LUIZ ANTONIO CUSTÓDIO MANGANELLI JUNQUEIRA

Estudo de Suavizadores para o Método Multigrid Algébrico baseado em
Wavelet

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para a obtenção do título de
Mestre em Engenharia Elétrica.

Área de concentração:
Sistemas de Potência

Orientador:
Prof. Dr. Silvio Ikuyo Nabeta

São Paulo
2008

Ficha Catalográfica

Junqueira, Luiz Antonio Custódio Manganelli

Estudo de suavizadores para o método multigrid algébrico baseado em wavelet / L.A.C.M. Junqueira. -- São Paulo, 2008. 219 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Energia e Automação Elétricas.

1. Multigrid algébrico 2. Suavizadores 3. Sistemas de equações lineares 4. Método dos elementos finitos I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Energia e Automação Elétricas II. t.

DEDICATÓRIA

Dedico este trabalho aos meus familiares e amigos

“Precisamos sempre rever nossos conceitos”

AGRADECIMENTOS

Agradeço, em especial ao professor Doutor Silvio Ikuyo Nabeta pela oportunidade e incentivo.

Ao Doutor Fábio Henrique Pereira pela sua colaboração.

Ao Doutor César Alberto Bravo Pariente, pela disponibilidade em discutir a apresentação formal do conteúdo desta dissertação.

Aos amigos que me apoiaram.

A minha noiva Cinthia, pelo amor e compreensão.

E, finalmente à banca examinadora pela disponibilidade, participação e interesse.

Resumo

Este trabalho consiste na análise do comportamento do método WAMG (Wavelet-Based Algebraic Multigrid), método numérico de resolução de sistemas de equações lineares desenvolvido no LMAG-Laboratório de Eletromagnetismo Aplicado, com relação a diversos suavizadores. O fato dos vetores que compõem os operadores matriciais Pronlongamento e Restrição do método WAMG serem ortonormais viabiliza uma série de análises teóricas e de dados experimentais, permitindo visualizar características não permitidas nos outros métodos Multigrid (MG), englobando o Multigrid Geométrico (GMG) e o Multigrid Algébrico (AMG). O método WAMG V-Cycle com Filtro Haar é testado em uma variedade de sistemas de equações lineares variando o suavizador, o coeficiente de relaxação nos suavizadores Damped Jacobi e Sobre Relaxação Sucessiva (SOR), e a configuração de pré e pós-suavização. Entre os suavizadores testados, estão os métodos iterativos estacionários Damped Jacobi, SOR, Esparsa Aproximada a Inversa tipo Diagonal (SPAI-0) e métodos propostos com a característica de suavização para-otimizada. A título de comparação, métodos iterativos não estacionários são testados também como suavizadores como Gradientes Conjugados, Gradientes Bi-Conjugados e ICCG. Os resultados dos testes são apresentados e comentados.

Palavras-chave: Multigrid Algébrico, Suavizadores, Sistemas de Equações Lineares, Método dos Elementos Finitos.

Abstract

This work is comprised of WAMG (Wavelet-Based Algebraic Multigrid) method behavioral analysis based on variety of smoothers, numerical method based on linear equation systems resolution developed at LMAG (Applied Electromagnetism Laboratory). Based on the fact that the vectors represented by WAMG Prolongation and Restriction matrix operators are orthonormals allows the use of a variety of theoretical and practical analysis, and therefore gain visibility of characteristics not feasible through others Multigrid (MG) methods, such as Geometric Multigrid (GMG) and Algebraic Multigrid (AMG). WAMG V-Cycle method with Haar Filter is tested under a variety of linear equation systems, by varying smoothers, relaxation coefficient at Damped Jacobi and Successive Over Relaxation (SOR) smoothers, and pre and post smoothers configurations. The tested smoothers are stationary iterative methods such as Damped Jacobi, SOR, Diagonal type-Sparse Approximate Inverse (SPAI-0) and suggested ones with optimized smoothing characteristic. For comparison purposes, the Conjugate Gradients, Bi-Conjugate Gradient and ICCG non-stationary iterative methods are also tested as smoothers. The testing results are formally presented and commented.

Key-Words: **Algebraic** Multigrid, Smoothers, Linear Equations System, Finite Elements Method.

Sumário

Capítulo 1	- 1 -
INTRODUÇÃO	- 1 -
1.1. Motivação	- 1 -
1.2. Estado da Arte	- 1 -
1.3. Contribuição	- 2 -
1.4. Organização do texto	- 3 -
Capítulo 2	- 4 -
MATRIZES E SISTEMAS DE EQUAÇÕES LINEARES	- 4 -
2.1 Matriz	- 4 -
2.1.1. Transformação entre Espaços de Dimensões Diferentes	- 4 -
2.1.2. Norma Frobenius e Para-Otimização	- 5 -
2.2. Sistemas de Equações	- 5 -
2.2.1. Complexidade na Resolução de Sistemas de Equações Lineares	- 7 -
2.2.2. Características de Sistemas Lineares oriundos do MEF	- 8 -
Capítulo 3	- 10 -
MÉTODOS DE RESOLUÇÃO	- 10 -
3.1. Métodos diretos	- 10 -
3.2. Métodos iterativos	- 11 -
3.2.1. Métodos Iterativos Estacionários (MIE)	- 11 -
3.2.2. Análise de Convergência	- 15 -
3.2.3. Análise dos Componentes de Frequência de Erro e Resíduo	- 17 -
3.2.4. Métodos em Destaque	- 19 -
3.3. Conclusão	- 42 -
Capítulo 4	- 44 -
TÉCNICA MULTIGRID	- 44 -
4.1 Notações de Grid Refinado e Grid Grosso	- 44 -
4.1. Operador Restrição R	- 45 -
4.2. Operador Prolongamento P	- 45 -
4.3. Obtenção da Matriz A^H	- 45 -
4.4. Correção em Grid Grosso	- 46 -
4.5. Princípio do Método Multigrid	- 46 -
4.6. Algumas Variações do Método MG	- 48 -
4.6.1. Multigrid Geométrico (GMG)	- 48 -
4.6.2. Multigrid Algébrico (AMG)	- 49 -
4.6.3. Wavelet based Algebraic Multigrid (WAMG)	- 49 -
4.7. Conclusões	- 50 -
Capítulo 5	- 51 -
METODOLOGIA DE ANÁLISE E PROPOSIÇÃO DE SUAVIZADORES	- 51 -
5.1. Análise de Subespaços do Método Multigrid	- 51 -
5.1.1. Matrizes associadas à correção em grid grosso com solução exata	- 51 -
5.1.2. Subespaços associados aos Operadores P e R	- 52 -
5.1.3. Análise de casos	- 55 -
5.1.4. Base de Transformação Ortonormal	- 63 -
5.1.5. Operadores P e R ideais	- 63 -

5.1.6. Operadores P e R ótimos	- 65 -
5.1.7. Sugestão de Obtenção de Operadores P e R Para-Ótimos	- 65 -
5.2. Análise da Característica de Suavização dos MIEs	- 66 -
5.3. Suavizadores Propostos	- 70 -
5.3.1. Características para-otimizadas	- 70 -
5.3.2. Combinando os critérios	- 73 -
5.3.3. WAMG com Filtro Haar em Sistemas de Equações Simétricos	- 73 -
5.3.4. Suavizadores com Características Para-Otimizadas	- 75 -
5.4. Indicadores de Auxílio à Decisão	- 78 -
5.4.1. Dominância da Diagonal	- 78 -
5.4.2. Densidade da Matriz	- 80 -
5.4.3. Largura de Banda	- 80 -
5.4.4. Projeções da Matriz A	- 80 -
5.4.5. Limite de Convergência do Erro na Correção em Grid Grosso	- 81 -
5.5. Conclusões	- 81 -
Capítulo 6	- 82 -
APLICAÇÕES E RESULTADOS	- 82 -
6.1. Matrizes testadas	- 84 -
6.1.1. Matrizes Poisson961, Poisson3721 e Poisson10201	- 84 -
6.1.2. Matrizes transformada Poisson961_T	- 85 -
6.1.3. Matrizes nos7 e bcsstk14	- 86 -
6.1.4. Matrizes Eletroima e MotorRelutancia	- 87 -
6.1.5. Matriz MotorRelutanciaBand	- 87 -
6.2. Resultados Obtidos	- 89 -
6.2.1. Indicadores das Matrizes	- 89 -
6.2.2. Distribuição de elementos não nulos	- 99 -
6.2.3. Comparação entre os Métodos Iterativos	- 101 -
6.2.4. Ranque das Configurações do Método WAMG	- 114 -
6.2.5. Componentes de Frequência	- 123 -
6.2.6. Filtro Haar x Vizinhança	- 143 -
6.2.7. Influência da Configuração de Pré e Pós-suavização	- 150 -
6.2.8. Coeficiente de Relaxação do Damped Jacobi	- 169 -
6.2.9. Coeficiente de Relaxação no Método SOR	- 182 -
6.3. Análise dos Resultados	- 195 -
6.3.1. Indicadores das Matrizes	- 196 -
6.3.2. Distribuição de elementos não nulos	- 196 -
6.3.3. Comparação entre os Métodos Iterativos	- 196 -
6.3.4. Ranque das Configurações do Método WAMG	- 197 -
6.3.5. Componentes de Frequência	- 198 -
6.3.6. Filtro Haar x Vizinhança	- 198 -
6.3.7. Influência da Configuração de Pré e Pós-suavização	- 199 -
6.3.8. Coeficiente de Relaxação do Método Damped Jacobi	- 199 -
6.3.9. Coeficiente de Relaxação nos Métodos SOR	- 200 -
CONCLUSÕES	- 201 -
SUGESTÃO PARA TRABALHOS FUTUROS	- 203 -
REFERÊNCIAS BIBLIOGRÁFICAS	- 204 -

Lista de Símbolos

A	Matriz dos coeficientes das incógnitas do sistema de equações
N	Ordem da matriz
x	Vetor das incógnitas do sistema de equações
$\bar{x}$	Vetor solução exata
$\tilde{x}$	Vetor solução aproximada
Δx	Vetor incremento da solução
e	Vetor erro
b	Vetor dos termos independentes do sistema de equações
r	Vetor resíduo
M	Matriz iteração
S	Matriz iteração do erro
Z	Matriz iteração do resíduo
A^T	Matriz transposta
x^T	Vetor transposto
A^*	Matriz complexo conjugado
x^*	Vetor complexo conjugado
A^{T*}	Matriz transposta complexo conjugado ou hermitiana da matriz
x^{T*}	Vetor transposto complexo conjugado ou hermitiana do vetor
A^h	Matriz A pertencente ao grid h
x^h	Vetor x pertencente ao grid h
a_{ij}	Elemento da linha i e coluna j da matriz A
x_i	i-ésimo elemento do vetor x
x_k	Vetor x na iteração k
v_i^A	i-ésimo autovetor da matriz A
λ_i^A	i-ésimo autovalor da matriz A
$\rho(A)$	Raio espectral da matriz A
P	Matriz Operador Prolongamento

$\bar{P}$ Matriz Complemento de P

$$P^{T*} \bar{P} = [0]$$

$$P(P^{T*}P)^{-1}P^{T*} + \bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*} = I$$

R Matriz Operador Restrição

$\bar{R}$ Matriz Complemento de R

$$\bar{R}R^{T*} = [0]$$

$$R^{T*}(RR^{T*})^{-1}R + \bar{R}^{T*}(\bar{R}\bar{R}^{T*})^{-1}\bar{R} = I$$

$\Omega(P)$ Subespaço obtido pela combinação linear dos vetores colunas da matriz P

$\Omega(\bar{P})$ Subespaço ortogonal ao subespaço $\Omega(P)$

$\Omega(R)$ Subespaço obtido pela combinação linear dos vetores colunas da matriz R

$\Omega(\bar{R})$ Subespaço ortogonal ao subespaço $\Omega(R)$

$|a|$ Módulo do escalar a.

