

UNIVERSIDADE FEDERAL DO PARANÁ
MÁRCIO ALEXANDRO MACIEL DE ANUNCIÇÃO

REDUÇÃO DO ERRO DE ITERAÇÃO E ACELERAÇÃO DO MÉTODO
***MULTIGRID* COM O USO DE EXTRAPOLADORES**

CURITIBA

2013

MÁRCIO ALEXANDRO MACIEL DE ANUNCIÇÃO

**REDUÇÃO DO ERRO DE ITERAÇÃO E ACELERAÇÃO DO MÉTODO
MULTIGRID COM O USO DE EXTRAPOLADORES**

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre em Ciências, pelo Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Setor de Ciências Exatas e Tecnologia, Universidade Federal do Paraná.

Orientador:

Prof. Dr. Marcio Augusto Villela Pinto

Co-orientador:

Prof. Dr. Luciano Kiyoshi Araki

CURITIBA

2013

TERMO DE APROVAÇÃO

MÁRCIO ALEXANDRO MACIEL DE ANUNCIÃO

REDUÇÃO DO ERRO DE ITERAÇÃO E ACELERAÇÃO DO MÉTODO *MULTIGRID* COM O USO DE EXTRAPOLADORES

Dissertação aprovada como requisito parcial para a obtenção do título do Mestre em Ciências, área de concentração: Mecânica Computacional, pelo Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Setor de Ciências Exatas e Tecnologia, Universidade Federal do Paraná, pela seguinte banca examinadora:

Prof. Dr. Marcio Augusto Villela Pinto
Universidade Federal do Paraná (UFPR)

Prof. Dra. Luciane Grossi
Universidade Estadual de Ponta Grossa – (UEPG)

Prof. Dr. Carlos Henrique Marchi
Universidade Federal do Paraná (UFPR)

Curitiba, 18 de fevereiro de 2013.

A Deus, por tudo.

Aos meus pais e irmã, pelo constante incentivo e credibilidade em minha capacidade.

Ao meu amor, pelo apoio interminável e compreensão ao longo desta jornada.

AGRADECIMENTOS

Agradeço imensamente ao meu orientador, Prof. Dr. Marcio Augusto Villela Pinto, por ter me mostrado o caminho da pós-graduação, ter aceitado me orientar neste trabalho, pelo apoio, pela paciência, didática e por todo o conhecimento recebido.

Ao meu co-orientador, Prof. Dr. Luciano Kiyoshi Araki, pela costumeira disposição em auxiliar, pela dedicação e pelo conhecimento recebido.

Ao meu colega Márcio André Martins, por permitir que eu continuasse o trabalho que ele havia iniciado, pelas dúvidas que sanou e pelas contribuições sempre valiosas.

Agradeço aos membros da banca examinadora, Prof. Dr. Carlos Henrique Marchi e Prof. Dra. Luciane Grossi pelo tempo despendido na leitura deste trabalho e pelas importantes sugestões apontadas.

Aos professores, funcionários e colegas do LENA-1 e do CESEC, pela amizade e disposição em ajudar.

Aos meus familiares, pela compreensão, apoio incondicional, credibilidade, incentivo, demonstração de felicidade a cada etapa vencida e pelo amor, acima de tudo.

À minha noiva Lyohana Ançay, por estar sempre próxima, me apoiar, me incentivar e demonstrar o seu amor a todo momento.

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

RESUMO

Determinar a solução de sistemas de equações lineares e não-lineares é um problema importante em Matemática Computacional. Métodos iterativos são amplamente utilizados para este fim. Entretanto, tais métodos podem convergir lentamente. Nas últimas décadas, um grande avanço na aceleração da taxa de convergência de processos iterativos se deu pelo desenvolvimento do método *Multigrid*. Outra forma de acelerar a convergência do método iterativo é utilizar um método de extrapolação associado ao processo iterativo. Alguns autores obtiveram resultados promissores com o estudo da combinação de métodos iterativos com métodos de extrapolação, o que mostra ser uma alternativa viável e promissora para aceleração de convergência. Neste trabalho foi resolvido numericamente o problema de condução de calor linear bidimensional, governado pela equação de Poisson, com condições de contorno de Dirichlet. Utilizou-se o Método das Diferenças Finitas (MDF), com esquema de aproximação de segunda ordem (CDS) para discretização do modelo matemático. Para a obtenção da solução, foi empregado o método *Multigrid* geométrico, *solver* Gauss-Seidel *red-black*, com esquema de correção (CS), restrição por ponderação completa, prolongação por interpolação bilinear e número máximo de níveis para os diversos casos estudados. Foram associados ao final do *Multigrid* os seguintes extrapoladores: Aitken, Empírico, Mitin, Épsilon escalar, Rho escalar, Épsilon topológico, Rho topológico, Múltipla extrapolação de Aitken e Múltipla extrapolação de Mitin. Durante o *Multigrid*, foi usado apenas o extrapolador Épsilon topológico. Os resultados podem ser considerados positivos, pois se verificou, entre outros, que o uso de extrapoladores associados ao método *Multigrid* reduz de forma satisfatória a magnitude do erro de iteração, do resíduo adimensionalizado com base na estimativa inicial e do fator de convergência, em um tempo praticamente equivalente ao da aplicação do método *Multigrid* puro ou apresentando uma leve melhoria de desempenho sobre o mesmo.

Palavras-chave: Métodos iterativos, métodos de extrapolação, método *Multigrid*, aceleração de convergência, erro de iteração.

ABSTRACT

The determination of solutions for linear and non-linear systems of equations is an important Computational Mathematics problem. Iterative methods are widely used in these situations. Otherwise, these methods present slow convergence. For the last decades, significant advances in the acceleration of the convergence rates have been obtained by the development of the Multigrid Method. Moreover, another way to speed-up the convergence of an iterative method is based on the use of an extrapolation method associated to the iterative process. Some authors have already obtained promising results by studying the combination of iterative methods and extrapolation ones, being a viable choice for the acceleration of convergence rates. In the present work a two-dimensional linear heat transfer conduction problem, given by a Poisson-type equation with Dirichlet boundary conditions, was numerically solved. The numerical model was obtained by the use of the Finite Difference Method (FDM) with a second-order (CDS) approximation scheme. In order to achieve the numerical solution, the Geometric Multigrid Method was used, associated to the Gauss-Seidel Red-Black solver, correction scheme (CS), full-weighting scheme for restriction, bilinear interpolation for prolongation and the maximum number of levels for each one of the studied cases. At the end of the Multigrid, the following interpolators were employed: Aitken, Empiric, Mitin, scalar Epsilon, scalar Rho, topological Epsilon, topological Rho, repeated Aitken extrapolation and repeated Mitin extrapolation. During the Multigrid, only the topological Epsilon extrapolator was used. Numerical results can be evaluated positively, once the extrapolators associated to the Multigrid Method satisfactorily reduce the magnitudes of the iteration error, the non-dimensional residual based on the initial guess and the convergence factor, spending a time interval nearly equivalent to the application of the pure Multigrid algorithm or even presenting a small performance improvement when compared to the basic Multigrid.

Key words: *Iterative methods, extrapolation methods, Multigrid method, convergence acceleration, iteration error.*

LISTA DE ALGORITMOS

Algoritmo 2.1: Procedimento de Gauss-Seidel <i>red-black</i> para $Au = f$	36
Algoritmo 2.2: Ciclo V com esquema CS para vários níveis de malhas. (adaptado de Briggs et al., 2000)	47
Algoritmo 2.3: Diversas chamadas do <i>Multigrid</i> com esquema CS	48

LISTA DE FIGURAS

Figura 2.1: Exemplo de transformação do domínio contínuo em domínio discreto	29
Figura 2.2: Convenção para nomenclatura dos vizinhos dos nós da malha bidimensional uniforme	34
Figura 2.3: Ordenação para Gauss-Seidel lexicográfico e <i>red-black</i>	35
Figura 2.4: Modos de Fourier com $k = 1, 3$ e 6 (adaptada de BRIGGS et al., 2000)	40
Figura 2.5: Jacobi ponderado com fator de ponderação igual a $2/3$ aplicado ao problema de condução de calor unidimensional com $N = 64$, estimativa inicial $(v_1 + v_6 + v_{32})/3$ e para 100 iterações (BRIGGS et al., 2000).....	41
Figura 2.6: Engrossamento padrão com razão de engrossamento $r = 2$	41
Figura 2.7: Processo de engrossamento e geração de malhas. (adaptado de http://www.mgnet.org/mgnet/tutorials/xwb/mg.html)	42
Figura 2.8: Operador de restrição (ponderação completa) da malha mais fina para a malha mais grossa (BRIGGS et al., 2000)	43
Figura 2.9: Prolongação bilinear (TROTTEBERG et al., 2001).....	44
Figura 2.10: Diagramas: (a) ciclo V, (b) F, (c) dente-de-serra, (d) W (TROTTEBERG et al., 2001).....	45
Figura 2.11: Representação esquemática da Múltipla extrapolação de Aitken com dois níveis	59
Figura 3.1: Domínio bidimensional para a equação de Poisson	62
Figura 4.1: Comportamento da $\ R\ _2$ em função de N para $\varepsilon = 10^{-15}$	74
Figura 4.2: Comportamento da $\ R\ _2$ em função de ε para $N = 4097 \times 4097$	75
Figura 4.3: Comportamento da $\ E_n\ _2$ em função de N para $\varepsilon = 10^{-15}$	76
Figura 4.4: Comportamento da $\ E_n\ _2$ em função de ε para $N = 4097 \times 4097$	77
Figura 4.5: Comportamento do t_{CPU} em função de N para $\varepsilon = 10^{-15}$	77
Figura 4.6: Comportamento da $\ R\ _2$ em função do número de iterações, para $N = 4097 \times 4097$	82
Figura 4.7: Aumento percentual da memória utilizada para MG + Épsilon topológico em relação a MG em função de N para $\varepsilon = 10^{-20}$	86
Figura A.1: Comportamento da $\ R\ _2$ em função de N para $\varepsilon = 10^{-6}$	96

Figura A.2: Comportamento da $\ R\ _2$ em função de N para $\varepsilon = 10^{-10}$	96
Figura A.3: Comportamento da $\ R\ _2$ em função de ε para $N = 129 \times 129$	97
Figura A.4: Comportamento da $\ R\ _2$ em função de ε para $N = 1025 \times 1025$	97
Figura A.5: Comportamento da $\ E_n\ _2$ em função de N para $\varepsilon = 10^{-6}$	98
Figura A.6: Comportamento da $\ E_n\ _2$ em função de N para $\varepsilon = 10^{-10}$	98
Figura A.7: Comportamento da $\ E_n\ _2$ em função de ε para $N = 129 \times 129$	99
Figura A.8: Comportamento da $\ E_n\ _2$ em função de ε para $N = 1025 \times 1025$	99
Figura A.9: Comportamento do t_{CPU} em função de N para $\varepsilon = 10^{-6}$	100
Figura A.10: Comportamento do t_{CPU} em função de N para $\varepsilon = 10^{-10}$	100
Figura B.1: Comportamento do t_{CPU} em função de N para $\varepsilon = 10^{-25}$	102

LISTA DE TABELAS

Tabela 1.1: Comparação entre os métodos de solução de problemas em engenharia (Adaptada de Tannehill et al., 1997).....	20
Tabela 4.1: Número de soluções utilizadas para extrapolações	67
Tabela 4.2: Casos estudados de acordo com N e ε	69
Tabela 4.3: Resultados dos parâmetros estudados para o caso 1-A ($N = 129 \times 129$ e $\varepsilon = 10^{-6}$)	70
Tabela 4.4: Resultados dos parâmetros estudados para o caso 1-B ($N = 129 \times 129$ e $\varepsilon = 10^{-10}$).....	70
Tabela 4.5: Resultados dos parâmetros estudados para o caso 1-C ($N = 129 \times 129$ e $\varepsilon = 10^{-15}$).....	71
Tabela 4.6: Resultados dos parâmetros estudados para o caso 2-A ($N = 1025 \times 1025$ e $\varepsilon = 10^{-6}$)	71
Tabela 4.7: Resultados dos parâmetros estudados para o caso 2-B ($N = 1025 \times 1025$ e $\varepsilon = 10^{-10}$).....	71
Tabela 4.8: Resultados dos parâmetros estudados para o caso 2-C ($N = 1025 \times 1025$ e $\varepsilon = 10^{-15}$).....	72
Tabela 4.9: Resultados dos parâmetros estudados para o caso 3-A ($N = 4097 \times 4097$ e $\varepsilon = 10^{-6}$)	72
Tabela 4.10: Resultados dos parâmetros estudados para o caso 3-B ($N = 4097 \times 4097$ e $\varepsilon = 10^{-10}$).....	72
Tabela 4.11: Resultados dos parâmetros estudados para o caso 3-C ($N = 4097 \times 4097$ e $\varepsilon = 10^{-15}$).....	72
Tabela 4.12: Resultados dos parâmetros estudados para o caso 1-A ($N = 129 \times 129$ e $\varepsilon = 10^{-6}$)	79
Tabela 4.13: Resultados dos parâmetros estudados para o caso 1-B ($N = 129 \times 129$ e $\varepsilon = 10^{-10}$).....	79
Tabela 4.14: Resultados dos parâmetros estudados para o caso 1-C ($N = 129 \times 129$ e $\varepsilon = 10^{-15}$	79

Tabela 4.15: Resultados dos parâmetros estudados para o caso 2-A ($N = 1025 \times 1025$ e $\varepsilon = 10^{-6}$)	79
Tabela 4.16: Resultados dos parâmetros estudados para o caso 2-B ($N = 1025 \times 1025$ e $\varepsilon = 10^{-10}$)	80
Tabela 4.17: Resultados dos parâmetros estudados para o caso 2-C ($N = 1025 \times 1025$ e $\varepsilon = 10^{-15}$)	80
Tabela 4.18: Resultados dos parâmetros estudados para o caso 3-A ($N = 4097 \times 4097$ e $\varepsilon = 10^{-6}$)	80
Tabela 4.19: Resultados dos parâmetros estudados para o caso 3-B ($N = 4097 \times 4097$ e $\varepsilon = 10^{-10}$)	80
Tabela 4.20: Resultados dos parâmetros estudados para o caso 3-C ($N = 4097 \times 4097$ e $\varepsilon = 10^{-15}$)	80
Tabela 4.21: Tempo de CPU (s) para MG e MG + Épsilon topológico para $\varepsilon = 10^{-20}$	83
Tabela 4.22: Valores de p e c para os algoritmos das metodologias em estudo	84
Tabela 4.23: <i>Speed-up</i> de MG + Épsilon topológico em relação ao MG para $\varepsilon = 10^{-20}$	84
Tabela 4.24: Memórias e aumento percentual da memória para as metodologias MG e MG + Épsilon topológico	85
Tabela B.1: Tempo de CPU (s) para MG e MG + Épsilon topológico para $\varepsilon = 10^{-25}$	101
Tabela B.2: Valores de p e c para os algoritmos das metodologias em estudo	102
Tabela B.3: <i>Speed-up</i> de MG + Épsilon topológico em relação ao MG para $\varepsilon = 10^{-25}$	103

LISTA DE ABREVIATURAS E SIGLAS

ADI	<i>Alternate Direction Implicit Method</i>
AMG	<i>Algebraic Multigrid</i>
CDS	<i>Central Difference Scheme</i>
CFD	<i>Computational Fluid Dynamics</i>
CPU	<i>Central Processing Unit</i>
CS	<i>Correction Scheme</i>
DDS	<i>Downstream Difference Scheme</i>
FAS	<i>Full Approximation Scheme</i>
FMG	<i>Full Multigrid</i>
GB	gigabytes
GMG	<i>Geometric Multigrid</i>
<i>max</i>	valor máximo
MB	megabytes
MDF	Método das Diferenças Finitas
MEC	Método dos Elementos de Contorno
MEF	Método dos Elementos Finitos
MG	<i>Multigrid</i>
MSI	<i>Modified Strongly Implicit Method</i>
MVF	Método dos Volumes Finitos
SEA	<i>Scalar Epsilon Algorithm</i>
SOR	<i>Sucessive Over-Relaxation</i>
TDMA	<i>Tridiagonal Matrix Algorithm</i>
TEA	<i>Topological Epsilon Algorithm</i>
UDS	<i>Upstream Difference Scheme</i>

LISTA DE SÍMBOLOS

A	matriz dos coeficientes
A^{-1}	inversa da matriz A
a	escalar
$a_{i,j}$	coeficientes da matriz A
c	coeficiente do ajuste de curvas
C_k	vetores pertencentes ao espaço de Hilbert
$cond(A)$	número de condicionamento da matriz A
D	vetor diferença
$\frac{d^n u}{dx^n}$	derivada de ordem n da função $u(x)$ em relação à coordenada x
$\frac{\partial^2 u}{\partial x^2}$	derivada parcial de segunda ordem da função $u(x,y)$ em relação a x
$\frac{\partial^2 u}{\partial y^2}$	derivada parcial de segunda ordem da função $u(x,y)$ em relação a y
e	erro algébrico
$E^{(n)}$	E-algorithm
e_j	j -ésima componente do vetor erro
e_k	transformação de Shanks
E_n	erro de iteração
E_π	erro de máquina
f	vetor independente
F	operador que define o processo iterativo
G	operador diagonal
h	distância entre dois nós consecutivos nas direções coordenadas na malha mais fina Ω^h
H	distância entre dois nós consecutivos nas direções coordenadas na malha imediatamente mais grossa Ω^H
H_k	determinante de Hankel
h_x	tamanho da malha na direção x

h_y	tamanho da malha na direção y
I	operador identidade
I_h^{2h}	operador de restrição
I_{2h}^h	operador de prolongação
ITE	iteração externa, ou ciclo V
ITE_{\max}	número máximo de iterações externas
k	número do modo de Fourier
L	número de níveis do <i>Multigrid</i>
l_2	norma Euclidiana
Li	operador linear
L_x	comprimento do domínio bidimensional: direção x
L_y	comprimento do domínio bidimensional: direção y
l_∞	norma infinito
M	pico de memória
N	número de incógnitas ou subintervalos do problema
$n!$	fatorial do número n
N_x	número de incógnitas: direção x
N_y	número de incógnitas: direção y
p	ordem de complexidade do solver dado pelo ajuste de curvas
Q	operador quadrático
$q^{(k)}$	fator de convergência empírico
$\hat{q}^{(k)}$	fator de convergência média empírico
r	razão de engrossamento
R	resíduo
\Re	conjunto dos números reais
$R(x)$	função racional
S	termo fonte
s	limite da sequência S_n
S_p	<i>speed-up</i>
S_n	sequência de números reais ou complexos
T	temperatura e operador da transformação de sequências

t_{CPU}	tempo de CPU
T_n	sequência transformada
u	vetor incógnita
$u(x)$	função genérica
v	solução aproximada para a variável u
x_n	sequência escalar
x, y	variáveis do sistema cartesiano bidimensional
x_i, y_i	componentes das coordenadas espaciais x e y
y	vetor arbitrário
z^{-1}	inverso do vetor z

Símbolos

$\ \cdot\ _\infty$	norma infinito
$\ \cdot\ _2$	norma euclidiana ou norma l_2
\approx	aproximadamente igual
\neq	diferente
\forall	para todo
\Leftrightarrow	se, e somente se
\exists	existe
$\langle \cdot, \cdot \rangle$	produto interno

Letras gregas

α, β	vetores quaisquer
Δ	operador diferença
Δ^2	transformação de Aitken
ε	tolerância
$\varepsilon^{(n)}$	solução extrapolada usando o algoritmo Épsilon
κ_T	núcleo da transformação T_n
λ	escalar

μ	ciclo <i>Multigrid</i>
ν_1	número de pré-suavização
ν_2	número de pós-suavização
$\rho^{(n)}$	solução extrapolada usando o algoritmo Rho
ϕ_n	solução obtida na n -ésima iteração
ϕ_∞^{Aitken}	solução extrapolada usando o extrapolador Aitken
$\phi_\infty^{Empírico}$	solução extrapolada usando o extrapolador Empírico
ϕ_∞^{Mitin}	solução extrapolada usando o extrapolador Mitin
$\phi_{g,m}$	g -ésima solução no m -ésimo nível de extrapolação
Ω	malha correspondente ao domínio contínuo
Ω^h	malha mais fina
Ω^H	malha imediatamente mais grossa

Subscritos

d, i, j, k, n, p	contadores
E	nó localizado à direita do nó P
k_{max}	número máximo de iterações do <i>solver</i> Gauss-Seidel
N	nó localizado ao norte do nó P
P	nó genérico da malha
S	nó localizado ao sul do nó P
W	nó localizado à esquerda do nó P

Sobrescritos

$h, 2h, 3h, \dots$	indicadores da posição da malha
it	iteração atual
k	k -ésima iteração do <i>solver</i> Gauss-Seidel
m, n, p	contadores

SUMÁRIO

1 INTRODUÇÃO	20
1.1 GENERALIDADES SOBRE CFD, MÉTODO <i>MULTIGRID</i> E MÉTODOS DE EXTRAPOLAÇÃO	20
1.2 MOTIVAÇÃO	24
1.3 REVISÃO BIBLIOGRÁFICA	24
1.4 DEFINIÇÃO DO PROBLEMA	27
1.5 OBJETIVOS	28
1.6 ORGANIZAÇÃO DA DISSERTAÇÃO	28
2 FUNDAMENTAÇÃO TEÓRICA	29
2.1 MÉTODOS NUMÉRICOS APLICADOS A EQUAÇÕES DIFERENCIAIS	29
2.1.1 Método das diferenças finitas	30
2.2 MÉTODOS PARA RESOLUÇÃO DE SISTEMAS LINEARES	33
2.2.1 Gauss-Seidel lexicográfico e <i>red-black</i>	33
2.3 ERROS NUMÉRICOS	35
2.4 MÉTODO <i>MULTIGRID</i>	37
2.4.1 Equação residual	37
2.4.2 A filosofia do método <i>Multigrid</i>	38
2.4.3 Operadores de Restrição	42
2.4.4 Operadores de Prolongação	43
2.4.5 Ciclos	45
2.4.6 Esquemas	45
2.4.7 Algoritmos	46
2.5 MÉTODOS DE EXTRAPOLAÇÃO	46
2.5.1 Extrapolador Aitken	52
2.5.2 Extrapolador Empírico	53
2.5.3 Extrapolador Mitin	53
2.5.4 Extrapolador Épsilon escalar	54
2.5.5 Extrapolador Épsilon topológico	55
2.5.6 Extrapolador Rho escalar	57
2.5.7 Extrapolador Rho topológico	58
2.5.8 Extrapolações múltiplas	59
2.6 RESUMO DO CAPÍTULO	60

3 MODELOS MATEMÁTICO E NUMÉRICO	61
3.1 MODELO MATEMÁTICO	61
3.2 MODELO NUMÉRICO	62
3.3 DADOS DE IMPLEMENTAÇÃO.....	63
3.4 RESUMO DO CAPÍTULO	65
4 MÉTODOS DE EXTRAPOLAÇÃO ASSOCIADOS AO MÉTODO <i>MULTIGRID</i>	66
4.1 USO DE EXTRAPOLADORES AO FINAL DO MÉTODO <i>MULTIGRID</i>	66
4.1.1 Parâmetros utilizados para análise dos resultados	68
4.1.2 Casos estudados	68
4.1.3 Resultados	69
4.1.4 Análise dos resultados e discussão	73
4.2 USO DE EXTRAPOLADORES DURANTE O MÉTODO <i>MULTIGRID</i>	78
4.2.1 Resultados	78
4.2.2 Análise dos resultados e discussão	80
4.3 TESTES ADICIONAIS	82
4.4 RESUMO DO CAPÍTULO 4	86
5 CONCLUSÃO.....	87
5.1 CONSTATAÇÕES GERAIS	87
5.2. CONTRIBUIÇÕES	88
5.3 TRABALHOS FUTUROS	88
REFERÊNCIAS	90
APÊNDICE A – FIGURAS	96
APÊNDICE B – TESTES ADICIONAIS	101

1 INTRODUÇÃO

Neste capítulo são descritas algumas generalidades da Dinâmica dos Fluidos Computacional (*Computational Fluid Dynamics*, CFD), sobre o método *Multigrid* e métodos de extrapolação. Será descrita a motivação para a realização do trabalho, a revisão bibliográfica, a definição do problema, os objetivos gerais e específicos desta pesquisa e a forma com que a dissertação está organizada.

1.1 GENERALIDADES SOBRE CFD, MÉTODO *MULTIGRID* E MÉTODOS DE EXTRAPOLAÇÃO

Segundo Maliska (2004) a dinâmica dos fluidos envolve a modelagem de fenômenos físico-químicos relacionados à mecânica dos fluidos, transferência de calor e massa, combustão, entre outros. A resolução desses problemas por meio de métodos numéricos é o campo de estudo da CFD.

De acordo com Fortuna (2000), o objetivo principal em CFD é reduzir o número de experimentos e explorar fenômenos que não poderiam ser estudados em laboratório de forma prática. Para obter a solução de um problema em engenharia podem ser empregados três tipos de métodos: experimentais, analíticos e numéricos, sendo que cada um possui suas vantagens e desvantagens. A Tab. 1.1, adaptada de Tannehill et al. (1997) apresenta uma comparação dos métodos acima citados.

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

Método	Vantagens	Desvantagens
Experimental	Reproduz a realidade.	Equipamento exigido, problemas de escala, dificuldades de medição, custo operacional.
Analítico	Mais geral, fórmula fechada.	Restrita a geometrias e processos físicos simples, geralmente restrita a problemas lineares, erros de modelagem.
Numérico	Não há restrição quanto à linearidade, geometrias e processos complicados, evolução temporal do processo, etc.	Erros numéricos, prescrição das condições de contorno de forma apropriada, custo computacional, erros de modelagem.

Em geral, na descrição qualitativa de um fenômeno físico, os modelos matemáticos resultantes são expressos por sistemas de equações diferenciais, válidas em todo o domínio do problema. A esse sistema são impostas condições de contorno e/ou condições iniciais apropriadas. Um sistema de equações diferenciais constitui um modelo contínuo, nos quais, com exceção de casos mais simples, as soluções analíticas não são conhecidas, ou são difíceis de serem determinadas. Recorre-se, então, aos modelos discretos (ou numéricos), nos quais o problema real, contínuo, é solucionado para uma quantidade finita de pontos do domínio (discretização do domínio). Esse conjunto de pontos é denominado malha.

Para cada ponto da malha, deve-se, então, realizar a discretização das equações diferenciais parciais que regem o fenômeno físico. Isso é feito através de aproximações, que podem ser feitas de diferentes formas, de acordo com o método numérico adotado. Dentre os métodos numéricos mais usuais, citam-se: o Método dos Elementos Finitos (MEF) (HUGHES, 2000; REDDY e GARTLING, 1994), o Método dos Volumes Finitos (MVF) (MALISKA, 2004; VERSTEEG e MALALASEKERA, 2007), o Método dos Elementos de Contorno (MEC) (BREBBIA et al., 1984), o Método das Diferenças Finitas (MDF) (CHAPRA e CANALE, 2008; FORTUNA, 2000; TANNEHILL et al., 1997), por exemplo.

Nesse trabalho será utilizado o Método das Diferenças Finitas (MDF), que aproxima as derivadas da equação diferencial original através de fórmulas discretas de diferenças. A aplicação dessas fórmulas aos pontos escolhidos para formar a malha gera um sistema de equações algébricas. Para casos mais simples, esses sistemas são lineares. Quanto maior for o número de pontos utilizados na discretização, mais fina será a malha, menor será o erro de discretização (definição de erro de discretização será dada no capítulo 2) associado à resolução do problema, porém, será necessário um maior esforço computacional, pois os sistemas lineares gerados serão maiores.

Os sistemas lineares possuem a forma $Au = f$, onde A é a matriz dos coeficientes, f é o vetor independente (também chamado de fonte) e u é o vetor incógnita. Ao utilizar o MDF, para uma malha muito refinada, esses sistemas lineares são de grande porte e, em geral, a matriz A é esparsa, ou seja, possui um número grande de termos nulos. Além disso, para alguns tipos de malhas, surgem matrizes tipo banda, ou seja, com elementos diferentes de zero apenas na diagonal principal e em algumas diagonais paralelas a essa.

