid* apresentam o fator de convergência assintótica ρ_{loc} utilizando-se o solver ILU muito melhor (muito mais próximo de zero) do que utilizando-se zebra-GS. Dessa forma, conclui-se que o solver ILU possui muito mais as propriedades de suavização que o solver zebra. Estes resultados serão apresentados na próxima seção.

7.2 Fator de convergência assintótica (ρ_{loc})

Nesta seção são apresentando os resultados referentes ao fator de convergência assintótica do método *multigrid*, calculado utilizando-se a Eq. (119). Para a obtenção de ρ_{loc} (daqui em diante por facilidade, será utilizada a notação $\rho_{loc}(M_h^{2h}) = \rho_{loc}$). Além dos solvers citados na seção 6.1 foram empregados os operadores de restrição por

ponderação completa (FW), injeção (INJ) e meia-ponderação (HW). Os operadores de prolongação empregados são bilinear e 7-pontos.

7.2.1 Fator de convergência assintótica utilizando os *solvers* do tipo Gauss-Seidel

A Figura 33 apresenta ρ_{loc} com os *solvers* GS-RB e GS-Lex; número de iterações internas $\nu=1$; restrição: FW, HW e INJ; prolongação: bilinear; $\mathcal{E}=10^\kappa$ e $\mathcal{E}=10^{-\kappa}$, com $\kappa \in \{0,1,2,3,4,5\}$.

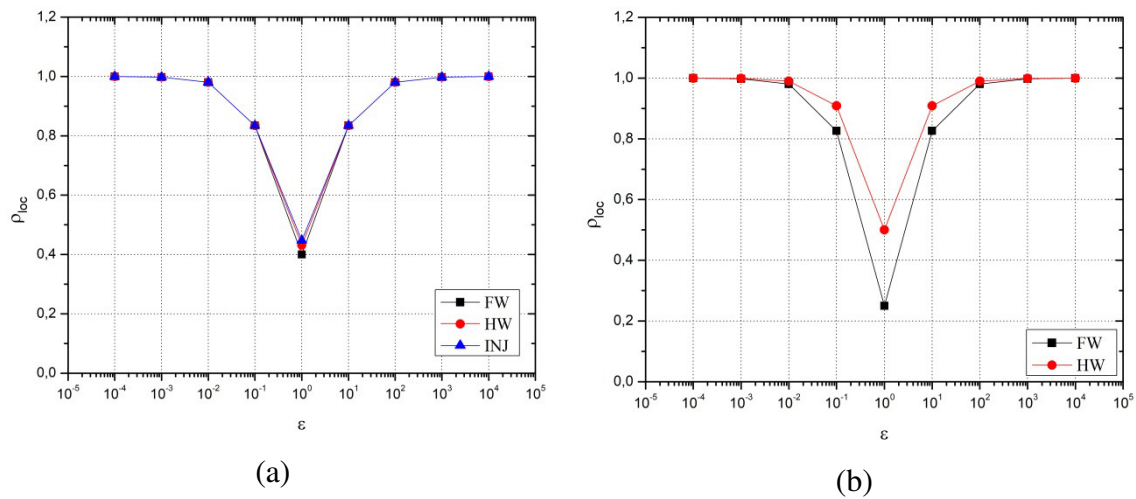


Figura 33: ρ_{loc} versus ϵ com os *solvers* (a) GS-Lex; (b) GS-RB.

Observa-se que o menor fator de convergência assintótica ocorre quando utiliza-se o *solver* GS-RB e restrição por ponderação completa (FW) e $\epsilon=1$. Para $\epsilon=10^{-\kappa}$ e $\epsilon=10^\kappa$, (anisotropias simétricas) ρ_{loc} apresenta o mesmo comportamento. Observa-se ainda que, para fortes anisotropias, ρ_{loc} está se aproximando de 1, ou seja, o *multigrid* começa a divergir.

Na Tabela 12 apresenta-se uma comparação dos resultados obtidos neste trabalho para o fator de convergência assintótica do *solver* GS-RB junto ao método *multigrid*, com os resultados apresentados por Trottenberg et al. (2001). Neste caso, tal coluna estará identificada com Trott.

Tabela 12: Fator de convergência assintótica utilizando GS-RB obtidos de Trottenberg (2001), pg. 133. Restrição: FW, Prolongação: Bilinear, $N = 128$, $\nu = 3$.

ε	ρ_{loc} (Trott)	ρ_{loc}
10^{-3}	0,99	0,9940209
10^{-2}	0,94	0,9420452
10^{-1}	0,56	0,5644739
1	0,053	0,05273436
10	0,56	0,5644739
10^2	0,94	0,9420452
10^3	0,99	0,9940209

A Figura 34 apresenta ρ_{loc} utilizando os *solvers* x-linha-GS e y-linha-GS com $\nu = 2$; alt-linha-GS com $\nu = 1$; restrição: FW; prolongação: bilinear e 7-pontos; $\varepsilon = 10^\kappa$ e $\varepsilon = 10^{-\kappa}$, com $\kappa \in \{0, 1, 2, 3, 4, 5\}$. Nota-se que os parâmetros de restrição foram fixados para que se pudesse avaliar o efeito da prolongação.

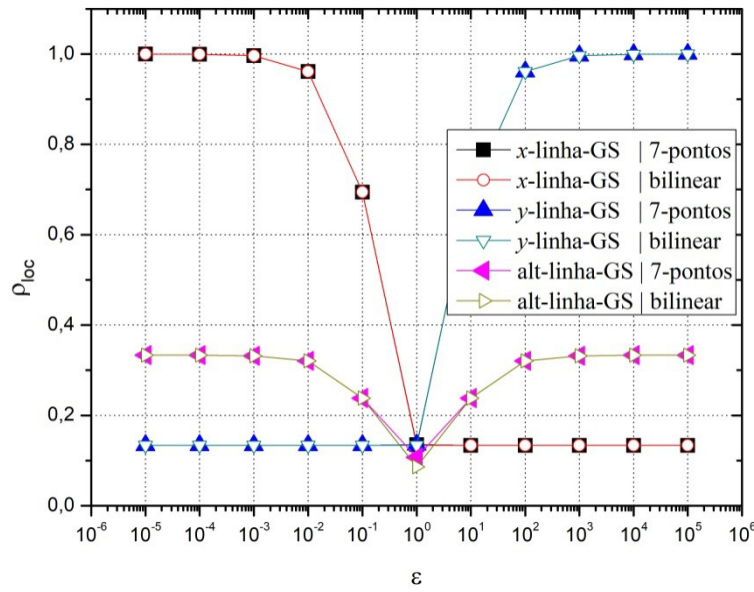


Figura 34: ρ_{loc} versus ε utilizando os *solvers* linha-GS com diferentes operadores de prolongação.

Observa-se que para $0 < \varepsilon \ll 1$ o menor valor para ρ_{loc} ocorre utilizando y-linha-GS e para $\varepsilon \gg 1$ o menor valor para ρ_{loc} ocorre utilizando x-linha-GS. Observa-se também que é indiferente utilizar o operador de prolongação 7-pontos ou bilinear.

A Figura 35 apresenta ρ_{loc} utilizando os *solvers* x-linha-GS e y-linha-GS; $\nu = 2$; restrição: FW, HW e INJ; prolongação: bilinear; $\varepsilon = 10^\kappa$ e $\varepsilon = 10^{-\kappa}$, com $\kappa \in \{0, 1, 2, 3, 4, 5\}$. Nota-se que se fixam os parâmetros da prolongação para se avaliar o efeito da restrição.

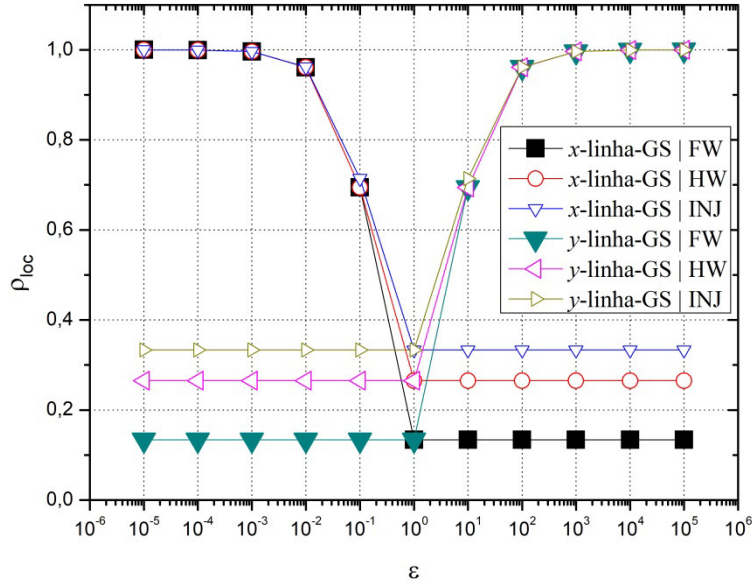


Figura 35: ρ_{loc} versus ε utilizando os *solvers* linha-GS com diferentes operadores de restrição.

Observa-se que o melhor operador de restrição é o FW, pois apresenta menor valor para ρ_{loc} , para todos os coeficientes de anisotropia analisados. Esse operador apresenta melhor desempenho, pois incorpora mais informações no processo de restrição, conforme pode ser observado na Eq.(55). Para $0 < \varepsilon \ll 1$ a combinação *y*-linha-GS | FW apresenta $\rho_{loc} \ll 1$ e para $\varepsilon \gg 1$ a combinação *x*-linha-GS | FW apresenta $\rho_{loc} \ll 1$.

Como já apresentado na seção anterior, para $\varepsilon \gg 1$, pela Eq.(148), tem-se um forte acoplamento na direção *x*, portanto, *solvers* que privilegiam essa direção têm um melhor desempenho para acoplamentos nessa direção, e isto independe do operador de prolongação ou restrição empregado.

A Figura 36 apresenta ρ_{loc} utilizando-se os *solvers* *x*-zebra-GS, *y*-zebra-GS com $\nu=2$; *alt*-zebra-GS com $\nu=1$; restrição FW; prolongação: bilinear e 7 pontos; $\varepsilon=10^\kappa$ e $\varepsilon=10^{-\kappa}$, com $\kappa \in \{0,1,2,3,4,5\}$. O operador de restrição está fixo para avaliar o desempenho dos operadores de prolongação.