$\|v\|$ Norma do vetor v.

$$\|A\|_F = \sqrt{\sum_{i=1}^N \sum_{j=1}^N |a_{ij}|^2} \quad \text{Norma Frobenius da matriz A}$$

∇ Operador Nabla

$\partial\omega$ Derivada de ω

Capítulo 1

INTRODUÇÃO

Com a evolução tecnológica, surge a necessidade de se resolver sistemas de equações com ordem cada vez maior e mal-condicionados, exigindo algoritmos de resolução de sistemas de equações que se comportem melhor para estes sistemas de equações. O uso de computadores baseados em arquitetura paralela (multiprocessador, cluster e grid) oferece maior velocidade de processamento, desde que o método de resolução permita paralelismo.

O método Multigrid tem se mostrado robusto na resolução de sistemas de equações esparsos (com matriz de coeficientes dependentes esparsa), de ordem elevada, além de permitir o paralelismo.

1.1. Motivação

O bom resultado do método Multigrid depende de suas configurações como: suavizador, coeficiente de relaxação do suavizador (caso existir) e a configuração de pré e pós-suavização.

Este trabalho foi motivado pelo objetivo de obter critérios que auxiliassem na escolha da melhor configuração do método WAMG.

1.2. Estado da Arte

Conforme descrito em [1], o método Multigrid (MG); foi proposto originalmente por Fedorenko em 1964 [3] e é atualmente um método numérico iterativo muito utilizado na resolução de sistemas de equações oriundos do método de Elementos Finitos (MEF). Brandt, Frederickson, Hackbush e Wesseling tiveram importante contribuição na sua divulgação [4] [5]. Em 1982, Brandt [12] propôs o método Multigrid Algébrico, do inglês ***Algebraic Multigrid*** (AMG), que passou a dispensar a necessidade de se obter informações do malhador (rotina do MEF que discretiza o domínio em estudo).

Em 2006, Pereira propôs o método Multigrid Algébrico baseado em Wavelet, do inglês **Wavelet based Algebraic Multigrid** (WAMG) [1] [2], que oferece vantagens na obtenção algébrica dos grids grosseiros.

Em 2003, Watanabe, Igarashi e Honma [24], o método iterativo não estacionário Gradientes Conjugados (CG) é sugerido como suavizador do método Multigrid para casos em que os elementos de discretização são disformes. Este artigo serviu como sugestão para o teste dos métodos iterativos não estacionários Gradientes Conjugados, Gradientes Bi-Conjugados (BiCGSTAB) e Gradientes Conjugados preconditionado pela Fatoração Incompleta de Cholesky (ICCG) como suavizadores do método WAMG.

São encontrados trabalhos que visam obter operadores de transformação de grids otimizados para o suavizador, como é o caso da técnica de **Smoothed Aggregation**, apresentada desde 1996 por Vanek, Mandel e Brezina [17] [18] [19].

Entre os suavizadores mais utilizados no método Multigrid são encontrados os métodos iterativos estacionários Jacobi, Damped Jacobi, Gauss-Seidel, SOR e SPAI.

Em 2001, Broecker, Grote, Mayer e Reusken [15] [16] apresentam o método iterativo estacionário SPAI(ϵ) como robusto suavizador do método MG que permite paralelismo eficiente. O fato de este método ser baseado em otimização inspirou na proposição e testes de suavizadores otimizados para o método WAMG neste trabalho.

1.3. Contribuição

Este trabalho complementa o trabalho que originou o método WAMG, através de uma amostragem rica em testes experimentais. São exploradas as particularidades dos wavelets, como viabilidade na decomposição de erro e resíduo em componentes de alta e baixa frequência e maior viabilidade de otimizações.

A análise proposta da característica de suavização dos métodos iterativos estacionários serve como motivação para a procura de novas estratégias de obtenção dos operadores Prolongamento e Restrição.

O modelo matemático proposto do método MG oferece uma compreensão intuitiva do método de forma abrangente e serve como ferramenta auxiliar no desenvolvimento de novas frentes a serem exploradas. Com base neste modelo, são sugeridas estratégias de otimização de suavizadores para o método WAMG com filtro Haar e otimização de Operadores Prolongamento e Restrição.

1.4. Organização do texto

No Capítulo 2, são apresentados os conceitos básicos e as notações adotadas para sistemas de equações lineares, classe de complexidade e as particularidades dos sistemas de equações lineares oriundos do MEF.

No Capítulo 3, são abordados os métodos diretos, os métodos iterativos estacionários e os métodos iterativos não estacionários. Os métodos iterativos estacionários (suavizadores) são apresentados de forma detalhada.

No Capítulo 4, é apresentada a técnica Multigrid.

No Capítulo 5, contém: a apresentação de um modelo matemático geral do método MG, baseado nos subespaços correlacionados aos operadores P e R ; a sugestão de uma estratégia de obtenção dos operadores P e R ; algumas propostas de suavizadores com características otimizadas para serem testados no método WAMG com Filtro Haar; uma análise da garantia de presença da característica de suavização dos métodos iterativos estacionários; e indicadores propostos para análise dos dados experimentais.

No Capítulo 6, são apresentados os resultados obtidos experimentalmente e a análise destes resultados.

Capítulo 2

MATRIZES E SISTEMAS DE EQUAÇÕES LINEARES

Neste capítulo, são apresentadas as notações adotadas e os conceitos básicos de matrizes e sistemas de equações lineares correlacionados ao

2.1 Matrizes

Uma matriz pode representar os coeficientes dependentes de um sistema de equações, pode representar uma função linear que permite determinar o vetor imagem a partir de um vetor domínio e pode transformar um vetor de um domínio para outro.

2.1.1. Transformação entre Espaços de Dimensões Diferentes

Para transformar um vetor de espaço de dimensão maior N em um vetor de espaço de dimensão menor n , utiliza-se uma matriz $n \times N$:

$$[v]_n = [R]_{nN} [V]_N$$

Onde $[V]_N$ é o vetor do espaço de dimensão maior, $[v]_n$ é o vetor do espaço de dimensão menor e $[R]_{nN}$ é a matriz que transforma o vetor do espaço de dimensão maior no vetor do espaço de dimensão menor.

Neste caso, apenas os componentes do vetor de ordem maior contidos no subespaço definido pela combinação linear dos vetores linha complexo conjugado da matriz R são “transferidos para o” (transformados em componentes não nulos no) espaço de dimensão menor.

Para transformar um vetor do subespaço de ordem menor n em um vetor do subespaço de ordem maior N , utiliza-se uma matriz $N \times n$:

$$[V]_N = [P]_{Nn} [v]_n$$

Onde $[P]_{Nn}$ é a matriz que transforma o vetor do espaço de dimensão menor no vetor do espaço de dimensão maior.

Neste caso, o vetor do subespaço de dimensão maior está contido no subespaço definido pela combinação linear dos vetores coluna da matriz P .

2.1.2. Norma Frobenius e Para-Otimização

A norma Frobenius de uma matriz A é determinada por:

$$\|A\|_F = \sqrt{\sum_{i=1}^N \sum_{j=1}^M |a_{ij}|^2} \quad (1)$$

É comum encontrar situações nas quais, uma condição ideal ou ausência de uma característica indesejável implica em uma matriz nula.

Nesta situação, a norma Frobenius desta matriz pode servir como indicador da presença desta característica indesejável.

Se esta matriz for obtida a partir de uma expressão matricial parametrizada, a minimização da norma de Frobenius desta matriz pode servir como critério de tentativa de otimização, ou para-otimização, obtendo valores que tendem ao ótimo para estes parâmetros. Para se minimizar a norma Frobenius de uma matriz, utiliza-se o método dos mínimos quadrados.

2.2. Sistemas de Equações

Um sistema de equações lineares de M equações e N incógnitas pode ser formulado matricialmente em:

$$\begin{aligned} [A]_{MN} [x]_N &= [b]_M \\ Ax &= b \end{aligned} \quad (2)$$

Os elementos da matriz A correspondem aos coeficientes das incógnitas, os elementos do vetor x correspondem às incógnitas e os elementos do vetor b correspondem aos termos independentes.

Cada linha i de A e b representa a equação i do sistema e cada coluna j de A e linha j de x representa a contribuição da incógnita j no equacionamento.

$$\sum_{j=1}^N a_{ij}x_j = b_i, i=1..M$$

O erro e é definido pela diferença entre a solução exata $\bar{x}$ e uma solução aproximada $\tilde{x}$.

$$\begin{aligned} [e]_N &= [\bar{x}]_N - [\tilde{x}]_N \\ e &= \bar{x} - \tilde{x} \end{aligned} \quad (3)$$

O resíduo r corresponde ao produto da matriz A pelo erro e :

$$[r]_M = [A]_{MN} [e]_N$$

Substituindo $e = \bar{x} - \tilde{x}$ (3), teremos:

$$\begin{aligned} [r]_M &= [A]_{MN} [\bar{x} - \tilde{x}]_N \\ [r]_M &= [A]_{MN} [\bar{x}]_N - [A]_{MN} [\tilde{x}]_N \end{aligned}$$

Substituindo $Ax = b$ (2), teremos:

$$\begin{aligned} [r]_M &= [b]_M - [A]_{MN} [\tilde{x}]_N \\ r &= Ae = A(\bar{x} - \tilde{x}) = b - A\tilde{x} \end{aligned} \quad (4)$$

Um sistema de equações lineares pode ter uma única solução exata, infinitas soluções exatas ou nenhuma solução exata. Se A for uma matriz quadrada (o número de equações M for igual ao número de incógnitas N) e não existir linhas que sejam combinação linear das demais, dizemos que a matriz A não é singular, garantindo a existência de uma única solução exata. Se a matriz A for Singular, dependendo dos valores dos coeficientes independentes, o sistema poderá não ter nenhuma solução exata ou infinitas soluções exatas.

Se a quantidade de elementos não nulos por linha for elevada, a matriz é denominada densa, caso contrário, a matriz é denominada esparsa. Vamos denominar por sistema de equações denso o sistema de equações representado por uma matriz densa e por sistema de equações esparsa o sistema de equações representado por uma matriz esparsa.

2.2.1. Complexidade na Resolução de Sistemas de Equações Lineares

A classe de complexidade de um algoritmo consiste na formulação simplificada que sinaliza o recurso (tempo ou memória) necessário para processar este algoritmo em função do tamanho da entrada de dados do problema.

Quando o recurso for o tempo, relacionado com a quantidade de operações necessárias para executar o algoritmo, será utilizada a terminologia complexidade de tempo do algoritmo, ou simplesmente, complexidade do algoritmo. Quando o recurso for quantidade de memória necessária para executar o algoritmo, será utilizada a terminologia complexidade de espaço.

A formulação simplificada pode ser obtida a partir da formulação exata, ocultando coeficientes e desprezando as parcelas menos significativas para o tamanho do problema tendendo ao tamanho máximo especificado para o algoritmo, que pode ser infinito, podendo também existir uma complexidade para cada faixa de tamanho de entrada de problema.

Na resolução de sistemas lineares, o tamanho do problema pode ser expresso pela ordem do sistema N , pela quantidade de elementos não nulos da matriz dos coeficientes dependentes Z ou por duas variáveis como ordem do sistema N e quantidade média de elementos não nulos por linha m .

Para o melhor caso do algoritmo, a complexidade é denotada pelo limite inferior $\Omega()$, para o pior caso, a complexidade é denotada pelo limite superior $O()$ e para o caso médio, é denotada por $\theta()$. A complexidade é denominada fechada quando seu limite superior coincidir com o seu limite inferior e é representada pelo limite superior $O()$.

Como exemplo, vamos supor um algoritmo de resolução de sistemas lineares que utiliza uma quantidade de instruções numéricas proporcional a Nm , a quantidade de iterações é proporcional a $\log(N)$ para sistemas bem condicionados e N para sistemas mal condicionados e a quantidade de memória necessária para executar o algoritmo é Nm , independentemente se o sistema é ou não bem condicionado. A complexidade inferior de tempo é $\Omega(Nm \log(N))$, a complexidade superior é $O(N^2m)$ e a complexidade de espaço fechada é $O(Nm)$.

2.2.2. Características de Sistemas Lineares oriundos do MEF

O Método de Elementos Finitos (MEF) permite resolver equações diferenciais parciais (EDP), discretizando o domínio do problema. O domínio é dividido em elementos finitos, através de uma malha de discretização. Esta malha corresponde a um grafo por possuir nós e arestas.