Métodos diretos, como a eliminação de Gauss, podem ser usados para resolver sistemas lineares de pequeno porte (da ordem de 10^2 incógnitas) com matrizes não esparsas. Por outro lado os métodos iterativos, como Gauss-Seidel, têm um bom desempenho com

sistemas maiores (da ordem de 10^5 incógnitas) com matrizes esparsas (BURDEN e FAIRES, 2007).

Os métodos iterativos produzem uma sequência de vetores a partir de uma aproximação inicial. Sob certas condições, esta sequência converge para a solução exata, caso ela exista. Do ponto de vista prático, busca-se resolver os problemas em malhas o mais refinadas possível, como forma de minimizar os erros de discretização. Neste caso tem-se sistemas lineares de grande porte, de modo que são os métodos iterativos os mais empregados. Neste trabalho, os métodos iterativos serão chamados de *solvers*, embora o termo possa também ser utilizado para métodos diretos.

Nas últimas décadas, um grande avanço na aceleração da taxa de convergência de processos iterativos se deu pelo desenvolvimento do método *Multigrid*, proposto originalmente por Fedorenko (1964), cuja filosofia está baseada no emprego de malhas auxiliares, mais grossas, nas quais as componentes do erro numérico são suavizadas, de modo que o processo iterativo torna-se mais rápido (detalhes serão apresentados no capítulo 2).

A implementação do método *Multigrid* é feita em duas fases: Na fase inicial, é feita a geração das malhas, construção dos operadores de transferência de informações entre malhas (restrição e prolongação). Na sequência, os elementos escolhidos na fase inicial são diretamente aplicados na resolução do problema. Essa resolução se dá com o uso de malhas auxiliares que apresentam uma quantidade menor de pontos (malhas mais grossas), que são visitadas conforme uma sequência previamente definida, a qual é chamada de ciclo (podendo ser de diversos tipos, entre V, W, F, dente-de-serra, etc).

O operador de restrição transfere as informações da malha fina para a malha imediatamente mais grossa, e o operador de prolongação transfere as informações da malha grossa para a malha imediatamente mais fina.

A suavização envolve a aplicação de um método iterativo (*solver*), também chamado de método de relaxação ou suavizador (como Gauss-Seidel, por exemplo). A correção na malha grossa envolve a resolução da equação residual nessa malha, obtendo uma aproximação para o erro. Essa aproximação é usada para corrigir a solução na malha imediatamente mais fina (o erro é transferido para essa malha) até que se retorne à malha inicial e a solução inicial lá armazenada seja corrigida.

Com o método *Multigrid* podem ser usados alguns tipos de esquemas de correção que se caracterizam por suas diferentes informações e a forma como as mesmas são transportadas (BRIGGS et al., 2000; TROTTEBERG et al., 2001; WESSELING, 1992): o esquema de correção (*Correction Scheme*, CS) e o esquema de aproximação completa (*Full*

Approximation Scheme, FAS) mais recomendados para problemas lineares e não-lineares, respectivamente. Ambos os esquemas, ainda podem usar soluções nas malhas grossas para gerar estimativas iniciais melhoradas para as malhas mais finas. O esquema que usa esta estratégia chama-se *Multigrid* completo (*Full Multigrid*, FMG).

De acordo com Trottenberg et al. (2001) e Briggs et al. (2000), o método *Multigrid* possui duas abordagens, que estão relacionadas às informações necessárias e à construção das malhas auxiliares: o método *Multigrid* geométrico (GMG), em que são necessárias informações a respeito das malhas auxiliares; e o método *Multigrid* algébrico (AMG), em que é necessária a matriz de coeficientes, que contém informações a respeito da malha inicial, convencionada como a malha com o maior número de incógnitas. O que os dois métodos têm em comum é o fato de o problema ser resolvido empregando-se uma hierarquia de malhas.

Outra forma que visa à obtenção de uma solução numérica acurada com menor esforço computacional está relacionada ao emprego de métodos de extrapolação, ou de aceleração de convergência associado ao método iterativo (BREZINSKI e ZAGLIA, 2002).

O problema de aceleração de sequências foi bem estudado nas décadas de 1960 e 1970, e resultou em uma série de métodos de extrapolação eficazes (KOKIOPOULOU e FROSSARD, 2007). Esses métodos não requerem um conhecimento explícito do gerador das sequências, e agem diretamente sobre os termos da sequência de vetores, podendo ser aplicados diretamente na sequência de sistemas lineares e não-lineares. Os métodos de extrapolação podem ainda ser classificados em escalares e vetoriais, de acordo com a forma com que suas informações são manipuladas.

Os extrapoladores (ou métodos de extrapolação) escalares utilizados nesse trabalho são: Aitken, que acelera a convergência de sequências que convergem linearmente; Empírico, que é uma forma modificada do extrapolador Aitken; Mitin, que é recomendado para sequências oscilantes; Épsilon escalar, recomendado na literatura como o melhor método para fins de aceleração de sequências que convergem lentamente e Rho escalar, que não acelera sequências com convergência linear, mas é muito poderoso para sequências logaritmicamente convergentes (GAO et al., 2010; GRAVES-MORRIS et al., 2000).

Os dois últimos métodos comentados (Épsilon e Rho) podem ser generalizados para o caso vetorial e, além disso, com base no estudo de Brezinski e Zaglia (2002), a formulação topológica destes dois métodos foi utilizada (Épsilon topológico e Rho topológico).

Além dos extrapoladores citados, extrapolações múltiplas (extrapolação de dados já extrapolados) também foram realizadas com Aitken e Mitin. Maiores detalhes serão apresentados no capítulo 2.

1.2 MOTIVAÇÃO

Encontrar a solução de sistemas de equações lineares e não-lineares é um problema importante em matemática computacional. Métodos iterativos são amplamente utilizados para encontrar essas soluções. Entretanto, esse problema é muitas vezes difícil de ser resolvido, devido à convergência lenta do processo iterativo adotado (MITIN, 1985). Uma possível solução é a associação de métodos de extrapolação (ou aceleradores de convergência), aos métodos iterativos. Métodos de extrapolação constituem-se em uma ferramenta poderosa para a obtenção de resultados numéricos de alta precisão com pouco esforço computacional (LIMA e GRAÇA, 1995).

O método *Multigrid* tem se mostrado extremamente eficiente na resolução de sistemas de equações lineares resultantes da discretização de equações diferenciais. Para uma ampla classe de diferentes problemas, o *Multigrid* é considerado um método eficiente, como pode ser verificado nos resultados de Fedorenko (1964) e Brandt (1977).

Tendo-se em vista que tanto os métodos de extrapolações quanto o método *Multigrid* visam a aceleração de convergência de sistemas lineares, mas por mecanismos diferentes, torna-se atraente a ideia de associar tais metodologias em busca de um método que agregue as características de ambos, aumentando ainda mais a taxa de convergência dos métodos e com isso obtendo uma redução no erro de iteração e redução no tempo de CPU.

1.3 REVISÃO BIBLIOGRÁFICA

A rapidez com que um sistema linear é resolvido por meio de um método iterativo pode ser avaliada através de um parâmetro denominado taxa de convergência. Por isso, diversos trabalhos têm como foco o aumento dessa taxa. Dentre as diferentes possibilidades, um método bastante eficiente e que tem sido muito empregado é o chamado método *Multigrid* (BRIGGS et al., 2000; WESSELING, 1992).

O objetivo do método *Multigrid* é acelerar a convergência de um processo iterativo (TANNEHILL et al., 1997).

As conclusões de Fedorenko (1964) mostraram que a velocidade de convergência do *Multigrid* é melhor que a dos métodos iterativos puros.

Segundo Wesseling (1992), os melhores desempenhos do método *Multigrid* são obtidos em problemas totalmente elípticos, ou seja, problemas dominados pela difusão; e os menores, de acordo com Ferziger e Peric (2002), em problemas dominados pela advecção.

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

Wesseling e Oosterlee (2001) citam vários desafios relacionados ao uso do *Multigrid* geométrico, como a solução das equações de Navier-Stokes, problemas com perturbações singulares, problemas de camada limite onde aparecem as malhas fortemente distorcidas (anisotropia geométrica), ou mesmo a paralelização de algoritmos.

Brandt (1977) mostrou preferência pelo esquema CS em relação ao esquema FAS no caso de problemas lineares. Segundo Brandt, cada ciclo iterativo do esquema FAS é mais caro computacionalmente se comparado ao esquema CS devido à transferência de informações a respeito do resíduo e da solução para as malhas mais grossas.

Segundo Roache (1998), um aumento na eficiência do método permite no mesmo tempo computacional, resolver um problema em uma malha mais refinada, isto é, com maior número de nós. Assim, pode-se obter uma solução numérica com menor erro de discretização, melhorando a qualidade e a confiabilidade dos resultados.

Em Trottenberg et al. (2001), experiências com o método *Multigrid* mostram que seus parâmetros (número de malhas, suavizador ou *solver*, número de iterações no *solver*, ciclos, formas de percorrer um ciclo e os esquemas de restrição e interpolação) podem ter uma forte influência na eficiência do algoritmo. Segundo Trottenberg et al. (2001), não há regras gerais na escolha destes parâmetros, porém certos valores podem ser recomendados para determinadas situações.

O grupo de pesquisa liderado pelo prof. Carlos Henrique Marchi, da Universidade Federal do Paraná, tem dedicado esforços no estudo de parâmetros ótimos do *Multigrid*; os principais resultados podem ser encontrados nas teses de doutorado de Oliveira (2010); Pinto (2006); nos trabalhos apresentados em congressos como Oliveira et al. (2006); Oliveira et al. (2010); Pinto et al. (2005); Pinto e Marchi (2006, 2007); Santiago et al. (2010); Santiago e Marchi (2007) e Suero et al. (2010); no artigo de revista de Oliveira et al. (2012) ou em <ftp://ftp.demec.ufpr.br/CFD>.

Muitos métodos iterativos são usados na resolução de sistemas de equações. Se o método produz uma convergência lenta, não é muito útil em situações práticas. Então, pode-se modificar o processo iterativo a fim de obter maior eficácia ou transformar a sequência que

converge lentamente em outra, com melhores propriedades de convergência. Uma das abordagens para este caso trata de métodos de extrapolação, que consistem em transformações de sequências.

De acordo com Brezinski e Zaglia (2008), em processos iterativos ‘lentos’, duas estratégias podem ser adotadas: modificar o processo iterativo ou transformar a sequência que converge lentamente em outra, com melhores propriedades de convergência. Para a primeira alternativa é necessário ter acesso ao processo que produz a sequência, o que nem sempre é possível (ou fácil). No segundo caso, o processo que produz a sequência é considerado uma “caixa preta”, e a sequência que converge lentamente se transforma em uma nova sequência sem que se modifique o processo iterativo.

Mitin (1985) considera um processo de aceleração de convergência de métodos iterativos, para solução de sistemas de equações através de métodos de extrapolação linear de sequências oscilatórias, propondo novos métodos de extrapolação linear e abordando a equivalência entre outros.

Evans (1963) descreve uma nova abordagem para extrapolação (com relaxamento) e aceleração linear do método iterativo de Aitken, e apresenta provas teóricas e experimentais na qual as aproximações das derivadas eram feitas com base em cinco pontos para equações diferenciais elípticas. Conclui-se que esta modificação do método de Aitken apresentou resultados comparáveis ao método de Sobre Relaxação Sucessiva (*Successive Over-Relaxation*, *SOR*) acelerado.

Os resultados de Homeier et al. (1995) sugerem o estudo da combinação de iteração direta com métodos de extrapolação. Ressalta-se que é desejável o desenvolvimento de novos processos de extrapolação vetoriais e que os propostos em seu trabalho ainda têm potencial para melhorias.

No trabalho de Jbilou e Sadok (2000) é proposta uma revisão dos métodos de extrapolação vetoriais mais conhecidos, ou seja, os polinomiais e os algoritmos Épsilon. Consideraram-se, também, algumas comparações numéricas entre esses métodos. Segundo os resultados apresentados por esses autores, os algoritmos Épsilon exigem mais capacidade de armazenamento e operações, em comparação aos métodos polinomiais, porém, há a vantagem de generalizarem de uma forma simples e linear, problemas não-lineares.

O trabalho de Mesgarani et al. (2010) utiliza Aitken e Épsilon em um problema que já havia sido acelerado com o método de extrapolação de Richardson. Através de resultados numéricos, verificou-se que uma maior precisão nos resultados dos testes é obtida através da aplicação de Aitken ou de Épsilon apenas uma vez do que usando Richardson duas vezes.

Em Lima e Graça (1995) é aplicado o algoritmo Épsilon para acelerar a convergência de um problema de valor de contorno modelado por uma equação diferencial de segunda ordem, onde o lado direito (termo fonte) pode apresentar várias singularidades. A equação foi discretizada usando-se o MDF. Uma expansão assintótica do erro é obtida, permitindo o uso do algoritmo Épsilon para acelerar a convergência. A aplicabilidade e a eficiência do algoritmo Épsilon são discutidas e ilustradas por exemplos numéricos, obtendo-se bons resultados, ao serem comparados às simulações realizadas quando é utilizado o extrapolador de Richardson.

Em Skelboe (1980) é apresentada uma análise unificada de três métodos de extrapolação: a extrapolação Épsilon escalar e vetorial e a extrapolação por polinômio mínimo. Como resultado principal, tem-se um teorema que garante a convergência quadrática dos métodos de extrapolação para problemas lineares.

A aplicação do extrapolador Épsilon vetorial e métodos de extrapolação Rho na aceleração do algoritmo de Richardson-Lucy e sua versão amortecida (métodos para restauração de imagens) são mostradas por Gao et al. (2010). Os resultados mostram que extrapolações Épsilon e Rho podem acelerar a convergência de maneira bastante eficiente, e no caso estudado, o método Rho mostrou-se mais eficiente que o Épsilon.

1.4 DEFINIÇÃO DO PROBLEMA

Nesta dissertação serão abordadas componentes algorítmicas do método *Multigrid* para o problema de condução de calor bidimensional em regime permanente descrito pela equação de Poisson (maiores detalhes no capítulo 3), usando-se extrapolações associadas ao processo iterativo.

Os parâmetros analisados são: tempo de CPU, resíduo adimensionalizado com base na estimativa inicial, normas do erro de iteração, fator de convergência empírico e memória de armazenamento. Esses parâmetros são determinados a partir da obtenção das soluções numéricas, e para serem comparados, diferentes variantes do método numérico de obtenção das soluções do modelo matemático são propostas.

As variantes nas metodologias de resolução do modelo matemático serão divididas em dois grupos: o primeiro, chamado de extrapolações ao final do *Multigrid*, no qual as extrapolações serão feitas ao final do processo iterativo, após a solução numérica ter sido obtida; e o segundo, denominado extrapolações durante o *Multigrid*, no qual o extrapolador de

melhor desempenho nas análises feitas com as soluções do primeiro grupo será utilizado durante o processo iterativo (maiores detalhes no capítulo 4).

Depois de realizadas todas as simulações, será possível determinar qual variante dos métodos de resolução do modelo matemático que apresentou os melhores resultados, ou seja, aquele que conseguiu acelerar o processo iterativo, reduzir o erro de iteração, analisando a memória utilizada na resolução do problema.

1.5 OBJETIVOS

Neste trabalho pretende-se apresentar os resultados de testes com alguns extrapoladores associados ao método *Multigrid*.

Os objetivos gerais são:

- acelerar o método *Multigrid* com o uso de extrapoladores (reduzir o tempo de CPU);
- reduzir o erro de iteração com o uso de extrapoladores associados ao método *Multigrid*.

Os objetivos específicos são:

- analisar o comportamento de diversos extrapoladores existentes na literatura;
- definir qual a melhor metodologia a ser empregada no processo de extrapolação associada ao método Multigrid.

1.6 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está organizada em seis capítulos e dois apêndices, da seguinte forma: No capítulo 1 foram apresentadas algumas generalidades em CFD, método *Multigrid* e métodos de extrapolação, a motivação, a revisão bibliográfica, a definição do problema e os objetivos desta pesquisa. O capítulo 2 traz a fundamentação teórica necessária para um bom desenvolvimento do texto. O capítulo 3 é destinado à descrição detalhada dos modelos matemático e numérico estudados e detalhes de implementação. O capítulo 4 apresenta os resultados das simulações realizadas para resolver o modelo matemático, a análise e discussão dos resultados dos parâmetros de interesse. O capítulo 5 é destinado à conclusão da dissertação e, em seguida, as referências bibliográficas. No apêndice A têm-se as figuras complementares e no apêndice B, os resultados dos testes adicionais que foram realizados.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentada uma fundamentação teórica sobre os métodos numéricos aplicados a equações diferenciais, métodos iterativos básicos, erros numéricos, princípios básicos do método *Multigrid* e os métodos de extrapolação.

2.1 MÉTODOS NUMÉRICOS APLICADOS A EQUAÇÕES DIFERENCIAIS

Os métodos numéricos aplicados a equações diferenciais têm por finalidade substituir os termos das derivadas envolvidas na equação por aproximações numéricas. Quando são utilizados, o problema contínuo é transformado em um problema discreto, em que o domínio espacial é dividido em um número finito de pontos (ou nós). Essa transformação é conhecida como discretização do domínio e resulta em um conjunto de pontos, que recebe o nome de malha computacional, ou simplesmente malha.

Em problemas bidimensionais, o domínio $\{(x, y) \in \mathfrak{R}^2 : 0 \leq x \leq L_x; 0 \leq y \leq L_y\}$ é dividido em subconjuntos através de um número de pontos dado por $N = N_x N_y$, onde N_x e N_y são os números de pontos nas direções coordenadas x e y , respectivamente. Introduce-se, então, uma malha com os pontos $(x_i, y_j) = ((i-1)h_x, (j-1)h_y)$, com $h_x = L_x / (N_x - 1)$ e $h_y = L_y / (N_y - 1)$, onde $i = 1, \dots, N_x$, $j = 1, \dots, N_y$ e h_x e h_y são os comprimentos de cada subintervalo do domínio, nas direções coordenadas x e y , respectivamente. A malha estabelecida é denotada por Ω^h . Neste trabalho, $L_x = L_y = 1$ e $N_x = N_y$, assim, $h_x = h_y = h$. A Fig. 2.1 a seguir ilustra essa ideia para um domínio quadrado.

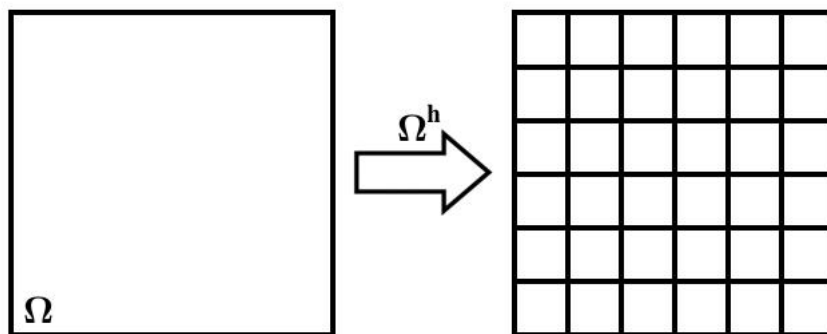


Figura 2.1: Exemplo de transformação do domínio contínuo em domínio discreto.

Na Fig. 2.1, Ω representa o domínio contínuo e Ω^h , o domínio discretizado, em que h é a distância entre dois nós consecutivos nas direções coordenadas (cruzamento entre as linhas horizontal e vertical). A malha resultante da discretização poderá ser classificada em:

- uniforme: caso o valor de h seja constante em uma mesma direção;
- não-uniforme: caso h varie em uma direção;

Para este trabalho, a malha obtida após a discretização do domínio será uniforme nas duas direções e estruturada, como a da Fig. 2.1, por exemplo.

Existem, na literatura, muitos métodos que podem ser utilizados na aproximação das equações diferenciais parciais do modelo matemático, resultando no modelo numérico, no qual as equações originais foram discretizadas. Estão entre esses métodos, o Método dos Elementos Finitos (MEF), o Método dos Volumes Finitos (MVF), o Método dos Elementos de Contorno (MEC), o Método das Diferenças Finitas (MDF), por exemplo, sendo que este último foi o método utilizado neste trabalho devido à facilidade de implementação, literatura bem estabelecida, resultados para comparação, etc.

Em todos os métodos citados, as discretizações do modelo matemático conduzem a um sistema de equações algébricas, lineares, ou não-lineares, do tipo $Au = f$, sendo que as propriedades físicas envolvidas passam a ser descritas em função de um conjunto de valores discretos, obtidos com a utilização de métodos e/ou aproximações adequados para cada ponto da malha computacional. Em Wesseling e Oosterlee (2001) podem ser encontradas mais informações sobre métodos de discretização.

2.1.1 Método das diferenças finitas

Segundo Patankar (1980) e Ferziger e Peric (2002), a ideia do MDF é obter a solução aproximada de uma equação diferencial em pontos ou nós do domínio, substituindo (ou aproximando) as derivadas da equação original por fórmulas discretas de diferenças. A aplicação dessas fórmulas aos pontos do domínio discretizado gera um sistema de equações algébricas, cuja solução fornece os valores das incógnitas do problema nesses pontos.

Expansões em série de Taylor ou ajuste polinomial (FORTUNA, 2000) são usados para obter as aproximações para as derivadas das variáveis com respeito às coordenadas.

Por definição, a derivada de uma função $u(x)$ em um ponto x_i é dada por

$$\left. \frac{du}{dx} \right)_{x=x_i} = \lim_{h \rightarrow 0} \frac{u(x_i + h) - u(x_i)}{h}. \quad (2.1)$$

A expansão em série de Taylor do valor de u em $x = x_i + h$ em torno do valor de u em $x = x_i$ é

$$u(x_i + h) = u(x_i) + h \left. \frac{du}{dx} \right)_{x=x_i} + \frac{h^2}{2!} \left. \frac{d^2u}{dx^2} \right)_{x=x_i} + \frac{h^3}{3!} \left. \frac{d^3u}{dx^3} \right)_{x=x_i} + \dots \quad (2.2)$$

A Eq. (2.2) pode ser reescrita como:

$$\left. \frac{du}{dx} \right)_{x=x_i} = \frac{u(x_i + h) - u(x_i)}{h} - \frac{h}{2!} \left. \frac{d^2u}{dx^2} \right)_{x=x_i} + \frac{h^2}{3!} \left. \frac{d^3u}{dx^3} \right)_{x=x_i} - \dots \quad (2.3)$$

Para pequenos valores de h , pode-se truncar a série, desprezando-se os termos relativos às derivadas de ordem igual ou superior a dois, obtendo-se assim, a forma aproximada, para a primeira derivada, que neste caso, chama-se diferença progressiva (*Forward Difference Scheme*, FDS):

$$\left. \frac{du}{dx} \right)_{x=x_i} \cong \frac{u(x_i + h) - u(x_i)}{h}. \quad (2.4)$$

De maneira análoga, pode-se definir uma aproximação para a primeira derivada por diferença regressiva (*Backward Difference Scheme*, BDS):

$$\left. \frac{du}{dx} \right)_{x=x_i} \cong \frac{u(x_i) - u(x_i - h)}{h} \quad (2.5)$$

e diferença central (*Central Difference Scheme*, CDS):

$$\left. \frac{du}{dx} \right)_{x=x_i} \cong \frac{u(x_i + h) - u(x_i - h)}{2h}. \quad (2.6)$$

Somando-se as expansões em série de Taylor que dão origem às aproximações das derivadas de primeira ordem por diferenças progressiva e regressiva

$$u(x_i + h) - u(x_i - h) = 2u(x_i) + h^2 \left. \frac{d^2 u}{dx^2} \right|_{x=x_i} + \frac{2h^4}{4!} \left. \frac{d^4 u}{dx^4} \right|_{x=x_i} + \dots \quad (2.7)$$

e isolando o termo que contém a derivada de segunda ordem, tem-se:

$$\left. \frac{d^2 u}{dx^2} \right|_{x=x_i} = \frac{u(x_i + h) - 2u(x_i) + u(x_i - h)}{h^2} - \frac{h^2}{12} \left. \frac{d^4 u}{dx^4} \right|_{x=x_i} + \dots \quad (2.8)$$

Truncando a série, desprezando-se os termos relativos às derivadas de ordem igual ou superior a quatro, chega-se à expressão da aproximação da derivada de segunda ordem (*Central Difference Scheme*, CDS) que será utilizada neste trabalho,

$$\left. \frac{d^2 u}{dx^2} \right|_{x=x_i} \cong \frac{u(x_i + h) - 2u(x_i) + u(x_i - h)}{h^2} \quad (2.9)$$

A ordem de acurácia (ou ordem assintótica) das aproximações das derivadas está relacionada ao expoente de h do primeiro termo desprezado da série de Taylor. A escolha por uma ou outra aproximação está relacionada, entre outros fatores, à acurácia desejada para o modelo numérico. Esse é o principal motivo para que neste trabalho seja usado o esquema CDS, pois assim, o modelo numérico apresenta ordem de acurácia igual a 2.

A Eq. (2.9) pode ser estendida para o caso de derivadas parciais:

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{(x,y)=(x_i,y_j)} \cong \frac{u(x_i + h_x, y_j) - 2u(x_i, y_j) + u(x_i - h_x, y_j)}{h_x^2} \quad (2.10)$$

e

$$\left. \frac{\partial^2 u}{\partial y^2} \right|_{(x,y)=(x_i,y_j)} \cong \frac{u(x_i, y_j + h_y) - 2u(x_i, y_j) + u(x_i, y_j - h_y)}{h_y^2} \quad (2.11)$$

2.2 MÉTODOS PARA RESOLUÇÃO DE SISTEMAS LINEARES

Segundo Cunha (2003), os métodos numéricos utilizados para a obtenção das soluções de sistemas lineares (também chamados de *solver*) podem ser divididos em dois grupos: os métodos diretos e os métodos iterativos. Com relação aos métodos diretos, seu uso apresenta a vantagem de fornecer, após um número finito de passos, a solução do sistema, não dependendo de critérios de convergência. Os métodos diretos mais utilizados (FORTUNA, 2000) são: eliminação de Gauss, fatoração LU, TDMA (*Tridiagonal Matrix Algorithm*), entre outros. O uso de métodos diretos pode ser inviável quando o sistema de equações é muito grande ou mal condicionado (BURDEN e FAIRES, 2007).

Os métodos iterativos produzem uma sequência de vetores a partir de uma aproximação inicial. Sob certas condições esta sequência converge para a solução exata, caso ela exista. Segundo Fortuna (2000), os métodos iterativos apresentam a vantagem de serem aplicáveis quando o sistema de equações é não-linear. De acordo com Burden e Faires (2007), eles são vantajosos quando se tem problemas com muitas incógnitas, além de aproveitarem a esparsidade da matriz de coeficientes. Alguns exemplos de métodos iterativos: Jacobi, Gauss-Seidel, (BURDEN e FAIRES, 2007), MSI (*Modified Strongly Implicit Method*) (SCHNEIDER e ZEDAN, 1981), ADI (*Alternate Direction Implicit Method*) (FERZIGER e PERIC, 2002), entre outros.

Neste trabalho, assim como em muitos problemas de interesse prático da engenharia, o sistema de equações resultante da discretização apresenta um número grande de variáveis, o que inevitavelmente gera matrizes de grande porte. Devido à natureza do método numérico escolhido para discretização, MDF, essas matrizes são esparsas. Isso motiva o uso de um método iterativo para resolver o problema, e o escolhido foi o Gauss-Seidel *red-black*. Tal ordenação beneficia a computação paralela, mas também pode apresentar bons resultados em computadores seriais (OLIVEIRA, 2010 e ZHANG, 1996).

2.2.1 Gauss-Seidel lexicográfico e *red-black*

Em CFD, matrizes pentadiagonais ocorrem com frequência, principalmente em problemas bidimensionais quando são utilizadas aproximações de segunda ordem para as derivadas com o uso do método das diferenças finitas.