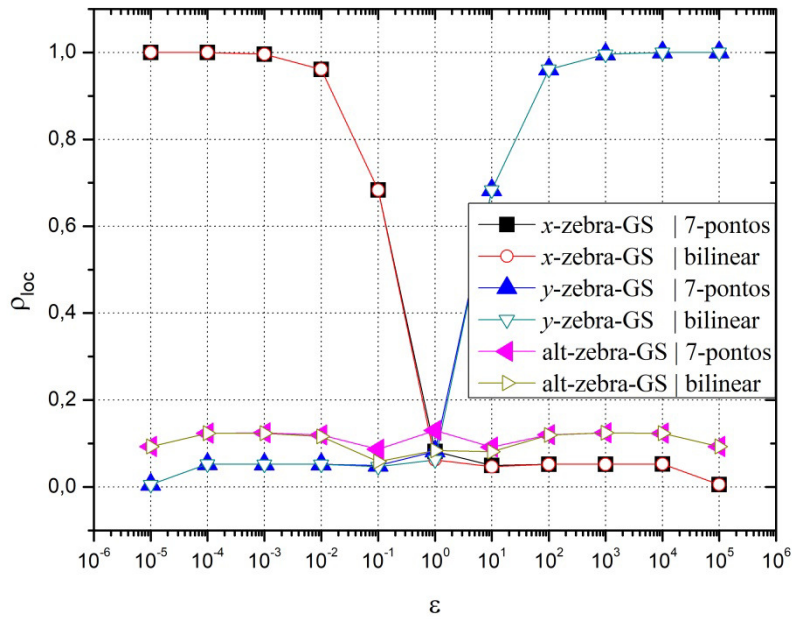


Figura 36: ρ_{loc} versus ε para os *solvers* zebra-GS e diferentes operadores de prolongação.

Observa-se que para $0 < \varepsilon \ll 1$ o menor valor para ρ_{loc} ocorre utilizando y-zebra-GS e para $\varepsilon \gg 1$ o menor valor para ρ_{loc} ocorre utilizando x-zebra-GS. Observa-se também que, como já foi visto anteriormente nos *solvers* linha, é indiferente utilizar o operador de prolongação 7-pontos ou bilinear.

A Figura 37 apresenta ρ_{loc} utilizando os *solvers* x-zebra-GS e y-zebra-GS; $\nu = 2$; restrição: FW e HW; prolongação: bilinear; $\varepsilon = 10^\kappa$ e $\varepsilon = 10^{-\kappa}$, com $\kappa \in \{0, 1, 2, 3, 4, 5\}$. O operador de prolongação foi fixado para avaliar o desempenho dos operadores de restrição.

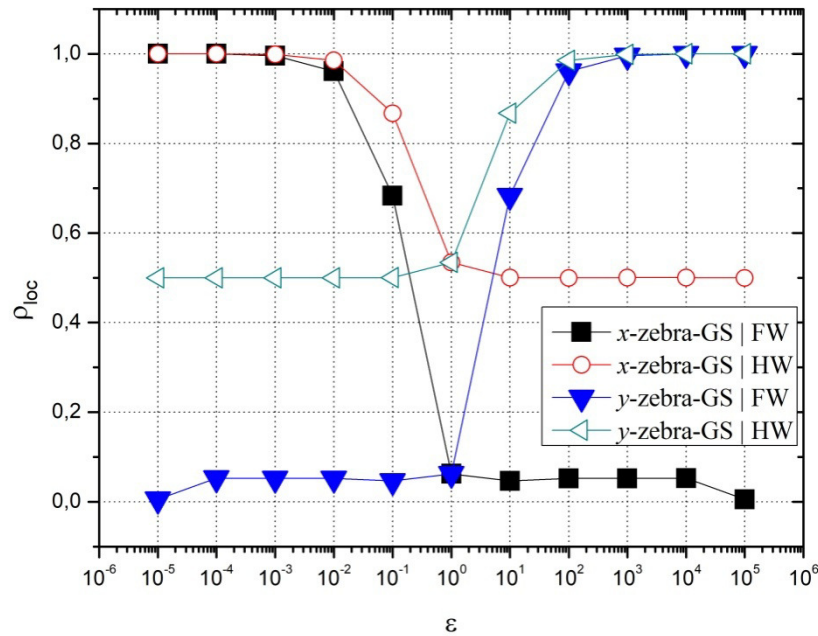


Figura 37: ρ_{loc} versus ε utilizando os solvers zebra-GS e diferentes operadores de restrição.

Observa-se que o melhor operador de restrição é o FW, pois apresenta menor valor para ρ_{loc} para todos os coeficientes de anisotropia analisados. Conforme apresentado na seção 7.1, esse operador apresenta melhor desempenho, pois incorpora mais informações no processo de restrição. Para $0 < \varepsilon < 1$ y-zebra-GS | FW apresenta ρ_{loc} pequeno e para $\varepsilon \gg 1$ x-zebra-GS | FW apresenta ρ_{loc} pequeno.

Este fato ocorre, pois para $\varepsilon \gg 1$, pela Eq.(148), tem-se forte acoplamento na direção x , portanto, solvers linha que privilegiam essa direção têm um melhor desempenho para acoplamentos nessa direção. Este fato já foi observado para o cálculo do fator de suavização, na seção anterior. Novamente, nesta seção, isto se confirma e, além disso, observa-se que é independente dos operadores de restrição e prolongação empregados.

Nas Tabelas 123 e 124 apresenta-se uma comparação dos resultados obtidos neste trabalho para o fator de convergência assintótica dos solvers tipo zebra-GS junto ao método *multigrid*, com os resultados apresentados por Trottenberg et al. (2001). Neste caso, tal coluna estará identificada com Trott.

Tabela 13: Fator de convergência assintótica utilizando x -zebra-GS e y -zebra-GS obtidos de Trottenberg (2001), pg. 137. Restrição: FW, Prolongação: Bilinear, $N=128$, $\nu=2$.

	x -zebra-GS	x -zebra-GS	y -zebra-GS	y -zebra-GS
ε	ρ_{loc} (Trott)	ρ_{loc}	ρ_{loc} (Trott)	ρ_{loc}
10^{-3}	0,996	0,99601	0,053	0,05222668
10^{-2}	0,96	0,9609803	0,052	0,05201303
10^{-1}	0,68	0,6830135	0,047	0,04659593
1	0,063	0,0625	0,063	0,0625
10	0,047	0,04659593	0,68	0,6830135
10^2	0,052	0,05201303	0,96	0,9609803
10^3	0,053	0,05222668	0,996	0,99601

Tabela 14: Fator de convergência assintótica utilizando alt -zebra-GS obtidos de Trottenberg (2002), pg. 138. Restrição: FW, Prolongação: Bilinear, $N=128$, $\nu=2$.

ε	ρ_{loc} (Trott)	ρ_{loc}
10^{-3}	0,053	0,051988
10^{-2}	0,051	0,049631
10^{-1}	0,038	0,02662
1	0,009	0,039171
10	0,038	0,03786949
10^2	0,051	0,05084
10^3	0,053	0,052108

A Figura 38 apresenta ρ_{loc} , utilizando os *solvers* tipo GS; restrição: FW; prolongação: bilinear; $\mathcal{E} = 10^\kappa$ e $\mathcal{E} = 10^{-\kappa}$, com $\kappa \in \{0, 1, 3, 5\}$. Os operadores de restrição e prolongação que obtiveram o melhor desempenho até então, serão fixados para comparar todos dos *solvers* do tipo GS que já foram avaliados.

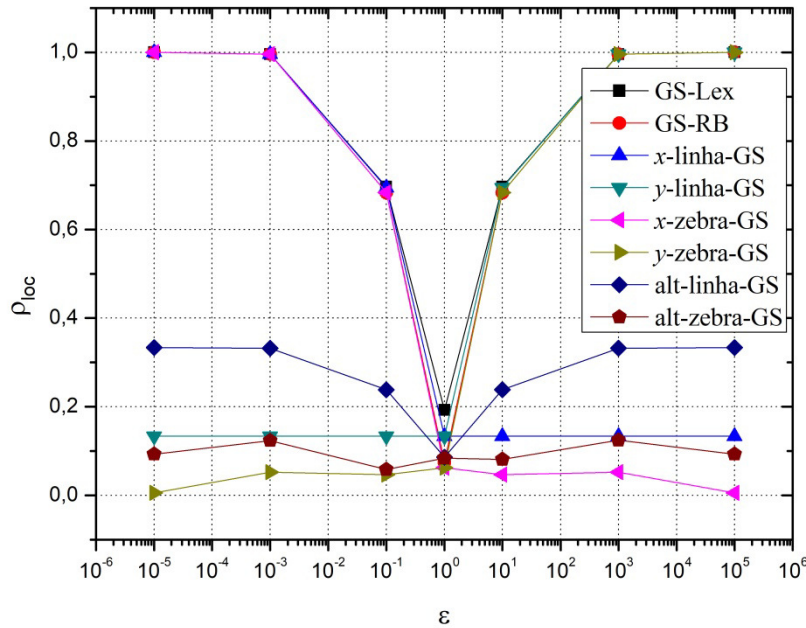


Figura 38: ρ_{loc} versus coeficiente de anisotropia \mathcal{E} com os *solver* tipo GS.

Para fins de comparação, considera-se $\nu=1$ para os *solvers* alt-linha-GS e alt-zebra-GS e $\nu=2$ para os demais *solvers*, isso porque, enquanto os demais *solvers* suavizam apenas uma vez em $\nu=1$ (uma iteração interna) os *solvers* alt-linha-GS e alt-zebra-GS suavizam duas vezes em $\nu=1$ (uma vez na direção de x e uma vez na direção de y). Comparando-se ρ_{loc} utilizando-se diferentes *solvers*, observa-se que para $\mathcal{E}=1$ (problema isotrópico) os menores fatores de suavização obtidos são $\rho_{loc}(\text{alt-zebra-GS})$, $\rho_{loc}(\text{x-zebra-GS})$, $\rho_{loc}(\text{y-zebra-GS})$ e $\rho_{loc}(\text{GS-RB})$.

Para $\varepsilon \gg 1$, os *solvers* que apresentam os melhores desempenhos ($\rho_{loc} \approx 0$), associados ao *multigrid* são alt-zebra-GS, x-zebra-GS e x-linha-GS. Observa-se ainda que $\rho_{loc}(x\text{-zebra-GS}) \ll \rho_{loc}(x\text{-linha-GS})$, $\rho_{loc}(x\text{-zebra-GS}) < \rho_{loc}(\text{alt-zebra-GS})$ e $\rho_{loc}(\text{alt-zebra-GS}) < \rho_{loc}(x\text{-linha-GS})$.

Para $0 < \varepsilon \ll 1$, os *solvers* que os melhores desempenhos ($\rho_{loc} \approx 0$), associados ao *multigrid* são alt-zebra-GS, y-zebra-GS e y-linha-GS. Observa-se ainda que $\rho_{loc}(y\text{-zebra-GS}) < \rho_{loc}(\text{alt-zebra-GS}) < \rho_{loc}(y\text{-linha-GS})$.

Diante dos resultados apresentados, conclui-se que o *solver* x-zebra-GS e y-zebra-GS são bons suavizadores e também apresentam um bom fator de convergência assintótica, quando acoplados ao *multigrid*.

Pode-se notar por estes resultados que os fatores de convergência assintótica empregando-se os *solvers* tipo GS estão de acordo com os valores disponíveis na literatura. Estes complementam os resultados apresentandos no capítulo 5 desta tese, em que foi apresentada a verificação do código computacional.