Uma equação diferencial pode ser classificada em:

Classe	Equação Diferencial	Tipo de Problema
Elíptica	$\nabla^2 x = 0$	Equação de Laplace (estado estacionário)
	$\nabla^2 x = f(p)$	Equação de Poisson
	Onde p é a posição.	
Hiperbólica	$\nabla^2 x = \frac{1}{v^2} \frac{\partial^2 x}{\partial t^2}$	Equação de Onda
Parabólica	$k \nabla^2 x = \frac{\partial x}{\partial t}$	Equação de Condução de Calor

Em um sistema de equações diferenciais parciais, na qual se deseja determinar os valores nodais de uma função, uma incógnita corresponde ao valor desta função na posição do nó correspondente.

No MEF, a função que se deseja determinar é interpolada por funções de forma. Cada função de forma é associada a um nó da malha, sendo que o valor da função de forma vale 1 para seu respectivo nó e 0 para os demais nós. No sistema de equações gerado pelo MEF, cada incógnita corresponde ao valor da função a ser determinada no respectivo nó e corresponde também ao coeficiente que multiplica a respectiva função de forma. Por conveniência, estas funções de forma são definidas localmente, gerando equações que relacionam apenas os nós vizinhos (conectados), resultando em sistemas de equações esparsos. Portanto, se os nós i e j forem vizinhos, ou seja, se existir uma conexão (aresta) c_{ij} entre os nós i e j , o elemento a_{ij} da matriz A será não nulo.

A partir da matriz A , é possível reproduzir um grafo correspondente, sem os atributos de coordenadas de nós e de dimensões das arestas, impondo a existência de um nó i para cada incógnita i e a existência de uma conexão c_{ij} para cada elemento a_{ij} não nulo.

No sistema de equações $Ax=b$, obtido pelo MEF, a matriz A é conhecida como matriz de Rigidez e o vetor b é composto pelo vetor das Condições de Contorno e pelo vetor Fonte. Os elementos da matriz de Rigidez A dependem das dimensões dos elementos discretizados e das propriedades do meio.

Capítulo 3

MÉTODOS DE RESOLUÇÃO

Os métodos de resolução de sistemas lineares podem ser classificados em métodos diretos e em métodos iterativos. Um método é dito direto se a solução for obtida em um número conhecido de operações e um método é dito iterativo, quando a cada iteração do método se obtém uma melhor aproximação da solução exata, até se obter uma solução que satisfaça a condição de mínimo resíduo.

Para se utilizar um método direto, é condição sine qua non que a matriz A não seja singular. O método iterativo permite a resolução de sistemas singulares. No caso em que não exista solução exata, a solução aproximada tenderá a solução que melhor satisfaça o sistema, segundo o critério de aproximação intrínseco ao método e, no caso em que existam várias soluções exatas, a solução aproximada poderá tender a uma das soluções exatas. Esta maior abrangência do método iterativo, de permitir resolver também sistemas singulares, caracteriza uma vantagem sobre o método direto.

3.1. Métodos diretos

A grande parte dos métodos diretos tem suas complexidades de tempo e espaço fechadas, sendo a complexidade de tempo $O(N^3)$ e complexidade de espaço $O(N^2)$, que correspondem a tempos de processamento e exigência de memória indesejáveis para sistemas de equações de ordem elevada.

Em particular, a fatoração completa LU tende a tirar proveito da esparsidade do sistema de equações, tendo complexidade de tempo inferior $\Omega(Nm^2)$ e complexidade de espaço inferior $\Omega(Nm)$, sendo eficiente para matrizes com banda estreita, podendo ser a melhor alternativa na resolução de alguns sistemas de equações esparsos não singulares, quando não se dispõe de processamento paralelo. No pior caso, tem complexidade de tempo superior $O(Nl_{BW}^2)$ e

complexidade de espaço superior $O(NI_{BW})$, onde I_{BW} é o indicador da largura de banda da matriz A .

3.2. Métodos iterativos

Em um método iterativo de resolução de sistemas de equações lineares, a cada iteração k , é determinada uma nova solução aproximada x_k até que o módulo do resíduo seja menor que um valor ϵ imposto.

A complexidade de tempo fechada no cálculo do resíduo é $O(Nm)$, e a complexidade de cada iteração para qualquer método iterativo não pode ser inferior à complexidade do cálculo do resíduo.

3.2.1. Métodos Iterativos Estacionários (MIE)

Uma propriedade dos métodos iterativos estacionários é de existirem uma matriz que associa o incremento da solução ao resíduo, uma matriz que relaciona o erro da iteração $k+1$ ao erro da iteração k e uma matriz que relaciona o resíduo da iteração $k+1$ ao resíduo da iteração k .

3.2.1.1 Matrizes Associadas ao MIE

Para todo o método iterativo estacionário é possível extrair a matriz iteração M , a matriz aproximação $\tilde{A}$, a iteração de erro S , a matriz iteração de resíduo Z e a matriz dos elementos desconsiderados Δ .

3.2.1.1.1 Matriz Iteração M

Vamos definir a matriz iteração M como sendo a matriz que relaciona o incremento da solução ao resíduo:

$$\Delta x_{k+1} = M r_k \quad (5)$$

$$\Delta x_{k+1} = x_{k+1} - x_k \quad (6)$$

Quanto mais a matriz M for aproximada da inversa da matriz A , melhor será a taxa de convergência por iteração.

$$M \approx A^{-1}$$

Podemos verificar que se aplica a propriedade distributiva:

$$M(r_{k1} + r_{k2}) = M r_{k1} + M r_{k2}$$

3.2.1.1.2 Matriz Aproximada $\tilde{A}$

Se a matriz M se aproxima da inversa da matriz A , a inversa da matriz M se aproxima de A .

$$\tilde{A} \approx M^{-1} \quad (7)$$

Para alguns métodos iterativos é possível obter a matriz $\tilde{A}$, aqui denominada de matriz aproximada ou matriz incompleta. A comparação entre matriz $\tilde{A}$ e a matriz A permite avaliar características de convergência do método, como será apresentado para os métodos Jacobi e Gauss-Seidel nas sessões [3.2.4.1] e [3.2.4.3].

3.2.1.1.3 Matriz Iteração de erro S

Vamos definir como matriz de iteração de erro S a matriz que relaciona o erro da iteração $k+1$ ao erro da iteração k :

$$e_{k+1} = S e_k$$

Substituindo $e = \bar{x} - \tilde{x}$ (3), teremos:

$$\bar{x}_{k+1} - x_{k+1} = S(\bar{x}_k - x_k)$$

$$-x_{k+1} = S(\bar{x}_k - x_k) - \bar{x}_k$$

$$x_{k+1} = S(x_k - \bar{x}_k) + \bar{x}_k$$

$$x_{k+1} - x_k = S(x_k - \bar{x}_k) + \bar{x}_k - x_k$$

Substituindo $\Delta x = x_{k+1} - x_k$ (6), teremos:

$$M r_k = S(x_k - \bar{x}_k) + \bar{x}_k - x_k$$

Substituindo $r = A(\bar{x} - \tilde{x})$ (4), teremos:

$$MA(x_k - \bar{x}_k) = S(x_k - \bar{x}_k) + \bar{x}_k - x_k$$

$$MA(x_k - \bar{x}_k) = (S - I)(x_k - \bar{x}_k)$$

$$MA = S - I$$

$$S = I - MA \quad (8)$$

Quanto mais a matriz S for aproximada da matriz nula, melhor será a convergência do método, portanto, a norma Frobenius da matriz S pode servir como indicador de convergência:

$$\|S\|_F = \sqrt{\sum_i^N \sum_j^N |s_{ij}|^2}$$

Podemos verificar que se aplica a propriedade distributiva:

$$S(e_{k1} + e_{k2}) = S e_{k1} + S e_{k2}$$

É possível relacionar um erro com o respectivo erro após K iterações:

$$e_{k+K} = S^K e_k$$

Para esta relação, podemos definir uma matriz iteração de erro equivalente:

$$S_{eq} = S^K$$

3.2.1.1.4 Matriz iteração de resíduo Z

Vamos definir como matriz de iteração de resíduo Z a matriz que relaciona o resíduo da iteração $k+1$ ao resíduo da iteração k :

$$r_{k+1} = Z r_k \quad (9)$$

Substituindo $r = A(\bar{x} - \tilde{x})$ (4), teremos:

$$A(\bar{x} - x_{k+1}) = ZA(\bar{x} - x_k)$$

$$A(\bar{x} - x_{k+1} + x_k - x_k) = ZA(\bar{x} - x_k)$$

Substituindo $\Delta x = x_{k+1} - x_k$ (6), teremos:

$$A(\bar{x} - \Delta x - x_k) = ZA(\bar{x} - x_k)$$

Substituindo $\Delta x = M r_k$ (5)

$$A(\bar{x} - M r_k - x_k) = ZA(\bar{x} - x_k)$$

$$A(\bar{x} - x_k) - AM r_k = ZA(\bar{x} - x_k)$$

Substituindo $r = A(\bar{x} - \tilde{x})$ (4), teremos:

$$\begin{aligned} A(\bar{x} - x_k) - AMA(\bar{x} - x_k) &= ZA(\bar{x} - x_k) \\ A - AMA &= ZA \\ Z &= I - AM \end{aligned} \quad (10)$$

Da mesma forma, quanto mais a matriz Z for aproximada à matriz nula, melhor será a taxa de convergência por iteração, portanto, a norma Frobenius da matriz Z pode servir como indicador de convergência:

$$\|Z\|_F = \sqrt{\sum_i^N \sum_j^N |z_{ij}|^2}$$

Podemos verificar que se aplica a propriedade distributiva:

$$Z(r_{k1} + r_{k2}) = Zr_{k1} + Zr_{k2}$$

É possível relacionar um resíduo com o respectivo resíduo após K iterações:

$$r_{k+K} = Z^K r_k$$

Para esta relação, podemos definir uma matriz iteração de resíduo equivalente:

$$Z_{eq} = Z^K$$

3.2.1.1.5 Matriz dos Elementos Desconsiderados Δ

A matriz dos elementos desconsiderados Δ consiste na diferença da matriz A pela matriz aproximação $\tilde{A}$:

$$\Delta = A - \tilde{A} \quad (11)$$

$$A = \Delta + \tilde{A} \quad (12)$$

Substituindo $A = \Delta + \tilde{A}$ (12) em $S = I - MA$ (8), teremos:

$$S = I - M(\Delta + \tilde{A})$$

Substituindo $\tilde{A} \approx M^{-1}$ (7), teremos:

$$S = I - M(\Delta + M^{-1})$$

$$\begin{aligned} S &= I - M\Delta - I \\ S &= -M\Delta \end{aligned} \quad (13)$$

Substituindo $A = \Delta + \tilde{A}$ (12) em $Z = I - AM$ (14), teremos:

$$Z = I - (\Delta + \tilde{A})M$$

Substituindo $\tilde{A} \approx M^{-1}$ (7), teremos:

$$\begin{aligned} Z &= I - (\Delta + M^{-1})M \\ Z &= I - I - \Delta M \\ Z &= -\Delta M \end{aligned} \quad (14)$$

3.2.2. Análise de Convergência

Na resolução de sistema de equações lineares, utilizando um método iterativo estacionário como solver, podemos verificar que nas primeiras iterações, a relaxação do erro é maior, sendo que a taxa de convergência tende a piorar com o avanço das iterações. Este comportamento é justificado pela análise dos autovetores de S . O mesmo comportamento ocorre com o resíduo, sendo justificado pela análise dos autovetores de Z .

3.2.2.1 Análise dos Autovetores de S

O erro inicial pode ser decomposto em autovetores da matriz S :

$$e_0 = \sum_{i=1}^N c_i^S v_i^S$$

Onde c_i^S é um escalar e v_i^S é o i -ésimo autovetor da matriz S .