Com a equação diferencial governante para o caso bidimensional (equação de Poisson – mais detalhes no capítulo 3) forma-se uma equação algébrica para cada nó. Um exemplo para esquemas de 5 pontos está apresentado na Eq. (2.12).

$$a_{i,j}u_{i,j} + a_{i-1,j}u_{i-1,j} + a_{i+1,j}u_{i+1,j} + a_{i,j-1}u_{i,j-1} + a_{i,j+1}u_{i,j+1} = f_{i,j}, \quad (2.12)$$

onde, de acordo com o esquema dado pela Fig. 2.2, tem-se:

$$a_P = a_{i,j}; a_W = a_{i-1,j}; a_E = a_{i+1,j}; a_S = a_{i,j-1}; a_N = a_{i,j+1} \quad (2.13)$$

e

$$u_P = u_{i,j}; u_W = u_{i-1,j}; u_E = u_{i+1,j}; u_S = u_{i,j-1}; u_N = u_{i,j+1}; f_{i,j} = f_P. \quad (2.14)$$

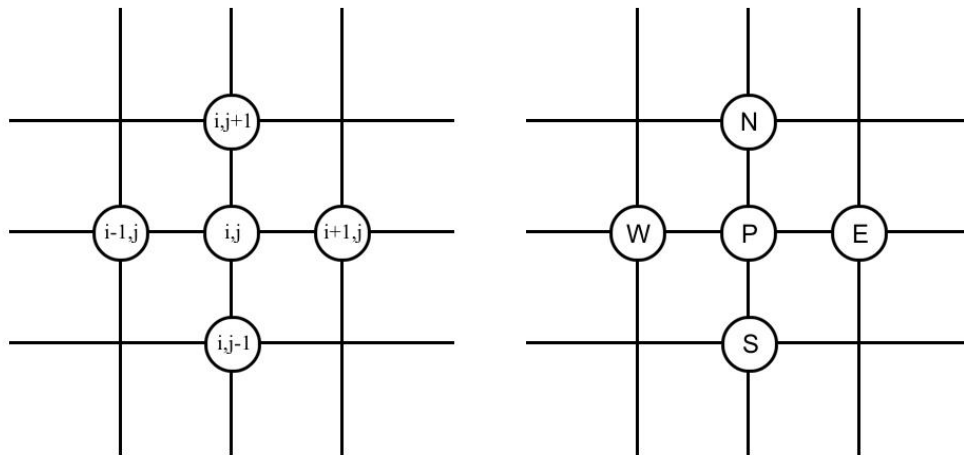


Figura 2.2: Convenção para nomenclatura dos vizinhos dos nós da malha bidimensional uniforme.

O conjunto de equações na forma da Eq. (2.12) escrita para cada par (i, j) do domínio, conduz a uma equação matricial da forma $Au = f$. O método de Gauss-Seidel (BURDEN e FAIRES, 2007) é um método iterativo usado para se resolver esse tipo de sistema de equações. Esse método resolve o sistema visitando equação por equação, iterativamente, usando-se em um mesmo ciclo, os valores das variáveis já calculadas nesse ciclo iterativo. A Eq. (2.12) pode ser reescrita, de acordo com o ordenamento lexicográfico (Fig. 2.3a), ou seja, de oeste (W) para leste (E) e do sul (S) para o norte (N), apresentada por Wesseling (1992), como:

$$u_P^{k+1} = -\left(a_W u_W^{k+1} + a_E u_E^k + a_S u_S^{k+1} + a_N u_N^k - f_P\right) / a_P, \quad (2.15)$$

onde o superíndice k representa a k -ésima iteração e o subíndice a posição do nó na malha computacional.

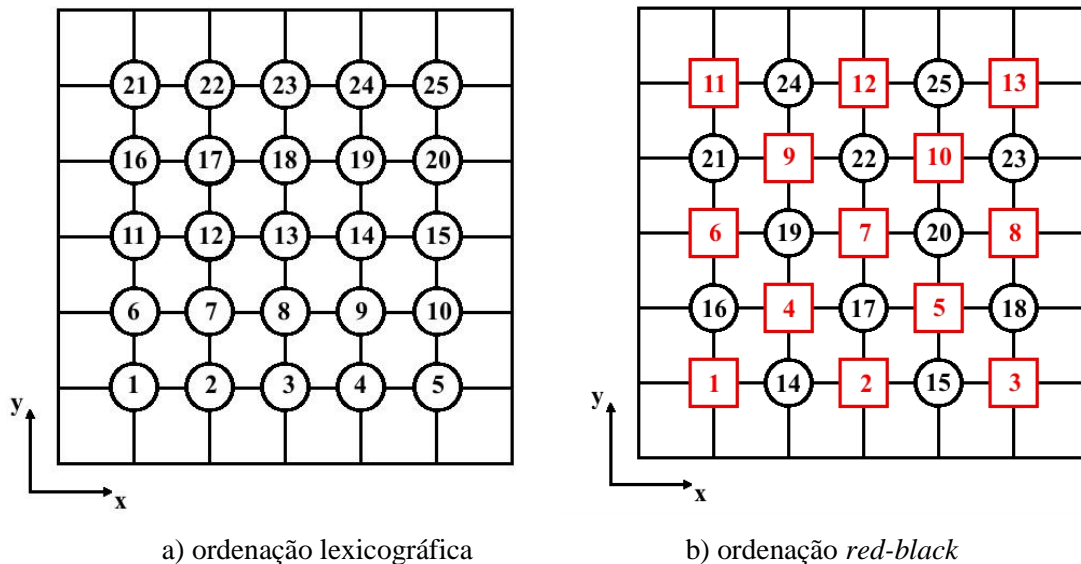


Figura 2.3: Ordenação para Gauss-Seidel lexicográfico e *red-black*.

Verifica-se na Eq. (2.15) que para atualizar cada ponto são necessários os quatro vizinhos mais próximos, de acordo com a convenção dada pela Fig. 2.2. Se a malha for reorganizada utilizando a ordenação *red-black*, (Fig. 2.3b) a atualização pode ser realizada simultaneamente nos pontos definidos como vermelhos (quadrados) e pretos (circunferências). Em ambos os casos, a ordenação é feita apenas nos pontos internos, pois para o modelo envolvido nesta dissertação, a solução nos contornos é conhecida. Mais detalhes sobre os tipos de ordenação podem ser encontrados em Trottenberg et al. (2001) e Wesseling (1992) (em cujo trabalho, a ordenação *red-black* é denominada *white-black*).

O algoritmo 2.1 apresentado a seguir descreve um procedimento para o *solver* Gauss-Seidel *red-black*, apresentada por Wesseling (1992). Maiores detalhes sobre esse algoritmo podem ser encontrados em Parter (1988) e Zhang (1996).

2.3 ERROS NUMÉRICOS

A solução de modelos matemáticos por intermédio de um método numérico pode causar diversas fontes de erro, sendo classificadas, de acordo com Marchi (2001), como:

Algoritmo 2.1: Procedimento de Gauss-Seidel *red-black* para $Au = f$

GAUSS-SEIDEL *RED-BLACK*

Inicializações: ($k = 0, k_{\max}$ e u_0)

1. Calcular u_p^{k+1} pela Eq. (2.15) para todos os pontos vermelhos (quadrados da Fig. 2.3b);
2. Calcular u_p^{k+1} pela Eq. (2.15) para todos os pontos pretos (circunferências da Fig. 2.3b);
3. Faça k receber $k + 1$;
4. Volte ao passo 1 até convergir ou até atingir k_{\max} .

Fim

- Erros de truncamento: ocorrem quando o modelo matemático é aproximado por um método numérico. Quando se tem apenas erros de truncamento (as outras fontes de erro são desprezíveis), este erro denomina-se erro de discretização.

- Erros de arredondamento: ocorrem devido à representação finita dos números reais nos cálculos. Segundo Fortuna (2000), não podem ser evitados, mas podem ter seu efeito reduzido pela utilização de precisão dupla ou quádrupla nos cálculos.

- Erros de iteração: é definido como sendo a diferença entre a solução exata das equações discretizadas (solução obtida em um processo iterativo, até que seja atingido o erro de máquina), e a solução numérica em uma determinada iteração (FERZIGER e PERIC, 2002).

- Erros de programação: são inerentes ao programador e à utilização do código computacional.

Para as simulações realizadas neste trabalho, os erros de arredondamento são minimizados devido ao emprego de precisão quádrupla nos cálculos. Os erros de programação também são minimizados devido à utilização de códigos computacionais confiáveis, ou seja, códigos que já passaram pelo processo de verificação computacional através da análise de erros numéricos (mais detalhes em Oliveira (2010)).

Segundo Martins (2002), o erro de iteração pode ser originado por diferentes fatores como: o emprego de métodos iterativos para solução de equações discretizadas, o uso de métodos segregados na obtenção de modelos constituídos por várias equações diferenciais, ou ainda pela existência de não-linearidades no modelo matemático. Em geral esse tipo de erro diminui com o aumento do número de iterações. O erro de máquina (E_π) é obtido ao

executar o programa até que os erros de iteração sejam minimizados, ou seja, sejam desprezíveis frente aos outros tipos de erro.

Os erros de truncamento são provenientes da aproximação do modelo matemático por um método numérico. Para reduzir seu efeito são usadas malhas com um grande número de nós. Neste trabalho, as simulações foram feitas usando malhas com $N = 3 \times 3$ nós até $N = 8193 \times 8193$ nós.

2.4 MÉTODO *MULTIGRID*

Nesta seção serão abordados os conceitos referentes ao método *Multigrid*, cuja filosofia está baseada no emprego de malhas auxiliares, mais grossas, nas quais as componentes do erro numérico são suavizadas, de tal modo que o processo iterativo torna-se mais rápido. Serão apresentados seus princípios fundamentais, operadores de transferência entre malhas, ciclos e algoritmos.

2.4.1 Equação residual

Considere o sistema na forma $Au = f$. Supondo que ele tenha solução única u e uma aproximação dada por v , o erro numérico é dado por:

$$e = u - v. \quad (2.16)$$

A magnitude do erro numérico pode ser medida através de normas de vetores (BRIGGS et al., 2000; BURDEN e FAIRES, 2007). As normas mais utilizadas são a norma infinito (l_∞) e a norma Euclidiana (l_2), definidas pelas Eqs. (2.17) e (2.18), respectivamente:

$$\|e\|_\infty = \max_{1 \leq j \leq N} |e_j| \quad (2.17)$$

e

$$\|e\|_2 = \left(\sum_{j=1}^N e_j^2 \right)^{\frac{1}{2}}, \quad (2.18)$$

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

Como na prática o erro, assim como a solução analítica, é desconhecido, uma forma de se avaliar sua magnitude seria avaliar o resíduo, que é utilizado para mostrar o quanto à aproximação v não satisfaz o sistema $Au = f$. O resíduo é dado por:

$$R = f - Av, \quad (2.19)$$

sendo que sua norma pode ser medida pelas mesmas normas utilizadas para o erro.

Tem-se que $e \approx 0$ implica em $R \approx 0$, porém $R \approx 0$ não implica em $e \approx 0$. Para matrizes bem condicionadas, ou seja, $cond(A) = \|A\| \cdot \|A^{-1}\| \approx 1$, se $R \approx 0$, então $e \approx 0$ (BRIGGS et al., 2000).

Lembrando que $Au = f$ e utilizando as definições de R e e , pode-se deduzir uma importante relação entre o erro e o resíduo denominada de equação residual:

$$Ae = R. \quad (2.20)$$

A equação residual informa que o erro satisfaz o mesmo conjunto de equações com a variável u quando f é substituída pelo resíduo R .

A equação residual tem uma grande vantagem. Supondo que uma aproximação v seja determinada através de um método iterativo, calcula-se o resíduo através da Eq. (2.19). Para melhorar a aproximação v , resolve-se a equação residual para e e então calcula-se uma nova aproximação usando a definição do erro:

$$u = v + e. \quad (2.21)$$

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

2.4.2 A filosofia do método *Multigrid*

Uma forma de reduzir as fortes oscilações do resíduo em cada malha é suavizá-los empregando um método de relaxação (método iterativo). Um dos métodos iterativos utilizados na literatura é o método *Multigrid*. Esse método tem como base a observação das propriedades dos métodos iterativos. A taxa de convergência depende dos autovalores da

matriz de iteração associada ao método. O raio espectral da matriz (autovalor de maior magnitude) determina quão rápida a solução é alcançada. O autovetor associado a esse autovalor determina a distribuição do erro de iteração, variando consideravelmente de método para método (FERZIGER e PERIC, 2002).

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

É importante conhecer o desempenho de métodos iterativos básicos utilizados na resolução de sistemas de equações. Esse é o motivo de se realizar uma análise de erros.

Para fazer uma análise de erros, Briggs et al. (2000) afirmam que é suficiente trabalhar com o sistema linear homogêneo $Au = 0$, usando estimativas iniciais quaisquer para suavizar o erro, pois a solução exata $u = 0$ é conhecida e o erro da aproximação v é $-v$.

Uma boa estimativa inicial consiste de vetores chamados de modos de Fourier, que têm a forma:

$$v_j = \text{sen}\left(\frac{jk\pi}{N}\right), \quad 0 \leq j \leq N, \quad 1 \leq k \leq N-1, \quad (2.22)$$

onde v_j representa a j -ésima componente do vetor \mathbf{v}_k , k é chamado de número de ondas (ou frequência) e indica o número de “meios senos” que constituem o vetor \mathbf{v}_k no domínio do problema, N é o número de pontos da malha.

Os modos de Fourier da parte inferior do espectro, com $1 \leq k \leq (N/2)$, são chamados de modos de Fourier de baixa frequência ou modos suaves. Os modos localizados na parte superior do espectro, com $(N/2) \leq k \leq (N-1)$, são chamado de modos de Fourier de alta frequência ou modos oscilatórios (BRIGGS et al., 2000).

A Fig. 2.4, apresenta algumas estimativas iniciais que usam modos de Fourier com $k = 1, 3$ e 6 .

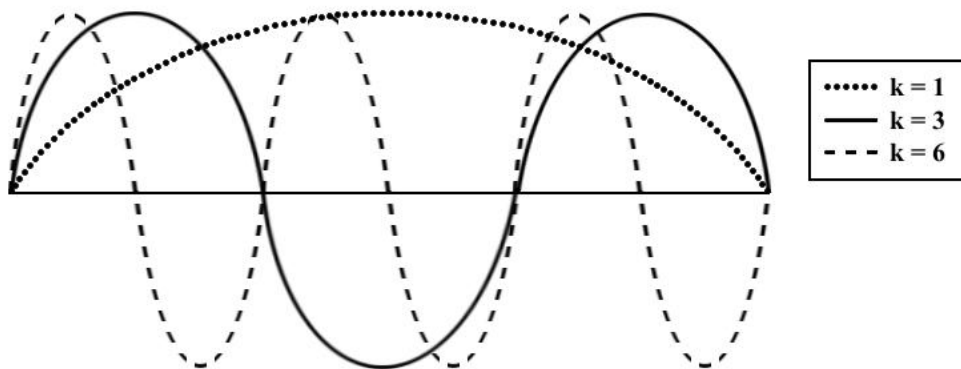


Figura 2.4: Modos de Fourier com $k=1, 3$ e 6 (adaptada de BRIGGS et al., 2000).

Nota-se na Fig. 2.4 que pequenos valores de k correspondem a ondas longas e suaves, enquanto valores maiores de k referem-se a ondas mais oscilatórias. De acordo com Briggs et al. (2000) e Trottenberg et al. (2001), os métodos iterativos, como Jacobi Ponderado e Gauss-Seidel, possuem propriedades de suavização, ou seja, são capazes de reduzir rapidamente as componentes de erros oscilatórios (ou de alta frequência) (BRIGGS et al., 2000).

Os métodos iterativos que possuem a propriedade de suavização convergem rapidamente já nas iterações iniciais, ou seja, reduzem rapidamente o erro nas primeiras iterações, porém, a redução torna-se lenta com aumento do número de iterações, conforme é apresentado na Fig. 2.5.

A ideia do método *Multigrid* é cobrir um espectro maior de comprimentos de onda através da iteração em uma sequência de malhas cada vez mais grossas, ou seja, uma malha com menor quantidade de pontos que a malha original. Nessas malhas mais grossas o processo de relaxação é repetido até que as componentes de erro tornem-se suaves. Quando isto ocorrer o problema é novamente transferido para uma malha mais grossa e assim sucessivamente até chegar à malha mais grossa estabelecida inicialmente ou na mais grossa possível. Os operadores que transferem tais informações são denominados operadores de transferência entre malhas.

A razão de engrossamento, para o caso bidimensional, considerando malhas uniformes, é definida como $r = H/h$, onde h representa o tamanho do espaçamento da malha fina Ω^h (também chamado de dimensão dos elementos da malha fina), H o tamanho do elemento da malha imediatamente mais grossa Ω^H .

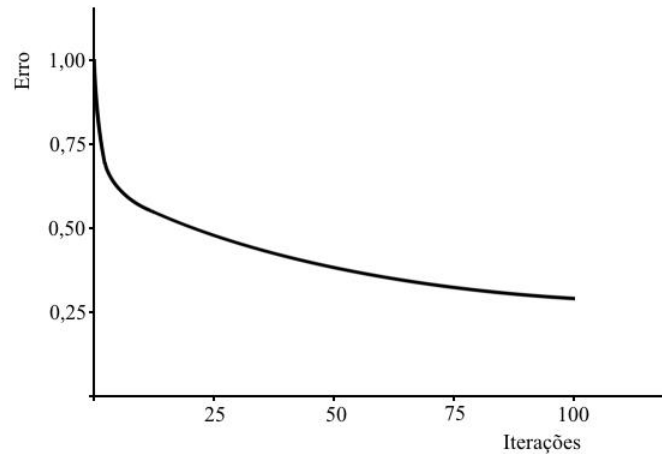
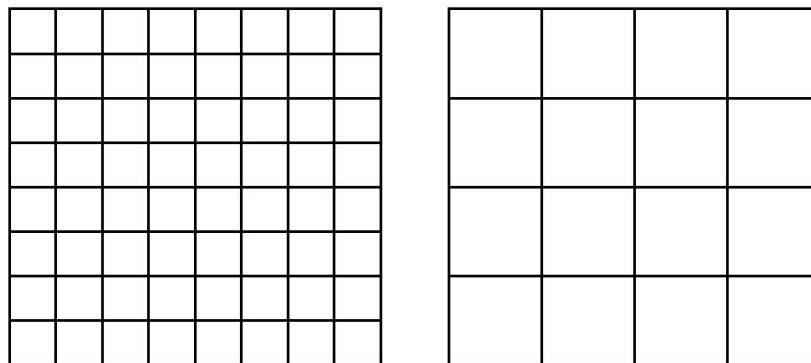


Figura 2.5: Jacobi ponderado com fator de ponderação igual a $2/3$ aplicado ao problema de condução de calor unidimensional com $N = 64$, estimativa inicial $(v_1 + v_6 + v_{32})/3$ e para 100 iterações (BRIGGS et al., 2000).

Brandt (1977) constatou que a razão $r = 2$ (caso no qual a malha grossa tem o dobro do espaçamento da malha fina, ou seja, $H = 2h$) é a recomendável, por ser mais conveniente e econômica para o processo de interpolação, além de ser de mais fácil implementação. Briggs et al. (2000) afirmam que $r \neq 2$, em geral, não traz vantagens, sem especificar para quais problemas ou classes de problemas.

A Fig. 2.6a representa uma malha fina original com $N_x = N_y = 9$ nós. Com o uso de $r = 2$, em ambas as direções, determina-se a malha imediatamente mais grossa com $N_x = N_y = 5$ nós, representada na Fig. 2.6b (engrossamento padrão). A malha mais fina é denotada por Ω^h , significando que cada elemento da malha tem comprimento h . Para o caso particular com razão de engrossamento $r = 2$, tem-se que $H = 2h$, logo a malha imediatamente mais grossa pode ser representada por Ω^{2h} . O engrossamento pode ainda ser realizado em uma única direção (semi-engrossamento) (OLIVEIRA et al., 2012).



a) Malha fina original (Ω^h) b) Malha imediatamente mais grossa (Ω^{2h})

Figura 2.6: Engrossamento padrão com razão de engrossamento $r = 2$.

Na Fig. 2.7 aparece uma sequência de malhas que serão visitadas no chamado processo de engrossamento utilizando-se $r = 2$. Neste exemplo, considera-se uma malha de $N = 17 \times 17$ nós (malha mais fina). O processo de engrossamento foi realizado até atingir a malha $N = 3 \times 3$ (a malha mais grossa possível). A quantidade de malhas utilizadas recebe o nome de número de níveis e será representado por L . No caso da Fig. 2.7, $L = 4$.

Segundo Ferziger e Peric (2002) e Roache (1998) a taxa de convergência ideal (teórica) do método *Multigrid* não depende do tamanho da malha, isto é, não depende do número de pontos da malha. Para obter um bom desempenho do *Multigrid*, diversos níveis de malha devem ser usados (TANNEHILL et al., 1997). Pinto e Marchi (2007) e Santiago e Marchi (2007) recomendam usar todos os níveis possíveis.

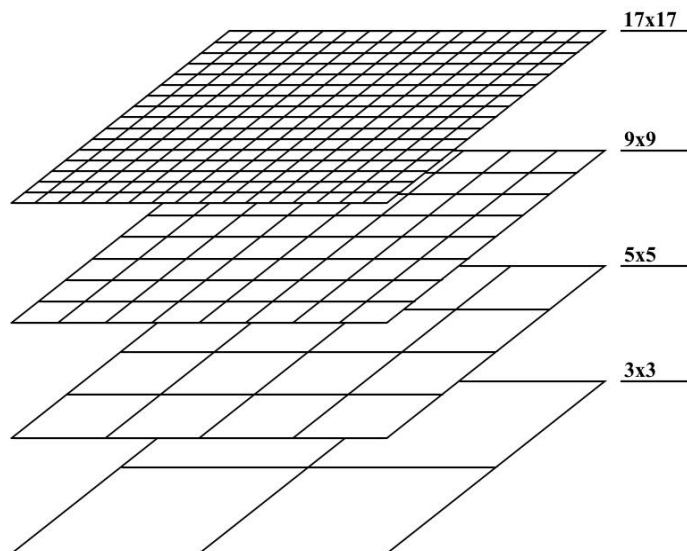


Figura 2.7: Processo de engrossamento e geração de malhas. (adaptado de <http://www.mgnet.org/mgnet/tutorials/xwb/mg.html>).

2.4.3 Operadores de Restrição

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

$$v^{2h} = [I_h]^{2h} v^h . \quad (2.23)$$

Entre os operadores de restrição conhecidos na literatura está o operador de restrição por ponderação completa (BRIGGS et al., 2000; TROTTEBERG et al., 2001; WESSELING, 1992), definido por

$$v_{i,j}^{2h} = \frac{1}{16} \left[v_{2i-1,2j-1}^h + v_{2i-1,2j+1}^h + v_{2i+1,2j-1}^h + v_{2i+1,2j+1}^h + 2 \left(v_{2i,2j-1}^h + v_{2i,2j+1}^h + v_{2i-1,2j}^h + v_{2i+1,2j}^h \right) + 4v_{2i,2j}^h \right], \quad (2.24)$$

com $0 \leq i \leq (N_x - 1)/2$ e $0 \leq j \leq (N_y - 1)/2$, onde N_x e N_y são os números dos nós da malha fina nas direções coordenadas x e y , respectivamente. Na Fig. 2.8 é apresentada a ação do operador de restrição por ponderação completa com razão de engrossamento $r = 2$ para o caso unidimensional.

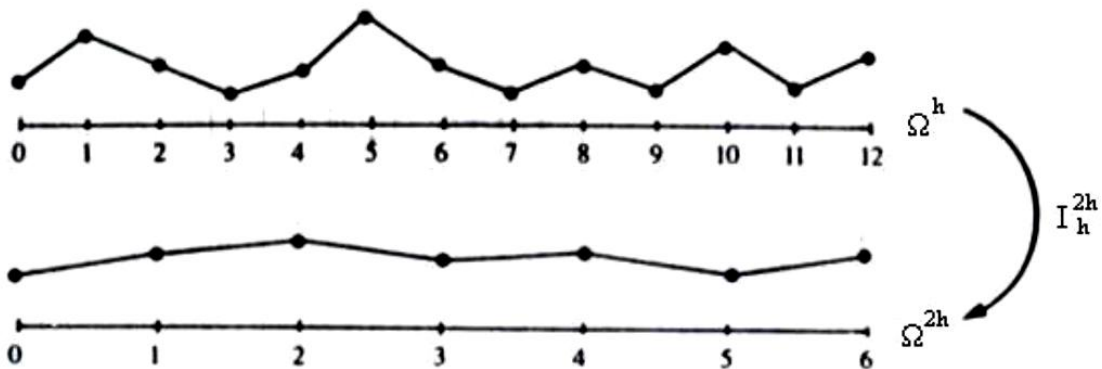


Figura 2.8: Operador de restrição (ponderação completa) da malha mais fina para a malha mais grossa (BRIGGS et al., 2000).

Outros operadores de restrição utilizados na literatura são: injeção e meia ponderação. Mais detalhes sobre os operadores de restrição podem ser encontrados em Briggs et al. (2000) e Trottenberg et al. (2001). Neste trabalho utilizou-se restrição por ponderação completa por ter eficiência comprovada (OLIVEIRA et al., 2012; TROTTEBERG et al., 2001).

2.4.4 Operadores de Prolongação

Os operadores que transferem informações da malha grossa Ω^{2h} para a malha fina Ω^h são chamados de operadores de prolongação. São representados genericamente por I_{2h}^h e definidos por

$$v^h = [I]_{2h}^h v^{2h}. \quad (2.25)$$

Uma forma simples de interpolação utilizada na literatura para problemas bidimensionais é a interpolação bilinear (BRIGGS et al., 2000; FERZIGER e PERIC, 2002; TANNEHILL et al., 1997; TROTTEBERG et al., 2001; WESSELING, 1992) dada por:

$$\left\{ \begin{array}{ll} v_{2i,2j}^h = v_{i,j}^{2h} & \text{para } \bullet \\ v_{2i+1,2j}^h = \frac{1}{2} \left(v_{i,j}^{2h} + v_{i+1,j}^{2h} \right) & \text{para } \diamond \\ v_{2i,2j+1}^h = \frac{1}{2} \left(v_{i,j}^{2h} + v_{i,j+1}^{2h} \right) & \text{para } \square \\ v_{2i+1,2j+1}^h = \frac{1}{4} \left(v_{i,j}^{2h} + v_{i+1,j}^{2h} + v_{i,j+1}^{2h} + v_{i+1,j+1}^{2h} \right) & \text{para } \circ \end{array} \right. , \quad (2.26)$$

onde i e j são índices da malha grossa com $0 \leq i \leq (N_x - 1)/2$ e $0 \leq j \leq (N_y - 1)/2$. Na Fig. 2.9 é apresentada uma malha fina com os símbolos referenciados na Eq. (2.26).

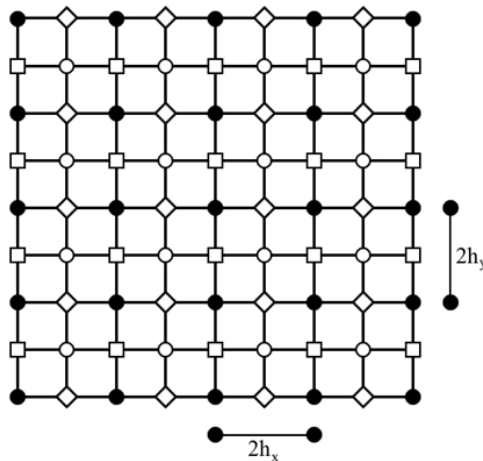


Figura 2.9: Prolongação bilinear (TROTTEBERG et al., 2001).

Outros operadores de prolongação podem ser vistos em Wesseling (1992), como: o operador de interpolação quadrática ou o operador de prolongação de De Zeeuw. Neste trabalho utilizou-se prolongação por interpolação bilinear, pela simplicidade de implementação e pela eficiência comprovada (BRIGGS et al., 2000; TROTTEBERG et al., 2001).