7.2.2 Fator de convergência assintótica utilizando os *solver* tipo ILU

Nesta seção serão apresentados os resultados utilizando ILU 7 pontos em contraponto ao ILU 5 pontos. Esse *solver* foi utilizado pelas boas propriedades de suavização, conforme pode ser comparado nas Tabs. Tabela 9 e Tabela 10. Além disso, esse *solver* é o mais indicado para problemas anisotrópicos com estêncil de 5 pontos (WESSELING, 1992; TROTTEMBERG et al., 2001; SAAD, 2003). Portanto, daqui para frente, para tratar do ILU 7 pontos, será usada simplesmente a notação ILU. A seguir serão avaliadas as diferentes direções de ILU.

A Figura 39 apresenta ρ_{loc} , utilizando ILU nas direções EN, NE, ES, SE, restrição por FW, prolongação bilinear, número de iterações internas $\nu = 2$, $\varepsilon = 10^\kappa$ e $\varepsilon = 10^{-\kappa}$, com $\kappa \in K$. Lembrando que, para a LFA serão considerados operadores discretos lineares gerais com coeficientes constantes, os quais são definidos em uma

malha **infinita**, onde a influência das fronteiras pode ser desprezada (TROTTENBERG et al., 2001; WESSELING, 1992).

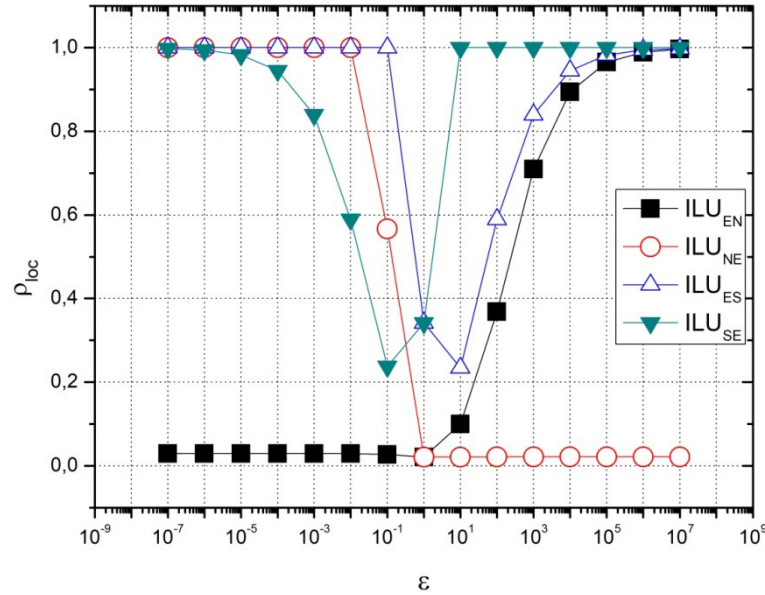


Figura 39: ρ_{loc} versus coeficiente de anisotropia ε com o *solver* ILU.

Observa-se que para $0 < \varepsilon \ll 1$, ILU_{EN} apresenta um bom desempenho ($\rho_{loc} \approx 0$) e para $\varepsilon \gg 1$, ILU_{NE} apresenta um bom desempenho. Ao se utilizar os *solvers* ILU nas direções ES e SE, o *multigrid* não apresentou melhor desempenho para nenhum dos coeficientes de anisotropia estudados.

Na Tabela 15 apresenta-se uma comparação dos resultados obtidos neste trabalho para o fator de convergência assintótica do *solver* ILU_{EN} , com os resultados apresentados por Trottenberg et al. (2001). Neste caso, tal coluna estará identificada com Trott.

Tabela 15: Fator de convergência assintótica utilizando ILU7 pontos na direção EN obtidos de Trottenberg (2001), pg. 261. Restrição: FW, Prolongação: Bilinear, $N = 128$, $\nu = 1$.

ε	ρ_{loc} (Trott)	ρ_{loc}
---------------	----------------------	--------------

10^{-4}	0,17	0,1715649
10^{-3}	0,17	0,1715048
10^{-2}	0,17	0,1708961
10^{-1}	0,17	0,1651007
1	0,13	0,1258859
10	0,27	0,2733908
10^2	0,61	0,6070012
10^3	0,84	0,8425066
10^4	0,94	0,9457425

Pode-se notar pelos resultados da Tabela 15 que os fatores de convergência assintótica empregando-se os *solvers* tipo ILU estão de acordo com os valores disponíveis na literatura. Estes complementam os resultados apresentandos no capítulo 5 desta tese, em que foi apresentada a verificação do código computacional.

A seguir, apresenta-se uma comparação entres os fatores de convergência assintótica empregando os *solvers* que obtiveram o melhor desempenho, dentre GS e ILU.

A Figura 40 apresenta ρ_{loc} , comparando os *solvers* ILU_{EN} , ILU_{NE} , x -zebra-GS e y -zebra-GS; restrição por FW, prolongação bilinear, número de iterações internas $v = 2$, $\varepsilon = 10^\kappa$ e $\varepsilon = 10^{-\kappa}$, com $\kappa \in \{0, 1, 2, 3, 4\}$.

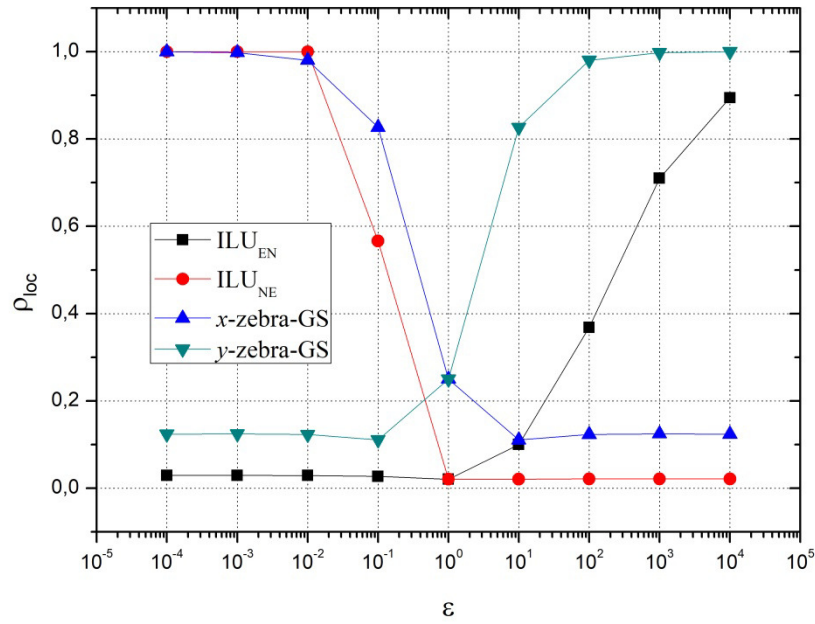


Figura 40: ρ_{loc} versus coeficiente de anisotropia ε comparando os melhores *solvers* tipo GS e os melhores *solvers* tipo ILU.

Observa-se que o *solver* ILU apresenta melhor desempenho (tem-se ρ_{loc} menor) em relação ao *solver* GS tanto para $0 < \varepsilon \ll 1$ como para $\varepsilon \gg 1$. Nota-se que para $0 < \varepsilon \ll 1$, com ILU_{EN} obtém-se o melhor ρ_{loc} ; e para $\varepsilon \gg 1$, com ILU_{NE} obtém-se melhor ρ_{loc} .

Após esta análise do fator de convergência assintótica, pretende-se propor um algoritmo que gere um método eficiente e robusto para qualquer anisotropia física. Isto será o objeto do próximo capítulo.

8.RESULTADOS

Neste capítulo será realizada uma análise detalhada utilizando-se o *solver* ILU e distintos operadores de restrição. Propõe-se um algoritmo que apresente melhores valores (mais próximos de zero) para o fator de convergência assintótica e o menor custo computacional do *multigrid*, portanto, um método eficiente e robusto para o problema de difusão anisotrópica.

O fator de convergência assintótica (ρ_{loc}) do método *multigrid* foi calculado mediante o estudo dos seguintes componentes: *solver* ILU nas direções EN e NE; operadores de restrição por injeção (INJ), meia-ponderação (HW), ponderação completa (FW), ponderação parcial em x (PW_x) e ponderação parcial em y (PW_y); prolongação bilinear e 7 pontos. Os resultados obtidos via LFA foram utilizados para prever e analisar a influência da anisotropia física sobre o custo computacional, o qual foi avaliado considerando-se o tempo de CPU e o número de operações em cada ciclo V e na etapa de restrição.

A Eq. (148) foi analisada para $\varepsilon = 10^\kappa$ e $\varepsilon = 10^{-\kappa}$, com $\kappa \in K = \{0, 1, 2, 3, 4, 5, 6, 7\}$. Assim como citado no capítulo anterior, quando $\varepsilon = 10^\kappa$ ou $\varepsilon = 10^{-\kappa}$, neste trabalho, denomina-se coeficiente de anisotropia simétrica. Por exemplo, $\varepsilon = 10^2$ é um coeficiente de anisotropia simétrica a $\varepsilon = 10^{-2}$.

Na seção 8.1 será realizada a análise de convergência utilizando LFA e o *solver* ILU. Somente os componentes ótimos obtidos via LFA serão utilizadas na construção de um novo algoritmo que irá proporcionar um método eficiente e robusto. Na seção 8.2 será realizada uma análise de complexidade confirmando as boas características do algoritmo construído. Na seção 8.3 a mesma análise de convergência da seção 8.1 será repetida para a versão modificada do ILU, denominada por MILU (WESSELING, 1992; SAAD, 2003). Na seção 8.4 será realizada a análise de complexidade do MILU utilizando-se os componentes ótimos obtidos na seção 8.3 e confirmando que o algoritmo proposto nesta tese se adapta bem ao MILU, apresentando-se assim, um método ainda mais eficiente e igualmente robusto.

8.1 Análise de convergência utilizando ILU

Na seção 7.2.2 observou-se que o *solver* ILU apresentou melhor desempenho em relação aos *solvers* tipo GS. Para $0 < \varepsilon \ll 1$, ILU_{EN} apresenta $\rho_{loc} \ll 1$ e para $\varepsilon \gg 1$, ILU_{NE} apresenta $\rho_{loc} \ll 1$. Para as análises apresentadas a seguir, os testes foram realizados utilizando-se somente esses *solvers* que apresentaram melhor desempenho em análises anteriores.

A Figura 41 apresenta o ρ_{loc} utilizando-se os *solvers* ILU_{EN} para $0 < \varepsilon \ll 1$ e ILU_{NE} para $\varepsilon \gg 1$; $\nu=2$; restrição FW; prolongação bilinear e 7pontos; $\varepsilon=10^\kappa$ e $\varepsilon=10^{-\kappa}$, com $\kappa \in K$. Nota-se que se fixam os parâmetros de restrição para se avaliar o efeito da prolongação.