Se multiplicarmos sucessivamente o erro pela matriz S , equivale a multiplicar sucessivamente cada componente de autovetor v_i^S pelo seu respectivo autovalor λ_i^S :

$$\begin{aligned} e_1 &= S e_0 = \sum_{i=1}^N c_i^S \lambda_i^S v_i^S \\ e_2 &= S e_1 = \sum_{i=1}^N c_i^S \lambda_i^{S^2} v_i^S \\ e_3 &= S e_2 = \sum_{i=1}^N c_i^S \lambda_i^{S^3} v_i^S \end{aligned}$$

$$e_k = \sum_{i=1}^N c_i^S \lambda_i^{S^k} v_i^S$$

Podemos observar que quanto menor for o módulo do autovalor, melhor será a convergência do respectivo componente de autovetor e quanto maior o módulo do autovalor, pior será a convergência, o que nos leva a concluir que a convergência do método iterativo estacionário depende do autovalor que tem o maior módulo, ou seja, do raio espectral da matriz S :

$$\rho(S) = \left| \lambda^S \right|_{\max} \quad (15)$$

3.2.2.2 Análise dos Autovetores de Z

O resíduo inicial pode ser decomposto em autovetores da matriz Z :

$$r_0 = \sum_{i=1}^N c_i^Z v_i^Z$$

Onde c_i^Z é um escalar e v_i^Z é o i -ésimo autovetor da matriz Z .

Se multiplicarmos sucessivamente o resíduo pela matriz Z , equivale a multiplicar sucessivamente cada componente de autovetor v_i^Z pelo seu respectivo autovalor λ_i^Z :

$$r_1 = Z r_0 = \sum_{i=1}^N c_i^Z \lambda_i^Z v_i^Z$$

$$r_2 = Z r_1 = \sum_{i=1}^N c_i^Z \lambda_i^{Z^2} v_i^Z$$

$$r_3 = Z r_2 = \sum_{i=1}^N c_i^Z \lambda_i^{Z^3} v_i^Z$$

$$r_k = \sum_{i=1}^N c_i^Z \lambda_i^{Z^k} v_i^Z$$

Podemos observar que quanto menor for o módulo do autovalor, melhor será a convergência do respectivo componente de autovetor e quanto maior o módulo do autovalor, pior será a convergência, o que nos leva a concluir que a

convergência do método depende do autovalor que tem o maior módulo, ou seja, do raio espectral da matriz Z :

$$\rho(Z) = \left| \lambda^Z \right|_{\max} \quad (16)$$

3.2.2.3 Correlação entre Autovetores de S e Z

Se o erro for composto apenas por um autovetor, teremos:

$$e_0 = c_i^S v_i^S \Rightarrow r_0 = c_i^Z A v_i^S$$

$$e_1 = c_i^S \lambda_i^S v_i^S \Rightarrow r_1 = \lambda_i^S c_i^Z A v_i^S$$

$$e_2 = c_i^S \lambda_i^{S^2} v_i^S \Rightarrow r_2 = \lambda_i^{S^2} c_i^Z A v_i^S$$

$$e_3 = c_i^S \lambda_i^{S^3} v_i^S \Rightarrow r_3 = \lambda_i^{S^3} c_i^Z A v_i^S$$

Podemos verificar que o i -ésimo autovetor/autovalor de S está associado ao i -ésimo autovetor/autovalor de Z , onde:

$$\lambda_i^Z = \lambda_i^S$$

$$v_i^Z = c_i A v_i^S$$

Portanto o raio espectral de S é igual ao raio espectral de Z

$$\rho(Z) = \rho(S)$$

No método iterativo estacionário, quando o erro e o resíduo atingem a pior taxa de convergência, a taxa de convergência do erro tende a ser a mesma do resíduo que tende a coincidir com o raio espectral de S e Z .

3.2.3. Análise dos Componentes de Freqüência de Erro e Resíduo

As figuras a seguir ilustram qualitativamente a característica de relaxação dos métodos iterativos estacionários.

Na figura 3.1, podemos verificar que o método iterativo Gauss-Seidel [Seção 3.2.4.3] está relaxando com mais eficiência os componentes oscilatórios do erro. Na figura 3.2, podemos verificar que o método iterativo Gauss-Seidel está relaxando

com maior eficiência os componentes oscilatórios do resíduo. Na figura 3.3, podemos observar a relaxação mais eficiente dos componentes oscilatórios do erro também para os métodos Damped Jacobi [Seção 3.2.4.1] e SPAI [Seção 3.2.4.5].

Fig. 3.1 – Erro na resolução pelo método iterativo estacionário Gauss-Seidel de um problema bidimensional (Equação Elíptica). [26]

Fig. 3.2 – Resíduo nas iterações 0, 2 e 100 na resolução pelo método iterativo estacionário Gauss-Seidel (Equação de Poisson) [25]

Fig. 3.3 – Erro após 3 iterações (Equação de Poisson) [27]

Como podemos notar nestes exemplos, os métodos iterativos estacionários têm a característica de serem mais eficientes na relaxação de componentes oscilatórios (Alta frequência) de erro e resíduo e menos eficiente na relaxação dos componentes de erro e resíduo suaves (Baixa frequência). Esta característica, de tornar suave o erro e o resíduo, fez com que estes métodos fossem conhecidos como suavizadores.

3.2.4. Métodos em Destaque

Os métodos iterativos estacionários mais utilizados como suavizador do método MG são Damped Jacobi, Gauss-Seidel, SOR e SPAI.

Neste trabalho, serão testados os métodos Damped Jacobi, SOR e SPAI-0.

Para cada método iterativo estacionário abordado, são apresentados a forma de obtenção, suas matrizes iteração M , S e Z , a matriz aproximação $\tilde{A}$, suas particularidades e suas variações.

3.2.4.1 Jacobi

O método Jacobi é obtido pela decomposição da matriz A nas matrizes diagonal D , triangular inferior L e triangular superior U :

$$A = D - L - U \quad (17)$$

Substituindo $Ax = b$ (2), teremos:

$$(D - L - U)x = b$$

$$Dx = b + (L + U)x$$

$$D x_{k+1} = b + (L + U) x_k \quad (18)$$

O algoritmo do método Jacobi, que permite obter x_{k+1} , da iteração $k+1$, em função de x_k , da iteração k , pode ter seu algoritmo representado da seguinte forma:

$$x_{k+1,i} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_{k,j} - \sum_{j=i+1}^N a_{ij} x_{k,j}}{a_{ii}} \text{ para } i \text{ variando de } 1 \text{ a } N \quad (19)$$

A matriz iteração M é obtida subtraindo e adicionando Dx_k em $Dx_{k+1} = b + (L+U)x_k$ (18):

$$Dx_{k+1} = b + (L+U)x_k - Dx_k + Dx_k$$

$$Dx_{k+1} = b + (L+U-D)x_k + Dx_k$$

$$Dx_{k+1} = b - Ax_k + Dx_k$$

$$Dx_{k+1} - Dx_k = b - Ax_k$$

Substituindo $r = b - Ax_k$ (4), teremos:

$$D(x_{k+1} - x_k) = r_k$$

Substituindo $\Delta x = x_{k+1} - x_k$ (6), teremos:

$$\Delta x_{k+1} = D^{-1} r_k$$

Substituindo $\Delta x_{k+1} = M r_k$ (5), teremos:

$$M_{Jacobi} = D^{-1} \quad (20)$$

O algoritmo do método Jacobi, que permite obter Δx em função do resíduo, pode ser representado da seguinte forma:

$$\Delta x_{k+1,i} = \frac{r_{ki}}{a_{ii}} \text{ para } i \text{ variando de } 1 \text{ a } N \quad (21)$$

A complexidade de tempo fechada na determinação de Δx a partir do resíduo é $O(N)$. Como a complexidade de tempo de cada iteração não pode ser menor que para o cálculo do resíduo, a complexidade de tempo fechada por iteração resulta em $O(Nm)$.

A matriz aproximação $\tilde{A}$ é obtida a partir de $\tilde{A} \approx M^{-1}$ (7):

$$\tilde{A}_{Jacobi} = M_{Jacobi}^{-1} \quad (22)$$

Substituindo $M_{Jacobi} = D^{-1}$ (20), teremos:

$$\tilde{A}_{Jacobi} = (D^{-1})^{-1}$$

$$\tilde{A}_{Jacobi} = D \quad (23)$$

Podemos verificar que a matriz aproximação $\tilde{A}$ corresponde à matriz A com os elementos não pertencentes à diagonal principal desconsiderados. Quanto mais significativos forem os elementos da diagonal principal em relação aos demais elementos, melhor será a aproximação, sendo recomendado o uso do método Jacobi como solver para matriz A diagonal dominante.

A transformação do sistema, reordenando as colunas, visando concentrar os elementos mais significativos na diagonal principal pode melhorar o desempenho do método.

A matriz iteração de erro é obtida a partir de $S = I - MA$ (8):

$$S_{Jacobi} = I - M_{Jacobi}A$$

Substituindo $M_{Jacobi} = D^{-1}$ (20), teremos:

$$\begin{aligned} S_{Jacobi} &= I - D^{-1}A \\ S_{Jacobi} &= I - D^{-1}(D - L - U) \\ S_{Jacobi} &= I - I + D^{-1}(L + U) \\ S_{Jacobi} &= D^{-1}(L + U) \end{aligned} \quad (24)$$

A matriz iteração de resíduo é obtida a partir de $Z = I - AM$ (10):

$$Z_{Jacobi} = I - AM_{Jacobi}$$

Substituindo $M_{Jacobi} = D^{-1}$ (20), teremos:

$$\begin{aligned} Z_{Jacobi} &= I - AD^{-1} \\ Z_{Jacobi} &= I - (D - L - U)D^{-1} \\ Z_{Jacobi} &= I - I + (L + U)D^{-1} \\ Z_{Jacobi} &= (L + U)D^{-1} \end{aligned} \quad (25)$$

Se a matriz A for simétrica, a transposta de $S_{Jacobi} = D^{-1}(L + U)$ (24) resulta em:

$$S_{Jacobi}^T = [D^{-1}(L + U)]^T$$

$$S_{Jacobi}^T = (L^T + U^T)D^{-1}$$

Para $A = A^T \Rightarrow L^T = U, U^T = L$

$$S_{Jacobi}^T = (U + L)D^{-1}$$

$$Z_{Jacobi} = S_{Jacobi}^T \quad (26)$$

$$\|Z_{Jacobi}\|_F = \|S_{Jacobi}\|_F \quad (27)$$

Uma característica interessante do método Jacobi, é que suas matrizes iteração de erro S_{Jacobi} e iteração de resíduo Z_{Jacobi} são menos densa que a matriz A , favorecendo a sua exploração.

A complexidade de tempo fechada na obtenção das matrizes S_{Jacobi} e Z_{Jacobi} é $O(Nm)$ e a complexidade de espaço fechada para representar as matrizes S_{Jacobi} e Z_{Jacobi} é $O(Nm)$. A complexidade de tempo fechada na determinação da norma Frobenius de S_{Jacobi} e Z_{Jacobi} é $O(Nm)$.

3.2.4.2 Damped Jacobi

Uma variação importante do método Jacobi é o Damped Jacobi, no qual é inserido o parâmetro ω :

$$M_{DJ} = \omega M_{Jacobi}$$

Substituindo $M_{Jacobi} = D^{-1}$ (20), teremos:

$$M_{DJ} = \omega D^{-1} \quad (28)$$

A complexidade de tempo fechada na determinação de Δx a partir do resíduo é $O(N)$. Como a complexidade de tempo de cada iteração não pode ser menor que para o cálculo do resíduo, a complexidade de tempo fechada por iteração resulta em $O(Nm)$.