2.4.5 Ciclos

A sequência na qual as malhas são visitadas é chamada de ciclo *Multigrid*. Neste trabalho será usado o ciclo V, pois o W é cerca de 50% mais caro em relação ao número de operações envolvidas (HIRSCH, 1988). Nele, todas as malhas são visitadas uma única vez, da mais refinada para a mais grossa (restrição) e da mais grossa para a mais refinada (prolongação). Existe uma generalização do ciclo V conhecida como ciclo μ (WESSELING, 1992). Desta generalização podem-se obter outros tipos de ciclos. Devido aos seus formatos, são chamados de ciclo W, ciclo dente-de-serra e ciclo F. A Fig. 2.10, apresenta alguns exemplos. Nessa figura o símbolo ● é utilizado para representar as suavizações realizadas em cada nível de malha e os traços que os unem representam os operadores de transferência entre malhas (operador de restrição e de prolongação).

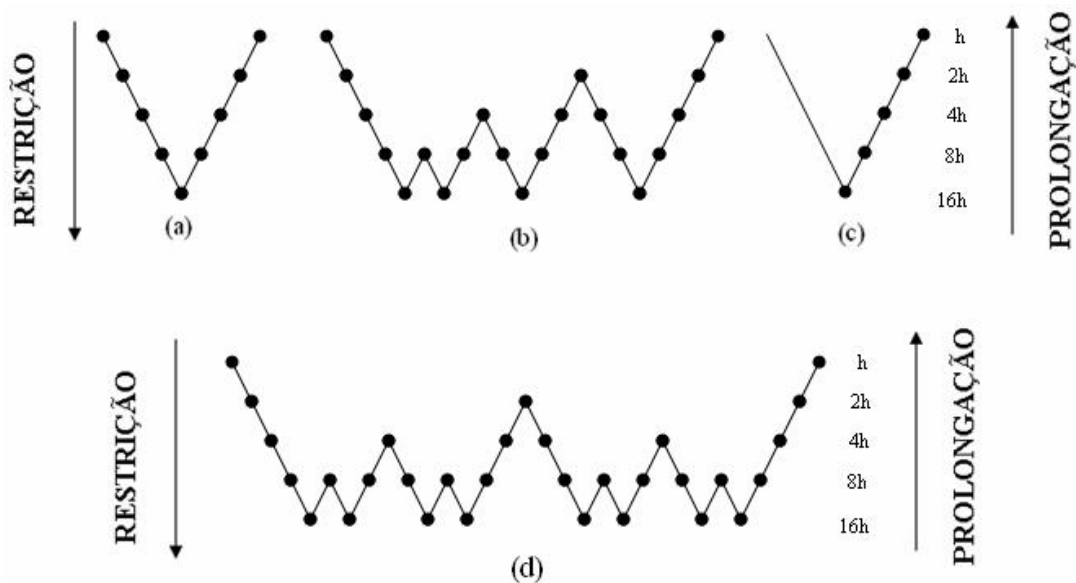


Figura 2.10: Diagramas: (a) ciclo V, (b) F, (c) dente-de-serra, (d) W (TROTTEBERG et al., 2001).

2.4.6 Esquemas

Dois tipos de esquemas podem ser usados com o método *Multigrid* de acordo com o tipo de informação transferida (BRIGGS et al., 2000): esquema de correção (*Correction Scheme*, CS) e o esquema de aproximação completa (*Full Approximation Scheme*, FAS).

O esquema CS transfere apenas o resíduo para a malha mais grossa e depois, retorna corrigindo a solução nas malhas mais refinadas. É mais adequado para problemas lineares,

como o problema de condução de calor modelado pela equação de Poisson, e por esse motivo, será utilizado esse esquema nesse trabalho.

O esquema FAS transfere o resíduo e a solução, e a equação $Au = f$ é resolvida em todas as malhas, sendo mais adequado a problemas não-lineares. Mais detalhes sobre o esquema FAS podem ser encontrados em Briggs et al. (2000).

2.4.7 Algoritmos

No algoritmo 2.2 são apresentados os passos para a implementação de um único ciclo V. O procedimento a seguir, descrito no algoritmo 2.3, desenvolve diversas chamadas do algoritmo 2.2 percorrendo vários ciclos V até se atingir um critério de parada ou alcançar o número máximo de ciclos escolhidos (ITE_{\max}). Os valores v_1 e v_2 representam o número de suavizações realizadas nos processos de restrição (pré-suavização) e prolongação (pós-suavização), respectivamente. Os superescritos $h, 2h, 4h, \dots$ indicam a malha onde se definem os vetores ou matrizes.

2.5 MÉTODOS DE EXTRAPOLAÇÃO

Muitos métodos iterativos são usados na resolução de sistemas de equações. Se o método produz uma convergência lenta, não é muito útil em situações práticas. Então, pode-se modificar o processo iterativo a fim de obter maior eficácia ou transformar a sequência que converge lentamente em outra, com melhores propriedades de convergência. Uma das abordagens para o segundo caso trata de métodos de extrapolação, que consistem em transformações de sequências.

De acordo com Brezinski e Zaglia (2002), se uma sequência (S_n) de números reais ou complexos converge lentamente para s , ela pode ser transformada em uma nova sequência (T_n) por uma transformação vetorial T . Por exemplo:

$$T_n = \frac{S_n + S_{n+1}}{2}, \quad n = 0, 1, 2, \dots, \quad (2.27)$$

ou

$$T_n = \frac{S_n S_{n+2} + S_{n+1}^2}{S_{n+2} - 2S_{n+1} + S_n}, \quad n = 0, 1, 2, \dots \quad (2.28)$$

**Algoritmo 2.2: Ciclo V com esquema CS para vários níveis de malhas
(adaptado de Briggs et al., 2000).**

Início Esquema de Correção (CS)

CSMG (v_0, v, f, h, v_1, v_2)

1. Suavizar $A^h u^h = f^h$ v_1 vezes em Ω^h com estimativa inicial v_0^h ;
2. Calcular o resíduo usando $R^h = f^h - A^h v^h$;
3. Restringir o resíduo da malha Ω^h para a malha Ω^{2h} : $f^{2h} = I_h^{2h} R^h$;
 - a) Suavizar $A^{2h} u^{2h} = f^{2h}$ v_1 vezes em Ω^{2h} com estimativa inicial $v_0^{2h} = 0$;
 - b) Calcular o resíduo usando $R^{2h} = f^{2h} - A^{2h} v^{2h}$;
 - c) Restringir o resíduo da malha Ω^{2h} para a malha Ω^{4h} : $f^{4h} = I_{2h}^{4h} R^{2h}$;
 - i) Suavizar $A^{4h} u^{4h} = f^{4h}$ v_1 vezes em Ω^{4h} com estimativa inicial $v_0^{4h} = 0$;
 - ii) Calcular o resíduo usando $R^{4h} = f^{4h} - A^{4h} v^{4h}$;
 - iii) Restringir o resíduo da malha Ω^{4h} para a malha Ω^{8h} : $f^{8h} = I_{4h}^{8h} R^{4h}$;
 - ⋮
 - Resolver $A^{Kh} u^{Kh} = f^{Kh}$;
 - ⋮
 - iv) Corrigir $v^{4h} \leftarrow v^{4h} + I_{8h}^{4h} v^{8h}$;
 - v) Suavizar $A^{4h} u^{4h} = f^{4h}$ v_2 vezes em Ω^{4h} com estimativa inicial v^{4h} ;
 - d) Corrigir $v^{2h} \leftarrow v^{2h} + I_{4h}^{2h} v^{4h}$;
 - e) Suavizar $A^{2h} u^{2h} = f^{2h}$ v_2 vezes em Ω^{2h} com estimativa inicial v^{2h} ;
4. Corrigir $v^h \leftarrow v^h + I_{2h}^h v^{2h}$;
5. Suavizar $A^h u^h = f^h$ v_2 vezes em Ω^h com estimativa inicial v^h .

Fim de CSMG

A fim de apresentar algum interesse prático, a nova sequência (T_n) deve apresentar as seguintes propriedades de interesse, em detrimento à sequência (S_n) (BREZINSKI e ZAGLIA, 2002):

Algoritmo 2.3: Diversas chamadas do *Multigrid* com esquema CS

Escolher v_0^h e ITE_{\max} .

1. $i = 1$

 Enquanto não atingir a convergência ou ITE_{\max} , faça:

 2. CSMG (v_0, v, f, h, v_1, v_2)

 3. $v_0 = v$

 4. $i = i + 1$.

Fim

- (T_n) deve convergir;
- (T_n) deve convergir para o mesmo limite de (S_n) . Nesse caso, T é chamada de transformação regular;
- (T_n) deve convergir para s mais rápido do que (S_n) , ou seja $\lim_{n \rightarrow \infty} (T_n - s)/(S_n - s) = 0$.

Segundo Brezinski e Zaglia (2002), normalmente essas propriedades não são válidas para todas as sequências (T_n) , mesmo que se tenha sequências (S_n) convergentes.

Deseja-se agora, rever a primeira transformação dada pela Eq. (2.27) e tentar encontrar a classe de sequências que pode ser acelerada. Tem-se:

$$\frac{T_n - s}{S_n - s} = \frac{1}{2} \left(1 + \frac{S_{n+1} - s}{S_n - s} \right). \quad (2.29)$$

Assim $\lim_{n \rightarrow \infty} (T_n - s)/(S_n - s) = 0$ se, e somente se $\lim_{n \rightarrow \infty} (S_{n+1} - s)/(S_n - s) = -1$, o que demonstra que esta transformação é capaz de acelerar a convergência de uma classe muito restrita de sequências, principalmente para os casos de somas.

Agora, para a transformação de sequência dada pela Eq. (2.28), pode ser demonstrado que ela acelera a convergência de todas as sequências para as quais exista um $\lambda \in [-1; 1]$, tal que $\lim_{n \rightarrow \infty} (S_{n+1} - s)/(S_n - s) = \lambda$. Desses exemplos, pode-se afirmar que transformações não-lineares de sequências geralmente têm melhores propriedades de aceleração de convergência do que os processos de soma linear (Eq. (2.27)), porém, nem sempre a transformação converge, ou pode ser que convirja, mas para um limite diferente.

Então, algumas propriedades importantes em relação à transformação de seqüências devem ser levadas em consideração.

Núcleo de uma transformação T , (κ_T) , é o conjunto de todas as seqüências que pertencem a S_n , tais que para todo n , $T_n \rightarrow s$ (BREZINSKI e ZAGLIA, 2002).

Para o processo de soma linear verifica-se que o seu núcleo é o conjunto de seqüências da forma

$$S_n = s + a(-1)^n, \quad (2.30)$$

onde a é um escalar.

Pode-se mostrar que para a transformação (2.28), o núcleo é o conjunto de seqüências da forma

$$S_n = s + a\lambda^n, \quad (2.31)$$

com a e λ escalares, $a \neq 0$ e $\lambda \neq 1$ (BREZINSKI e ZAGLIA, 2002).

Nota-se, então, que o núcleo da segunda transformação (Eq. (2.28)) contém o núcleo da primeira (Eq. (2.27)). Além disso, fica evidente também, que em ambos os casos, o núcleo depende de alguns parâmetros arbitrários s e a no primeiro caso, e a e $\lambda \neq 1$ no segundo. Se a seqüência (S_n) for acelerada por outra seqüência pertencente ao núcleo da transformação utilizado, então, por construção, tem-se para todo n , $T_n \rightarrow s$. Geralmente, s é o limite da seqüência (S_n) , mas nem sempre este é o caso.

O núcleo também pode ser dado de forma implícita por meio de uma relação que existe entre os termos consecutivos da seqüência. Assim, para a primeira transformação, é equivalente escrever que, para todo n ,

$$S_{n+1} - s = -(S_n - s), \quad (2.32)$$

enquanto, para o segundo caso, para todo n ,

$$S_{n+1} - s = \lambda(S_n - s). \quad (2.33)$$

As equações de diferenças Eq. (2.32) e Eq. (2.33) são chamadas de forma implícita do núcleo porque não fornecem diretamente (isto é, explicitamente) o núcleo, mas apenas implicitamente através da solução da equação diferença. Resolver essas equações levam à forma explícita do núcleo. Percebe-se que ambas as formas são equivalentes e dependem de parâmetros.

Em diversas aplicações, os resultados dependem de certo parâmetro escalar, que tende a um limite, para a obtenção do resultado exato (por exemplo, os autovalores de uma matriz, cujos elementos dependem de um parâmetro).

A transformação de sequência $T : (S_n) \mapsto (T_n)$ é dita método de extrapolação se é tal que, para todo n , $T_n \rightarrow s$ se, e somente se $(S_n) \in \kappa_T$. Assim, qualquer transformação de sequência pode ser vista como um método de extrapolação.

Mitin (1985) considera um processo iterativo em um espaço de Hilbert, onde os vetores $C_1, C_2, \dots, C_k, \dots, C_\infty$ são obtidos em passos desse processo. Define-se o vetor $\Delta_k = C_k - C_\infty$. Em geral, tem-se: $\Delta_{k+1} = F(\Delta_k)$, onde F é o operador que define o processo iterativo. Nas proximidades de uma solução consistente, ou seja, uma solução próxima da solução real, F pode ser expandido em uma série de potências dos operadores L linear, Q quadrático, etc. Ou seja,

$$\Delta_{k+1} = L(\Delta_k) + Q(\Delta_k) + \dots \quad (2.34)$$

Ao se considerar, apenas, o termo linear tem-se:

$$\Delta_{k+1} \cong L(\Delta_k). \quad (2.35)$$

Definindo-se o vetor diferença como:

$$D_k = C_k - C_{k-1} = \Delta_k - \Delta_{k-1}, \quad (2.36)$$

este satisfaz a equação:

$$D_{k+p} = (L)^p D_k, \quad (2.37)$$

e, por construção, o vetor C_{k+pm} , obtido na $k + pm$ iteração (m inteiro e positivo), pode ser escrito como:

$$C_{k+pm} = C_k + D_{k+1} + \dots + D_{k+pm}. \quad (2.38)$$

Com as Eqs. (2.37) e (2.38), pode-se escrever:

$$C_{k+pm} = C_k + \left[I + (L)^p + \dots + (L)^{p(m-1)} \right] (D_{k+1} + \dots + D_{k+p}), \quad (2.39)$$

onde I é o operador identidade.

Assim, tem-se:

$$C_{k+pm} = C_k + \left[\sum_{l=0}^{m-1} (L)^{pl} \right] (D_{k+1} + \dots + D_{k+p}), \quad (2.40)$$

onde a série

$$\left[\sum_{l=0}^{\infty} (L)^{pl} \right], \quad (2.41)$$

é a expansão do operador $\left[I - (L)^p \right]^{-1}$ em série de potências de $(L)^p$.

Considerando-se o limite com relação a m na Eq. (2.40), tem-se:

$$C_{\infty} = \lim_{m \rightarrow \infty} C_{k+pm} = \lim_{m \rightarrow \infty} \left\{ C_k + \left[\sum_{l=0}^{m-1} (L)^{pl} \right] (D_{k+1} + \dots + D_{k+p}) \right\} \quad (2.42)$$

ou

$$C_{\infty} = C_k + \left[I - (L)^p \right]^{-1} (C_{k+p} - C_k). \quad (2.43)$$

Ao se aproximar o operador L pelo operador diagonal G , e considerando-se sua j -ésima componente, a Eq. (2.43) se torna:

$$C_{\infty,j} = C_{k,j} + \left[I - (G_j)^p \right]^{-1} (C_{k+p,j} - C_{k,j}). \quad (2.44)$$

Em Mitin (1985), mostrou-se que é possível eliminar G_j dessa equação. Para isso consideram-se as iterações $k+d+p$ e $k+d$. Então, substituindo na Eq. (2.44), origina-se um sistema de equações que resulta em:

$$C_{\infty,j} = \frac{C_{k+d+p,j}C_{k,j} - C_{k+p,j}C_{k+d,j}}{C_{k+d+p,j} - C_{k+p,j} - C_{k+d,j} + C_{k,j}}, \quad p=1,2,\dots \quad d=1,2,\dots \quad (2.45)$$

Assim, para $p=d=2$,

$$C_{\infty,j} = \frac{C_{k+4,j}C_{k,j} - C_{k+2,j}^2}{C_{k+4,j} - 2C_{k+2,j} + C_{k,j}}, \quad (2.46)$$

e para $p=d=1$,

$$C_{\infty,j} = \frac{C_{k+2,j}C_{k,j} - C_{k+1,j}^2}{C_{k+2,j} - 2C_{k+1,j} + C_{k,j}}, \quad (2.47)$$

que são expressões que servirão de suporte para que sejam definidas as fórmulas dos três primeiros extrapoladores que serão estudados a seguir.

2.5.1 Extrapolador Aitken

Considerando-se as iterações subsequentes, com $k=1$, $C_1 = \phi_1$, $C_2 = \phi_2$, $C_3 = \phi_3$ e para todas as j -ésimas componentes satisfazendo a relação descrita pela Eq. (2.47), pode-se escrever:

$$\phi_{\infty}^{Aitken} = \frac{\phi_1\phi_3 - \phi_2^2}{\phi_3 - 2\phi_2 + \phi_1}, \quad (2.48)$$

que é a fórmula usada para que seja realizada a extrapolação de Aitken (BREZINSKI e ZAGLIA, 2002; BURDEN e FAIRES, 2007).

2.5.2 Extrapolador Empírico

Com base na teoria do estimador Empírico (MARTINS, 2002), tem-se:

$$\phi_{\infty}^{Empírico} = \phi_3 + \frac{(\phi_3 - \phi_2)^2}{2\phi_2 - \phi_3 - \phi_1}, \quad (2.49)$$

que é uma relação com as mesmas propriedades do extrapolador de Aitken.

2.5.3 Extrapolador Mitin

Considerando-se agora as iterações subsequentes com $k=1$, $C_1 = \phi_1$, $C_2 = \phi_2$, $C_3 = \phi_3$, $C_4 = \phi_4$, $C_5 = \phi_5$ e para todas as j -ésimas componentes satisfazendo a relação descrita para a Eq. (2.46), pode-se escrever:

$$\phi_{\infty}^{Mitin} = \frac{\phi_1\phi_5 - \phi_3^2}{\phi_5 - 2\phi_3 + \phi_1}, \quad (2.50)$$

que é a fórmula para que seja realizada a extrapolação de Mitin (MITIN, 1985).

O extrapolador Aitken e o extrapolador Empírico aceleram a convergência de sequências que convergem linearmente. O extrapolador Mitin é recomendado para sequências oscilantes.

Outros dois métodos de extrapolação existentes na literatura (conforme serão descritos a seguir) são o algoritmo Épsilon e o algoritmo Rho que apresentam uma semelhança formal, mas diferem significativamente em sua capacidade de acelerar a convergência. As propriedades dos algoritmos Épsilon e Rho são, de certo modo, complementares entre si.

A importância de estudar o algoritmo Épsilon reside em parte no seu potencial para aplicação na aceleração da convergência da solução iterativa de equações diferenciais discretizadas. Entretanto, todos os algoritmos têm seus domínios de validade. O algoritmo Épsilon falha, por exemplo, em sequências logaritmicamente convergentes (que convergem muito lentamente) e não consegue atingir o ponto fixo de geradores de sequências que divergem muito rapidamente. O algoritmo Épsilon geralmente falha para tais sequências e Delahaye e Germain-Bonne (1982) e Delahaye (1988) demonstraram que não há um acelerador universal para sequências logaritmicamente convergentes (GRAVES-MORRIS et al., 2000). O algoritmo Rho não acelera sequências com convergência linear, mas é muito poderoso para sequências logaritmicamente convergentes (GAO et al., 2010; GRAVES-MORRIS et al., 2000).

2.5.4 Extrapolador Épsilon escalar

O algoritmo Épsilon foi introduzido por Wynn (1956) como uma implementação eficiente do método de Shanks (SHANKS, 1955).

Seja (x_n) uma sequência escalar e considere o determinante de Hankel (BREZINSKI e ZAGLIA, 2002):

$$H_k(x_n) = \begin{vmatrix} x_n & \cdots & x_{n+k-1} \\ x_{n+1} & \cdots & x_{n+k} \\ \vdots & \vdots & \vdots \\ x_{n+k-1} & \cdots & x_{n+2k-2} \end{vmatrix}, \quad (2.51)$$

com $H_0(x_n) = 0, \forall n, k, n = 0, 1, \dots$

A transformação de Shanks e_k é definida por:

$$e_k(x_n) = \frac{H_{k+1}(x_n)}{H_k(\Delta^2 x_n)}, \quad k = 1, 2, \dots, \quad (2.52)$$

onde $\Delta^2 x_n$ refere-se à extrapolação de Aitken para a sequência x_n .

Para o núcleo da transformação e_k , em Brezinski e Zaglia (2002) provou-se que $\forall n, e_k(x_n) = x \Leftrightarrow \exists a_0, \dots, a_k$ com $a_k \neq 0$ e $a_0 + \dots + a_k \neq 0$ de tal forma que $\forall n,$

$$\sum_{i=0}^k a_i (x_{n+i} - x) = 0. \quad (2.53)$$

Para implementar a transformação de Shanks sem calcular determinantes, Wynn (1956) descobriu uma forma recursiva chamada de Algoritmo Épsilon escalar (*Scalar Epsilon Algorithm*, SEA), definido por:

$$\varepsilon_{-1}^{(n)} = 0; \quad \varepsilon_0^{(n)} = x_n; \quad \varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{1}{\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}}, \quad k, n = 0, 1, \dots \quad (2.54)$$

O extrapolador Épsilon escalar está relacionado com a transformação de Shanks por:

$$\varepsilon_{2k}^{(n)} = e_k(x_n) \quad \text{e} \quad \varepsilon_{2k+1}^{(n)} = \frac{1}{e_k(\Delta x_n)}; \quad k, n = 0, 1, \dots \quad (2.55)$$

2.5.5 Extrapolador Épsilon topológico

Para que o extrapolador Épsilon escalar possa ser aplicado a uma sequência de vetores (S_n) , este deveria ser aplicado a cada componente do vetor. No entanto, uma desvantagem de usar esta técnica é que as conexões entre as componentes são ignoradas, além do fato de que algumas componentes transformadas deixam de existir, ou ainda, são muito grandes, numericamente. Tais desvantagens limitam a aplicação do Épsilon escalar para sequências vetoriais (JBILOU e SADOK, 2000).

Para generalizar o extrapolador Épsilon escalar no caso vetorial, é necessário definir o inverso de um vetor. Uma possibilidade considerada por Wynn (1962) é o uso do inverso definido por

$$z^{-1} = \frac{z}{\|z\|^2}, \quad z \in \mathbf{R}^N. \quad (2.56)$$

Portanto, para sequências vetoriais, o extrapolador Épsilon vetorial é definido por:

$$\varepsilon_{-1}^{(n)} = 0; \quad \varepsilon_0^{(n)} = S_n; \quad \varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + [\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}]^{-1}, \quad k, n = 0, 1, \dots \quad (2.57)$$

Brezinski (1975) propõe outra generalização do Épsilon escalar para seqüências vetoriais, que se trata de uma forma recursiva do algoritmo Épsilon vetorial, no qual não é necessário o cálculo do inverso de vetores, o chamado Épsilon topológico (*Topological Epsilon Algorithm*, TEA):

$$\left\{ \begin{array}{l} \varepsilon_{-1}^{(n)} = 0; \quad \varepsilon_0^{(n)} = S_n; \quad \varepsilon_{2k+1}^{(n)} = \varepsilon_{2k-1}^{(n+1)} + \frac{y}{\langle y, \Delta \varepsilon_{2k}^{(n)} \rangle}; \\ \varepsilon_{2k+2}^{(n)} = \varepsilon_{2k}^{(n+1)} + \frac{\Delta \varepsilon_{2k}^{(n)}}{\langle \Delta \varepsilon_{2k+1}^{(n)}, \Delta \varepsilon_{2k}^{(n)} \rangle}, \quad k, n = 0, 1, \dots \end{array} \right. \quad (2.58)$$

sendo y um vetor arbitrário, ortogonal à j -ésima componente do resíduo generalizado entre $e_{k,j}(S_n)$ e $e_{k,j-1}(S_n)$ e

$$\langle \alpha, \beta \rangle = \sum_{i=1}^p \alpha_i \beta_i \quad (2.59)$$

o produto interno de dois vetores quaisquer α e β , onde α_i e β_i são as componentes dos vetores α e β , respectivamente e p é o número de componentes dos vetores.

O operador de diferença Δ atua sobre o sobrescrito n e tem-se

$$\varepsilon_{2k}^{(n)} = e_k(S_n) = s, \quad (2.60)$$

e

$$\varepsilon_{2k+1}^{(n)} = \frac{y}{\langle y, e_k(\Delta S_n) \rangle}, \quad k, n = 0, 1, \dots \quad (2.61)$$

2.5.6 Extrapolador Rho escalar

Constrói-se uma transformação de seqüência, cujo núcleo é a seqüência (\tilde{S}_n) definida na forma:

$$\tilde{S}_n = \frac{\tilde{S}_n^k + a_1 x_n^{k-1} + \dots + a_k}{x_n^k + b_1 x_n^{k-1} + \dots + b_k}, k, n = 1, 2, \dots, \quad (2.62)$$

onde (x_n) é uma seqüência escalar auxiliar dada.

A relação acima (Eq. (2.62)) pode ser escrita como

$$\tilde{S}_n = \tilde{S} - b_1 x_n^{-1} \tilde{S}_n + a_1 x_n^{-1} - \dots - b_k x_n^{-k} \tilde{S}_n + a_k x_n^{-k}. \quad (2.63)$$

Fazendo

$$g_{2i-1}(n) = x_n^{-i} S_n, \quad g_{2i}(n) = x_n^{-i}, \quad i = 1, \dots, k, \quad (2.64)$$

e usando a estrutura do E-algorithm (BREZINSKI e ZAGLIA, 2002), define-se o número $E_k^{(n)}$, denotado neste caso particular por $\rho_{2k}^{(n)}$, por

$$\rho_{2k}^{(n)} = \frac{\begin{vmatrix} 1 & \tilde{S}_n & x_n & x_n \tilde{S}_n & \dots & x_n^{k-1} & x_n^{k-1} \tilde{S}_n & x_n^k \tilde{S}_n \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 1 & \tilde{S}_{n+2k} & x_{n+2k} & x_{n+2k} \tilde{S}_{n+2k} & \dots & x_{n+2k}^{k-1} & x_{n+2k}^{k-1} \tilde{S}_{n+2k} & x_{n+2k}^k \tilde{S}_{n+2k} \end{vmatrix}}{\begin{vmatrix} 1 & \tilde{S}_n & x_n & x_n \tilde{S}_n & x_n^{k-1} & x_n^{k-1} \tilde{S}_n & x_n^k \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \tilde{S}_{n+2k} & x_{n+2k} & x_{n+2k} \tilde{S}_{n+2k} & x_{n+2k}^{k-1} & x_{n+2k}^{k-1} \tilde{S}_{n+2k} & x_{n+2k}^k \end{vmatrix}}, \quad k, n = 0, 1, \dots \quad (2.65)$$

onde $|\cdot|$ representa o determinante das matrizes.

Assim, por construção, a função racional

$$R_{2k}(x) = \frac{\rho_{2k}^{(n)} x^k + a_1 x^{k-1} + \dots + a_k}{x^k + b_1 x^{k-1} + \dots + b_k}, \quad (2.66)$$

satisfaz a condição de interpolação

$$R_{2k}(x_{n+1}) = S_{n+1}, \quad i = 0, \dots, 2k, \quad (2.67)$$

uma vez que

$$\lim_{x \rightarrow \infty} R_{2k}(x) = S_{n+1}. \quad (2.67)$$

O número $\rho_{2k}^{(n)}$, em conjunto com seu intermediário $\rho_{2k+1}^{(n)}$, são chamados de diferenças recíprocas e estão relacionados por um algoritmo muito semelhante ao algoritmo Épsilon escalar. É o chamado algoritmo Rho escalar, dado por:

$$\rho_{-1}^{(n)} = 0; \quad \rho_0^{(n)} = S_n; \quad \rho_{k+1}^{(n)} = \rho_{k-1}^{(n+1)} + \frac{x_{n+k+1} - x_n}{\rho_k^{(n+1)} - \rho_k^{(n)}}, \quad k, n = 0, 1, \dots \quad (2.68)$$

2.5.7 Extrapolador Rho topológico

Assim como ocorre com o algoritmo Épsilon escalar, o Rho escalar também pode ser generalizado para o caso vetorial (BREZINSKI e ZAGLIA, 2002):

$$\rho_{-1}^{(n)} = 0; \quad \rho_0^{(n)} = S_n; \quad \rho_{2k+1}^{(n)} = \rho_{k-1}^{(n+1)} + (k+1) \left(\Delta \rho_k^{(n)} \right)^{-1}, \quad k, n = 0, 1, \dots \quad (2.69)$$

e tem-se o Rho topológico:

$$\rho_{-1}^{(n)} = 0; \quad \rho_0^{(n)} = S_n; \quad \rho_{2k+1}^{(n)} = \rho_{2k-1}^{(n+1)} + \frac{(2k+1)y}{\langle y, \Delta \rho_{2k}^{(n)} \rangle}, \quad k, n = 0, 1, \dots \quad (2.70)$$

sendo y um vetor arbitrário.