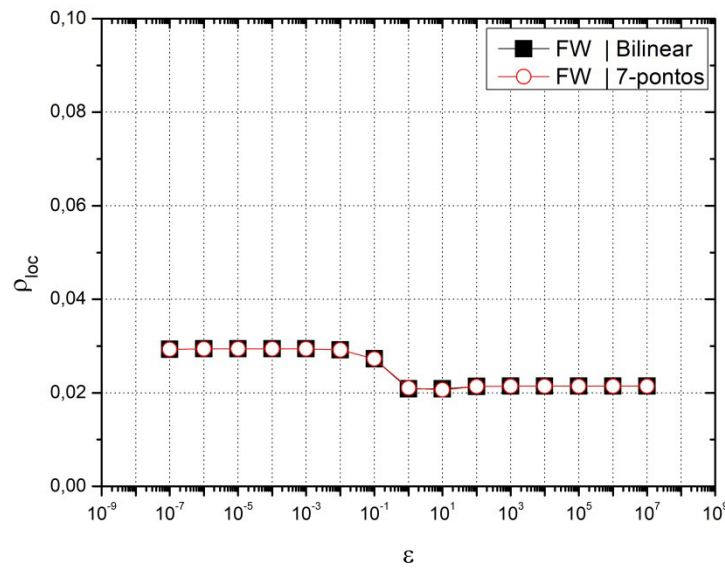


Figura 41: ρ_{loc} versus ε com diferentes operadores de interpolação.

Observa-se que as combinações da restrição FW com as prolongações bilinear e 7pontos, apresentaram bom desempenho ($\rho_{loc} \ll 1$) e possuem fator de convergência muito próximos.

Considerando-se que o operador de prolongação bilinear é de fácil programação e demanda menos recursos de memória, o mesmo foi empregado na análise a seguir. Em tal análise, foram comparados os fatores de convergência assintótica mediante diferentes operadores de restrição. Nota-se que se fixam os parâmetros de prolongação para se avaliar o efeito da restrição.

A Figura 42 apresenta o ρ_{loc} utilizando os *solvers* ILU_{EN} com $0 < \varepsilon \ll 1$ e ILU_{NE} com $\varepsilon \gg 1$; $\nu=2$; restrições FW, HW, INJ, PW_x e PW_y ; prolongação bilinear; $\varepsilon = 10^\kappa$ e $\varepsilon = 10^{-\kappa}$, com $\kappa \in K$.

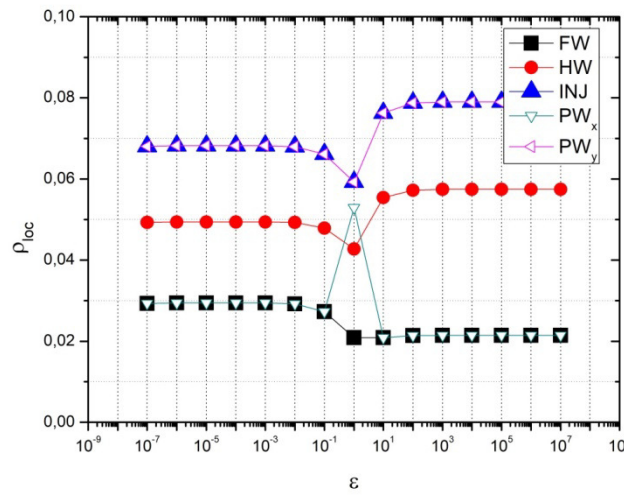


Figura 42: ρ_{loc} versus ε com diferentes operadores de restrição.

Na Figura 42 nota-se que, para problemas anisotrópicos ($\kappa \neq 0$) os menores valores para ρ_{loc} são alcançados utilizando-se as restrições FW e PW_x .

A partir dos resultados apresentados até então, propõe-se o Algoritmo 3 o qual combina o *solver* ILU em diferentes direções com as restrições FW e PW_x e que será utilizado daqui para frente. A sigla **REST**, utilizada neste algoritmo, representa qualquer uma das restrições FW ou PW_x , já definidas anteriormente.

Algoritmo 3: ILU_{EN-NE}

Início

se $\mathcal{E} > 1$ então

 Aplique a relaxação ILU_{NE} com a restrição **REST**

senão se $\mathcal{E} = 1$ então

 Aplique a relaxação ILU_{NE} com a restrição FW

senão

 Aplique a relaxação ILU_{EN} com a restrição **REST**

fim

É possível calcular o fator de convergência do *multigrid* empiricamente com base no resíduo. Tal fator, como descrito em Trottenberg et al. (2001) é dado por

$$\rho_h = \frac{\|r_h^m\|}{\|r_h^{m-1}\|}. \quad (157)$$

É possível também calcular o fator de convergência médio, como sendo a média geométrica dos fatores de convergência da Eq. (157)

$$\hat{\rho}_h = \sqrt[m]{\rho_h^m \rho_h^{m-1} \dots \rho_h^1} = \sqrt[m]{\frac{|r_h^m|}{|r_h^0|}}. \quad (158)$$

Neste texto se usa a Eq. (157) para calcular o “fator de convergência empírico” também chamado de “fator de convergência calculado experimentalmente”. Observa-se que ρ_{loc} é uma precisão teórica de ρ_h e, portanto, esses valores tendem a ser próximos para malhas muito refinadas.

A seguir apresenta-se uma comparação do fator de convergência assintótico de duas malhas (ρ_{loc}), previsto pela LFA com o fator de convergência empírico (ρ_h),

calculado para uma malha fina, obtida com 10 níveis de refino ($N = 1025 \times 1025$) e para diferentes valores para o coeficiente de anisotropia ε .

Na Figura 43, utilizou-se o Algoritmo 3 com **REST** = PW_x , $\nu = 2$ e prolongação bilinear. Deve-se ressaltar aqui que, os fatores de convergência utilizando FW ou PW_x são muito próximos, portanto, neste momento utilizar qualquer uma destas restrições é indiferente.

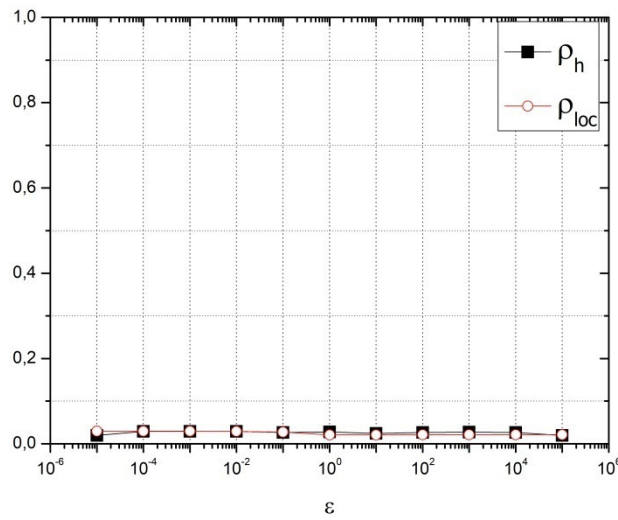


Figura 43: ρ_{loc} e ρ_h versus coeficiente de anisotropia ε .

Nota-se que $\rho_{loc} \approx \rho_h \ll 1$ para todos os coeficientes de anisotropia estudados. A escala em tal figura, foi feita de forma proposital, a fim de mostrar que ambos estão muito próximos de zero e distante da unidade (propriedade desejável). Observa-se que ρ_{loc} calculado via LFA está de acordo com o ρ_h calculado experimentalmente. A Tabela 16, apresenta-se alguns valores para ρ_{loc} e ρ_h que foram utilizados na Figura 43.

Tabela 16: ρ_{loc} e ρ_h da Figura 43.

ε	ρ_h	ρ_{loc}
10^{-4}	0,02846	0,02943
10^{-3}	0,02823	0,02941
10^{-2}	0,02841	0,02921

$$\frac{10^{-1} \quad 0,02646 \quad 0,02726}{\quad}$$

Testes adicionais foram feitos para ε muito próximos de 1 e os resultados, qualitativamente permaneceram os mesmos.

A Figura 44 apresenta o fator de convergência assintótica ρ_{loc} calculado via LFA e o fator de convergência experimental ρ_h para diferentes coeficientes de anisotropias e diferentes níveis de refino de malhas. Utilizou-se o Algoritmo 3 com **REST** = PW_x , $\nu=2$ e prolongação bilinear. Observa-se que à medida que a malha torna-se mais refinada $\rho_h \rightarrow \rho_{loc}$ para todos os coeficientes de anisotropia estudados, mostrando a robustez da metodologia estudada.

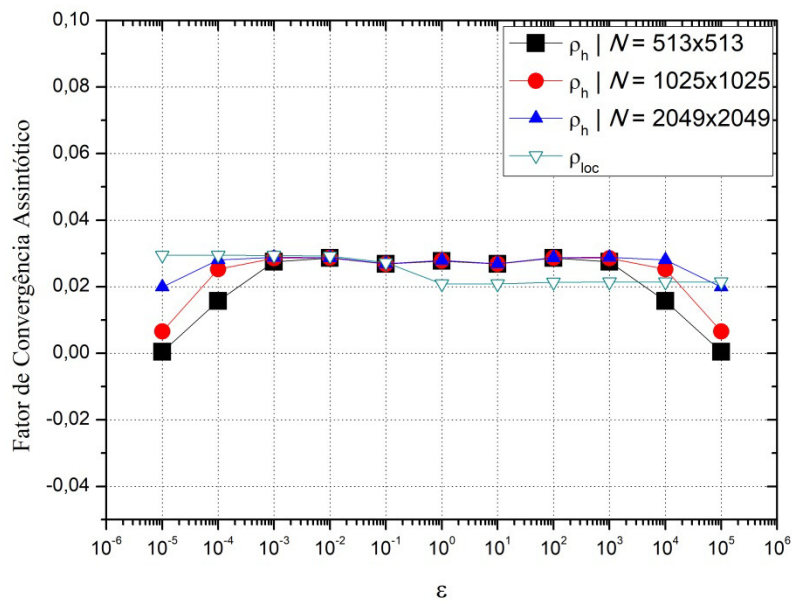


Figura 44: ρ_{loc} e ρ_h versus ε para malhas com diferentes números de pontos.

Na próxima seção será apresentada a análise de complexidade do *multigrid*. Para essa análise o *multigrid* foi construído com os componentes que apresentaram os melhores fatores de convergência, de acordo com a LFA desta seção. Será realizada também uma comparação entre as restrições por ponderação parcial (PW_x) e ponderação completa (FW).

8.2 Análise de complexidade utilizando ILU

Na Figura 45 utilizou-se o Algoritmo 3 com $\mathbf{REST} = \mathbf{PW}_x$, $\mathbf{REST} = \mathbf{FW}$, $\nu = 2$, prolongação bilinear. Apresenta-se o tempo de CPU para diversos coeficientes de anisotropia em um problema com $N = 512 \times 512 = 262.144$ incógnitas. Verifica-se que à medida que o problema torna-se mais anisotrópico, o tempo de CPU diminui. Para todos os coeficientes de anisotropia, o t_{CPU} utilizando a restrição FW está muito próximo do t_{CPU} utilizando \mathbf{PW}_x , ou seja, $t_{\text{CPU}}(\mathbf{FW}) \approx t_{\text{CPU}}(\mathbf{PW}_x)$.