Utilizando a expressão $\Delta x_{k+1} = M_{DJ} r_k$ (5) no método Damped Jacobi, teremos:

$$\Delta x_{k+1} = M_{DJ} r_k$$

Substituindo $M_{DJ} = \omega D^{-1}$ (28), teremos:

$$\Delta x_{k+1} = \omega D^{-1} r_k \quad (29)$$

Substituindo $\Delta x_{k+1} = x_{k+1} - x_k$ (6), teremos:

$$x_{k+1} - x_k = \omega D^{-1} r_k$$

Substituindo $r = b - Ax$ (4), teremos:

$$\begin{aligned} x_{k+1} - x_k &= \omega D^{-1} (b - Ax_k) \\ x_{k+1} &= x_k + \omega D^{-1} (b - Ax_k) \end{aligned} \quad (30)$$

A matriz aproximação $\tilde{A}$ é obtida a partir de $\tilde{A} \approx M^{-1}$ (7):

$$\tilde{A}_{DJ} = M_{DJ}^{-1} \quad (31)$$

Substituindo $M_{DJ} = \omega D^{-1}$ (28), teremos:

$$\begin{aligned} \tilde{A}_{DJ} &= (\omega D^{-1})^{-1} \\ \tilde{A}_{DJ} &= \frac{1}{\omega} D \end{aligned} \quad (32)$$

A matriz iteração de erro é obtida a partir de $S = I - MA$ (8):

$$S_{DJ} = I - M_{DJ} A$$

Substituindo $M_{DJ} = \omega D^{-1}$ (28), teremos:

$$\begin{aligned} S_{DJ} &= I - \omega D^{-1} A \\ S_{DJ} &= I - \omega D^{-1} (D - L - U) \\ S_{DJ} &= I - \omega I + \omega D^{-1} (L + U) \\ S_{DJ} &= (1 - \omega) I + \omega D^{-1} (L + U) \end{aligned} \quad (33)$$

A matriz iteração de resíduo é obtida a partir de $Z = I - AM$ (10):

$$Z_{DJ} = I - A M_{DJ}$$

Substituindo $M_{DJ} = \omega D^{-1}$ (28), teremos:

$$\begin{aligned} Z_{DJ} &= I - \omega A D^{-1} \\ Z_{DJ} &= I - \omega (D - L - U) D^{-1} \\ Z_{DJ} &= I - \omega I + \omega (L + U) D^{-1} \end{aligned}$$

$$Z_{DJ} = (1 - \omega)I + \omega(L + U)D^{-1} \quad (34)$$

Uma característica interessante do método Damped Jacobi, é que suas matrizes iteração de erro S_{Jacobi} e iteração de resíduo Z_{Jacobi} têm a mesma esparsidade que a matriz A , favorecendo a sua exploração.

A complexidade de tempo fechada na obtenção das matrizes S_{DJ} e Z_{DJ} é $O(Nm)$ e a complexidade de espaço fechada para representar as matrizes S_{DJ} e Z_{DJ} é $O(Nm)$. A complexidade de tempo fechada na determinação da norma Frobenius de S_{DJ} e Z_{DJ} é $O(Nm)$.

Se a matriz A for simétrica, a transposta de $S_{DJ} = (1 - \omega)I + \omega D^{-1}(L + U)$ (33) resulta em:

$$\begin{aligned} S_{DJ}^T &= [(1 - \omega)I + \omega D^{-1}(L + U)]^T \\ S_{DJ}^T &= (1 - \omega)I + \omega(L^T + U^T)D^{-1} \\ \text{Para } A = A^T &\Rightarrow L^T = U, U^T = L \\ S_{DJ}^T &= (1 - \omega)I + \omega(U + L)D^{-1} \\ Z_{DJ} &= S_{DJ}^T \end{aligned} \quad (35)$$

$$\|Z_{DJ}\|_F = \|S_{DJ}\|_F \quad (36)$$

3.2.4.3 Gauss-Seidel

O método Gauss-Seidel possui 4 variações em destaque: Forward Gauss-Seidel (FGS), Backward Gauss-Seidel (BGS), Red Black Gauss-Seidel (RBGS) e Symmetric Gauss-Seidel (SGS). Tanto o FGS, quanto o BGS são obtidos a partir da decomposição da matriz A nas matrizes D , L e U . O SGS corresponde à combinação do FGS com o BGS.

O FGS é obtido a partir da decomposição $A = D - L - U$ (17), substituindo em $Ax = b$ (2):

$$(D - L - U)x = b$$

$$(D - L)x = Ux + b$$

$$(D-L) x_{k+1} = U x_k + b \quad (37)$$

O algoritmo do método FGS, que permite obter a solução posterior em função da anterior, pode ser representado da seguinte forma:

$$x_{k+1} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_{k+1j} - \sum_{j=i+1}^N a_{ij} x_{kj}}{a_{ii}} \text{ para } i \text{ variando de } 1 \text{ a } N \quad (38)$$

A matriz iteração M é obtida adicionando e subtraindo $A x_k$ em $(D-L) x_{k+1} = U x_k + b$ (37):

$$(D-L) x_{k+1} = U x_k + b + A x_k - A x_k$$

$$(D-L) x_{k+1} = (A+U) x_k + b - A x_k$$

Substituindo $A = D - L - U$ (17), teremos:

$$(D-L) x_{k+1} = (D-L) x_k + b - A x_k$$

Substituindo $r = b - A x_k$ (4), teremos:

$$(D-L) x_{k+1} = (D-L) x_k + r$$

$$(D-L)(x_{k+1} - x_k) = r$$

Substituindo $\Delta x = x_{k+1} - x_k$ (6), teremos:

$$\Delta x_{k+1} = (D-L)^{-1} r$$

Substituindo $\Delta x_{k+1} = M r_k$ (5), teremos:

$$M_{FGS} = (D-L)^{-1} \quad (39)$$

O algoritmo do método FGS, que permite obter Δx em função do resíduo, pode ser representado da seguinte forma:

$$\Delta x_{k+1} = \frac{r_i - \sum_{j=1}^{i-1} a_{ij} \Delta x_{kj}}{a_{ii}} \text{ para } i \text{ variando de } 1 \text{ a } N \quad (40)$$

A complexidade de tempo fechada no processamento de cada iteração é $O(Nm)$.

A matriz aproximação $\tilde{A}$ é obtida a partir de $\tilde{A} \approx M^{-1}$ (7):

$$\tilde{A}_{FGS} = M_{FGS}^{-1}$$

Substituindo $M_{FGS} = (D - L)^{-1}$ (39), teremos:

$$\tilde{A}_{FGS} = [(D - L)^{-1}]^{-1}$$

$$\tilde{A}_{FGS} = D - L \quad (41)$$

Podemos verificar que a matriz aproximação $\tilde{A}$ corresponde à matriz A com os elementos da triangular superior desconsiderados. O método FGS é um método direto para o caso particular em que a triangular superior da matriz A possuir todos os elementos nulos.

A matriz iteração de erro é obtida a partir de $S = I - MA$ (8):

$$S_{FGS} = I - M_{FGS}A$$

Substituindo $A = D - L - U$ (17), teremos:

$$S_{FGS} = I - M_{FGS}(D - L - U)$$

Substituindo $M_{FGS} = (D - L)^{-1}$ (39), teremos:

$$S_{FGS} = I - (D - L)^{-1}(D - L - U)$$

$$S_{FGS} = I - I - (D - L)^{-1}U$$

$$S_{FGS} = -(D - L)^{-1}U \quad (42)$$

Para que a matriz S seja próxima da matriz nula, sinalizando boa eficiência do método, é necessário que os elementos de U sejam próximos a zero e os elementos da matriz D sejam dominantes.

A matriz iteração de resíduo é obtida a partir de $Z = I - AM$ (10):

$$Z_{FGS} = I - AM_{FGS}$$

Substituindo $A = D - L - U$ (17), teremos:

$$Z_{FGS} = I - (D - L - U)M_{FGS}$$

Substituindo $M_{FGS} = (D - L)^{-1}$ (39), teremos:

$$\begin{aligned}
 Z_{FGS} &= I - (D - L - U)(D - L)^{-1} \\
 Z_{FGS} &= I - I - U(D - L)^{-1} \\
 Z_{FGS} &= -U(D - L)^{-1}
 \end{aligned} \tag{43}$$

A complexidade de tempo superior na determinação das matrizes S e Z é $O(N^2m)$ e a complexidade de espaço superior para representar as matrizes S e Z é $O(N^2)$, não justificando a determinação destas matrizes.

A complexidade de tempo superior na determinação da norma Frobenius de S e Z é $O(N^2m)$, não justificando utilizá-la como indicador de convergência.

O BGS é obtido a partir da decomposição $A = D - L - U$ (17), substituindo em $Ax = b$ (2):

$$\begin{aligned}
 (D - L - U)x &= b \\
 (D - U)x &= Lx + b \\
 (D - U) x_{k+1} &= Lx_k + b
 \end{aligned} \tag{44}$$

O algoritmo do método BGS, que permite obter a solução posterior em função da anterior, pode ser representado da seguinte forma:

$$x_{k+1i} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_{k+1j} - \sum_{j=i+1}^N a_{ij} x_{kj}}{a_{ii}} \text{ para } i \text{ variando de } N \text{ a } 1 \tag{45}$$

A matriz iteração M é obtida adicionando e subtraindo Ax_k em $(D - U) x_{k+1} = Lx_k + b$ (44):

$$\begin{aligned}
 (D - U) x_{k+1} &= Lx_k + b + Ax_k - Ax_k \\
 (D - U) x_{k+1} &= (A + L)x_k + b - Ax_k
 \end{aligned}$$

Substituindo $A = D - L - U$ (17), teremos:

$$(D - U) x_{k+1} = (D - U)x_k + b - Ax_k$$

Substituindo $r = b - Ax_k$ (4), teremos:

$$(D-U)_{k+1} x = (D-U)_k x + r_k$$

$$(D-U)_{k+1} (x_{k+1} - x_k) = r_k$$

Substituindo $\Delta x = x_{k+1} - x_k$ (6), teremos:

$$(D-U)_{k+1} \Delta x = r_k$$

$$\Delta x_{k+1} = (D-U)^{-1}_k r_k$$

Substituindo $\Delta x = M_{k+1} r_k$ (5), teremos:

$$M_{BGS} = (D-U)^{-1} \quad (46)$$

O algoritmo do método BGS, que permite obter Δx em função do resíduo, pode ser representado da seguinte forma:

$$\Delta x_{k+1 i} = \frac{r_{k i} - \sum_{j=i+1}^N a_{ij} \Delta x_{k+1 j}}{a_{ii}} \quad \text{para } i \text{ variando de } N \text{ a } 1 \quad (47)$$

A matriz aproximação $\tilde{A}$ é obtida a partir de $\tilde{A} \approx M^{-1}$ (7):

$$\tilde{A}_{BGS} = M_{BGS}^{-1}$$

Substituindo $M_{BGS} = (D-U)^{-1}$ (46), teremos:

$$\begin{aligned} \tilde{A}_{BGS} &= [(D-U)^{-1}]^{-1} \\ \tilde{A}_{BGS} &= D-U \end{aligned} \quad (48)$$

Podemos verificar que a matriz aproximação $\tilde{A}$ corresponde à matriz A com os elementos da triangular inferior desconsiderados. O método BGS é um método direto para o caso particular em que os elementos da matriz L forem nulos.

A matriz iteração de erro é obtida a partir de $S = I - MA$ (8):

$$S_{BGS} = I - M_{BGS} A$$

Substituindo $A = D - L - U$ (17), teremos:

$$S_{BGS} = I - M_{BGS} (D - L - U)$$

Substituindo $M_{BGS} = (D-U)^{-1}$ (46), teremos:

$$\begin{aligned}
 S_{BGS} &= I - (D - U)^{-1}(D - L - U) \\
 S_{FGS} &= I - I - (D - U)^{-1}L \\
 S_{BGS} &= -(D - U)^{-1}L
 \end{aligned} \tag{49}$$

Para que a matriz S seja próxima da matriz nula, sinalizando boa eficiência do método, é necessário que os elementos da matriz L sejam próximos a zero e os elementos da matriz D sejam dominantes.

A matriz iteração de resíduo é obtida a partir de $Z = I - AM$ (10):

$$Z_{BGS} = I - AM_{BGS}$$

Substituindo $A = D - L - U$ (17), teremos:

$$Z_{BGS} = I - (D - L - U)M_{BGS}$$

Substituindo $M_{BGS} = (D - U)^{-1}$ (46), teremos:

$$\begin{aligned}
 Z_{BGS} &= I - (D - L - U)(D - U)^{-1} \\
 Z_{BGS} &= I - I - L(D - U)^{-1} \\
 Z_{BGS} &= -L(D - U)^{-1}
 \end{aligned} \tag{50}$$

A complexidade de tempo superior na determinação das matrizes S e Z é $O(N^2m)$ e a complexidade de espaço superior para representar as matrizes S e Z é $O(N^2)$, não justificando a determinação destas matrizes.