2.5.8 Extrapolações múltiplas

Para usar extrapolações múltiplas, pode-se aplicar um determinado método de extrapolação de forma repetida. Para isso são necessárias soluções numéricas para a variável de interesse. O número de soluções depende do extrapolador que está sendo usado nas extrapolações múltiplas.

Neste trabalho, as extrapolações múltiplas foram realizadas para os extrapoladores Aitken e Mitin e tiveram dois níveis de extrapolação.

Para a Múltipla extrapolação de Aitken, por exemplo, são usadas as cinco últimas soluções obtidas pelo método *Multigrid* ($\phi_{1,0}$, $\phi_{2,0}$, $\phi_{3,0}$, $\phi_{4,0}$ e $\phi_{5,0}$), onde $\phi_{g,m}$ representa a g -ésima solução e m o nível de extrapolação. Para o primeiro nível de soluções, onde ainda não foi efetuada nenhuma extrapolação ($m=0$), são realizadas extrapolações aplicando as soluções $\phi_{1,0}$, $\phi_{2,0}$ e $\phi_{3,0}$; $\phi_{2,0}$, $\phi_{3,0}$ e $\phi_{4,0}$ e $\phi_{3,0}$, $\phi_{4,0}$ e $\phi_{5,0}$, na Eq. (2,48), gerando as soluções, $\phi_{3,1}$, $\phi_{4,1}$ e $\phi_{5,1}$, respectivamente, onde $m=1$ representa o primeiro nível de extrapolação. Em seguida, essas três soluções extrapoladas ($\phi_{3,1}$, $\phi_{4,1}$ e $\phi_{5,1}$), são usadas para uma nova extrapolação, gerando a solução extrapolada no segundo nível, $\phi_{5,2}$. A Fig 2.11 apresenta um esquema da Múltipla extrapolação de Aitken para dois níveis.

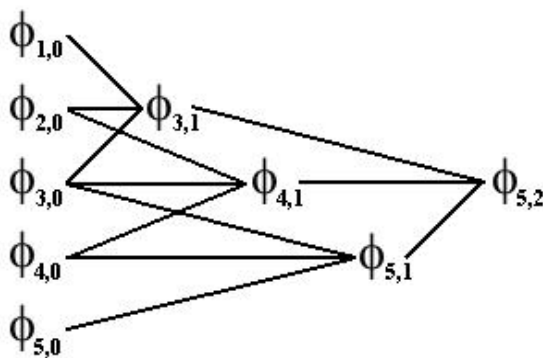


Figura 2.11: Representação esquemática da Múltipla extrapolação de Aitken com dois níveis.

Para a Múltipla extrapolação de Mitin o procedimento é análogo, porém, neste caso é necessário um mínimo de nove soluções para efetuar dois níveis de extrapolação.

2.6 RESUMO DO CAPÍTULO

Neste capítulo apresentaram-se os conceitos relacionados aos métodos numéricos, dando ênfase ao MDF. Também foram retratados de forma geral os métodos para resolução de sistemas de equações, dando um enfoque especial no método Gauss-Seidel *red-black*, que será utilizado neste trabalho. Na sequência vieram as definições e tipos de erros numéricos. Em seguida, os principais aspectos do método *Multigrid*, método este que se destaca pela aceleração (melhora na taxa de convergência) da resolução de problemas lineares e não-lineares e que desfruta da propriedade de suavização de certos métodos iterativos. Foi mostrada a filosofia desse método, os operadores de transferência entre malhas, tipos de ciclos, esquemas de correção e o algoritmo referente ao esquema CS. Em seguida, explorou-se a teoria dos métodos de extrapolação que serão utilizados neste trabalho (Aitken, Empírico, Mitin, Épsilon escalar e topológico, Rho escalar e topológico). O capítulo encerrou-se com o conceito de extrapolações múltiplas.

3 MODELOS MATEMÁTICO E NUMÉRICO

Este capítulo apresenta os modelos matemático e numérico utilizados neste trabalho e os detalhes de implementação. Utilizou-se neste estudo a equação de Poisson, por ser uma equação simples, de fácil implementação e que possui solução analítica conhecida, pois assim, as análises dos parâmetros feitas neste trabalho, podem servir para possíveis generalizações, como recomenda Trottenberg et al. (2001).

3.1 MODELO MATEMÁTICO

O modelo matemático considerado neste trabalho refere-se ao problema de condução de calor linear bidimensional em regime permanente descrito pela equação de Poisson (INCROPERA et al., 2008):

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

com condições de contorno de Dirichlet:

$$T(0, y) = T(x, 0) = T(1, y) = T(x, 1) = 0, \quad (3.2)$$

onde x e y são as direções coordenadas (variáveis independentes) e T a temperatura (variável dependente), com $(0 \leq x \leq 1, 0 \leq y \leq 1)$, conforme Fig. 3.1. O termo fonte é dado por:

$$S(x, y) = -2 \left[(1 - 6x^2)y^2(1 - y^2) + (1 - 6y^2)x^2(1 - x^2) \right]. \quad (3.3)$$

Com isso, tem-se a seguinte solução analítica:

$$T(x, y) = (x^2 - y^4)(y^4 - x^2). \quad (3.4)$$

O fenômeno físico associado à condução de calor refere-se à propagação do calor em meio sólido ou líquido em repouso, através da vibração molecular, nas quais as fronteiras estejam submetidas a diferentes temperaturas (INCROPERA et al., 2008).

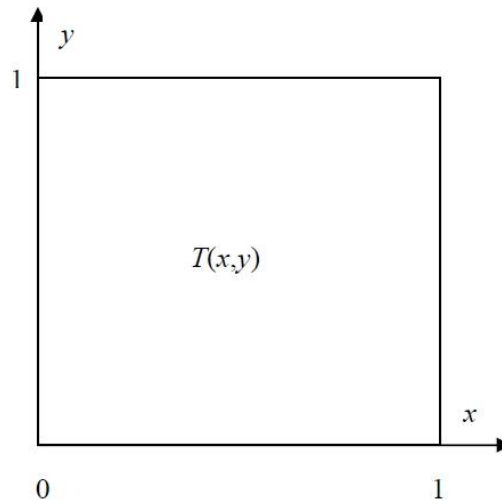


Figura 3.1: Domínio bidimensional para a equação de Poisson.

3.2 MODELO NUMÉRICO

A discretização do domínio de cálculo se dará em uma malha uniforme nas duas direções, com $N = N_x N_y$ nós, onde N_x e N_y representam os números de pontos nas direções coordenadas x e y , respectivamente, gerando uma malha com o espaçamento entre os nós (também chamado de dimensão dos elementos da malha) $h_x = 1/(N_x - 1)$ e $h_y = 1/(N_y - 1)$. Neste trabalho serão usadas malhas uniformes, tais que $N_x = N_y$ e $h_x = h_y = h$.

Para cada um dos $(N_x - 2)(N_y - 2)$ pontos interiores da malha, a Eq. (3.1) é discretizada com o método das diferenças finitas (MDF), com aproximações por diferença central (CDS), onde as derivadas são aproximadas usando as Eqs. (2.10) e (2.11), sendo que a posição dos termos aproximados serão dadas conforme descrito na Fig. 2.2. O termo fonte, que não tem derivadas será aproximado diretamente no ponto P , gerando:

$$\frac{T_W - 2T_P + T_E}{h^2} + \frac{T_S - 2T_P + T_N}{h^2} = S_P. \quad (3.5)$$

Rearranjando os termos da Eq. (3.5), obtém-se:

$$\left(\frac{4}{h^2}\right)T_P = \left(\frac{1}{h^2}\right)T_W + \left(\frac{1}{h^2}\right)T_E + \left(\frac{1}{h^2}\right)T_S + \left(\frac{1}{h^2}\right)T_N - S_P, \quad (3.6)$$

ou

$$a_P T_P = a_W T_W + a_E T_E + a_N T_N + a_S T_S + b_P. \quad (3.7)$$

Considerando-se as Eqs. (3.6) e (3.7), tem-se:

$$a_W = a_E = a_S = a_N = \left(\frac{1}{h^2}\right), \quad a_P = \left(\frac{4}{h^2}\right) \text{ e } b_P = -S_P. \quad (3.8)$$

Estes coeficientes são válidos para os pontos internos da malha, ou seja, para $i = 2, 3, \dots, N_x - 1$ e $j = 2, 3, \dots, N_y - 1$. Para os contornos, com as condições impostas, considera-se:

$$T_{1,j} = T_{N_x,j} = T_{i,1} = T_{i,N_y} = 0. \quad (3.9)$$

3.3 DADOS DE IMPLEMENTAÇÃO

Nesta seção encontram-se as características básicas do método *Multigrid* padrão utilizado nas simulações deste trabalho.

Neste trabalho utilizou-se o esquema CS, por se tratar de um problema linear (BRIGGS et al., 2000), e conforme Fletcher (1991), o esquema de aproximação completa FAS é cerca de 5% a 10% mais caro, em relação ao tempo de CPU, que o CS por restringir o resíduo e a solução para as malhas mais grossas. O tipo de ciclo utilizado é o V, pois o W é cerca de 50% mais caro em relação ao número de operações envolvidas (HIRSCH, 1988).

Utiliza-se restrição por ponderação completa e prolongação através de interpolação bilinear (TROTTEBERG et al., 2001). Entre os operadores de restrição conhecidos, o mais comum é o operador de restrição por injeção (BRIGGS et al., 2000; TROTTEBERG et al., 2001). O operador de restrição por injeção também é utilizado em diversos trabalhos entre eles Gerolymos e Vallet (2005), Oliveira (2010) e Pinto (2006). O operador de prolongação bilinear também é muito utilizado na literatura; entre eles pode-se citar: Briggs et al. (2000);

Ferziger e Peric (2002); Tannehill et al. (1997); Trottenberg et al. (2001) e Wesseling (1992). A razão de engrossamento de malhas (r) usada é a padrão, ou seja, $r = 2$ (BRANDT, 1977; BRIGGS et al., 2000).

O suavizador (ou *solver*) utilizado foi o método de Gauss-Seidel *red-black* descrito em Wesseling (1992). No presente trabalho, o número de iterações internas feitas em cada malha é representado por $v_1 = v_2 = v$.

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

$$\|R\|_2 = \frac{\|R^{it}\|_2}{\|R^0\|_2}, \quad (3.10)$$

onde R^{it} é o resíduo da iteração atual e R^0 o resíduo da estimativa inicial. Entre os trabalhos que usam essa norma pode-se citar: Briggs et al. (2000); Oliveira (2010); Oliveira et al. (2012); Trottenberg et al. (2001); Wesseling (1992) e Zhang (2002). O processo iterativo é interrompido quando a norma l_2 é menor ou igual à tolerância ε ($\varepsilon > 0$).

Para resolver o modelo matemático proposto a fim de se obter os parâmetros para análise, em todas as simulações, o método *Multigrid* partiu da malha mais fina (L) e foi até a malha mais grossa, ou seja, a malha com $N = 3 \times 3$ nós, portanto, o número de pontos dessa malha é $N = N_x N_y$, onde $N_x = N_y = 2^L + 1$. Por exemplo, ao tomar-se $L = 5$, tem-se $N = 33 \times 33$ nós na malha mais fina.

Um dos objetivos deste trabalho é analisar o comportamento do tempo de CPU (t_{CPU}) para os diversos casos que serão descritos no capítulo 4. Entende-se por tempo de CPU o tempo gasto para realizar a geração de malhas, atribuir a estimativa inicial, calcular os coeficientes e resolver o sistema linear representado pela Eq. (3.7) até atingir a tolerância estabelecida. Em alguns casos, a resolução destes sistemas será realizada usando apenas o *Multigrid* e em outros, efetuando as extrapolações associadas ao método *Multigrid*. Este tempo é medido em segundos (s) usando-se a sub-rotina CPU_TIME do Intel Fortran 11.1.

Em algumas situações, o tempo de CPU foi muito pequeno, menor do que um segundo. Quando isto ocorre aumenta-se a imprecisão da função CPU_TIME. Por este motivo, utilizou-se um procedimento (SANTIAGO e MARCHI, 2007) a fim de minimizar a incerteza da função CPU_TIME na medida do tempo de CPU, realizado da seguinte forma: para todas as malhas cujo tempo de CPU foi inferior a 1 segundo, acrescentou-se ao programa principal um ciclo externo. Este ciclo faz com que o número de simulações se repita automaticamente até que seja obtido um tempo de CPU igual ou superior a 1 segundo. Nestes casos, o tempo de CPU de uma simulação é uma média do tempo obtido em todas as repetições.

Outro objetivo deste trabalho é a análise do comportamento do erro de iteração para os diversos casos estudados. Para que esta análise fosse possível, primeiramente, foram obtidas as soluções exatas dos sistemas gerados pelos modelos matemáticos, de acordo com o número de incógnitas (N). Estas soluções foram obtidas executando-se o programa até que fosse atingido o erro de máquina. De posse destas soluções, pôde-se avaliar o comportamento do erro de iteração para todos os casos estudados.

O pico de memória de armazenamento também foi utilizado nas análises. Entende-se por pico de memória de armazenamento de um programa, como o maior valor da memória utilizada por ele no armazenamento de dados durante sua execução. Para acompanhar estes valores nas diversas simulações, utilizou-se o Gerenciador de Tarefas do *Windows*.

A forma com que foram realizadas as extrapolações ao final e durante o processo iterativo serão detalhadas no capítulo 4.

3.4 RESUMO DO CAPÍTULO

Neste capítulo apresentaram-se os modelos matemático e numérico abordados neste trabalho. O modelo matemático refere-se ao problema de condução de calor linear bidimensional em regime permanente descrito pela equação de Poisson, com condições de contorno de Dirichlet. A discretização se deu sobre uma malha bidimensional uniforme, utilizando-se o MDF. Foram apresentados os coeficientes e termos fontes do sistema de equações algébricas resultante da discretização para os pontos internos e contornos do domínio. Em seguida, foram apresentados os dados utilizados nos códigos computacionais implementados, citando as características do método *Multigrid* empregado, definindo o critério de convergência e os principais parâmetros que foram alvos de análise neste trabalho.

4 MÉTODOS DE EXTRAPOLAÇÃO ASSOCIADOS AO MÉTODO *MULTIGRID*

Neste capítulo são mostrados os resultados obtidos nas simulações que envolveram a equação de Poisson utilizando métodos de extrapolação associados ao método *Multigrid*, discretizada em uma malha uniforme, usando o MDF. São apresentados e descritos detalhadamente os diversos casos que serão utilizados nas seções a seguir.

O objetivo inicial dos testes, cujos resultados estão presentes na seção 4.1, é descobrir, entre os extrapoladores utilizados, qual é o mais eficiente, ou seja, o extrapolador de melhor desempenho quando são analisados os seguintes parâmetros: norma do resíduo adimensionalizado com base na estimativa inicial, norma do erro de iteração, fator de convergência empírico, tempo de CPU e memória de armazenamento.

Na sequência, após o extrapolador de melhor desempenho ser determinado, este foi usado durante o processo iterativo (seção 4.2). Desta forma, pode-se determinar qual metodologia deverá ser utilizada para associar métodos de extrapolação ao método *Multigrid* e com isso, reduzir o erro de iteração e o tempo de CPU.

Para confirmar os resultados, testes adicionais e análises envolvendo *speed-up* e ajustes de curvas foram realizados (seção 4.3).

4.1 USO DE EXTRAPOLADORES AO FINAL DO MÉTODO *MULTIGRID*

Para simular o uso dos extrapoladores ao final do *Multigrid*, verificaram-se primeiramente, quantos ciclos V eram necessários para atingir a tolerância desejada usando apenas o *Multigrid*. Então, de acordo com o número de soluções utilizadas para cada extrapolador, estas foram armazenadas e extrapoladas.

As soluções foram encontradas de acordo com os dados fixos e variáveis que serão listados na subseção 4.1.2, e estes deram origem a nove casos que diferem entre si pelo número de incógnitas (N), ou pelo critério de parada (ϵ). Cada caso foi resolvido de três maneiras distintas:

- usando apenas o método *Multigrid* até ser atingido o critério de parada;
- usando o método *Multigrid* com um ciclo V além daqueles necessários para se atingir o critério de parada;

• usando o método *Multigrid* até atingir o critério de parada e extrapolando as soluções obtidas nas últimas iterações, através do emprego dos seguintes extrapoladores: Aitken, Empírico, Mitin, Épsilon escalar, Rho escalar, Épsilon topológico, Rho topológico, Múltipla extrapolação de Aitken e Múltipla extrapolação de Mitin, conforme definidos na seção 2.5.

O número de soluções necessárias para efetuar uma extrapolação para cada extrapolador é dado na Tab. 4.1:

Tabela 4.1: Número de soluções utilizadas para extrapolações

Extrapolador	Número de soluções
Aitken	3
Empírico	3
Mitin	5
Épsilon escalar	5
Rho escalar	5
Épsilon topológico	5
Rho topológico	5
Múltipla extrapolação de Aitken	5
Múltipla extrapolação de Mitin	9

Em todas as simulações, o critério de parada utilizado foi a norma l_2 do resíduo adimensionalizado com base na estimativa inicial, dada pela Eq. (3.10).

Para análise da eficiência dos extrapoladores, primeiramente, adotaram-se como parâmetros: a norma do resíduo adimensionalizado (que será designada por $\|R\|_2$) e as normas infinito e euclidiana do erro de iteração ($\|E_n\|_\infty$ e $\|E_n\|_2$, respectivamente), sendo que E_n é obtido através da diferença entre a solução obtida na iteração n e a solução obtida levando-se o processo iterativo até o erro de máquina. Considera-se, então, a comparação entre a solução extrapolada e a solução com um ciclo adicional do *Multigrid*.

Também são utilizados como parâmetros de comparação, o tempo de processamento (t_{CPU}), o pico de memória (M), o fator de convergência empírico ($q^{(k)}$) e o fator de convergência média empírico ($\hat{q}^{(k)}$), estes últimos definidos, respectivamente (TROTTEBERG et al., 2001) por:

$$q^{(k)} = \frac{\|R^k\|_2}{\|R^{k-1}\|_2} \quad (4.1)$$

$$\hat{q}^{(k)} = \sqrt[k]{\frac{\|R^k\|_2}{\|R^0\|_2}} \quad (4.2)$$

onde $\|R^k\|_2$ é a norma euclidiana do vetor resíduo na k -ésima iteração, com $R^0 \neq 0$. Se $R^0 = 0$, deve-se tomar $R^0 = 1$ por questões puramente computacionais.

4.1.1 Parâmetros utilizados para análise dos resultados

Espera-se de um bom método iterativo que o tempo de processamento dos dados para obtenção da solução do problema seja pequeno, o menor possível.

Em relação ao pico de memória, quanto menor o valor deste parâmetro, mais vantajosa é a metodologia utilizada, pois o programa estará exigindo um esforço menor do computador que realiza as simulações.

Quanto à norma do resíduo adimensionalizada com base na estimativa inicial (que é o mais utilizado), tem-se os resultados dos testes usando-se três casos: o método *Multigrid*, o método *Multigrid* com um ciclo V adicional e com o uso dos extrapoladores. Nesta etapa do trabalho, o extrapolador de melhor desempenho é o que mais reduzir a norma do resíduo.

Espera-se para o fator de convergência empírico e o fator de convergência média empírico que apresentem valores próximo de zero (BRIGGS et al., 2000; BURDEN e FAIRES, 2007), ou seja a norma do resíduo na iteração considerada tem um pequeno valor quando comparado ao valor da norma do resíduo na iteração anterior (fator de convergência empírico) ou quando comparado ao valor da norma do resíduo na iteração inicial (fator de convergência média empírico).

Para o parâmetro erro de iteração, o melhor extrapolador é aquele que ao ser aplicado ao método iterativo, provoca uma maior redução deste erro, para qualquer norma utilizada.

4.1.2 Casos estudados

Para resolver o modelo matemático, foi utilizado o método *Multigrid* geométrico, sendo fixados os seguintes dados:

- Número de pontos nas direções x e y da malha mais grossa: 3.
- Número máximo de iterações externas (ciclos V): 25.

- Número máximo de iterações internas do *solver* Gauss Seidel *red-black*: 3.
- Razão de engrossamento: 2.

Em todas as simulações, o método *Multigrid* partiu da malha mais fina e foi até a malha mais grossa possível, ou seja, a malha com $N = 3 \times 3$ nós. O número de incógnitas na malha mais fina (N) e o critério de parada (ε), aparecem na Tab. 4.2, dando origem a nove casos distintos:

Tabela 4.2: Casos estudados de acordo com N e ε

Caso	N	ε
1-A	129x129	10^{-6}
1-B	129x129	10^{-10}
1-C	129x129	10^{-15}
2-A	1025x1025	10^{-6}
2-B	1025x1025	10^{-10}
2-C	1025x1025	10^{-15}
3-A	4097x4097	10^{-6}
3-B	4097x4097	10^{-10}
3-C	4097x4097	10^{-15}

4.1.3 Resultados

As soluções foram obtidas com precisão quádrupla. As tabelas a seguir apresentam os resultados das simulações feitas para resolução do modelo matemático, com apenas três algoritmos decimais.

Em todas as tabelas, a primeira linha apresenta os resultados dos parâmetros estudados na resolução do problema com o *Multigrid* sem o uso de extrapoladores, até que o critério de parada fosse atingido (esta linha será designada por MG); a segunda linha contém os resultados dos parâmetros obtidos com o *Multigrid* sem extrapoladores, porém, com mais um ciclo V (ou iteração externa) após ser atingido o critério de parada (esta linha será designada por MG + 1 ITE) e as linhas seguintes os resultados dos parâmetros obtidos com o uso dos extrapoladores nomeados (estas linhas serão designadas por MG + nome do extrapolador).

As colunas apresentam a metodologia utilizada na resolução do problema; o tempo de CPU (t_{CPU}) em segundos (s); o pico de memória de armazenamento (M) em megabytes (MB) ou gigabytes (GB), dependendo do caso; a norma euclidiana do resíduo adimensionalizada

pela norma euclidiana do resíduo na estimativa inicial ($\|R\|_2$); o fator de convergência empírico ($q^{(k)}$); o fator de convergência média empírico ($\hat{q}^{(k)}$); a norma infinito erro de iteração ($\|E_n\|_\infty$) e a norma euclidiana do erro de iteração ($\|E_n\|_2$).

A seguir são apresentadas as tabelas com os resultados dos parâmetros estudados, onde, em alguns resultados, o sinal e os dois algarismos que aparecem depois da letra “E” referem-se ao expoente na base 10 pela qual deve ser multiplicada a mantissa (número anterior à letra “E”).

Tabela 4.3: Resultados dos parâmetros estudados para o caso 1-A ($N = 129 \times 129$ e $\varepsilon = 10^{-6}$)

Metodologia	$t_{CPU}(s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	0,163	6,272	1,148E-07	4,052E-02	4,092E-02	6,817E-09	3,042E-09
MG + 1 ITE	0,195	6,272	4,715E-09	4,106E-02	4,095E-02	2,813E-10	1,284E-10
MG + Aitken	0,811	7,996	4,178E-09	3,638E-02	4,013E-02	1,277E-10	4,646E-11
MG + Empírico	0,811	7,996	4,178E-09	3,638E-02	4,013E-02	1,277E-10	4,646E-11
MG + Mitin	0,827	8,500	4,028E-08	3,507E-01	5,854E-02	9,597E-10	3,197E-10
MG + Épsilon escalar	0,234	8,680	1,068E-06	9,307E+00	1,011E-01	9,287E-10	1,568E-11
MG + Rho escalar	0,234	8,680	7,110E-05	6,191E+02	2,035E-01	4,071E-06	1,732E-06
MG + Épsilon topológico	0,835	10,896	1,475E-09	1,284E-02	3,374E-02	3,453E-12	1,212E-12
MG + Rho topológico	0,254	11,124	7,084E-05	6,168E+02	2,034E-01	4,065E-06	1,727E-06
MG + Múltipla ext. Aitken	0,234	8,688	1,598E-05	1,392E+02	1,587E-01	6,689E-09	6,447E-10

Obs.: Não foi possível realizar a Múltipla extrapolação de Mitin, pois o processo iterativo convergiu antes de completar as nove iterações usadas por este extrapolador.

Tabela 4.4: Resultados dos parâmetros estudados para o caso 1-B ($N = 129 \times 129$ e $\varepsilon = 10^{-10}$)

Metodologia	$t_{CPU}(s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	0,257	6,272	8,168E-12	4,176E-02	4,111E-02	4,865E-13	2,307E-13
MG + 1 ITE	0,289	6,272	3,429E-13	4,198E-02	4,121E-02	2,037E-14	9,807E-15
MG + Aitken	1,295	7,964	1,234E-13	1,511E-02	3,679E-02	4,430E-15	1,691E-15
MG + Empírico	1,295	7,964	1,234E-13	1,511E-02	3,679E-02	4,430E-15	1,691E-15
MG + Mitin	1,279	8,492	2,022E-11	2,476E+00	6,482E-02	2,704E-14	8,348E-15
MG + Épsilon escalar	0,354	8,680	1,700E-11	2,081E+00	6,358E-02	1,975E-14	2,515E-16
MG + Rho escalar	0,354	8,684	4,726E-09	5,786E+02	1,188E-01	2,810E-10	1,284E-10
MG + Épsilon topológico	1,310	10,856	1,065E-14	1,304E-03	2,802E-02	6,682E-17	2,466E-17
MG + Rho topológico	0,374	11,120	4,706E-09	5,761E+02	1,187E-01	2,808E-10	1,283E-10
MG + Múltipla ext. Aitken	0,359	8,700	1,208E-09	1,480E+02	1,021E-01	4,689E-13	2,775E-14
MG + Múltipla ext. Mitin	0,364	9,740	6,087E-10	7,452E+01	9,463E-02	5,248E-13	9,906E-15

Tabela 4.5: Resultados dos parâmetros estudados para o caso 1-C ($N = 129 \times 129$ e $\varepsilon = 10^{-15}$)

Metodologia	$t_{CPU} (s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	0,348	6,272	6,110E-16	4,226E-02	4,139E-02	3,609E-17	1,776E-17
MG + 1 ITE	0,380	6,272	2,587E-17	4,235E-02	4,147E-02	1,525E-18	7,567E-19
MG + Aitken	1,732	7,960	4,842E-18	7,924E-03	3,606E-02	1,780E-19	7,075E-20
MG + Empírico	1,732	7,960	4,830E-18	7,905E-03	3,605E-02	1,780E-19	7,075E-20
MG + Mitin	1,747	8,488	2,266E-17	3,709E-02	4,101E-02	8,533E-19	3,260E-19
MG + Épsilon escalar	0,473	8,680	1,518E-15	2,485E+00	5,822E-02	1,942E-18	2,237E-20
MG + Rho escalar	0,473	8,684	3,429E-13	5,612E+02	9,146E-02	2,035E-14	9,807E-15
MG + Épsilon topológico	1,778	10,856	1,471E-19	2,407E-04	2,695E-02	1,208E-21	5,121E-22
MG + Rho topológico	0,494	11,120	3,427E-13	5,608E+02	9,146E-02	2,035E-14	9,804E-15
MG + Múltipla ext. Aitken	0,478	8,692	1,023E-13	1,674E+02	8,269E-02	3,539E-17	1,498E-18
MG + Múltipla ext. Mitin	0,478	9,736	2,653E-14	4,342E+01	7,390E-02	2,425E-17	4,024E-19

Tabela 4.6: Resultados dos parâmetros estudados para o caso 2-A ($N = 1025 \times 1025$ e $\varepsilon = 10^{-6}$)

Metodologia	$t_{CPU} (s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	10,156	119,636	1,159E-07	4,062E-02	4,100E-02	6,880E-09	3,086E-09
MG + 1 ITE	12,106	119,636	4,772E-09	4,117E-02	4,103E-02	2,844E-10	1,304E-10
MG + Aitken	54,413	216,400	2,120E-06	1,829E+01	1,133E-01	2,380E-10	4,687E-11
MG + Empírico	54,413	216,400	2,120E-06	1,829E+01	1,133E-01	2,380E-10	4,687E-11
MG + Mitin	54,476	251,368	3,661E-05	3,159E+02	1,822E-01	5,789E-09	3,226E-10
MG + Épsilon escalar	14,570	251,564	4,295E-03	3,706E+04	4,031E-01	6,759E-07	1,003E-09
MG + Rho escalar	14,586	251,556	1,750E-03	1,510E+04	3,471E-01	4,094E-06	1,752E-06
MG + Épsilon topológico	56,160	399,448	8,179E-10	7,057E-03	3,058E-02	3,141E-12	1,103E-12
MG + Rho topológico	15,819	399,660	7,118E-05	6,141E+02	2,035E-01	4,088E-06	1,748E-06
MG + Múltipla ext. Aitken	14,586	251,572	5,554E-03	4,792E+04	4,208E-01	8,792E-07	1,444E-09

Obs.: Não foi possível realizar a Múltipla extrapolação de Mitin, pois o processo iterativo convergiu antes de completar as nove iterações externas que seriam usadas por este extrapolador.