Observa-se também que para anisotropias simétricas ($\varepsilon = 10^{-\kappa}$ e $\varepsilon = 10^{\kappa}$) com $\kappa \in \{1, 2, 3, 4\}$ os valores obtidos para t_{CPU} são muito próximos entre si.

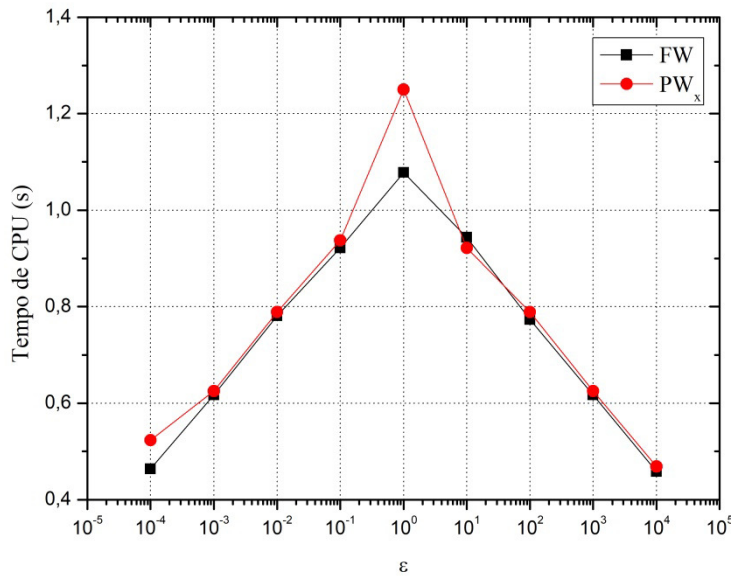


Figura 45: t_{CPU} versus ε para $L=9$ e ILU.

Para avaliar o efeito do número de incógnitas sobre o tempo de CPU foram utilizados os componentes ótimos obtidos via LFA. As Figs. Figura 46(a) e Figura 46(b), respectivamente para as restrições FW e \mathbf{PW}_x , mostram a influência do número de incógnitas e do coeficiente de anisotropia no tempo de CPU utilizando-se o Algoritmo 3, $\nu = 2$, prolongação bilinear, $\varepsilon = 10^{-\kappa}$ com $\kappa \in \{1, 2, 3, 4, 5, 6\}$. Os resultados para $\varepsilon = 10^{\kappa}$ são similares.

Observa-se que o coeficiente de anisotropia não influencia significativamente no tempo de CPU, ou seja, para um mesmo tamanho de problema, o tempo de CPU é muito próximo, independente do coeficiente de anisotropia.

Observa-se também que o tempo de CPU aumenta em função do tamanho do problema.

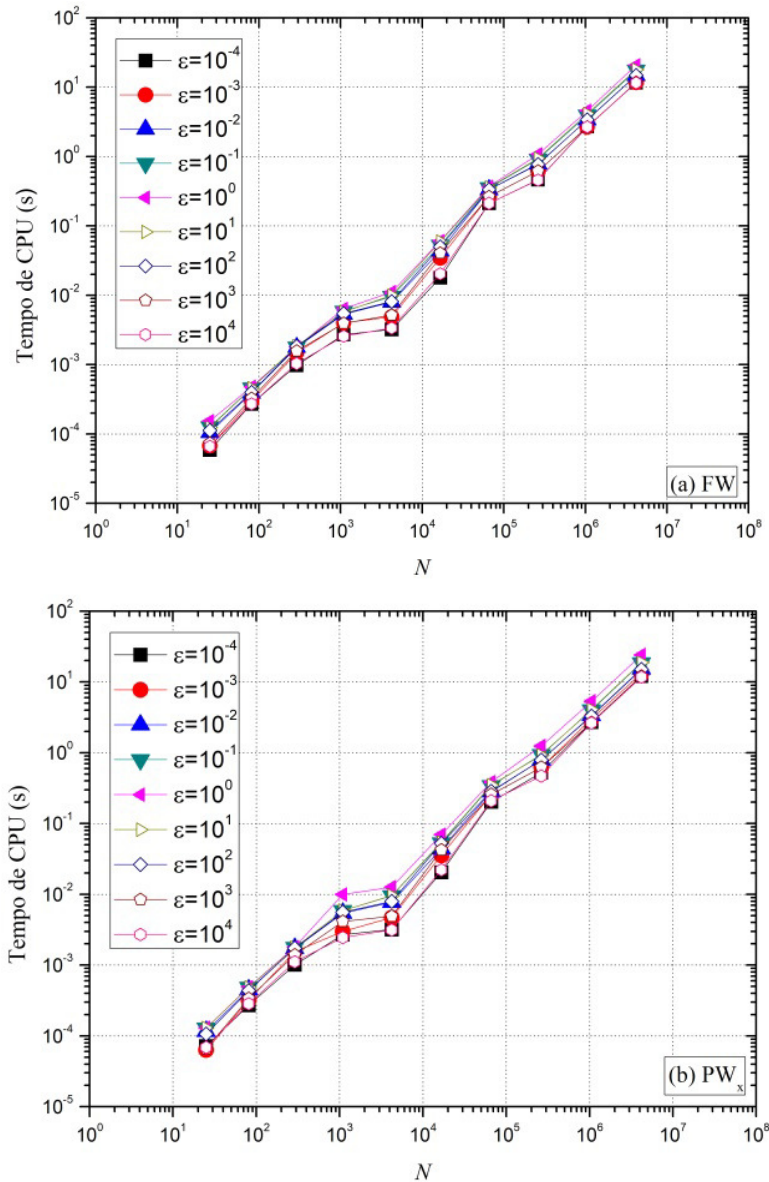


Figura 46: Tempo de CPU versus número de pontos N .

Para avaliar o desempenho do *multigrid*, mediante diferentes coeficientes de anisotropia, foi feito um ajuste de curva do tipo $t_{\text{CPU}} = cN^p$, onde p representa a ordem

de complexidade do *solver*, N é o número de pontos na malha e c é uma constante que depende do método. Quanto mais próximo da unidade o valor de p , melhor é o desempenho do método empregado. No caso ideal, o *multigrid* apresenta $p=1$, o que significa que o tempo de CPU cresce linearmente com o aumento de N (WESSELING, 1992; TROTTERBERG et al., 2001). O resultado é mostrado na Tabela 17, considerando as duas restrições abordadas (FW e PW_x).

Pode-se observar na Tabela 17 que, para as todas as anisotropias empregadas, o método *multigrid* tem um bom desempenho, dado que $p \approx 1$ em todos os casos. Esses resultados demonstram a eficiência e a robustez do Algoritmo 3, proposto neste trabalho.

Tabela 17: Ordem de complexidade (p) para diferentes coeficientes de anisotropia (ε).

ε	$p(PW_x)$	$p(FW)$
10^{-4}	1,07747	1,07733
10^{-2}	1,05023	1,05894
1	1,05940	1,04380
10^2	1,07255	1,06199
10^4	1,07627	1,07775

Os valores obtidos para o fator de convergência, apresentados na Tabela 16 e para a ordem de complexidade apresentados na Tabela 17 referente aos operadores de restrição PW_x e FW, são muito próximos entre si para as anisotropias analisadas e, portanto, insuficientes para decidir qual deles resulta em um algoritmo mais eficiente.

A Figura 47 mostra o tempo de CPU para dois coeficientes de anisotropia com as restrições FW e PW_x . Observa-se que, empregando-se a restrição PW_x tem-se t_{CPU} levemente menor do que empregando-se FW.

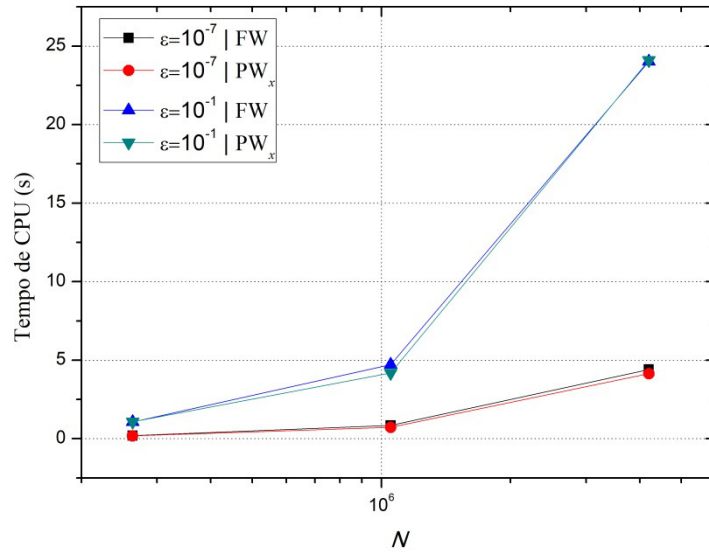


Figura 47: Tempo de CPU *versus* número de pontos N .

Para complementar a análise, foi calculado o número de operações aritméticas efetuadas cada um dos operadores (FW e PW_x), tanto para o ciclo V, quanto para a etapa de restrição somente. As operações aritméticas calculadas referem-se às operações de ponto flutuante (flops) realizadas durante o processo iterativo, as quais independem do *hardware* empregado. Cada operação de adição, subtração, multiplicação e divisão corresponde a 1 flop.

Foram realizados testes para $N_{10} = 1024 \times 1024 = 1.048.576$ (N_{10} é o número de pontos da malha mais fina, considerando um problema em que o número máximo de níveis seja $L_{\max} = 10$) e para alguns valores de ε . A Tabela 18 apresenta a razão entre o número de flops de um ciclo V e o número de pontos da malha mais fina N_{10} . A Tabela 19 apresenta a razão entre o número de flops realizados em cada restrição e o número de pontos na malha mais fina N_{10} . Nestas tabelas, os menores valores estão em negrito.

Tabela 18: Razão entre o número de flops de um ciclo V e N_{10} .

ε	$\text{flops}(PW_x) / N_{10}$	$\text{flops}(FW) / N_{10}$
10^{-2}	1510,221554	1543,296057
10^{-1}	1798,299366	1837,98877

1	2368,277388	2126,896785
10^1	1860,997604	1900,687008
10^2	1564,958717	1598,03322

Tabela 19: Razão entre o número de flops da restrição e N_{10} .

ε	flops(PW_x) / N_{10}	flops(FW) / N_{10}
10^{-2}	11,57607615	44,65057942
10^{-1}	13,8912913	53,5806953
1	18,37833483	62,02688007
10^1	13,89129137	53,5806953
10^2	11,57607615	44,65057942

Conforme Tabela 18, o ciclo *multigrid* que requer o menor número de flops é aquele com o operador PW_x , exceto para o caso isotrópico ($\varepsilon = 1$). Quando apenas a etapa de restrição é considerada, os resultados apresentados na Tabela 19 demonstram a grande vantagem do operador PW_x com relação ao operador FW, com relação ao número de flops realizados. Para todos os casos, o número de flops para o PW_x é cerca de 75% menor do que o número de flops para o FW.

Desempenho semelhante ao apresentado na Figura 42 e nas Tabs. Tabela 18 e Tabela 19 foram observados para os demais coeficientes de anisotropia estudados, confirmando assim a eficiência e a robustez do algoritmo proposto, além do baixo custo computacional do operador PW_x .