A complexidade de tempo superior na determinação da norma Frobenius de S é $O(N^2m)$, não justificando utilizá-la como indicador de convergência.

Ao compararmos o algoritmo (38) do método FGS com o algoritmo (45) do método BGS, podemos ver que são semelhantes, se diferenciando apenas pela ordem que são determinados os elementos do vetor x , ou seja, os elementos do vetor x são determinados em ordem crescente no método FGS e em ordem decrescente no método BGS.

O método RBGS tem como ordem de determinação dos elementos de x , primeiro todos os elementos ímpares e depois todos os componentes pares, ou vice versa.

O método SGS corresponde em aplicar, a cada iteração, uma iteração do FGS e outra iteração do BGS.

O algoritmo do SGS, que permite obter a solução posterior em função da anterior, pode ser representado da seguinte forma:

$$\begin{aligned} x_{i, k+\frac{1}{2}} &= \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_{j, k+\frac{1}{2}} - \sum_{j=i+1}^N a_{ij} x_{j, k}}{a_{ii}} \quad \text{para } i \text{ variando de } 1 \text{ a } N \\ x_{i, k+1} &= \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_{j, k+1} - \sum_{j=i+1}^N a_{ij} x_{j, k+\frac{1}{2}}}{a_{ii}} \quad \text{para } i \text{ variando de } N \text{ a } 1 \end{aligned} \quad (51)$$

O algoritmo do SGS, que permite obter Δx em função do resíduo, pode ser representado da seguinte forma:

$$\begin{aligned} \Delta x_{i, k+\frac{1}{2}} &= \frac{r_{i, k} - \sum_{j=1}^{i-1} a_{ij} \Delta x_{j, k+\frac{1}{2}}}{a_{ii}} \quad \text{para } i \text{ variando de } 1 \text{ a } N \\ \Delta x_{i, k+1} &= \frac{r_{i, k+\frac{1}{2}} - \sum_{j=i+1}^N a_{ij} \Delta x_{j, k+1}}{a_{ii}} \quad \text{para } i \text{ variando de } N \text{ a } 1 \end{aligned} \quad (52)$$

A matriz iteração de erro S corresponde ao produto das matrizes iteração de erro do FGS e do BGS:

$$S_{SGS} = S_{BGS} S_{FGS} \quad (53)$$

Substituindo $S_{BGS} = -(D-U)^{-1}L$ (49) e $S_{FGS} = -(D-L)^{-1}U$ (42), teremos:

$$\begin{aligned} S_{SGS} &= [-(D-U)^{-1}L][-(D-L)^{-1}U] \\ S_{SGS} &= (D-U)^{-1}LU(D-L)^{-1} \end{aligned} \quad (54)$$

Podemos verificar que, no caso em que a matriz A seja simétrica, a matriz S será simétrica:

$$\begin{aligned}
 S_{SGS}^T &= [(D-U)^{-1}LU(D-L)^{-1}]^T \\
 S_{SGS}^T &= (D^T - L^T)^{-1}U^T L^T (D^T - U^T)^{-1} \\
 A = A^T &\Rightarrow L^T = U, U^T = L \\
 S_{SGS}^T &= (D-U)^{-1}LU(D-L)^{-1} \\
 S_{SGS}^T &= S_{SGS}
 \end{aligned} \tag{55}$$

A matriz iteração de resíduo corresponde ao produto das matrizes iteração de resíduo do FGS e do BGS:

$$Z_{SGS} = Z_{BGS}Z_{FGS} \tag{56}$$

Substituindo $Z_{BGS} = -L(D-U)^{-1}$ (50) e $Z_{FGS} = -U(D-L)^{-1}$ (43), teremos:

$$\begin{aligned}
 Z_{SGS} &= [-L(D-U)^{-1}][-(D-L)^{-1}U] \\
 Z_{SGS} &= L(D-U)^{-1}(D-L)^{-1}U
 \end{aligned} \tag{57}$$

Podemos verificar que, caso a matriz A seja simétrica, Z será simétrica:

$$\begin{aligned}
 Z_{SGS}^T &= [L(D-U)^{-1}(D-L)^{-1}U]^T \\
 Z_{SGS}^T &= U^T (D^T - L^T)^{-1} (D^T - U^T)^{-1} L^T \\
 Z_{SGS}^T &= L(D-U)^{-1}(D-L)^{-1}U \\
 Z_{SGS}^T &= Z_{SGS}
 \end{aligned} \tag{58}$$

A complexidade de tempo superior na determinação das matrizes S e Z é $O(N^2m)$ e a complexidade de espaço superior para representar as matrizes S e Z é $O(N^2)$, não justificando a determinação destas matrizes. A complexidade de tempo superior na determinação da norma Frobenius de S é $O(N^2m)$, não justificando utilizá-la como indicador de convergência.

3.2.4.4 SOR (Successive Over Relaxation)

O método SOR conta com o coeficiente de relaxação ω , oferecendo maior flexibilidade na sua utilização. Impor valor 1 para este coeficiente de relaxação ω , equivale a aplicar o método de Gauss-Seidel.

Assim como o método de Gauss-Seidel, o método SOR possui 4 variações em destaque: Forward SOR (FSOR), Backward SOR (BSOR), Red Black SOR e Symmetric SOR (SSOR).

O FSOR é obtido a partir da decomposição $A = D - L - U$ (17), substituindo em $Ax = b$ (2):

$$(D - L - U)x = b$$

$$\omega(D - L - U)x = \omega b$$

$$-\omega Lx = \omega b - \omega(D - U)x$$

Inserindo Dx dos dois lados, teremos:

$$Dx - \omega Lx = Dx + \omega b - \omega(D - U)x$$

$$(D - \omega L)x = Dx + \omega b - \omega(D - U)x$$

$$(D - \omega L) x_{k+1} = Dx_k + \omega b - \omega(D - U) x_k \quad (59)$$

O algoritmo do método FSOR, que permite obter a solução da iteração posterior a partir da anterior, pode ser representado da seguinte forma:

$$x_{i,k+1} = (1 - \omega) x_{i,k} + \omega \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_{j,k+1} - \sum_{j=i+1}^N a_{ij} x_{j,k}}{a_{ii}} \quad \text{para } i \text{ variando} \quad (60)$$

de 1 a N

A matriz iteração M é obtida subtraindo e adicionando $\omega L x_k$ em

$$(D - \omega L) x_{k+1} = Dx_k + \omega b - \omega(D - U) x_k \quad (59):$$

$$(D - \omega L) x_{k+1} = Dx_k + \omega b - \omega(D - U) x_k - \omega L x_k + \omega L x_k$$

$$(D - \omega L) x_{k+1} = (D - \omega L) x_k + \omega L x_k + \omega b - \omega(D - U) x_k$$

$$(D - \omega L)(x_{k+1} - x_k) = \omega b - \omega(D - L - U) x_k$$

Substituindo $\Delta x = x_{k+1} - x_k$ (6), teremos:

$$(D - \omega L) \Delta x_{k+1} = \omega r_k$$

$$\Delta x_{k+1} = \omega (D - \omega L)^{-1} r_k$$

Substituindo $\Delta x_{k+1} = M r_k$ (5), teremos:

$$\begin{aligned} M_{FSOR} &= \omega (D - \omega L)^{-1} \\ M_{FSOR} &= \left(\frac{1}{\omega} D - L \right)^{-1} \end{aligned} \quad (61)$$

O algoritmo do método FSOR, que permite obter Δx em função do resíduo, pode ser representado da seguinte forma:

$$\Delta x_{k+1} = \omega \frac{r_i^k - \sum_{j=1}^{i-1} a_{ij} \Delta x_{j,k+1}}{a_{ii}} \text{ para } i \text{ variando de } 1 \text{ a } N \quad (62)$$

A matriz aproximação $\tilde{A}$ é obtida a partir de $\tilde{A} \approx M^{-1}$ (7):

$$\tilde{A}_{FSOR} = M_{FSOR}^{-1}$$

Substituindo $M_{FSOR} = \left(\frac{1}{\omega} D - L \right)^{-1}$ (61), teremos:

$$\begin{aligned} \tilde{A}_{FSOR} &= \left[\left(\frac{1}{\omega} D - L \right)^{-1} \right]^{-1} \\ \tilde{A}_{FSOR} &= \frac{1}{\omega} D - L \end{aligned} \quad (63)$$

A matriz iteração de erro é obtida a partir de $S = I - MA$ (8):

$$S_{FSOR} = I - M_{FSOR} A$$

Substituindo $A = D - L - U$ (17), teremos:

$$S_{FGS} = I - M_{FSOR} (D - L - U)$$

Substituindo $M_{FSOR} = \left(\frac{1}{\omega} D - L \right)^{-1}$ (61), teremos:

$$\begin{aligned} S_{FSOR} &= I - \left(\frac{1}{\omega} D - L \right)^{-1} (D - L - U) \\ S_{FSOR} &= I - \left(\frac{1}{\omega} D - L \right)^{-1} \left(\frac{1}{\omega} D - \frac{1}{\omega} D + D - L - U \right) \\ S_{FSOR} &= I - I - \left(\frac{1}{\omega} D - L \right)^{-1} \left(-\frac{1}{\omega} D + D - U \right) \end{aligned}$$

$$S_{FSOR} = -\left(\frac{1}{\omega}D - L\right)^{-1}\left[\left(1 - \frac{1}{\omega}\right)D - U\right] \quad (64)$$

A matriz iteração de resíduo é obtida a partir de $Z = I - AM$ (10):

$$Z_{FSOR} = I - AM_{FSOR}$$

Substituindo $A = D - L - U$ (17), teremos:

$$Z_{FSOR} = I - (D - L - U)M_{FSOR}$$

Substituindo $M_{FSOR} = \left(\frac{1}{\omega}D - L\right)^{-1}$ (61), teremos:

$$\begin{aligned} Z_{FSOR} &= I - (D - L - U)\left(\frac{1}{\omega}D - L\right)^{-1} \\ Z_{FSOR} &= I - \left(\frac{1}{\omega}D - \frac{1}{\omega}D + D - L - U\right)\left(\frac{1}{\omega}D - L\right)^{-1} \\ Z_{FSOR} &= I - I - \left(-\frac{1}{\omega}D + D - U\right)\left(\frac{1}{\omega}D - L\right)^{-1} \\ Z_{FSOR} &= -\left[\left(1 - \frac{1}{\omega}\right)D - U\right]\left(\frac{1}{\omega}D - L\right)^{-1} \end{aligned} \quad (65)$$

A complexidade de tempo superior na determinação das matrizes S e Z é $O(N^2m)$ e a complexidade de espaço superior para representar as matrizes S e Z é $O(N^2)$, não justificando a determinação destas matrizes. A complexidade de tempo superior na determinação da norma Frobenius de S é $O(N^2m)$, não justificando utilizá-la como indicador de convergência.