Tabela 4.7: Resultados dos parâmetros estudados para o caso 2-B ($N = 1025 \times 1025$ e $\varepsilon = 10^{-10}$)

Metodologia	$t_{CPU} (s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	16,037	119,636	8,313E-12	4,188E-02	4,120E-02	4,933E-13	2,350E-13
MG + 1 ITE	17,956	119,636	3,500E-13	4,210E-02	4,130E-02	2,068E-14	1,000E-14
MG + Aitken	86,440	218,468	4,753E-10	5,718E+01	9,206E-02	4,599E-14	1,724E-15
MG + Empírico	86,440	218,468	4,753E-10	5,718E+01	9,206E-02	4,599E-14	1,724E-15
MG + Mitin	87,267	251,364	3,405E-09	4,096E+02	1,145E-01	4,350E-13	8,516E-15
MG + Épsilon escalar	22,293	251,560	6,745E-09	8,113E+02	1,236E-01	8,114E-13	1,575E-15
MG + Rho escalar	22,261	251,556	9,300E-08	1,118E+04	1,654E-01	2,841E-10	1,304E-10
MG + Épsilon topológico	89,139	399,448	1,080E-14	1,300E-03	2,806E-02	6,435E-17	2,399E-17
MG + Rho topológico	23,447	399,652	4,763E-09	5,729E+02	1,189E-01	2,839E-10	1,303E-10
MG + Múltipla ext. Aitken	22,246	251,576	4,697E-07	5,650E+04	1,980E-01	7,880E-11	1,131E-13
MG + Múltipla ext. Mitin	22,745	317,372	7,103E-07	8,544E+04	2,074E-01	7,811E-11	1,652E-13

Tabela 4.8: Resultados dos parâmetros estudados para o caso 2-C ($N = 1025 \times 1025$ e $\varepsilon = 10^{-15}$)

Metodologia	$t_{CPU} (s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	21,871	119,640	6,272E-16	4,238E-02	4,149E-02	3,672E-17	1,816E-17
MG + 1 ITE	23,775	119,636	2,663E-17	4,246E-02	4,157E-02	1,553E-18	7,749E-19
MG + Aitken	119,076	218,468	3,145E-14	5,014E+01	7,495E-02	4,118E-18	7,301E-20
MG + Empírico	119,076	218,468	3,145E-14	5,014E+01	7,495E-02	4,118E-18	7,301E-20
MG + Mitin	119,029	251,360	8,701E-14	1,387E+02	8,158E-02	9,630E-18	3,349E-19
MG + Épsilon escalar	29,874	251,556	3,044E-11	4,854E+04	1,329E-01	5,152E-15	7,111E-18
MG + Rho escalar	29,921	251,564	4,511E-12	7,193E+03	1,133E-01	2,066E-14	1,000E-14
MG + Épsilon topológico	121,525	399,444	1,648E-19	2,628E-04	2,721E-02	1,213E-21	5,165E-22
MG + Rho topológico	31,169	399,656	3,498E-13	5,577E+02	9,161E-02	2,066E-14	9,998E-15
MG + Múltipla ext. Aitken	29,952	251,576	4,716E-12	7,519E+03	1,137E-01	7,045E-16	1,856E-18
MG + Múltipla ext. Mitin	30,405	317,368	4,702E-10	7,497E+05	1,669E-01	6,402E-14	1,098E-16

Fazendo uma análise prévia dos resultados dos parâmetros mostrados nas tabelas anteriores, o extrapolador Épsilon topológico foi o que apresentou os melhores resultados em relação aos demais no que diz respeito à queda do erro de iteração, do resíduo e dos fatores de convergência. Com base neste fato, somente este extrapolador será utilizado para uma análise adicional onde $N = 4097 \times 4097$ e para os três valores da tolerância.

Tabela 4.9: Resultados dos parâmetros estudados para o caso 3-A ($N = 4097 \times 4097$ e $\varepsilon = 10^{-6}$)

Metodologia	$t_{CPU} (s)$	$M(GB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	164,004	1,844	1,158E-07	4,062E-02	4,099E-02	6,881E-09	3,088E-09
MG + 1 ITE	195,142	1,844	4,770E-09	4,117E-02	4,102E-02	2,844E-10	1,305E-10
MG + Épsilon topológico	841,813	6,311	6,946E-10	5,996E-03	2,975E-02	3,137E-12	1,103E-12

Tabela 4.10: Resultados dos parâmetros estudados para o caso 3-B ($N = 4097 \times 4097$ e $\varepsilon = 10^{-10}$)

Metodologia	$t_{CPU} (s)$	$M(GB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	258,416	1,844	8,309E-12	4,188E-02	4,120E-02	4,934E-13	2,352E-13
MG + 1 ITE	288,945	1,844	3,499E-13	4,211E-02	4,130E-02	2,068E-14	1,001E-14
MG + Épsilon topológico	1337,631	6,311	1,080E-14	1,300E-03	2,806E-02	6,431E-17	2,400E-17

Tabela 4.11: Resultados dos parâmetros estudados para o caso 3-C ($N = 4097 \times 4097$ e $\varepsilon = 10^{-15}$)

Metodologia	$t_{CPU} (s)$	$M(GM)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	351,579	1,844	6,270E-16	4,238E-02	4,148E-02	3,673E-17	1,818E-17
MG + 1 ITE	382,140	1,844	2,663E-17	4,247E-02	4,157E-02	1,554E-18	7,756E-19
MG + Épsilon topológico	1847,473	6,311	1,668E-19	2,661E-04	2,723E-02	1,213E-21	5,169E-22

4.1.4 Análise dos resultados e discussão

Na coluna $\|R\|_2$ da Tab. 4.3 (caso 1-A) observa-se que os extrapoladores mais eficientes foram: Aitken, Empírico e Épsilon topológico. Todos apresentaram a mesma ordem de grandeza (10^{-9}) para este parâmetro, resultado semelhante à metodologia que utilizou mais uma iteração externa (MG + 1 ITE). Para $\|E_n\|_\infty$, os extrapoladores Aitken e Empírico e a metodologia MG + 1 ITE mantiveram ordem de grandeza 10^{-9} , enquanto que o extrapolador Épsilon topológico reduziu a ordem de grandeza para 10^{-12} . Em relação a $\|E_n\|_2$, o uso dos extrapoladores Aitken e Empírico apresentou ordem de grandeza de 10^{-11} , MG + 1 ITE manteve a ordem 10^{-10} e o uso do Épsilon topológico reduziu a ordem de grandeza do erro de iteração dada pela norma euclidiana para 10^{-12} . Ao se considerar a análise da eficiência do processo iterativo pelo fator de convergência empírico ($q^{(k)}$), nota-se em todas as tabelas que o extrapolador Épsilon topológico apresentou os melhores resultados, ou seja, os menores valores para este parâmetro. Quanto ao fator de convergência média empírico ($\hat{q}^{(k)}$), a ordem de grandeza deste parâmetro para uma iteração adicional do *Multigrid* (MG + 1 ITE) e para o uso do extrapolador Épsilon topológico (MG + Épsilon topológico) foi a mesma em todos os casos estudados, sendo que nenhum outro extrapolador apresentou resultados melhores.

Quanto ao tempo de processamento, pode-se observar em todas as tabelas que o tempo necessário para efetuar a extrapolação (para qualquer extrapolador) é maior que o tempo necessário para que seja realizado um ciclo adicional do *Multigrid*, o que indica que em relação a este parâmetro, todos os extrapoladores mostraram-se menos eficientes que um ciclo adicional do *Multigrid*.

Quanto à memória de armazenamento dos dados (M), quando um extrapolador é utilizado, as últimas soluções devem ser armazenadas para que esses valores sejam utilizados no cálculo referente à extrapolação. Desta forma, para todos os casos com extrapoladores, o pico de memória sempre foi maior, se comparado à metodologia com uma iteração externa adicional do método *Multigrid*.

Sendo assim, tendo em vista os resultados obtidos nos parâmetros estudados, principalmente a redução que ocorreu no erro de iteração; neste item especificamente o extrapolador Épsilon topológico mostrou ser o mais eficiente, apesar do alto custo computacional (t_{CPU} e memória RAM).

Foram feitas análises similares nas Tabs. 4.4 a 4.8, e os resultados foram análogos, ou seja, o extrapolador Épsilon topológico ao final do método *Multigrid* foi a metodologia que apresentou o melhor desempenho. Por esse motivo, as tabelas Tabs. 4.9, 4.10 e 4.11 apresentaram os resultados dos casos 3-A, 3-B e 3-C, respectivamente para as metodologias MG, MG + 1 ITE e MG + Épsilon topológico. Nos três casos, o uso do extrapolador Épsilon topológico mostrou-se mais eficiente do que uma iteração adicional do *Multigrid* (MG + 1 ITE).

A Fig. 4.1 apresenta o comportamento da $\|R\|_2$, para $\varepsilon = 10^{-15}$, em função de N usando as metodologias MG, MG + 1 ITE e MG + Épsilon topológico. A figura apresenta-se em escala bilogárítmica. Para os outros valores de ε o resultado é análogo, conforme pode ser verificado no apêndice A.

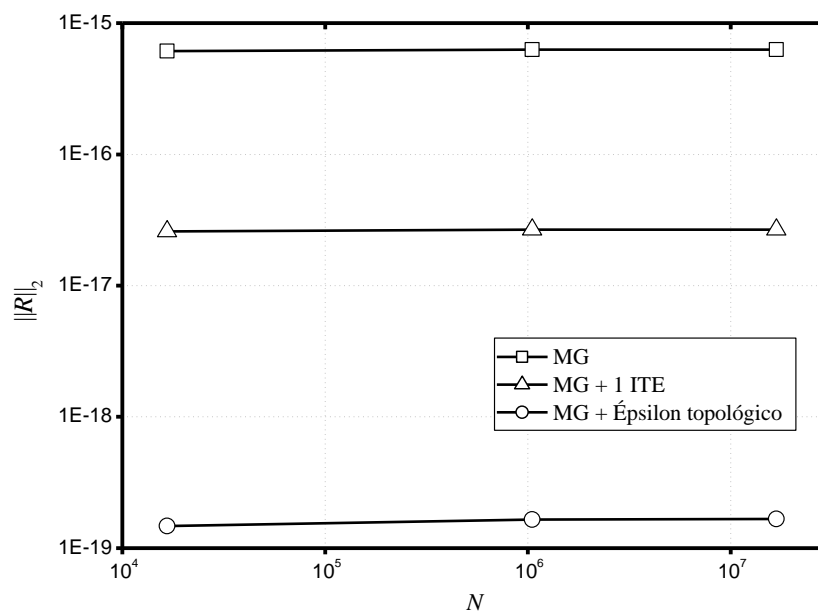


Figura 4.1 – Comportamento da $\|R\|_2$ em função de N para $\varepsilon = 10^{-15}$.

Nesta figura, percebe-se que $\|R\|_2$ ficou praticamente constante para cada uma das metodologias estudadas, ou seja, o número de incógnitas não interfere significativamente na queda de $\|R\|_2$ ao se resolver o problema proposto usando as metodologias MG, MG + 1 ITE e MG + Épsilon topológico, o que era esperado, visto que o critério de parada utilizado foi o mesmo.

Pode-se perceber ainda (ver apêndice A) que a $\|R\|_2$ apresentou valores menores quando o extrapolador Épsilon topológico associado ao método *Multigrid* foi utilizado, independentemente do ε utilizado, sendo que esta melhora foi tornando-se cada vez mais

significativa a partir do momento em que os valores de ε foram se tornando mais rigorosos (menores valores de ε).

A Fig. 4.2 apresenta o comportamento da $\|R\|_2$, para $N = 4097 \times 4097$, de acordo com os valores de ε usando as metodologias MG, MG + 1 ITE e MG + Épsilon topológico. A figura apresenta-se em escala bilogarítmica. Para os outros valores de N o resultado é análogo, conforme pode ser verificado no apêndice A.

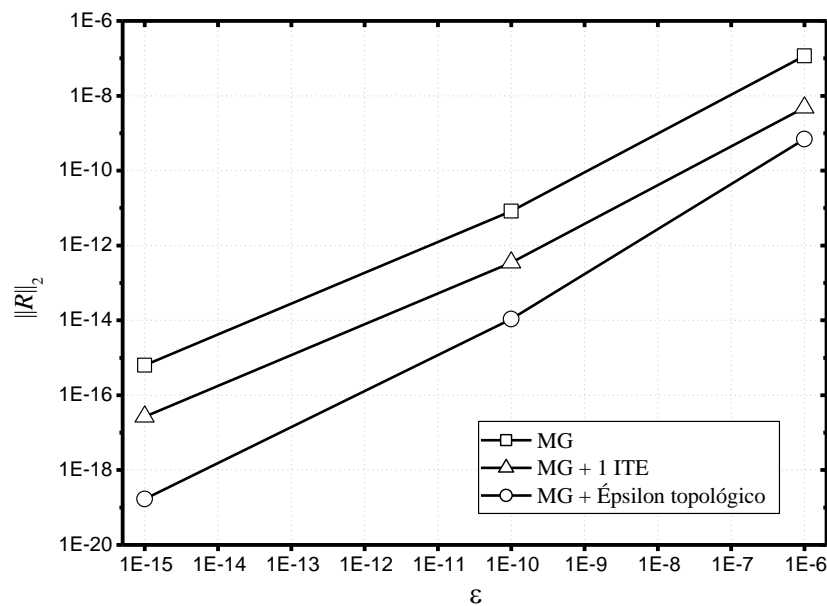


Figura 4.2 – Comportamento da $\|R\|_2$ em função de ε para $N = 4097 \times 4097$.

Nesta Fig. 4.2, comparando-se as curvas das três diferentes metodologias de resolução do problema, percebe-se que $\|R\|_2$ apresentou valores menores quando o extrapolador Épsilon topológico foi associado ao *Multigrid*, quando comparado a uma iteração adicional do *Multigrid*, independentemente dos valores de ε .

De acordo com as Figs. 4.1 e 4.2 e as figuras presentes no apêndice A, pode-se observar de maneira mais clara que para todos os valores de ε e para todos os valores de N , o uso do extrapolador Épsilon topológico ao final do *Multigrid* foi a metodologia que mais reduziu $\|R\|_2$.

A Fig. 4.3 apresenta o comportamento da $\|E_n\|_2$, para $\varepsilon = 10^{-15}$, de acordo com os valores de N usando as metodologias MG, MG + 1 ITE e MG + Épsilon topológico. A figura apresenta-se em escala bilogarítmica. Para os outros valores de ε o resultado é análogo, conforme pode ser verificado no apêndice A.

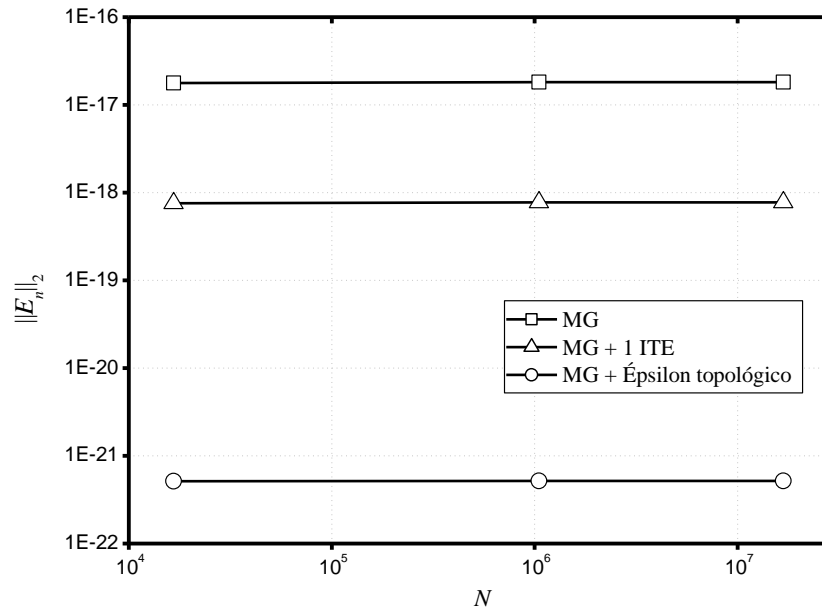


Figura 4.3 – Comportamento da $\|E_n\|_2$ em função de N para $\varepsilon = 10^{-15}$.

Na Fig. 4.3, percebe-se que $\|E_n\|_2$, assim como aconteceu com $\|R\|_2$ (Fig. 4.1), ficou praticamente constante para cada metodologia estudada, ou seja, o número de incógnitas não interfere significativamente na queda de $\|E_n\|_2$ ao se resolver o problema proposto usando as metodologias MG, MG + 1 ITE e MG + Épsilon topológico.

Pode-se perceber ainda (ver apêndice A) que a $\|E_n\|_2$ apresentou valores menores quando o extrapolador Épsilon topológico associado ao *Multigrid* foi utilizado, independentemente dos valores de ε , sendo que esta melhora foi tornando-se cada vez mais significativa a partir do momento em que os critérios de convergência foram se tornando mais rigorosos (menores valores de ε).

Na Fig. 4.4 é apresentado o comportamento da $\|E_n\|_2$, para $N = 4097 \times 4097$, de acordo os valores de ε , usando as metodologias MG, MG + 1 ITE e MG + Épsilon topológico. A figura apresenta-se em escala bilogarítmica. Para os outros valores de N o resultado é análogo.

Na Fig. 4.4, comparando-se as curvas das três diferentes metodologias de resolução do problema, percebe-se que $\|E_n\|_2$ apresentou valores menores quando o extrapolador Épsilon topológico foi associado ao *Multigrid*, quando comparado a uma iteração adicional do *Multigrid*, independentemente dos valores de ε .

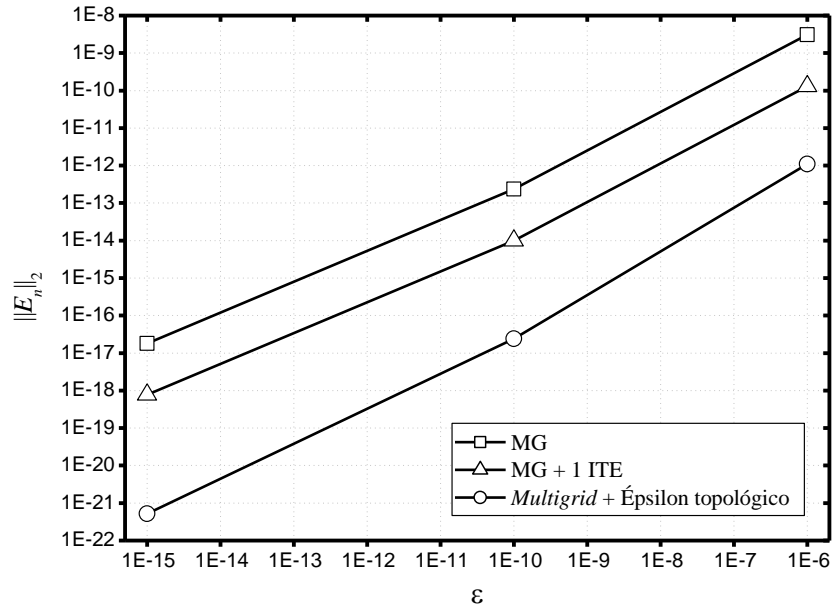


Figura 4.4 – Comportamento da $\|E_n\|_2$ em função de ε para $N = 4097 \times 4097$.

De acordo com as Figs. 4.3 e 4.4 e as figuras presentes no apêndice A, pode-se observar que para todos os valores de ε e para todos os valores de N , o uso do extrapolador Épsilon topológico ao final do *Multigrid* foi a metodologia que mais reduziu $\|E_n\|_2$.

O gráfico da Fig. 4.5 apresenta a variação do t_{CPU} em função de N , com $\varepsilon = 10^{-15}$, para as três metodologias de resolução do problema proposto. O gráfico apresenta-se em escalar bilogarítmica. Para os outros valores de ε o resultado é análogo, conforme pode ser verificado no apêndice A.

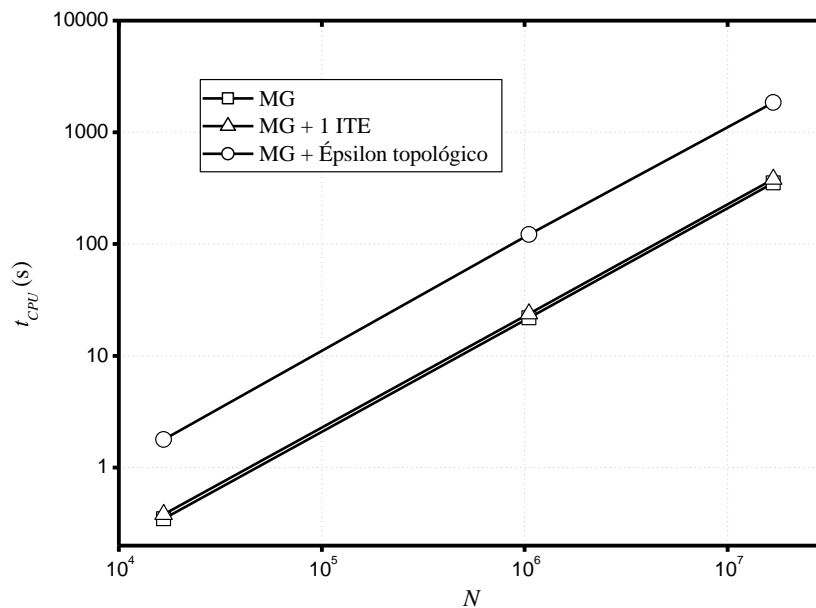


Figura 4.5 – Comportamento do t_{CPU} em função de N para $\varepsilon = 10^{-15}$.

Nessa figura pode-se observar o quanto o uso do extrapolador Épsilon topológico faz aumentar o tempo de processamento dos dados e obtenção das soluções numéricas e esse aumento se torna maior quando o problema é resolvido usando um número maior de incógnitas. Porém, por mais que o tempo de CPU seja maior, a acurácia da solução também é maior.

Com base nos resultados apresentados nas tabelas, gráficos e na discussão desenvolvida, conclui-se que o extrapolador que se mostrou mais eficiente na maioria dos parâmetros ($\|\mathbf{R}\|_2$, $\mathbf{q}^{(k)}$, $\hat{\mathbf{q}}^{(k)}$, $\|\mathbf{E}_n\|_\infty$ e $\|\mathbf{E}_n\|_2$) foi o Épsilon topológico.

4.2 USO DE EXTRAPOLADORES DURANTE O MÉTODO *MULTIGRID*

O objetivo dos testes apresentados nas seções anteriores era determinar qual o extrapolador é o mais eficiente em relação aos parâmetros abordados (ou pelo menos, na maioria destes parâmetros) e verificou-se que o extrapolador Épsilon topológico foi o que apresentou os melhores resultados.

Então, o próximo passo da presente pesquisa foi o de inserir este extrapolador durante o processo iterativo do método *Multigrid*. Portanto, cinco soluções das iterações externas (ciclos V) do *Multigrid* são armazenadas (não considerando a estimativa inicial), e estas são extrapoladas com o uso do Épsilon topológico. Assim, será gerada uma solução que servirá de estimativa inicial para a próxima iteração externa e novamente cinco soluções das iterações externas do *Multigrid* são armazenadas e estas extrapoladas com o uso do Épsilon topológico. O processo se repete até que o critério de parada seja atingido.

Para se determinar a eficiência dessa metodologia, comparam-se os resultados obtidos com o uso do *Multigrid* até se atingir o critério de convergência (MG), com os resultados obtidos com o uso do extrapolador Épsilon topológico durante o *Multigrid* (MG + Épsilon topológico) para os nove casos anteriores (dados pela Tab. 4.2).

4.2.1 Resultados

As soluções foram obtidas com precisão quádrupla. As tabelas a seguir apresentam os resultados das simulações feitas para resolução do modelo matemático, com apenas três algarismos decimais.

Em todas as tabelas, a primeira linha apresenta os resultados dos parâmetros estudados na resolução do problema com o *Multigrid* até que o critério de convergência fosse atingido (esta linha será designada por MG) e a segunda linha contém os resultados dos parâmetros obtidos com o extrapolador Épsilon topológico efetuando as extrapolações durante o *Multigrid* até ser atingido o mesmo critério de parada (esta linha será designada por MG + Épsilon topológico).

As colunas apresentam a metodologia utilizada na resolução do problema, o tempo de CPU (t_{CPU}), em segundos; o pico de memória de armazenamento (M); a norma euclidiana do resíduo adimensionalizada pela norma euclidiana do resíduo na estimativa inicial ($\|R\|_2$); o fator de convergência empírico ($q^{(k)}$); o fator de convergência média empírico ($\hat{q}^{(k)}$); a norma infinito erro de iteração ($\|E_n\|_\infty$) e a norma euclidiana do erro de iteração ($\|E_n\|_2$).

Tabela 4.12: Resultados dos parâmetros estudados para o caso 1-A ($N = 129 \times 129$ e $\varepsilon = 10^{-6}$)

Metodologia	$t_{CPU}(s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	0,163	6,272	1,148E-07	4,052E-02	4,092E-02	6,817E-09	3,042E-09
MG + Épsilon topológico	0,182	8,620	1,148E-07	4,052E-02	4,092E-02	6,817E-09	3,042E-09

Obs.: Para este caso, com cinco iterações externas do *Multigrid*, o critério de convergência já foi atingido, ou seja, não foram realizadas extrapolações. A diferença no t_{CPU} ocorre devido ao armazenamento das soluções que seriam utilizadas na extrapolação.