8.3 Análise de convergência utilizando MILU

Nesta seção os resultados apresentados empregam a técnica apresentada na seção 3.2.3, em que é introduzido um parâmetro σ , a fim de reduzir o efeito causado pelo descarte dos elementos durante o processo de eliminação LU incompleta. Nota-se que $\sigma = 0$ equivale ao método ILU padrão, ou seja, ILU sem modificação.

A Figura 48 apresenta ρ_{loc} para $\varepsilon = 10^{-\kappa}$ com $\kappa \in \{1, 2, 3, 4\}$ para diferentes valores de σ , com $0 \leq \sigma \leq 1$. Utilizou-se restrição FW, interpolação bilinear e $\nu = 2$. Observa-se que o menor valor para ρ_{loc} ocorre quando $\sigma \approx 0,2$. Os valores obtidos para $\varepsilon = 10^{\kappa}$ são similares a $\varepsilon = 10^{-\kappa}$.

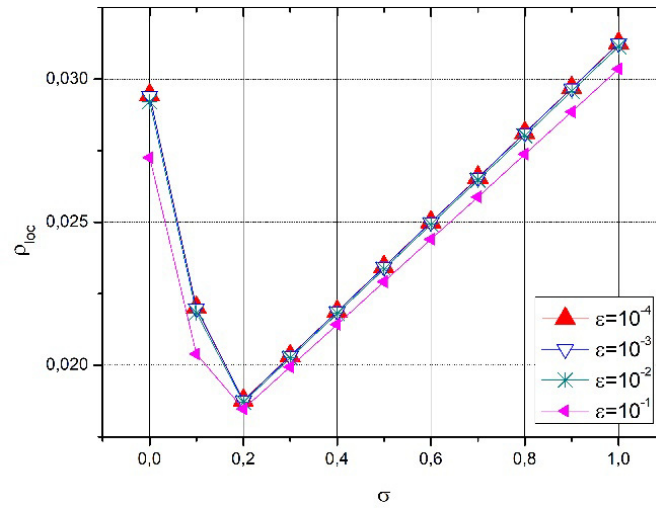


Figura 48: ρ_{loc} versus σ para diferentes valores de ε .

A Figura 49 apresenta ρ_h para $\varepsilon = 10^{-\kappa}$ com $\kappa \in \{1, 2, 3, 4\}$ para diferentes valores de σ , com $0 \leq \sigma \leq 1$. Observa-se que os menores valores para ρ_h também ocorrem quando $\sigma \approx 0,2$. Os valores obtidos para $\varepsilon = 10^{\kappa}$ são similares a $\varepsilon = 10^{-\kappa}$. Desta forma, daqui para frente será utilizado $\sigma = 0,2$ na formulação do MILU.

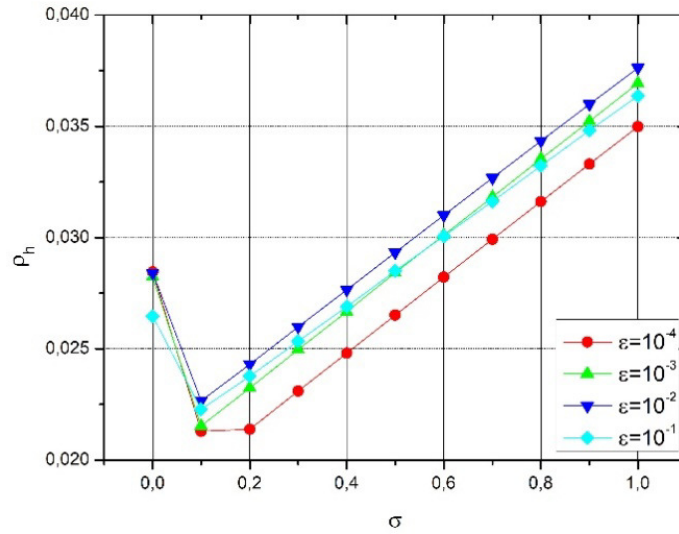


Figura 49: ρ_h versus σ para diferentes valores de ϵ .

A partir dos resultados apresentados até então, propõe-se o Algoritmo 4, o qual combina o *solver*MILU com $\sigma=0,2$ em diferentes direções com as restrições FW e PW_x e que será utilizado daqui para frente. A sigla **REST**, assim como no Algoritmo 3, representa qualquer uma das restrições (FW ou PW_x) já definidas anteriormente.

Nas seções 8.1 e 8.2, para $0 < \epsilon \ll 1$, o *solver* empregado foi ILU_{EN} com $\sigma=0$. Nesta seção, para $0 < \epsilon \ll 1$, será denominado $MILU_{EN}$ e utiliza-se $\sigma=0,2$. Da mesma forma, para $\epsilon \gg 1$, será denominado $MILU_{NE}$.

Algoritmo 4: $MILU_{EN-NE}$

Início: para $\sigma=0,2$ faça

se $\epsilon > 1$ então

Aplicar a relaxação $MILU_{NE}$ com a restrição **REST**

senão se $\epsilon = 1$ então

Aplique a relaxação $MILU_{NE}$ com a restrição FW

senão

Aplique a relaxação $MILU_{EN}$ com a restrição **REST**

fim

fim

A seguir apresenta-se uma comparação do fator de convergência assintótico de duas malhas (ρ_{loc}), previsto pela LFA com o fator de convergência empírico (ρ_h), calculado para uma malha fina, obtida com 10 níveis de refino ($N = 1025 \times 1025$) e para diferentes valores para o coeficiente de anisotropia ε .

Na Figura 50, utilizou-se o Algoritmo 4 ($\sigma = 0,2$) com **REST** = PW_x , $\nu = 2$ e prolongação bilinear. Deve-se ressaltar aqui que os fatores de convergência utilizando FW ou PW_x são muito próximos, portanto, neste momento utilizar qualquer uma dessas restrições é indiferente.

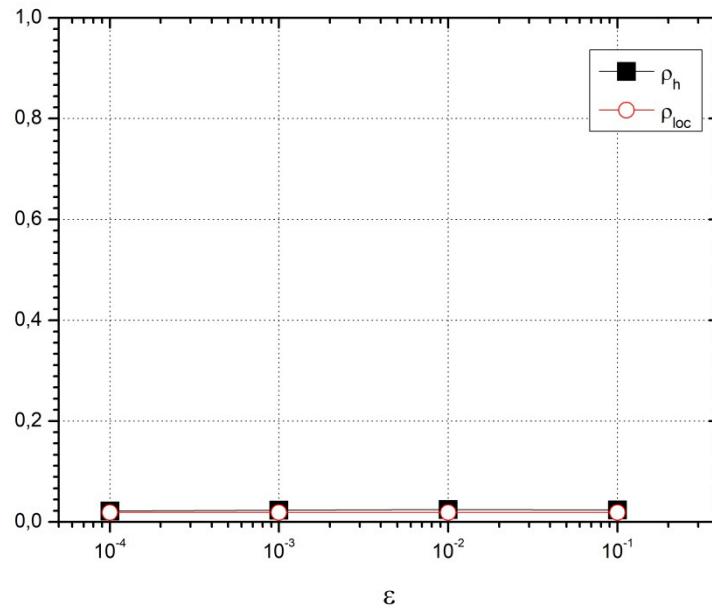


Figura 50: ρ_{loc} e ρ_h versus ε para $\sigma = 0,2$.

Nota-se que $\rho_{loc} \approx \rho_h \ll 1$ para todos os coeficientes de anisotropia estudados. Observa-se que ρ_{loc} calculado via LFA está de acordo com o ρ_h calculado experimentalmente. Os resultados observados nas Tabs. Tabela 16 e Tabela 20 confirmam que utilizando-se MILU com $\sigma = 0,2$ obtém-se um método *multigrid* ainda melhor em relação ao método apresentado no Algoritmo 3.

Tabela 20: ρ_{loc} e ρ_h para $\sigma = 0,2$.

ε	ρ_h	ρ_{loc}
10^{-4}	0,02139	0,01876
10^{-3}	0,02324	0,01876
10^{-2}	0,02431	0,01870
10^{-1}	0,02378	0,01847

Na próxima seção será apresentada a análise de complexidade do *multigrid*. Para essa análise o *multigrid* foi construído com os componentes que apresentaram os melhores fatores de convergência, de acordo com a LFA desta seção. Será realizada também uma comparação entre as restrições por ponderação parcial (PW_x) e ponderação completa (FW).

8.4 Análise de complexidade utilizando MILU

Na Figura 51 utilizou-se o Algoritmo 4 com **REST** = PW_x , **REST** = FW, $\nu = 2$, prolongação bilinear. Apresenta-se o tempo de CPU para diversos coeficientes de anisotropia em um problema com $N = 512 \times 512 = 262.144$ incógnitas. Verifica-se que à medida que o problema torna-se mais anisotrópico, o tempo de CPU diminui. Para todos os coeficientes de anisotropia, o t_{CPU} utilizando a restrição FW está muito próximo do t_{CPU} utilizando PW_x , ou seja, $t_{CPU}(FW) \approx t_{CPU}(PW_x)$.

Utilizando os algoritmos conhecidos na literatura, à medida que o problema torna-se mais anisotrópico o *multigrid* degrada. Ao se utilizar o Algoritmo 4, escolhendo o melhor *solver* de acordo com o coeficiente de anisotropia, este problema está contornado.

Observa-se também que para anisotropias simétricas ($\varepsilon = 10^{-\kappa}$ e $\varepsilon = 10^{\kappa}$) com $\kappa \in \{1, 2, 3, 4, 5, 6, 7\}$ os valores obtidos para t_{CPU} são muito próximos entre si.

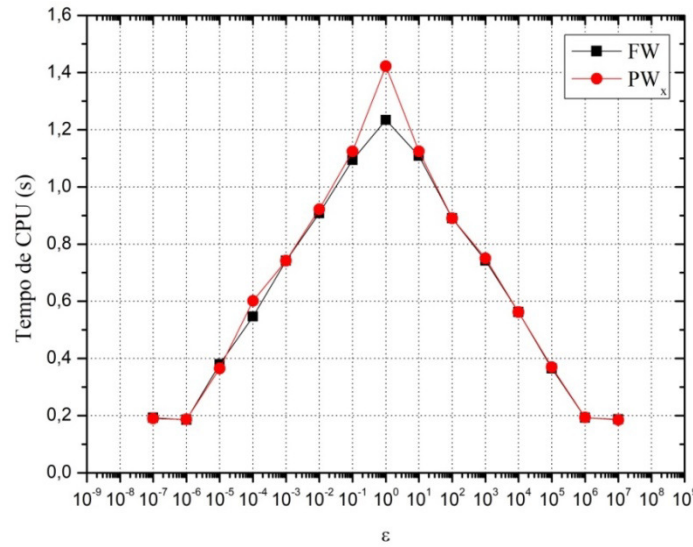


Figura 51: t_{CPU} versus ε para $N = 512 \times 512$ e MILU.