O método BSOR é obtido a partir da decomposição $A = D - L - U$ (17), substituindo em $Ax = b$ (2):

$$\begin{aligned} (D - L - U)x &= b \\ \omega(D - L - U)x &= \omega b \\ -\omega Ux &= \omega b - \omega(D - L)x \end{aligned}$$

Inserindo Dx dos dois lados, teremos:

$$\begin{aligned} Dx - \omega Ux &= Dx + \omega b - \omega(D - L)x \\ (D - \omega U)x &= Dx + \omega b - \omega(D - L)x \\ (D - \omega U) x_{k+1} &= Dx_k + \omega b - \omega(D - L) x_k \end{aligned} \quad (66)$$

O algoritmo do método BSOR, que permite obter a solução da iteração posterior a partir da anterior, pode ser representado da seguinte forma:

$$x_{k+1} = (1 - \omega) x_k + \omega \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_{j,k} - \sum_{j=i+1}^N a_{ij} x_{j,k+1}}{a_{ii}} \text{ para } i \text{ variando de } N \text{ a } 1 \quad (67)$$

A matriz iteração M é obtida subtraindo e adicionando $\omega L x_k$ em $(D - \omega U) x_{k+1} = D x_k + \omega b - \omega(D - L) x_k$ (66):

$$\begin{aligned} (D - \omega U) x_{k+1} &= D x_k - \omega U x_k + \omega U x_k + \omega b - \omega(D - L) x_k \\ (D - \omega U) x_{k+1} &= (D - \omega U) x_k + \omega U x_k + \omega b - \omega(D - L) x_k \\ (D - \omega U)(x_{k+1} - x_k) &= \omega b - \omega(D - L - U) x_k \end{aligned}$$

Substituindo $A = D - L - U$ (17), teremos:

$$(D - \omega U)(x_{k+1} - x_k) = \omega b - \omega A x_k$$

Substituindo $\Delta x = x_{k+1} - x_k$ (6), teremos:

$$\begin{aligned} (D - \omega U) \Delta x_{k+1} &= \omega r_k \\ \Delta x_{k+1} &= \omega(D - \omega U)^{-1} r_k \end{aligned}$$

Substituindo $\Delta x = M r_k$ (5), teremos:

$$\begin{aligned} M_{BSOR} &= \omega(D - \omega U)^{-1} \\ M_{BSOR} &= \left(\frac{1}{\omega} D - U \right)^{-1} \end{aligned} \quad (68)$$

O algoritmo do método BSOR, que permite obter Δx em função do resíduo, pode ser representado da seguinte forma:

$$\Delta x_{k+1} = \omega \frac{r_k - \sum_{j=i+1}^N a_{ij} \Delta x_{j,k+1}}{a_{ii}} \text{ para } i \text{ variando de } N \text{ a } 1 \quad (69)$$

A matriz aproximação $\tilde{A}$ é obtida a partir de $\tilde{A} \approx M^{-1}$ (7):

$$\tilde{A}_{BSOR} = M_{BSOR}^{-1}$$

Substituindo $M_{BSOR} = (\frac{1}{\omega} D - U)^{-1}$ (68), teremos:

$$\tilde{A}_{BSOR} = [(\frac{1}{\omega} D - U)^{-1}]^{-1}$$

$$\tilde{A}_{BSOR} = \frac{1}{\omega} D - U \quad (70)$$

A matriz iteração de erro é obtida a partir de $S = I - MA$ (8):

$$S_{BSOR} = I - M_{BSOR} A$$

Substituindo $A = D - L - U$ (17), teremos:

$$S_{BSOR} = I - M_{BSOR} (D - L - U)$$

Substituindo $M_{BSOR} = (\frac{1}{\omega} D - U)^{-1}$ (68), teremos:

$$S_{BSOR} = I - (\frac{1}{\omega} D - U)^{-1} (D - L - U)$$

$$S_{BSOR} = I - (\frac{1}{\omega} D - U)^{-1} (\frac{1}{\omega} D - \frac{1}{\omega} D + D - L - U)$$

$$S_{BSOR} = I - I - (\frac{1}{\omega} D - U)^{-1} (-\frac{1}{\omega} D + D - L)$$

$$S_{BSOR} = -(\frac{1}{\omega} D - U)^{-1} [(1 - \frac{1}{\omega}) D - L] \quad (71)$$

A matriz iteração de resíduo é obtida a partir de $Z = I - AM$ (10):

$$Z_{BSOR} = I - AM_{BSOR}$$

Substituindo $A = D - L - U$ (17), teremos:

$$Z_{BSOR} = I - (D - L - U) M_{BSOR}$$

Substituindo $M_{BSOR} = (\frac{1}{\omega} D - U)^{-1}$ (68), teremos:

$$Z_{BSOR} = I - (D - L - U) (\frac{1}{\omega} D - U)^{-1}$$

$$Z_{BSOR} = I - (\frac{1}{\omega} D - \frac{1}{\omega} D + D - L - U) (\frac{1}{\omega} D - U)^{-1}$$

$$Z_{BSOR} = I - I - (-\frac{1}{\omega} D + D - L) (\frac{1}{\omega} D - U)^{-1}$$

$$Z_{BSOR} = -[(1 - \frac{1}{\omega}) D - L] (\frac{1}{\omega} D - U)^{-1} \quad (72)$$

A complexidade de tempo superior na determinação das matrizes S e Z é $O(N^2m)$ e a complexidade de espaço superior para representar as matrizes S e Z é $O(N^2)$, não justificando a determinação destas matrizes. A complexidade de tempo superior na determinação da norma Frobenius de S é $O(N^2m)$, não justificando utilizá-la como indicador de convergência.

Conforme observado para o método Gauss-Seidel, ao compararmos os algoritmos (60) e (67), podemos observar que são equivalentes, exceto pela ordem inversa de determinação dos elementos do vetor x .

O método RBSOR tem como ordem de determinação dos elementos de x , primeiro todos os elementos ímpares e depois todos os componentes pares, ou vice versa.

O método SSOR corresponde em aplicar, a cada iteração, uma iteração do FSOR e outra iteração do BSOR.

O algoritmo do SSOR, que permite obter a solução da iteração posterior a partir da anterior, pode ser representado da seguinte forma:

$$x_i = (1 - \omega) x_i + \omega \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^N a_{ij} x_j}{a_{ii}} \quad \text{para } i$$

variando de 1 a N

$$x_i = (1 - \omega) x_i + \omega \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^N a_{ij} x_j}{a_{ii}} \quad \text{para } i$$

variando de N a 1

O algoritmo do SSOR, que permite obter Δx em função do resíduo, pode ser representado da seguinte forma:

$$\Delta x_i = \omega \frac{r_i - \sum_{j=1}^{i-1} a_{ij} \Delta x_j}{a_{ii}} \quad \text{para } i \text{ variando de } 1 \text{ a } N \quad (74)$$

$$\Delta x_i = \omega \frac{r_i - \sum_{j=i+1}^N a_{ij} \Delta x_j}{a_{ii}} \quad \text{para } i \text{ variando de } N \text{ a } 1$$

A matriz iteração de erro S corresponde ao produto das matrizes iteração de erro do FSOR e do BSOR:

$$S_{SSOR} = S_{BSOR} S_{FSOR} \quad (75)$$

Substituindo $S_{BSOR} = -(\frac{1}{w}D - U)^{-1}[(1 - \frac{1}{w})D - L]$ (71) e $S_{FSOR} = -(\frac{1}{w}D - L)^{-1}[(1 - \frac{1}{w})D - U]$ (64),

teremos:

$$S_{SSOR} = [-(\frac{1}{w}D - U)^{-1}][(1 - \frac{1}{w})D - L][-(\frac{1}{w}D - L)^{-1}][(1 - \frac{1}{w})D - U]$$

$$S_{SSOR} = (\frac{1}{w}D - U)^{-1}[(1 - \frac{1}{w})D - L](\frac{1}{w}D - L)^{-1}[(1 - \frac{1}{w})D - U] \quad (76)$$

Podemos verificar que a matriz S_{SSOR} não é simétrica.

A matriz iteração de resíduo corresponde ao produto das matrizes iteração de resíduo do FSOR e do BSOR:

$$Z_{SSOR} = Z_{BSOR} Z_{FSOR} \quad (77)$$

Substituindo $Z_{BSOR} = -(1 - \frac{1}{w})D - L)(\frac{1}{w}D - U)^{-1}$ (72) e $Z_{FSOR} = -(1 - \frac{1}{w})D - U)(\frac{1}{w}D - L)^{-1}$

(65), teremos:

$$Z_{SSOR} = \{-(1 - \frac{1}{w})D - L\}(\frac{1}{w}D - U)^{-1}\{-(1 - \frac{1}{w})D - U\}(\frac{1}{w}D - L)^{-1}$$

$$Z_{SSOR} = [(1 - \frac{1}{w})D - L](\frac{1}{w}D - U)^{-1}[(1 - \frac{1}{w})D - U](\frac{1}{w}D - L)^{-1} \quad (78)$$

Podemos verificar que a matriz Z_{SSOR} não é simétrica.

A complexidade de tempo superior na determinação das matrizes S e Z é $O(N^2m)$ e a complexidade de espaço superior para representar as matrizes S e Z é $O(N^2)$, não justificando a determinação destas matrizes. A complexidade de tempo superior na determinação da norma Frobenius de S é $O(N^2m)$, não justificando utilizá-la como indicador de convergência.

3.2.4.5 Sparse Approximate Inverse (SPAI)

O método SPAI consiste em determinar uma matriz M esparsa que se aproxime da inversa de A .

Inicialmente, são escolhidos quais elementos da matriz M serão não nulos e depois, são obtidos os valores para estes elementos não nulos que resultam no menor valor da norma Frobenius da matriz iteração de erro S .

$$\begin{aligned} \min_M \|S\|_F &\rightarrow M_{SPAI} \\ \min_M \|I - MA\|_F &\rightarrow M_{SPAI} \end{aligned} \quad (79)$$

O valor dos elementos não nulos da matriz M é calculado pelo método dos mínimos quadrados.

As variações mais comuns do método SPAI são SPAI-n e SPAI(ϵ).

O SPAI-n tem a mesma distribuição de elementos não nulos que A^n , ou seja, o SPAI-0 é uma matriz diagonal, o SPAI-1 tem a mesma esparsidade que A , o SPAI-2 tem a mesma disposição que $A^T.A$ e assim por diante.

No SPAI-0, os elementos de M são determinados por:

$$\begin{aligned} s_{ij} &= i_{ij} - m_{ii}a_{ij} \\ |s_{ij}|^2 &= (i_{ij} - m_{ii}a_{ij})(i_{ij} - m_{ii}^*a_{ij}^*) \\ \frac{\delta |s_{ij}|^2}{\delta m_{ii}} &= -a_{ij}(i_{ij} - m_{ii}^*a_{ij}^*) - \frac{\delta m_{ii}^*}{\delta m_{ii}} a_{ij}^*(i_{ij} - m_{ii}a_{ij}) \\ \frac{\delta \sum_{j=1}^N |s_{ij}|^2}{\delta m_{ii}} &= -\sum_{j=1}^N a_{ij}(i_{ij} - m_{ii}^*a_{ij}^*) - \frac{\delta m_{ii}^*}{\delta m_{ii}} \sum_{j=1}^N a_{ij}^*(i_{ij} - m_{ii}a_{ij}) \end{aligned} \quad (80)$$

$$\frac{\delta \sum_{j=1}^N |s_{ij}|^2}{\delta m_{ii}} = -a_{ii} + m_{ii}^* \sum_{j=1}^N |a_{ij}|^2 + \frac{\delta m_{ii}^*}{\delta m_{ii}} [-a_{ii}^* + m_{ii} \sum_{j=1}^N |a_{ij}|^2]$$

Para $\frac{\delta \sum_{j=1}^N |s_{ij}|^2}{\delta m_{ii}} = 0$, teremos:

$$-a_{ii} + m_{ii}^* \sum_{j=1}^N |a_{ij}|^2 + \frac{\delta m_{ii}^*}{\delta m_{ii}} [-a_{ii}^* + m_{ii} \sum_{j=1}^N |a_{ij}|^2] = 0$$

$$-a_{ii} + m_{ii}^* \sum_{j=1}^N |a_{ij}|^2 = -\frac{\delta m_{ii}^*}{\delta m_{ii}} [-a_{ii}^* + m_{ii} \sum_{j=1}^N |a_{ij}|^2]$$

$$-a_{ii}^* + m_{ii} \sum_{j=1}^N |a_{ij}|^2 = 0$$

$$m_{ii} \sum_{j=1}^N |a_{ij}|^2 = a_{ii}^*$$

$$m_{ii} = \frac{a_{ii}^*}{\sum_{j=1}^N \|a_{ij}\|^2} \quad (81)$$

A matriz M é uma matriz diagonal. A complexidade de tempo fechada da rotina de determinação da matriz M é $O(Nm)$. A complexidade de espaço fechada na reprodução da matriz M é $O(N)$. A complexidade de tempo fechada de cada iteração é $O(Nm)$. A complexidade de tempo fechada na obtenção das matrizes S e Z é $O(Nm)$ e a complexidade de espaço fechada para representar as matrizes S e Z é $O(Nm)$. A complexidade de tempo fechada na determinação da norma Frobenius de S e Z é $O(Nm)$.