Tabela 4.13: Resultados dos parâmetros estudados para o caso 1-B ($N = 129 \times 129$ e $\varepsilon = 10^{-10}$)

Metodologia	$t_{CPU}(s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	0,257	6,272	8,168E-12	4,176E-02	4,111E-02	4,865E-13	2,307E-13
MG + Épsilon topológico	0,254	10,916	2,358E-12	2,053E-04	1,693E-02	9,427E-14	3,323E-14

Tabela 4.14: Resultados dos parâmetros estudados para o caso 1-C ($N = 129 \times 129$ e $\varepsilon = 10^{-15}$)

Metodologia	$t_{CPU}(s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	0,348	6,272	6,110E-16	4,226E-02	4,139E-02	3,609E-17	1,776E-17
MG + Épsilon topológico	0,359	10,900	2,162E-16	2,271E-02	1,817E-02	1,986E-18	8,514E-19

Tabela 4.15: Resultados dos parâmetros estudados para o caso 2-A ($N = 1025 \times 1025$ e $\varepsilon = 10^{-6}$)

Metodologia	$t_{CPU}(s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	10,156	119,636	1,159E-07	4,062E-02	4,100E-02	6,880E-09	3,086E-09
MG + Épsilon topológico	11,544	251,360	1,159E-07	4,062E-02	4,100E-02	6,880E-09	3,086E-09

Obs.: Para este caso, com cinco iterações externas do *Multigrid*, o critério de convergência já foi atingido, ou seja, não foram realizadas extrapolações. A diferença no t_{CPU} ocorre devido ao armazenamento das soluções que seriam utilizadas na extrapolação.

Tabela 4.16: Resultados dos parâmetros estudados para o caso 2-B ($N = 1025 \times 1025$ e $\varepsilon = 10^{-10}$)

Metodologia	$t_{CPU}(s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	16,037	119,636	8,313E-12	4,188E-02	4,120E-02	4,933E-13	2,350E-13
MG + Épsilon topológico	15,975	399,432	1,816E-12	1,567E-04	1,621E-02	8,779E-14	3,110E-14

Tabela 4.17: Resultados dos parâmetros estudados para o caso 2-C ($N = 1025 \times 1025$ e $\varepsilon = 10^{-15}$)

Metodologia	$t_{CPU}(s)$	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	21,871	119,640	6,272E-16	4,238E-02	4,149E-02	3,672E-17	1,816E-17
MG + Épsilon topológico	22,651	399,436	2,305E-16	2,398E-02	1,830E-02	1,913E-18	8,249E-19

Tabela 4.18: Resultados dos parâmetros estudados para o caso 3-A ($N = 4097 \times 4097$ e $\varepsilon = 10^{-6}$)

Metodologia	$t_{CPU}(s)$	$M(GB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	164,004	1,844	1,158E-07	4,062E-02	4,099E-02	6,881E-09	3,088E-09
MG + Épsilon topológico	186,842	3,946	1,158E-07	4,062E-02	4,099E-02	6,881E-09	3,088E-09

Obs.: Para este caso, com cinco iterações externas do *Multigrid*, o critério de convergência já foi atingido, ou seja, não foram realizadas extrapolações. A diferença no t_{CPU} ocorre devido ao armazenamento das soluções que seriam utilizadas na extrapolação.

Tabela 4.19: Resultados dos parâmetros estudados para o caso 3-B ($N = 4097 \times 4097$ e $\varepsilon = 10^{-10}$)

Metodologia	$t_{CPU}(s)$	$M(GB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	258,416	1,844	8,309E-12	4,188E-02	4,120E-02	4,934E-13	2,352E-13
MG + Épsilon topológico	257,480	6,311	1,726E-12	1,490E-04	1,607E-02	8,770E-14	3,109E-14

Tabela 4.20: Resultados dos parâmetros estudados para o caso 3-C ($N = 4097 \times 4097$ e $\varepsilon = 10^{-15}$)

Metodologia	$t_{CPU}(s)$	$M(GB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	351,579	1,844	6,270E-16	4,238E-02	4,148E-02	3,673E-17	1,818E-17
MG + Épsilon topológico	364,699	6,311	2,320E-16	2,417E-02	1,831E-02	1,912E-18	8,251E-19

4.2.2 Análise dos resultados e discussão

Tendo em vista que não foram realizadas extrapolações durante o *Multigrid* para os casos 1-A, 2-A e 3-A, os dados das Tabs. 4.12, 4.15 e 4.18 não farão parte da discussão.

Analisando a Tab. 4.13, observa-se que o uso do extrapolador Épsilon topológico durante o *Multigrid* atingiu o critério de parada em um tempo menor que o *Multigrid*, assim como o fator de convergência empírico, que apresentou um resultado melhor quando comparado ao valor obtido apenas com o *Multigrid*. O mesmo ocorre com o erro de iteração nas duas normas utilizadas. A ordem de grandeza do fator de convergência média empírico manteve-se constante para as duas metodologias abordadas, assim como, a ordem de grandeza do resíduo adimensionalizado com base na estimativa inicial (porém com uma leve vantagem para a aplicação da metodologia MG + Épsilon topológico). A análise das Tabs. 4.16 e 4.19 é análoga.

Analisando os resultados que aparecem na Tab. 4.14, observa-se que o uso do extrapolador Épsilon topológico durante o *Multigrid* atingiu o critério de parada em um tempo maior que o *Multigrid*, cerca de 3,7% acima, o que é um valor baixo tendo em vista que a norma infinito do erro de iteração caiu cerca de 95,5%. A ordem de grandeza do fator de convergência empírico das duas metodologias manteve-se constante, assim como a ordem de grandeza do fator de convergência média empírico e do resíduo adimensionalizado com base na estimativa inicial (porém com uma leve vantagem para a aplicação da metodologia MG + Épsilon topológico). Os resultados para o erro de iteração usando o extrapolador Épsilon topológico durante o *Multigrid* foram favoráveis, visto que nas duas normas utilizadas, apresentaram resultados melhores que aqueles obtidos apenas com o uso do *Multigrid*. A análise das Tabs. 4.17 e 4.20 é análoga.

A figura 4.6 apresenta o comportamento da norma l_2 do resíduo (em escala logarítmica) em função do número de iterações.

De acordo com a Fig. 4.6, percebe-se que até a quinta iteração, o resíduo cai igualmente para as duas metodologias. Porém, como na próxima iteração ocorre a primeira extrapolação, a norma do resíduo da metodologia MG + Épsilon topológico fica com magnitude menor que a metodologia MG (aproximadamente 200 vezes menor). Com toda esta redução, o critério de parada é atingido na 9ª iteração com o uso do MG + Épsilon topológico, enquanto são necessárias 11 iterações para interromper o processo no caso do MG. Para as outras malhas e outros critérios de parada, os resultados são análogos.

De acordo com as análises realizadas, pode-se concluir que o uso do extrapolador Épsilon topológico durante o *Multigrid*, reduz de forma satisfatória a magnitude do erro de iteração e do resíduo. Para confirmar os resultados, testes adicionais foram realizados e estão descritos na seção 4.3 a seguir.

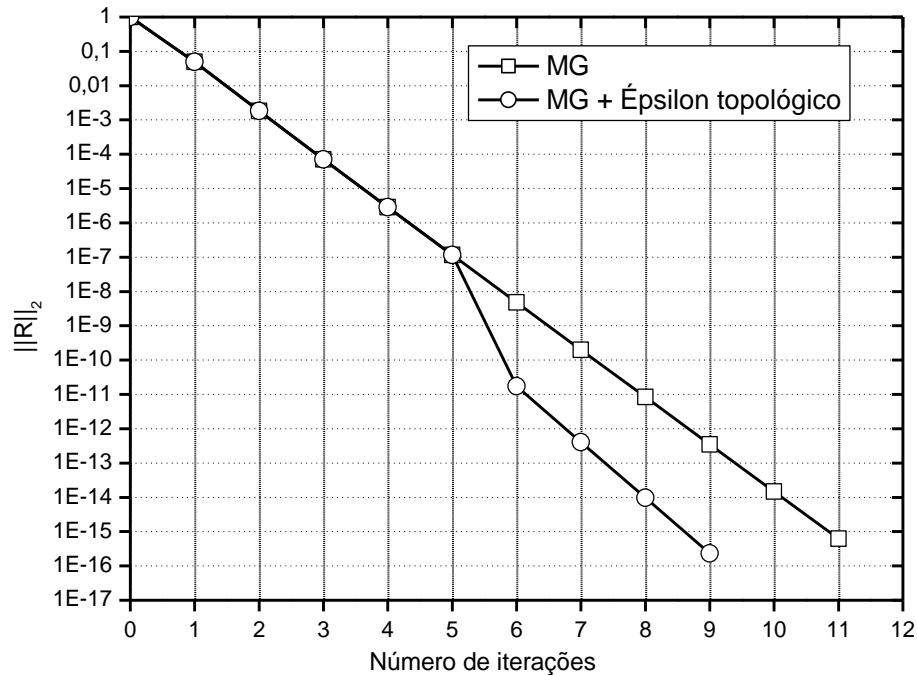


Figura 4.6 – Comportamento da $\|R\|_2$ em função do número de iterações, para $N = 4097 \times 4097$.

4.3 TESTES ADICIONAIS

Nos testes anteriores, o extrapolador Épsilon topológico reduziu bem a norma do erro de iteração, em todos os casos estudados, quando a extrapolação foi realizada ao final do *Multigrid* e também quando as extrapolações foram feitas durante o *Multigrid*. Desta forma, um dos objetivos do trabalho foi alcançado: reduzir o erro de iteração usando extrapoladores.

Na busca de uma análise mais precisa em relação à redução do tempo de CPU, testes adicionais foram realizados a fim de se chegar a uma conclusão sobre esse parâmetro (t_{CPU}). Isso porque os casos abordados não apresentaram resultados satisfatórios quando as extrapolações foram realizadas tanto ao final quanto durante o *Multigrid*.

Foram realizadas novas simulações usando o *Multigrid* com números de incógnitas de $N = 9 \times 9$ até $N = 8193 \times 8193$ e $\varepsilon = 10^{-20}$, a fim de se obter os resultados para o parâmetro t_{CPU} . Em seguida, aplicou-se o extrapolador Épsilon topológico durante o *Multigrid* na obtenção dos resultados para os mesmos dados.

A Tab. 4.21 apresenta os resultados obtidos para o tempo de CPU para as duas metodologias abordadas: *Multigrid* (denotada por MG) e Épsilon topológico durante o *Multigrid* (denotada por MG + Épsilon topológico):

Analisando a Tab. 4.21 verifica-se que os resultados obtidos para o t_{CPU} para as duas metodologias, MG e MG + Épsilon topológico, foram muito próximos.

Tabela 4.21: Tempo de CPU (s) para MG e MG + Épsilon topológico para $\varepsilon = 10^{-20}$

N	MG	MG + Épsilon topológico
9x9	0,001	0,001
17x17	0,007	0,007
33x33	0,029	0,030
65x65	0,118	0,127
129x129	0,468	0,515
257x257	1,872	2,012
513x513	7,426	7,909
1025x1025	29,625	29,219
2049x2049	118,374	117,063
4097x4097	476,146	470,811
8193x8193	1897,737	1876,767

Para análise da complexidade dos algoritmos que envolvem as metodologias MG e MG + Épsilon topológico, calculou-se o expoente p , obtido pelo método dos mínimos quadrados, para a função dada por:

$$t_{CPU}(N) = cN^p, \quad (4.3)$$

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

Para o método *Multigrid* ideal, $p = 1$, significando que o esforço computacional cresce linearmente com o tamanho da malha (BRANDT, 1977; TROTTENBERG et al., 2001). Assim, para um dado hardware e compilador, quanto menor for p , mais eficiente é o algoritmo e quanto menor o valor de c , mais rápido é o algoritmo.

A Tab. 4.22 apresenta os valores de p e c dos ajustes de curvas obtidos para os algoritmos em estudo.

Na Tab. 4.22 observa-se que o valor de p é levemente menor quando é utilizada a metodologia MG + Épsilon topológico, mas essa diferença é praticamente insignificante a ponto de que se pode afirmar que as duas metodologias tem praticamente a mesma ordem de

complexidade. Portanto, essa metodologia é levemente mais rápida em relação ao uso do *Multigrid* quando o número de incógnitas aumenta (veja Tab. 4.21).

Tabela 4.22: Valores de p e c para os algoritmos das metodologias em estudo

Metodologia	c	p
MG	0,0755	1,03258
MG + Épsilon topológico	0,0851	1,02957

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

$$S_p = \frac{t_{CPU}(\text{algoritmo A})}{t_{CPU}(\text{algoritmo B})}. \quad (4.4)$$

A Tab. 4.23 apresenta o *speed-up* da metodologia *Multigrid* (algoritmo A) em relação à metodologia MG + Épsilon topológico (algoritmo B) para $\varepsilon = 10^{-20}$.

Tabela 4.23: *Speed-up* de MG em relação ao MG + Épsilon topológico para $\varepsilon = 10^{-20}$

N	S_p
9x9	1
17x17	1
33x33	0,966
65x65	0,929
129x129	0,908
257x257	0,930
513x513	0,938
1025x1025	1,013
2049x2049	1,011
4097x4097	1,011
8193x8193	1,011

Observa-se que o algoritmo que associa o extrapolador Épsilon topológico durante o *Multigrid* mostrou-se mais rápido a partir de $N = 1025 \times 1025$ (*speed-up* maior que 1), o que permite concluir que associar este extrapolador durante o processo iterativo além de reduzir o erro de iteração para qualquer valor de N , também reduz o tempo de CPU, para $N \geq 1025 \times 1025$, para o caso estudado.

Um estudo semelhante foi realizado para o critério de convergência $\varepsilon = 10^{-25}$ e os resultados podem ser vistos no apêndice B, o que permite afirmar que o uso do extrapolador Épsilon topológico durante o processo iterativo do método *Multigrid*, é mais efetivo na redução do tempo de CPU à medida que aumenta o número de incógnitas e também à medida que o critério de convergência torna-se mais rigoroso.

Com base nos resultados obtidos com o uso do extrapolador Épsilon topológico durante o *Multigrid*, pode-se concluir que essa metodologia é vantajosa, se comparada à metodologia que utiliza apenas o método *Multigrid* (MG), pois reduz o erro de iteração, melhora os fatores de convergência empíricos e, pode ser considerado equivalente ao tempo de CPU da metodologia MG.

A desvantagem em utilizar tal metodologia, é que o uso de memória aumenta quando é usado tal extrapolador. A tabela a seguir (Tab. 4.24) apresenta os valores das memórias (M) para os números de incógnitas de $N = 9 \times 9$ até $N = 8193 \times 8193$.

Tabela 4.24: Memórias e aumento percentual da memória para as metodologias MG e MG + Épsilon topológico

N	MG	MG + Épsilon topológico	Aumento na memória
9x9	4,408 MB	4,416 MB	0,181%
17x17	4,448 MB	4,568 MB	2,698%
33x33	4,556 MB	4,860 MB	6,673%
65x65	4,920 MB	6,132 MB	24,634%
129x129	6,272 MB	10,844 MB	72,895%
257x257	11,708 MB	29,428 MB	151,350%
513x513	33,304 MB	103,496 MB	210,761%
1025x1025	119,636 MB	399,400 MB	233,846%
2049x2049	0,464 GB	1,582 GB	240,948%
4097x4097	1,844 GB	6,311 GB	242,245%
8193x8193	7,361 GB	25,384 GB	244,844%

A Fig. 4.8 a seguir, em escala bilogarítmica, ilustra o aumento percentual no uso de memória da metodologia MG + Épsilon topológico, em relação à metodologia MG.

Com base na Tab. 4.24 e na Fig. 4.8, percebe-se que à medida que N cresce, a taxa de variação da memória da metodologia MG + Épsilon topológico em relação à metodologia MG aumenta, sendo que esta variação percentual torna-se assintótica, que é uma boa propriedade.

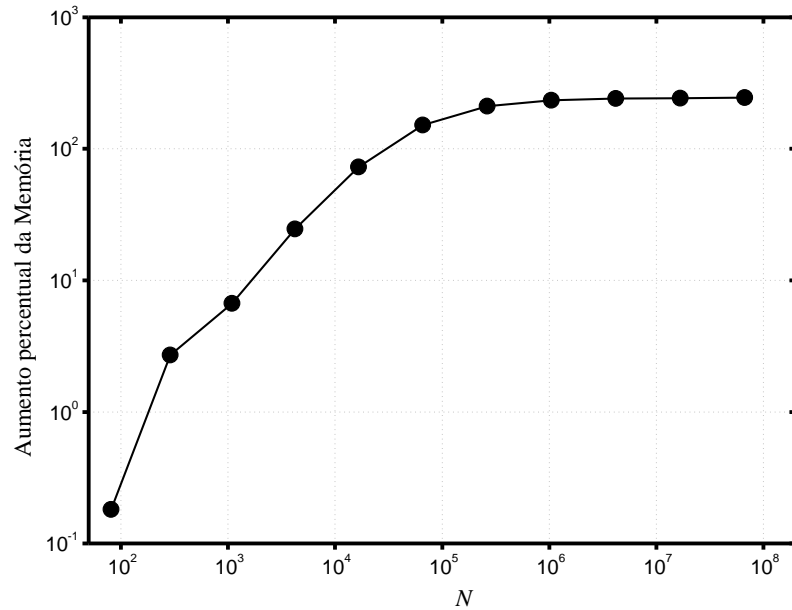


Figura 4.7 – Aumento percentual da memória utilizada para MG + Épsilon topológico em relação a MG em função de N para $\varepsilon = 10^{-20}$.

4.4 RESUMO DO CAPÍTULO 4

Neste capítulo foram apresentados os resultados das simulações para o problema bidimensional linear de condução de calor, governado pela equação de Poisson, usando o método *Multigrid* geométrico. Foram associados ao final do *Multigrid* os seguintes extrapoladores: Aitken, Empírico, Mitin, Épsilon escalar, Rho escalar, Épsilon topológico, Rho topológico, Múltipla extrapolação de Aitken e Múltipla extrapolação de Mitin, sendo que o extrapolador Épsilon topológico foi considerado o mais eficiente na maioria dos parâmetros estudados. Em seguida foram apresentados os resultados das simulações realizadas para resolução numérica do mesmo problema, mas agora, usando o extrapolador Épsilon topológico durante o *Multigrid*. Verificou-se que em ambos os casos, o uso do extrapolador Épsilon topológico reduziu de forma satisfatória o erro de iteração. Porém, quando as extrapolações foram realizadas no final do processo iterativo, o tempo de CPU apresentou valores elevados, mas quando as extrapolações foram realizadas durante o *Multigrid*, este parâmetro foi praticamente equivalente ao uso do *Multigrid* puro. Além disso, a memória de armazenamento de dados quando um extrapolador é usado, é sempre maior do que aquela necessária para armazenar os dados quando o *Multigrid* puro é utilizado.

5 CONCLUSÃO

A conclusão deste trabalho está dividida em três partes: constatações gerais, contribuições e trabalhos futuros, descritas a seguir.

5.1 CONSTATAÇÕES GERAIS

Neste trabalho foi resolvido numericamente o problema de condução de calor linear bidimensional, governado pela equação de Poisson, com condições de contorno de Dirichlet. Utilizou-se o MDF, com esquema de aproximação de segunda ordem (CDS) para discretização do modelo matemático, e o *solver* Gauss-Seidel *red-black*. Para a obtenção da solução, foi empregado o método *Multigrid* geométrico, com esquema de correção CS, restrição por ponderação completa, prolongação por interpolação bilinear e número máximo de níveis para os diversos casos estudados. Foram associados ao final do *Multigrid* os seguintes extrapoladores: Aitken, Empírico, Mitin, Épsilon escalar, Rho escalar, Épsilon topológico, Rho topológico, Múltipla extrapolação de Aitken e Múltipla extrapolação de Mitin. Durante o *Multigrid*, foi usado apenas o extrapolador Épsilon topológico.

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

1) A redução do valor da magnitude do resíduo adimensionalizada com base na estimativa inicial e também da magnitude do erro de iteração ao final do *Multigrid* foi maior com o uso do extrapolador Épsilon topológico do que com os outros extrapoladores e também com iteração adicional do *Multigrid*.

2) O tempo de CPU ao final do *Multigrid* foi maior para todos os extrapoladores estudados do que com uma iteração adicional do *Multigrid*.

3) O uso do extrapolador Épsilon topológico durante o *Multigrid* reduz mais a norma do resíduo adimensionalizada, as normas do erro de iteração e os fatores de convergência, quando comparado ao uso do *Multigrid* puro para resolução do problema proposto.

4) O tempo de CPU das extrapolações realizadas durante o *Multigrid* são praticamente equivalentes ao uso do *Multigrid* puro, sendo que os valores deste parâmetro se tornam melhores a partir do momento em que o número de incógnitas aumenta para um dado ε .

5) O pico de memória de armazenamento é sempre maior quando qualquer um dos extrapoladores é utilizado, mas seu crescimento percentual é assintótico com N .

5.2. CONTRIBUIÇÕES

Este trabalho contribui com a literatura no sentido que:

1) Obteve-se uma redução do erro de iteração e melhora no fator de convergência da resolução numérica do problema de condução de calor bidimensional, governado pela equação de Poisson, em malhas uniformes usando extrapolador Épsilon topológico durante os ciclos do método *Multigrid*.

2) O tempo de CPU necessário para obtenção das soluções numéricas do problema usando o extrapolador Épsilon topológico durante o *Multigrid* foi praticamente o mesmo tempo de CPU do *Multigrid* puro, sendo que a primeira metodologia mostrou-se mais vantajosa em relação a este parâmetro (Tempo de CPU) com o aumento do número de incógnitas.

5.3 TRABALHOS FUTUROS

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

- Ao se considerar a aplicação dos extrapoladores sobre as iterações finais de um processo iterativo, por se tratar de um pós-processamento, é razoável se considerar que o esforço computacional dependa apenas do número de pontos da malha, e independa da complexidade dos cálculos envolvidos no processo iterativo. Com isso, espera-se que tal aplicação seja mais eficaz em problemas que envolvam processos iterativos complexos – com alto custo computacional. Ou seja, presume-se que a aplicação do extrapolador tenha um custo computacional menor do que a realização de uma iteração externa que envolva um processo iterativo complexo (de alto custo computacional), como problemas modelados pelas equações de advecção-difusão, Burgers, Navier-Stokes, etc.

- Na literatura, existem outros métodos de extrapolação, como o extrapolador de Richardson, que poderiam ser utilizados juntamente ao método *Multigrid* com o intuito de se comparar novos resultados com os obtidos neste trabalho.

- Utilizou-se o *solver* Gauss-Seidel *red-black*. Propõe-se o estudo do *solver* Gauss-Seidel quatro cores.

- Utilizar técnicas de programação paralela com o objetivo de melhorar o desempenho do método *Multigrid*.

- Utilizaram-se extrapolações ao final do processo iterativo, após o mesmo atingir um critério de convergência pré-estabelecido e extrapolações durante o processo iterativo. Propõem-se outras formas de associar o extrapolador durante o *Multigrid*.
- Propõe-se realizar extrapolações múltiplas usando o extrapolador que apresentou os resultados mais favoráveis neste trabalho.
- Propõe-se verificar a aceleração do *Multigrid* usando recombinação de iteradas através dos subespaços de Krylov.

REFERÊNCIAS

BRANDT, A. Multi-level adaptive solutions to boundary-value problems. **Mathematics of Computation**, v. 31, pp. 333-390, 1977.

BREBBIA, C. A.; TELLES J. C. F.; WROBEL, L. C. **Boundary Element Techniques - Theory and Applications in Engineering**. Berlin: Springer-Verlag, 1984.

BREZINSKI, C. Forme confluente de l'epsilon-algorithme topologique. **Numer. Math.**, v.23, pp. 363-370, 1975.

BREZINSKI, C.; ZAGLIA, R. M. **Extrapolation Methods** - Theory and Practice. 2 ed., Amsterdam: Elsevier Science Publishers, 2002.

BREZINSKI, C.; ZAGLIA, R. M., A Review of Vector Convergence Acceleration Methods, With Applications to Linear Algebra Problems, **International Journal of Quantum Chemistry**, v. 109, pp. 1631–1639, 2008.

BRIGGS, W. L.; HENSON, V. E.; MCCORMICK, S. F. **A Multigrid Tutorial**, 2 ed. Philadelphia: SIAM, 2000.

BURDEN, R. L.; FAIRES, J. D. **Análise Numérica**. São Paulo: Pioneira Thomson Learning, 2007.

CHAPRA, S. C.; CANALE, R. P. **Introduction to Computing for Engineers**, 5 ed. New York: Mc-Graw-Hill, 2008.

CUNHA, C. **Métodos Numéricos**, 2 ed. Campinas, SP: Editora da UNICAMP, 2003.

DELAHAYE. J. P. E GERMAIN-BONNE, B. **The set of logarithmically convergent sequences cannot be accelerated**. Philadelphia: SIAM, 1982.

DELAHAYE. J. P. **Sequence Transformations**. Berlin: Springer, 1988.

EVANS, D. J. The extrapolated modified Aitken iteration method for solving elliptic difference equations. **The Computer Journal**, v. 6, n. 2, pp. 193-201, 1963.

FEDORENKO, R. P. On the Speed of Convergence of an Iteration Process, **USSR Comput. Math. And Math. Phys.**, v. 4, n. 3, 1964.

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

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

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

GALANTE, G. **Métodos *Multigrid* Paralelos em Malhas Não-Estruturadas Aplicados à Simulação de Problemas de Dinâmica de Fluidos Computacional e Transferência de Calor**. Dissertação (Mestrado). Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 2006.

GAO, Q.; JIANG, Z.; LIAO, T.; SONG, K., Application of the vector ε and ρ extrapolation methods in the acceleration of the Richardson–Lucy algorithm, **Optics Communications**, v. 283, pp. 4224–4229, 2010.

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

GRAVES-MORRIS, P. R; ROBERTS, D. E.; SALAMC, A., The epsilon algorithm and related topics, **Journal of Computational and Applied Mathematics**, v. 122, pp. 51-80, 2000.

HIRSCH, C. **Numerical Computational of Internal and External Flows**, v.1. New York: John Wiley & Sons, 1988.

HOMEIER, H.; HERBERT, H.; SEBASTIAN, R.; HARTMUL, K., Iterative solution of the Ornstein-Zernike equation with various closures using vector extrapolation, **Computer Physics Communications**, v.92, pp. 188-202, 1995.

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

INCROPERA, F. P.; DEWITT, D. P. BERGMAN, T.L.; LAVINE, A. S. **Fundamentos de Transferência de Calor e de Massa**, 6 ed. Rio de Janeiro: LTC Editora, 2008.

JBILOU, K.; SADOK, H. Vector extrapolation methods. Applications and numerical comparison. **Journal of Computational and Applied Mathematics**, v.122 , pp. 149-165, 2000.

KOKIOPOULOU, E.; FROSSARD, P. Accelerating Distributed Consensus Using Extrapolation, **IEEE Signal Processing Letters**, v. 14, n. 10, pp. 665-668, 2007.

LIMA, P.M.; GRAÇA, M.M. Convergence acceleration for boundary value problems with singularities using the E-algorithm, **Journal of Computational and Applied Mathematics**, v. 61, pp. 139-164, 1995.

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

MARCHI, C. H. **Verificação de soluções numéricas unidimensionais em dinâmica dos fluidos**. Tese (Doutorado em Engenharia Mecânica). Universidade Federal de Santa Catarina. Florianópolis, SC, 2001.

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

MESGARANI, H.; AGHAZADEH, N; PARMOUR, P. Aitken extrapolation and epsilon algorithm for an accelerated solution of weakly singular nonlinear Volterra integral equations. **Physical Scripta**, v.81, 2010

MITIN, A. V. Linear extrapolation in an iterative method for solving systems of equations, **U.S.S.R. Comput. Maths. Math. Phys.**, v.25, n.2, pp.1-6, 1985.

OLIVEIRA, F.; PINTO, M. A. V.; SANTIAGO, C. D.; MARCHI, C. H. Efeito de parâmetros do método Multigrid CS e FAS sobre o tempo de CPU em problemas 1D lineares e não-lineares. In: 27th Iberian Latin-American Congress on Computational Methods in Engineering (CILAMCE 2006), **Proceedings** Belém, PA, Brasil, 2006.

OLIVEIRA, F. **Efeitos de malhas anisotrópicas bidimensionais sobre o desempenho do método *Multigrid* geométrico**. Tese (Doutorado em Engenharia Mecânica). Universidade Federal do Paraná. Curitiba, PR, 2010.

OLIVEIRA, F.; PINTO, M. A. V.; MARCHI, C. H.; ARAKI, L. K. Efeito de malhas anisotrópicas bidimensionais sobre o desempenho do *Multigrid* Geométrico. In VI Congresso Nacional de Engenharia Mecânica. **Proceedings** Campina Grande, 2010.