As Figs. Figura 52(a) e Figura 52(b), respectivamente para as restrições FW e PW_x, mostram a influência do número de incógnitas e do coeficiente de anisotropia no tempo de CPU utilizando-se o Algoritmo 4, $\nu = 2$, prolongação bilinear, $\varepsilon = 10^{-\kappa}$ com $\kappa \in \{1, 2, 3, 4, 5, 6\}$. Os resultados para $\varepsilon = 10^{\kappa}$ são similares.

Observa-se que o coeficiente de anisotropia não influencia significativamente no tempo de CPU. Observa-se também que o tempo de CPU aumenta em função do tamanho do problema.

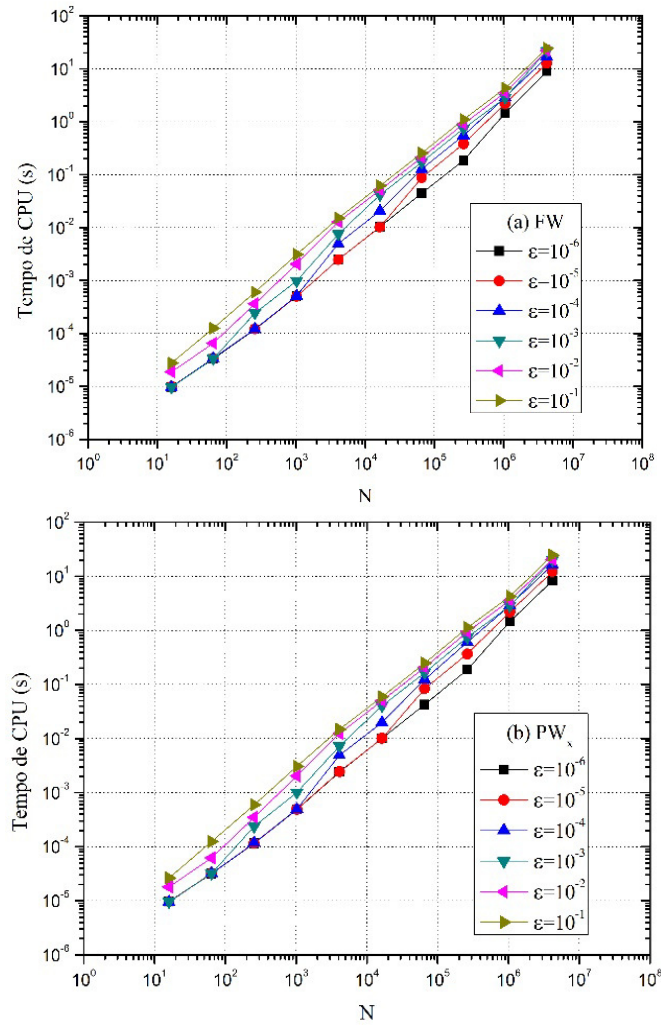


Figura 52: t_{CPU} versus número de incógnitas N .

Assim como foi apresentando na seção 8.2, para avaliar o desempenho do *multigrid*, mediante diferentes coeficientes de anisotropia, foi feito um ajuste de curva do tipo $t_{\text{CPU}} = cN^p$. O resultado é mostrado na Tabela 21, considerando as duas restrições abordadas (FW e PW_x).

Pode-se observar na Tabela 21 que, para as todas as anisotropias empregadas, o método *multigrid* tem um bom desempenho, dado que $p \approx 1$ em todos os casos. Esses resultados mostram a eficiência e a robustez do Algoritmo 4, proposto neste trabalho.

Tabela 21: Ordem de complexidade (p) para diferentes coeficientes de anisotropia (ε) e MILU.

ε	$p(PW_x)$	$p(FW)$
10^{-4}	1,13	1,17
10^{-2}	1,04	1,02
1	1,01	1,04
10^2	1,03	1,03
10^4	1,17	1,14

Os valores obtidos para o fator de convergência, apresentado na Tabela 20 e para a ordem de complexidade, apresentado na Tabela 21, referente aos operadores de restrição PW_x e FW são muito próximos entre si e, portanto, também são insuficientes para decidir qual deles resulta em um algoritmo mais eficiente.

Assim como foi apresentado na seção 8.2, para complementar a análise, foi calculado o número de operações aritméticas efetuadas por cada um dos operadores (FW e PW_x), tanto para o ciclo V, quanto para a etapa de restrição somente.

Foram realizados testes para $N_9 = 512 \times 512 = 262.144$ (N_9 é o número de pontos da malha mais fina, considerando um problema onde o número máximo de níveis seja $L_{\max} = 9$) e para alguns valores de ε . A Tabela 22 apresenta a razão entre o número de flops de um ciclo V e o número de pontos da malha mais fina N_9 . A Tabela 23 apresenta a razão entre o número de flops realizados em cada restrição e o número de pontos na malha mais fina N_9 . Nessas tabelas, os menores valores estão em negrito.

Conforme Tabela 22 o ciclo *multigrid* que requer o menor número de flops é aquele com o operador PW_x , exceto para o caso isotrópico ($\varepsilon = 1$). Quando apenas a etapa de restrição é considerada, os resultados apresentados na Tabela 23 mostram a grande vantagem do operador PW_x com relação ao operador FW, com relação ao número de flops realizados. Para todos os casos, o número de flops para o PW_x é cerca de 75% menor do que o número de flops para o FW.

Tabela 22: Razão entre o número de flops de um ciclo V e N_9 - MILU.

ε	flops(PW _x) / N_9	flops(FW) / N_9
10^{-6}	334,5341	341,0978
10^{-5}	626,3522	639,4796
10^{-4}	918,1703	937,8613
10^{-3}	1205,497	1231,751
10^{-2}	1506,298	1525,642
10^{-1}	1793,625	1833,007
1	2368,277	2126,897
10^1	1856,023	1895,405
10^2	1541,355	1574,174
10^3	1246,107	1272,361
10^4	950,8577	970,5488
10^5	649,136	662,2634
10^6	348,7089	353,9781

Tabela 23: Razão entre o número de flops da restrição e o número de pontos na malha mais fina N_9 - MILU.

ε	flops(PW_x) / N_9	flops(FW) / N_9
10^{-6}	2,297292	8,860983
10^{-5}	4,594584	17,72197
10^{-4}	6,891876	26,58295
10^{-3}	9,189167	35,44393
10^{-2}	11,48646	44,30491
10^{-1}	13,78375	53,1659
1	18,37833	62,02688
10^1	13,78375	53,1659
10^2	11,48646	44,30491
10^3	9,189167	35,44393
10^4	6,891876	26,58295
10^5	4,594584	17,72197
10^6	2,297292	8,860983

Desempenho semelhante ao apresentado nas Tabs. Tabela 22 e Tabela 23 foram observados para os demais coeficientes de anisotropia estudados, confirmando assim a eficiência e a robustez do algoritmo proposto, além do baixo custo computacional mediante o uso do operador PW_x associado ao *solver* MILU ($\sigma = 0, 2$).

9.CONCLUSÕES

No presente trabalho foi proposto um método robusto e eficiente para a resolução de problemas com anisotropia física. Para auxiliar no projeto do método *multigrid* eficiente foi empregada análise de Fourier Local. O modelo matemático considerado foi de difusão bidimensional. Para a discretização das equações foi utilizado o MDF com esquema de segunda ordem CDS.

Foi realizada análise de suavização, empregando-se os seguintes *solvers*: Gauss-Seidel por pontos com ordenação lexicográfica (GS-Lex); Gauss-Seidel por pontos com ordenação *red-black* (GS-RB); Gauss-Seidel por linhas com ordenação lexicográfica (*x*-linha-GS, *y*-linha-GS e alt-linha-GS); Gauss-Seidel por linhas com ordenação zebra (*x*-zebra-GS, *y*-zebra-GS e alt-zebra-GS); decomposição LU incompleta de 7 pontos.

Foi realizada análise de convergência assintótica empregando-se os seguintes operadores de restrição: INJ, HW, FW, PW_x e PW_y . Os operadores de prolongação utilizados foram: bilinear e 7 pontos. Para realizar análise de complexidade foram empregados os parâmetros ótimos obtidos na análise de convergência. As conclusões obtidas foram:

- Quanto ao fator de suavização, calculado via LFA, demonstrou-se que os *solvers* zebra-GS e ILU são bons suavizadores para problemas anisotrópicos.
- O fator de convergência assintótica, calculado via LFA, mostra que os operadores de restrição FW e PW_x e o operador de prolongação bilinear são adequados para problemas de anisotropia física.
- O fator de convergência assintótica calculado via LFA e empiricamente ficaram muito próximos, o que confirma a eficiência da LFA para prever e propor método *multigrid* eficiente.
- O *solver* ILU, além de ser bom suavizador para problemas anisotrópicos, quando combinado com bons operadores de restrição e prolongação, faz com que o fator de convergência assintótico do *multigrid* apresente ótimo desempenho.

- Uma pequena alteração na formulação do *solver* ILU, introduzindo um parâmetro σ , definiu-se o *solver* MILU, o qual apresentou fator de convergência assintótico menor do que ILU, para $\sigma=0,2$.
- Dentre as direções de ILU estudadas (EN, NE, ES, SE) para o *multigrid* padrão (operador de restrição FW e prolongação bilinear), obteve-se $\rho_{loc} \approx 0,02$ tanto para $0 < \varepsilon \ll 1$ com ILU_{EN} como para $\varepsilon \gg 1$ com ILU_{NE}.
- Utilizando-se restrição FW, ILU_{EN} para $0 < \varepsilon \ll 1$ e ILU_{NE} $\varepsilon \gg 1$, observou-se que os operadores de prolongação bilinear e 7 pontos apresentam desempenho similar. Nesse caso, para todos os valores de ε estudados obteve-se $\rho_{loc} \ll 1$.
- Utilizando-se interpolação bilinear, ILU_{EN} para $0 < \varepsilon \ll 1$ e ILU_{NE} $\varepsilon \gg 1$, observou-se que os menores valores para ρ_{loc} ocorrem com os operadores de restrição FW e PW_x e, para estes operadores, os valores para ρ_{loc} são similares.
- Utilizando-se os algoritmos Algoritmo 3 e Algoritmo 4, $\rho_{loc} \approx \rho_h \ll 1$ para todos os coeficientes de anisotropia estudados e $\rho_h \rightarrow \rho_{loc}$ a medida que a malha torna-se mais refinada.
- O $t_{CPU}(FW) \approx t_{CPU}(PW_x)$ para todos os valores de ε estudados utilizando-se os algoritmos Algoritmo 3 e Algoritmo 4.
- A ordem de complexidade p do método *multigrid* utilizando-se o Algoritmo 3 está próxima da unidade para todas as anisotropias estudadas. Para $\varepsilon=10^{-4}$, por exemplo, tem-se $p=1,07747$, utilizando-se tal algoritmo.
- O custo computacional do *multigrid* relacionado ao número de flops da restrição em um ciclo V, ao se utilizar os algoritmos Algoritmo 3 e Algoritmo 4 com a restrição dada por PW_x é 75% menor do que com a restrição dada por FW.
- Os algoritmos com a restrição dada por PW_x, propostos neste trabalho, são eficientes, robustos e apresentam baixo custo computacional.

9.1 Contribuições

Nesta tese foi desenvolvido um método robusto e eficiente para o problema de difusão com anisotropia nos coeficientes (anisotropia física). Foi empregado engrossamento padrão e foi feita uma seleção de suavizadores e operadores de restrição e prolongação adequados, dependendo da anisotropia do problema. A escolha destes componentes foi realizada com a ajuda de uma análise de Fourier local (LFA) e comprovada através de experimentos numéricos.

Para a LFA os suavizadores utilizados foram Gauss-Seidel por pontos (GS-RB e GS-Lex); Gauss-Seidel por linhas (y-linha-GS, x-linha-GS, alt-linha-GS, y-zebra-GS, x-zebra-GS e alt-zebra-GS); Decomposição LU incompleta (ILU 5 pontos e ILU 7 pontos). Os operadores de restrição testados foram INJ, HW, FW, PW_x e PW_y ; e os operadores de prolongação foram bilinear e 7 pontos.

Uma grande contribuição desta tese refere-se a uma coletânea de testes com diversos componentes do método *multigrid* para vários coeficientes de anisotropia, os quais não são encontrados na literatura.

Porém, sua principal contribuição é a obtenção de um algoritmo robusto e eficiente para resolver problemas com forte anisotropia difusiva. Foi proposto um algoritmo que é robusto para uma gama enorme de anisotropias e também para qualquer nível de refino de malha; além de se ter um algoritmo com fator de convergência próximo de 0,02 com baixo custo computacional se comparado aos métodos disponíveis na literatura, portanto, um método altamente eficiente.

Outra importante contribuição, é que o algoritmo proposto tende a ficar cada vez mais eficiente à medida que se tem anisotropias mais fortes.

9.2 Trabalhos futuros

A seguir, são apresentadas algumas propostas de trabalhos futuros, para complementar e ampliar os resultados obtidos neste trabalho.

- Realizar análise de convergência e análise de complexidade utilizando a equação de advecção-difusão com anisotropia física.

- Utilizar os mesmos componentes *multigrid* descritos neste trabalho e aplicar à equação de difusão anisotrópica tridimensional.
- Propor problemas em que a anisotropia apareça na solução analítica, gerando um complicador a mais para a sua solução.
- Empregar engrossamento agressivo de malhas afim de reduzir os recursos com memória;
- Propor uma extensão das técnicas estudadas nesta tese para o ILU por blocos, BILU.
- Utilizar outros tipos de ciclos *multigrid*, tais como, W e F.
- Aplicar a programação paralela a fim de melhorar o desempenho do método *multigrid*.

REFERÊNCIAS

- BAKHVALOV, N. S., 1966. On the Convergence of Relaxation Method with Natural Constraints on the Elliptic operator. URSS Comp. Math. Phys, v.1, n.5.
- BRANDT, A., 1977. Multi-Level Adaptive Solutions to Boundary-Value Problems. Mathematics of Computation, v. 31, p. 333-390.
- BRIGGS, W. L.; HENSON, V.E.; MCCORMICK, S.F., 2000. A *Multigrid* Tutorial. 2^a ed., SIAM.
- BURDEN, R. L.; FAIRES, J. D., 2009. Análise Numérica. São Paulo: Pioneira Thomson Learning.
- CORDAZZO, J., 2006. Simulação de Reservatórios de Petróleo utilizando o método EBFVM e *Multigrid* Algébrico. Tese de doutorado em Engenharia Mecânica. UFSC. Florianópolis, SC, Brasil.
- CRAIG C. D., 1996. *Multigrid* Methods in Science and Engineering. IBM T.F. Watson Research Center and Yale University, IEE.
- DEDNER, A.; MULLER, E.; SCHEICHL, R., 2014. Efficient *Multigrid* Preconditioners for Anisotropic Problems in Geophysical Modelling. 14 Aug. 2014 <<http://www.arXiv.org>>.
- FEDORENKO, R. P., 1964. On the Speed of Convergence of an Iteration Process. USSR. Comput. Math. and Math. Phys, v. 4, p.3.
- FERZIGER, J. H.; PERIC, M., 2002. Computational Methods for Fluid Dynamics. Berlin: Springer-Verlag.
- FISCHER, R; HUCKLE, T., 2006. *Multigrid* Methods for Anisotropic BTTB Systems. Linear Algebra and its Applications. v. 417, p. 314-333.
- FISCHER, R; HUCKLE, T., 2008. *Multigrid* Solution Techniques for Anisotropic Structured Linear Systems. Applied Numerical Mathematic. v. 58, p. 407-421.
- FORTUNA, A. O., 2000. Técnicas Computacionais para Dinâmica dos Fluidos. São Paulo: Edusp.

- FRANCO, S. R.; GASPAR, F. J.; PINTO, M. A. V.; RODRIGO, C., 2018. *Multigrid* method based on a space-time approach with standard coarsening for parabolic problems. *Applied Mathematics and Computation*. v. 317, p. 25-34.
- GEE, W.M.; HU J.; TUMINARO S.R., 2009. A New Smoothed Aggregation *Multigrid* for Anisotropic Problems. *Numerical Linear Algebra Appl*, v.16, p.19-37.
- GHIA, U.; GHIA, K. N.; GHIN, C. T., 1982. High-Re solutions for incompressible flow using the Navier-Stokes equations and a *multigrid* method. *Journal of Computational Physics*, v.48, p.387-411.
- GOLUB, G. H; ORTEGA, J. M., 1992. *Scientific Computing and Differential Equations: an Introduction to Numerical Methods*, Academic Press.
- HUGHES, T. J. R., 2000. *The Finite Element Method – Linear Static and Dynamics Finite Element Analysis*. Mineola (USA): Dover Publications.
- HUTCHINSON, B. R.; RAITHBY., 1986. A *Multigrid* Method based on the Additive Correction Strategy. *Numerical Heat Transfer*, v. 9, p. 511-537.
- JOHANNSEN, K., 2005. A robust 9-point ILU smoother for anisotropic problems, IWR Preprint, University of Heidelberg.
- KARAA, S. E ZHANG, J., 2002. Convergence and performance of iterative methods for solving variable coefficient convection-diffusion equation with a fourth-order compact difference scheme. *International Journal of Computers and Mathematics with applications*, v. 44, p.457-479.
- LARSSON, J.; LIEN, F.S.; YEEE., 2005. Conditional semicoarsening multigrid algorithm for the Poisson equation on anisotropic grids, *J. Comput. Phys.* 208, p.368–383.
- MALISKA, C. R., 2004. *Transferência de Calor e Mecânica dos Fluidos Computacional*. LTC, 2ª ed.
- MARCHI, C. H., 2001. Verificação de soluções numéricas unidimensionais em dinâmica dos fluidos. Tese de Doutorado em Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil.

- OLIVEIRA, F., 2010. Efeito de Malhas Anisotrópicas bidimensionais sobre o Desempenho do Método *Multigrid* Geométrico. Tese de doutorado em Engenharia Mecânica. UFPR. Curitiba, PR, Brasil.
- OLIVEIRA, F.; PINTO, M.A.V.; MARCHI, C.H.; ARAKI, L.K., 2012. Optimized partial semicoarsening *multigrid* algorithm for heat diffusion problems and anisotropic grids. *Applied Mathematical Modelling*. v. 36, p. 4665–4676.
- PINTO, M. A. V.; MARCHI, C. H., 2006. Efeito dos Parâmetros do Método *Multigrid* CS e FAS sobre o tempo de CPU para a Equação de Laplace Bidimensional. *Proceedings of ENCIT*.
- PINTO, M. A. V.; RODRIGO, C.; GASPAR, F. J.; OOSTERLEE, C.W., 2016. On the robustness of ILU smoothers on triangular grids, *Applied Numerical Mathematics*. v. 106, p. 37-52.
- PLETCHER, R, H; TANNEHILL, J, C.; ANDERSON, D, A. *Computational Fluid Mechanics and Heat Transfer*, 3^a ed, CRC Press, USA, 2013
- RABI, J. A.; LEMOS, M. J. S., 2001. Optimization of convergence acceleration in *multigrid* numerical solutions of conductive-convective problems. *Applied Mathematics and Computational*. v. 124, p. 215-226.
- REDDYE, J. N.; GARTLING, D. K., 1994. *The finite Element Method in Heat Transfer and Fluid Dynamics*. Boca Raton (USA): CRC Press.
- RUGE, J. STÜBEN, K., 1986. Algebraic *Multigrid* (AMG). *S. F. McCormick, Multigrid Methods*, v. 5 de *Frontiers in Applied Mathematics*, SIAM.
- SAAD, Y. *Iterative Methods for Sparse Linear Systems*, 2^aed., PWS, Philadelphia, 2003
- SCHNEIDER, G. E., ZEDAN, M., 1981. A modified strongly implicit procedure for the numerical solution of fiel problems. *Numerical Heat Transfer*.
- SETTARI, A.; AZIZ, K. A., 1973. Generalization of Additive Correction Methods for the Iterative Solution of Matrix Equations. *SIAM Journal of Numerical Analysis*. v.10, p.506-521.
- SUERO, R., PINTO, M. A. V., MARCHI, C. H., 2008. Effect of physical anisotropy of the *multigrid* method in solving the advection-diffusion equation 2D. XXIX CILAMCE.

- TANNEHILL, J. C.; ANDERSON, D. A.; PLETCHER, R. H., 1997. Computational Fluid Mechanics and Heat Transfer. 2 ed. Washington: Taylor & Francis.
- THEKALE, T.; GRADL, K.; KLAMROTH, U.; RÜDE, U., 2010. Optimizing the number of *multigrid* cycles in the full *multigrid* algorithm, Numerical Linear Algebra with Applications. v. 17, p. 199–210.
- TROTTEBERG, U.; OOSTERLEE, C.; SCHÜLLER, A., 2001. *Multigrid*. Academic Press.
- VERSTEEG, H. K.; MALALASEKERA, W., 2007. An Introduction to Computational Fluid Dynamic, The Finite Volume Method, England: Longman, 2 ed.
- VINOGRADOVA, S. A.; KRUKIER, L. A., 2013. The use of incomplete LU decomposition in modeling convection-diffusion processes in an anisotropic medium. Mathematical Models and Computer Simulations. v. 5, p. 190-197.
- WESSELING, P., 1992. An Introduction to *Multigrid* Methods. John Wiley and Sons.
- WESSELING, P.; OOSTERLEE, C. W., 2001. Geometric *Multigrid* with Applications to Computational Fluid Dynamics. Journal of Computation and Applied Mathematics, v. 128, p. 311-334.
- WIENANDS, R.; JOPPICH, W., 2005. Practical Fourier Analysis for *Multigrid* Methods, CRC Press, USA.
- ZHANG, J., 2002. *Multigrid* Method and Fourth-Order Compact Scheme for 2D Poisson Equation with Unequal Mesh-Size Discretization. Journal of Computational Physics, v. 179, p. 170-179.