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

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

PATANKAR, S. **Numerical heat transfer and fluid flow**. New York: Taylor and Francis, 1980.

PINTO, M. A. V.; MARCHI, C. H. Efeito dos parâmetros do método *Multigrid* CS e FAS sobre o tempo de CPU para a equação de Laplace bidimensional. In: 11th Brazilian Congress of Thermal Sciences and Engineering (ENCIT 2006). **Proceedings** Paper CIT 06-0348. Curitiba, 2006.

PINTO, M. A. V. **Comportamento do Método *Multigrid* Geométrico em Problemas de Transferência de Calor**. Tese (Doutorado em Métodos Numéricos em Engenharia). Universidade Federal do Paraná, Curitiba, PR, 2006.

PINTO, M. A. V.; MARCHI, C. H. Optimum parameters of a geometric Multigrid for the two-dimensional Laplace's equation. In 19th International Congress of Mechanical Engineering (COBEM 2007). **Proceedings** Brasília, 2007.

PINTO, M. A. V.; SANTIAGO, C. D.; MARCHI, C. H. Effect of Parameters of a Multigrid Method on CPU Time for One-dimensional Problems, In 19th International Congress of Mechanical Engineering (COBEM 2005). **Proceedings...** Ouro Preto, 2005.

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

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

SANTIAGO, C. D.; MARCHI, C. H. Optimum Parameters of a Geometric *Multigrid* for a Two-Dimensional Problem of Two-Equations, In 19th International Congress of Mechanical Engineering (COBEM 2007). **Proceedings** Brasília, 2007.

SANTIAGO, C. D.; MARCHI, C. H.; SOUZA, L. F. Análise do desempenho do método *Multigrid* geométrico com a formulação função de corrente e velocidade. In VI Congresso Nacional de Engenharia Mecânica. **Proceedings** Campina Grande, 2010.

SCHNEIDER, G, E., ZEDAN, M. A modified strongly implicit procedure for the numerical solution of field problems. **Numerical Heat Transfer**, v.4, pp. 1-19, 1981.

SHANKS, D. Non linear transformations of divergent and slowly convergent sequences. **Math. Phys.**, v. 34, pp. 1-42, 1955.

SKELBOE, S. Computation of the Periodic Steady-State Response of Nonlinear Networks by Extrapolation Methods, **IEEE Transactions on Circuits and Systems**, v. 27, n. 3, 1980.

SUERO, R.; PINTO, M. A. V.; MARCHI, C. H.; ARAKI, L. K.; ALVES, A. C. Otimização do método *Multigrid* algébrico para as equações bidimensionais de Laplace e Poisson. In VI Congresso Nacional de Engenharia Mecânica. **Proceedings** Campina Grande, 2010.

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

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

VERSTEEG, H. K.; MALALASEKERA, W. **An introduction to Computational Fluid Dynamics: The Finite Volume Method**, ed. 2, Harlow, England: Pearson/Prentice Hall, 2007.

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

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

WYNN, P. On a procrustean technique for the numerical transformation of slowly convergent sequences and series. **Proc. Cambridge Phil. Soc.**, v. 52, pp. 663-671, 1956.

WYNN, P. Acceleration techniques for iterated vector and matrix problems. **Math. Comput.**, v. 16, pp. 301-322, 1962.

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

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

APÊNDICE A – FIGURAS

Neste apêndice podem ser encontradas as figuras correspondentes aos gráficos do comportamento de alguns parâmetros em relação ao número de incógnitas do problema (N) ou em relação ao critério de parada (ε). Tais figuras são complementares em relação às que podem ser encontradas no capítulo 4. Todas as figuras deste apêndice estão em escala bilogarithmica.

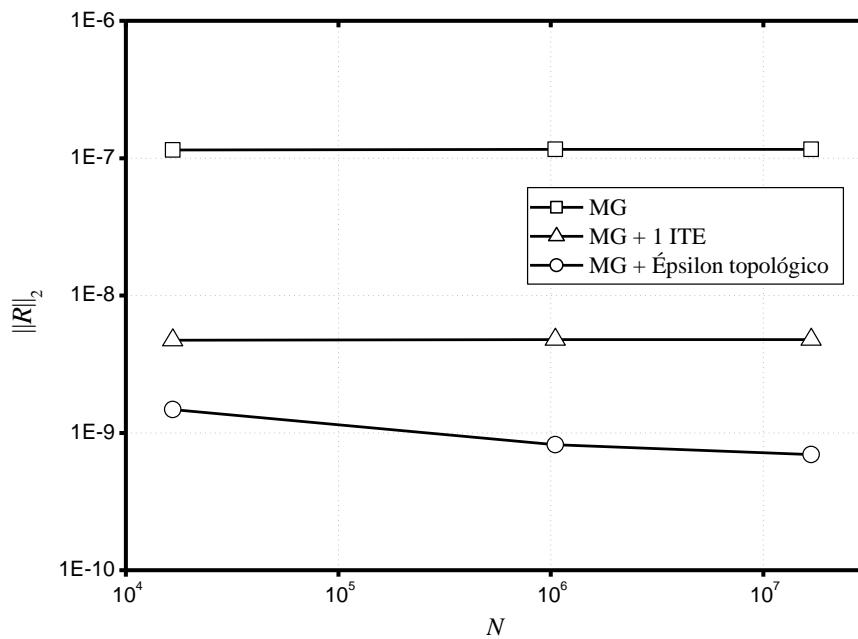


Figura A.1 – Comportamento da $\|R\|_2$ em função de N para $\varepsilon = 10^{-6}$.

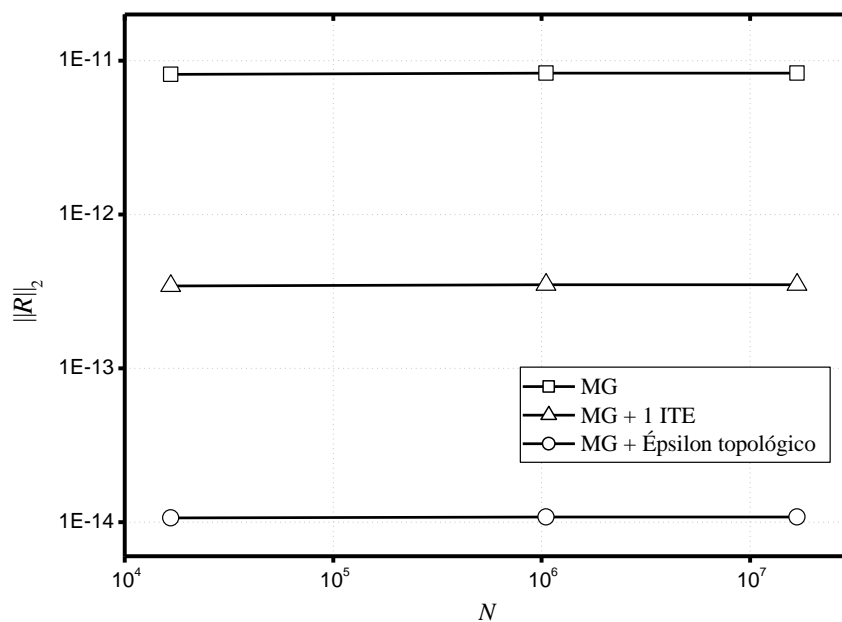


Figura A.2 – Comportamento da $\|R\|_2$ em função de N para $\varepsilon = 10^{-10}$.

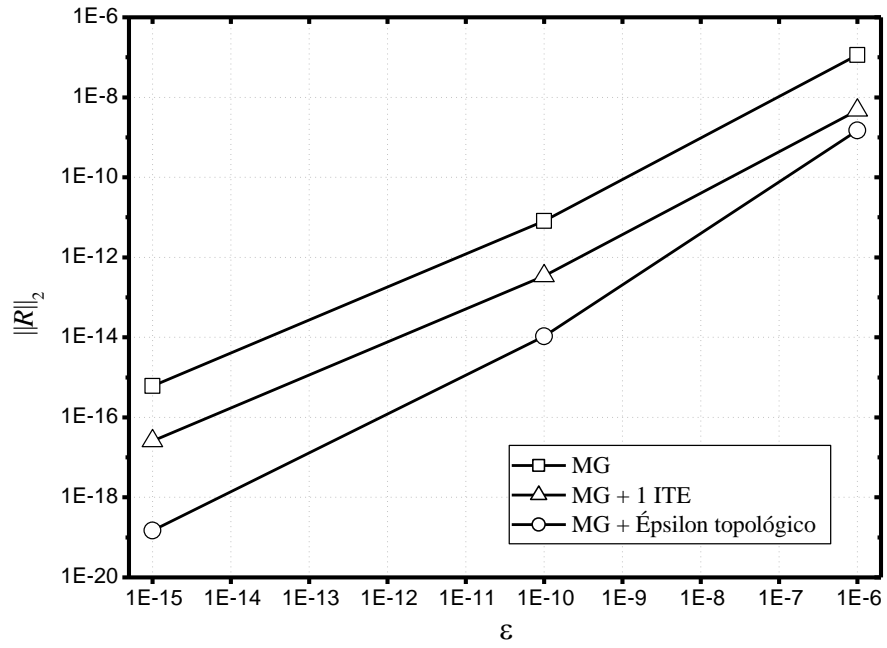


Figura A.3 – Comportamento da $\|R\|_2$ em função de ϵ para $N = 129 \times 129$.

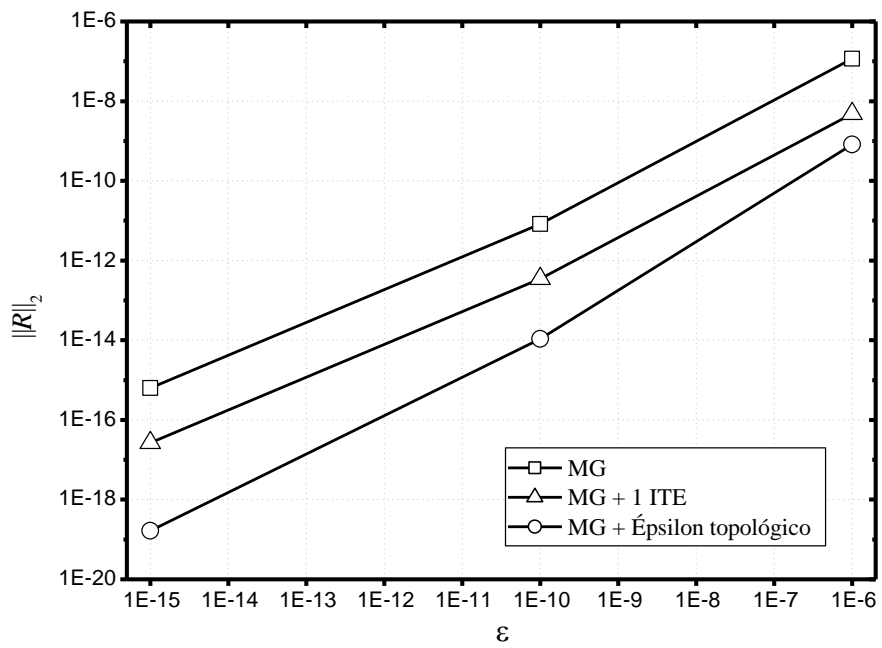


Figura A.4 – Comportamento da $\|R\|_2$ em função de ϵ para $N = 1025 \times 1025$.

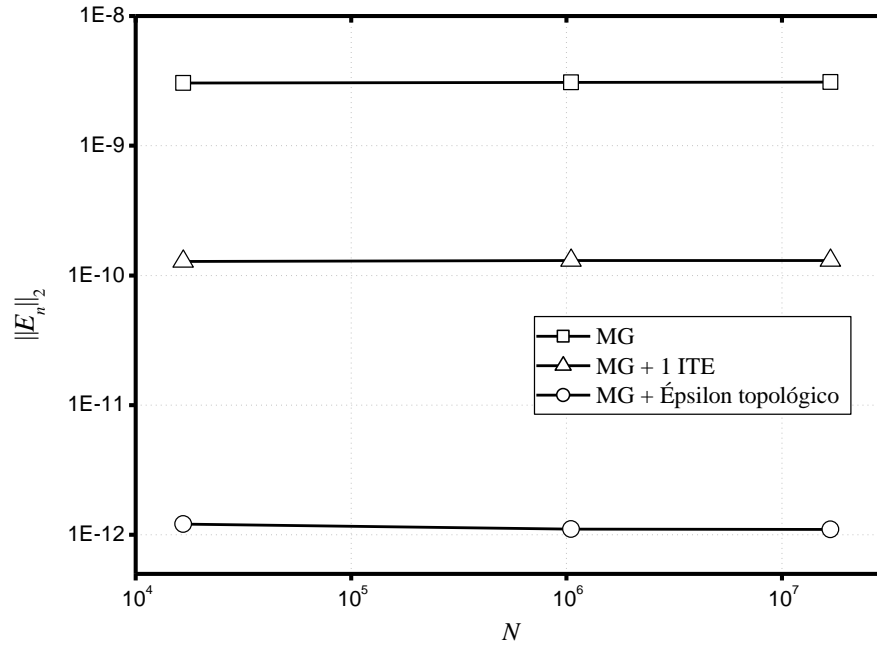


Figura A.5 – Comportamento da $\|E_n\|_2$ em função de N para $\epsilon = 10^{-6}$.

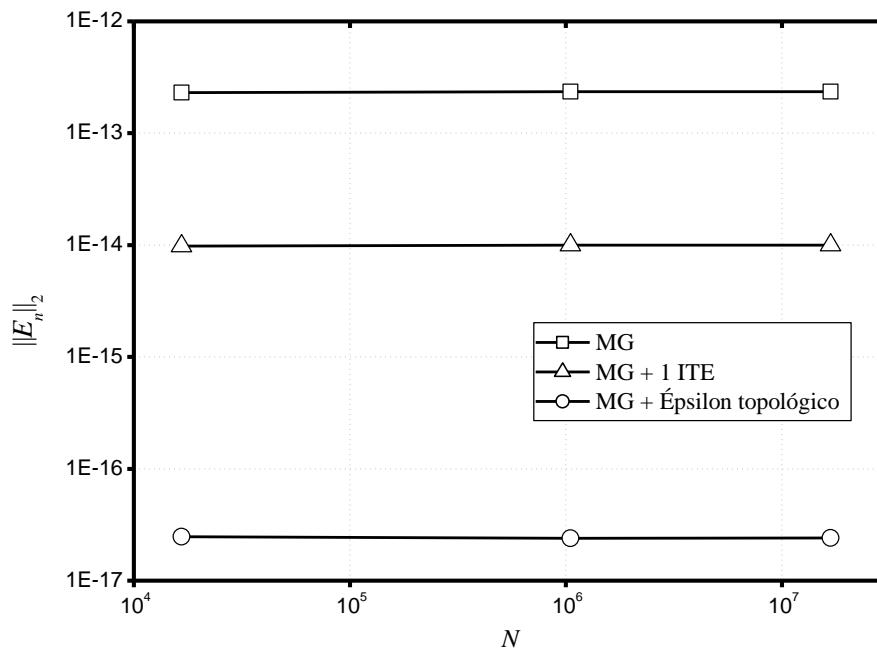


Figura A.6 – Comportamento da $\|E_n\|_2$ em função de N para $\epsilon = 10^{-10}$.

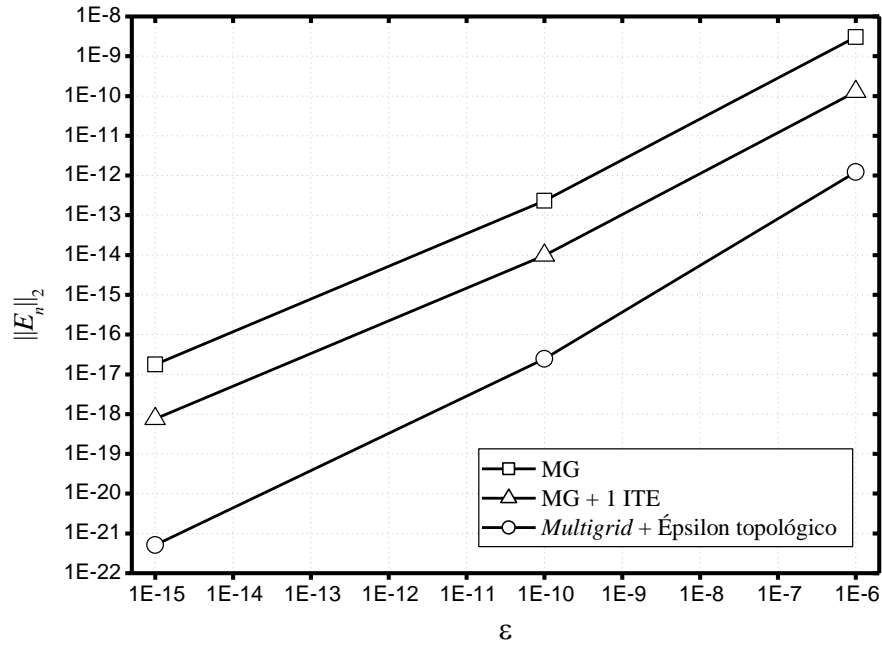


Figura A.7 – Comportamento da $\|E_n\|_2$ em função de ϵ para $N = 129 \times 129$.

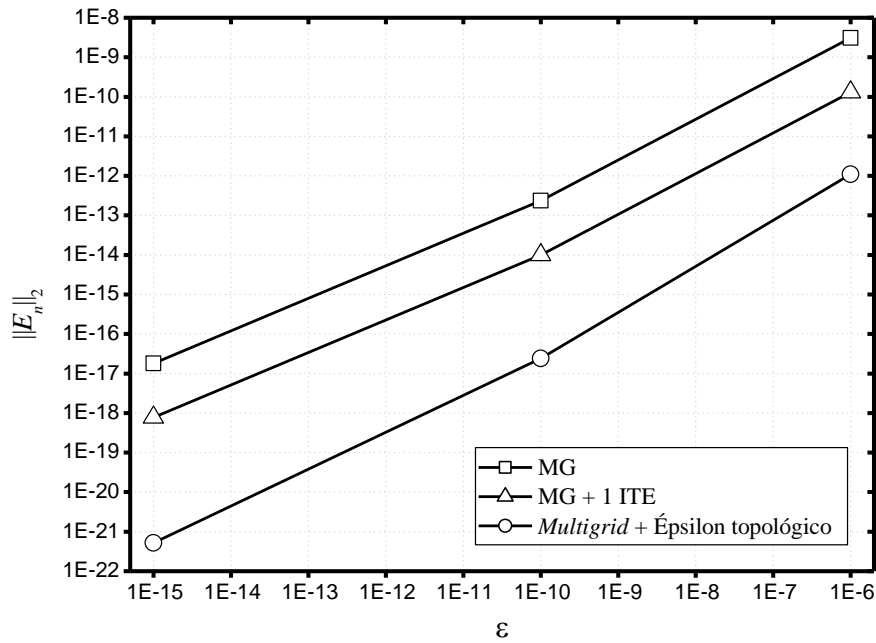


Figura A.8 – Comportamento da $\|E_n\|_2$ em função de ϵ para $N = 1025 \times 1025$.

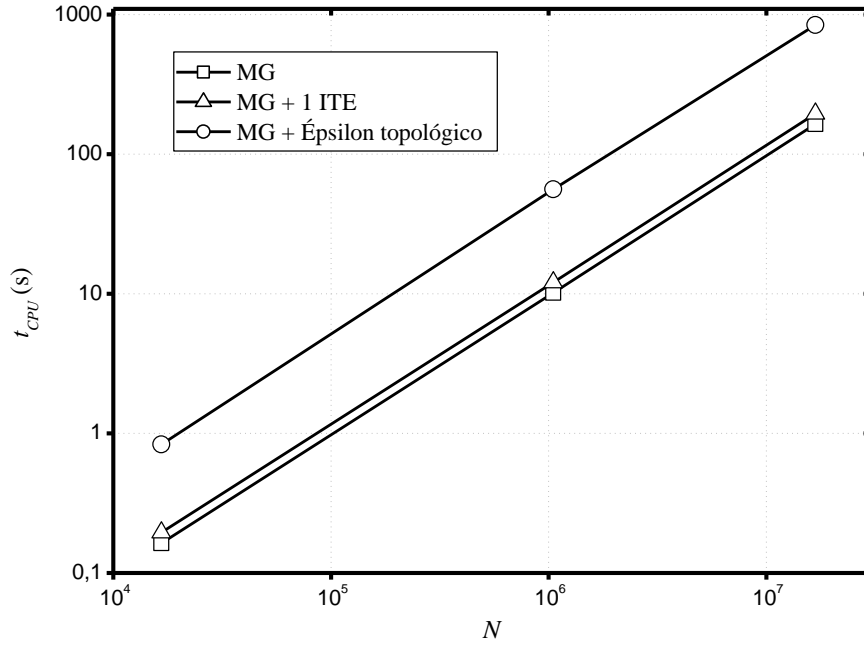


Figura A.9 – Comportamento do t_{CPU} em função de N para $\epsilon = 10^{-6}$.

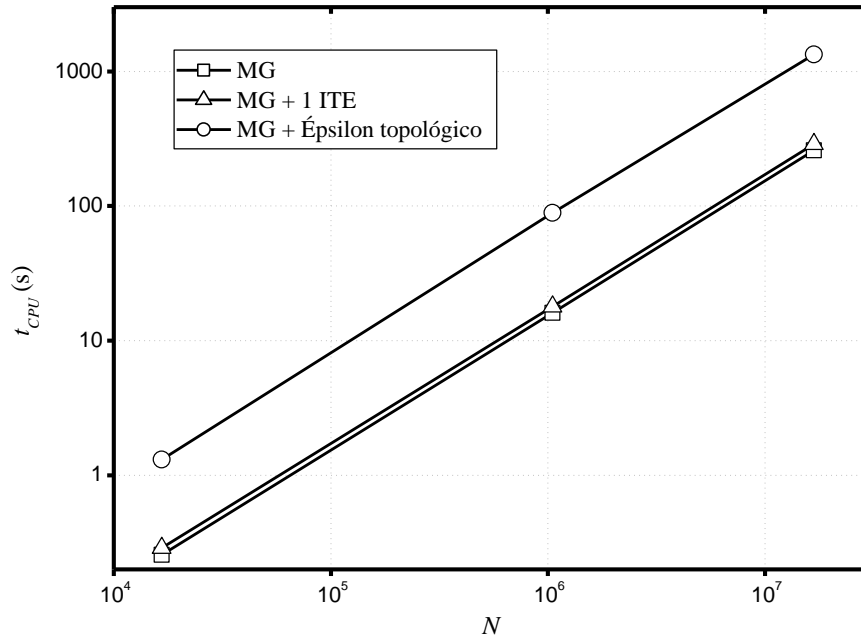


Figura A.10 – Comportamento do t_{CPU} em função de N para $\epsilon = 10^{-10}$.

APÊNDICE B – TESTES ADICIONAIS

Neste apêndice podem ser encontrados os resultados de alguns parâmetros e análises de testes adicionais semelhantes aos que aparecem na seção 4.3.

Nesses testes, primeiramente aplicou-se o método *Multigrid* para obter os resultados para o t_{CPU} , com números de incógnitas variando de $N = 9 \times 9$ até $N = 8193 \times 8193$ e $\varepsilon = 10^{-25}$. Em seguida, aplicou-se o extrapolador Épsilon topológico durante o *Multigrid* na obtenção dos resultados para o t_{CPU} para os mesmos dados.

A Tab. B.1 apresenta os resultados obtidos para o t_{CPU} para as duas metodologias abordadas: *Multigrid* (denotada por MG) e Épsilon topológico durante o *Multigrid* (denotada por MG + Épsilon topológico):

Tabela B.1: Tempo de CPU (s) para MG e MG + Épsilon topológico para $\varepsilon = 10^{-25}$

N	MG	MG + Épsilon topológico
9x9	0,002	0,002
17x17	0,008	0,009
33x33	0,034	0,036
65x65	0,149	0,142
129x129	0,593	0,562
257x257	2,340	2,262
513x513	9,344	8,986
1025x1025	37,284	35,896
2049x2049	148,934	143,537
4097x4097	598,326	577,797
8193x8193	2388,875	2306,907

Na Fig. B.1 é apresentado o comportamento dos dados que aparecem na Tab. B.1. A figura apresenta-se em escala bilogarítmica.

Analisando a Tab. B.1 e a Fig. B.1 verifica-se que os resultados obtidos para o t_{CPU} para as duas metodologias MG e MG + Épsilon topológico foram muito próximos.

A Tab. B.2 apresenta os valores de p e c dos ajustes de curvas (dados pela Eq. (4.3)), obtidos para os algoritmos em estudo.

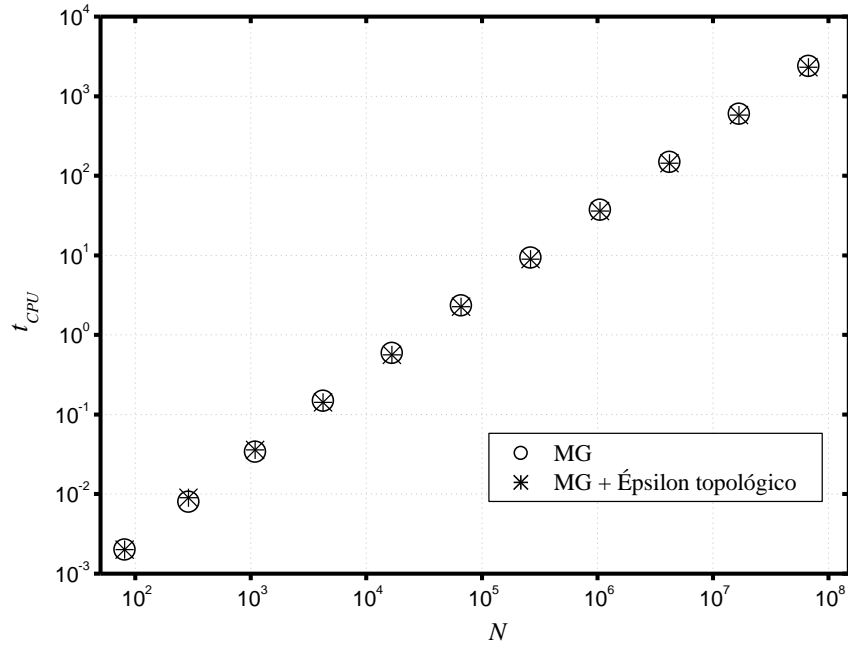


Figura 4.6 – Comportamento do t_{CPU} em função de N para $\varepsilon = 10^{-25}$.

Tabela B.2: Valores de p e c para os algoritmos das metodologias em estudo

Metodologia	c	p
MG	2,618E-05	1,021
MG + Épsilon topológico	2,789E-05	1,014

Na Tab. B.2 observa-se que o valor de p é levemente menor quando é utilizada a metodologia MG + Épsilon topológico, mas essa diferença é praticamente insignificante a ponto de podermos afirmar que as duas metodologias tem praticamente a mesma ordem de complexidade. Portanto, essa metodologia é levemente mais rápida em relação ao uso do *Multigrid* quando o número de incógnitas aumenta.

A Tab. B.3 apresenta o *speed-up* (dado pela Eq. (4.4)) da metodologia MG + Épsilon topológico em relação à metodologia MG para o modelo matemático estudado e $\varepsilon = 10^{-25}$.

Observa-se que o algoritmo que associa o extrapolador Épsilon topológico durante o *Multigrid* mostrou-se mais rápido a partir de $N = 65 \times 65$ (*speed-up* maior que 1), o que permite concluir que associar este extrapolador durante o processo iterativo além de reduzir o erro de iteração para qualquer valor de N , também reduz o tempo de CPU, para $N \geq 65 \times 65$, para o caso estudado.

Tabela B.3: *Speed-up* de MG + Épsilon topológico em relação ao MG para $\varepsilon = 10^{-25}$

N	S_p
9x9	1
17x17	0,888
33x33	0,944
65x65	1,049
129x129	1,055
257x257	1,034
513x513	1,039
1025x1025	1,038
2049x2049	1,037
4097x4097	1,035
8193x8193	1,035