Para a norma Frobenius da matriz S_{SPAI-0} , teremos:

$$\|S_{SPAI-0}\|^2 = \sum_{i=1}^N \sum_{j=1}^N |s_{ij}|^2$$

Substituindo $|s_{ij}|^2 = (i_{ij} - m_{ii} a_{ij})(i_{ij} - m_{ii}^* a_{ij}^*)$ (80), teremos:

$$\|S_{SPAI-0}\|^2 = \sum_{i=1}^N \sum_{j=1}^N (i_{ij} - m_{ii} a_{ij})(i_{ij} - m_{ii}^* a_{ij}^*)$$

$$\|S_{SPAI-0}\|^2 = \sum_{i=1}^N \sum_{j=1}^N (i_{ij}^2 - i_{ij} m_{ii} a_{ij} - i_{ij} m_{ii}^* a_{ij}^* + |m_{ii}|^2 |a_{ij}|^2)$$

$$\|S_{SPAI-0}\|^2 = N - \sum_{i=1}^N (m_{ii} a_{ii} + m_{ii}^* a_{ii}^*) + \sum_{i=1}^N m_{ii} m_{ii}^* \sum_{j=1}^N |a_{ij}|^2$$

Substituindo $m_{ii} = \frac{a_{ii}^*}{\sum_{j=1}^N \|a_{ij}\|^2}$ (81), teremos:

$$\begin{aligned} \|S_{SPAI-0}\|^2 &= N - \sum_{i=1}^N \left(\frac{a_{ii}^*}{\sum_{j=1}^N \|a_{ij}\|^2} a_{ii} + \frac{a_{ii}}{\sum_{j=1}^N \|a_{ij}\|^2} a_{ii}^* \right) + \sum_{i=1}^N \frac{a_{ii}^*}{\sum_{j=1}^N \|a_{ij}\|^2} \frac{a_{ii}}{\sum_{j=1}^N \|a_{ij}\|^2} \sum_{j=1}^N |a_{ij}|^2 \\ \|S_{SPAI-0}\|^2 &= N - \sum_{i=1}^N \frac{|a_{ii}|^2}{\sum_{j=1}^N \|a_{ij}\|^2} \end{aligned} \quad (82)$$

No SPAI-1, a matriz M tem a mesma esparsidade de A . A complexidade de tempo superior da rotina de determinação da matriz M é $O(Nm_3^3)$. A complexidade de espaço fechada na reprodução da matriz M é $O(Nm)$. A complexidade de tempo fechada de cada iteração é $O(Nm)$. A complexidade de tempo superior na obtenção das matrizes S e Z é $O(Nm_2^2)$ e a complexidade de espaço superior para representar as matrizes S e Z é $O(Nm_2^2)$. A complexidade de tempo superior na determinação da norma Frobenius de S e Z é $O(Nm_2^2)$.

No método SPAI(ϵ), a distribuição de elementos não nulos na M é determinada pelo algoritmo SPAI-Algorithm [12]. Este algoritmo impõe que a norma de cada vetor linha k da matriz S^T seja menor ou igual a ϵ :

$$\|S_k^T\|_2 \leq \epsilon, k = 1, \dots, N \quad (83)$$

$$\|I_k - A^T M_k^T\|_2 \leq \epsilon, k = 1, \dots, N \quad (84)$$

Para cada linha k da matriz M , associada à coluna k da matriz S^T , o algoritmo inicia com um elemento não nulo na diagonal principal e vai inserindo elementos não nulos, segundo um critério de melhor candidato, recalculando os valores dos elementos na linha k da matriz M , até satisfazer a condição (83).

No setup do método SPAI(ϵ), a classe de complexidade inferior é $\Omega(Nm)$ e a superior é $O(N^5)$ e, para representar a matriz M , a classe de complexidade de espaço inferior é $\Omega(N)$ e a superior é $\Omega(N^2)$, dependendo do valor de ϵ e da matriz A .

3.2.4.6 Métodos iterativos não estacionários

Não existem matrizes M , S , Z e $\tilde{A}$ associadas aos métodos iterativos não estacionários, portanto as análises de convergência baseada nos autovetores de S e Z não se aplicam para estes métodos.

A propriedade de relaxar eficientemente os componentes de alta frequência não é garantida nestes métodos, não sendo indicados como suavizador do método Multigrid.

No artigo [24], o método dos Gradientes Conjugados (CG) é sugerido como suavizador do método MG em casos particulares. Neste presente trabalho, a título de complementação de análise, sejam testados os métodos iterativos não estacionários CG, Gradientes Bi-Conjugados (BiCGSTAB) e Gradientes Conjugados Precondicionado pela Fatoração Incompleta de Cholesky (ICCG) como solver e como suavizador do método Multigrid WAMG.

3.3. Conclusão

Geralmente, os métodos iterativos são sensíveis à dominância da diagonal principal da matriz A . Para o caso extremo em que existe pelo menos um elemento nulo na diagonal principal, os métodos como Jacobi, Damped Jacobi, Gauss-Seidel e SOR resultam na divisão por zero; para o método SPAI-0, o valor das incógnitas referentes aos elementos nulos da diagonal principal não converge, sendo indicados os métodos SPAI- n (para $n > 0$) e SPAI(ϵ) para este caso extremo. Para o outro caso extremo, no qual a matriz A é diagonal, os métodos Jacobi e Gauss-Seidel chegam à solução exata em apenas uma iteração. Segundo esta linha de raciocínio, para sistemas de equações cuja diagonal principal da matriz A é pouco dominante, sugere-se os métodos SPAI- n ($n > 0$) e SPAI(ϵ) e para os sistemas de equações cuja

diagonal da matriz A é significativamente dominante, sugere-se os métodos Jacobi e Gauss-Seidel, ou Damped Jacobi e SOR com coeficiente de relaxação próximo a 1.

Os métodos Gauss-Seidel e SOR são sensíveis à ordenação de linhas e colunas. Um critério que auxilie na obtenção da ordenação ótima poderá potencializá-los.

Quanto ao paralelismo, os métodos Jacobi, Damped Jacobi e SPAI permitem paralelismo eficiente, enquanto, os métodos Gauss-Seidel e SOR, exigem processamento serial, a menos de alguma adaptação.

Capítulo 4

TÉCNICA MULTIGRID

Neste capítulo, são apresentados: as notações usuais para representar grids, as formulações correlacionadas aos operadores P e R, a obtenção da matriz do grid grosso, as formulações correlacionadas à correção em grid grosso, o princípio do método MG e o princípio dos métodos GMG, AMG e WAMG.

4.1 Notações de Grid Refinado e Grid Grosso

No método Multigrid, cada nível corresponde a um grid, que por sua vez, equivale a uma malha de discretização do domínio em estudo pelo MEF. Um grid fino corresponde a uma malha refinada e um grid grosso corresponde a uma malha pouco refinada, que podem ser obtidas no processo de refinamento do malhador.

Existem duas notações usuais para diferenciar os grids quanto ao refinamento:

- A primeira notação é usual quando só se tem a necessidade de representar um grid refinado e um grid grosso:

$$A^h x^h = b^h \quad (85)$$

$$A^H x^H = b^H \quad (86)$$

$$A^h e^h = r^h \quad (87)$$

$$A^H e^H = r^H \quad (88)$$

O sistema de equações relativo ao grid refinado é representado por h e o sistema de equações relativo ao grid grosso é representado por H .

- A segunda notação é usual quando existe a necessidade de diferenciar mais de dois grids:

$$A^{2^l h} x^{2^l h} = b^{2^l h} \quad (89)$$

$$A^{2^l h} e^{2^l h} = r^{2^l h} \quad (90)$$

Nesta representação, o grid mais refinado (nível 0) é representado por h , o grid do nível 1 por $2h$, o grid do nível 2 por $4h$ e o grid do nível l por $2^l h$.

4.1. Operador Restrição R

Para transportar (transformar) o resíduo do grid refinado para o grid grosso, utiliza-se a matriz não quadrada R , denominada operador Restrição:

$$r^H = Rr^h \text{ ou } b^H = Rr^h \quad (91)$$

Substituindo $A^H e^H = r^H$ (88), teremos:

$$e^H = A^{H-1} r^H = A^{H-1} Rr^h \quad (92)$$

$$e^H = A^{H-1} R A^h e^h \quad (93)$$

4.2. Operador Prolongamento P

Para transformar o erro do grid grosso e^H em componentes de erro do grid grosso no grid refinado e_h^h , utiliza-se a matriz P , denominada operador Prolongamento:

$$e_h^h = P e^H \quad (94)$$

4.3. Obtenção da Matriz A^H

Existem dois métodos de obtenção da matriz A^H : Pelo MEF e a partir de A^h .

A obtenção de A^H pelo MEF consiste na montagem do sistema de equações a partir do equacionamento do MEF para respectiva malha.

Para deduzir como se dá a obtenção da matriz A^H a partir de A^h , vamos transformar o erro e^H do grid grosso em componente de erro no grid refinado $P e^H$, determinar o respectivo resíduo $A^h P e^H$, transformar este resíduo para o grid grosso $R A^h P e^H$ e obter o respectivo erro no grid grosso $A^{H-1} R A^h P e^H$, que deve ser o próprio erro e^H do grid grosso:

$$\begin{aligned}
 e^H &= A^{H-1} R A^h P e^H \\
 I &= A^{H-1} R A^h P \\
 A^H &= R A^h P
 \end{aligned} \tag{95}$$

Portanto, a segunda forma de obter A^H é a partir da matriz A^h e dos operadores P e R .

4.4. Correção em Grid Grosso

Para corrigir a solução em grid grosso, transforma-se o resíduo do grid refinado para o grid grosso, obtém-se uma solução no grid grosso e adiciona esta solução transformada para o grid refinado à solução anterior:

$$\begin{aligned}
 r_k^H &= R r_k^h \rightarrow x_{k+1}^H \\
 x_{k+1}^h &= x_k^h + P x_{k+1}^H \\
 x_{k+1}^h - x_k^h &= P x_{k+1}^H
 \end{aligned} \tag{96}$$

Substituindo $\Delta x = x_{k+1} - x_k$ (6), teremos:

$$\Delta x_{k+1}^h = x_{k+1}^h - x_k^h = P x_{k+1}^H \tag{97}$$

Se no grid grosso, for obtida a solução exata $x_{k+1}^H = A^{H-1} b_k^H$ (2), onde $b^H = R r^h$ (91) teremos:

$$\Delta x_{k+1}^h = P A^{H-1} r_k^H \tag{98}$$

Substituindo $r^H = R r^h$ (91), teremos:

$$\Delta x_{k+1}^h = P A^{H-1} R r_k^h \tag{99}$$

4.5. Princípio do Método Multigrid

No método Multigrid, o suavizador fica encarregado de relaxar os componentes de erro de alta frequência e os componentes de baixa frequência são corrigidos nos grids mais grosseiros (níveis inferiores). O operador R é obtido de forma a “transferir” (transformar) os componentes de baixa frequência do resíduo para

o grid grosso e o operador P é obtido de forma a transformar o erro do grid grosso em componentes de erro de baixa frequência do grid refinado.

O método Multigrid pode variar com a sequência em que as transformações de grids ocorrem, tendo como exemplos: Full Multigrid, V-Cycle e W-Cycle (fig. 4.1).

Fig. 4.1

Em cada nível do método Multigrid V-Cycle, existem 3 etapas: pré-suavização, correção em grid grosso e pós-suavização. Na pré-suavização, são executadas v_1 passos de suavização (iterações do MIE) e na pós-suavização, são executados v_2 passos de suavização.

No método Multigrid W-Cycle, em cada nível, podem ser feitas mais de uma correção em grid grosso.

No método Full Multigrid, a resolução se dá do grid mais grosseiro até o mais refinado.

Se o operador P for real e transformar o erro do grid grosso em componentes de erro de baixa frequência do grid refinado, o operador $R = P^T$ irá “transferir” (transformar) os componentes de resíduo de baixa frequência para o grid grosso.

$$RA^h P = P^T A^h R^T$$

$$R = P^T \quad (100)$$

Portando, $A^H = RA^hP$ (95), resulta em:

$$A^H = P^T A^h P$$

A condição para que A^H seja simétrica é:

$$A^H = A^{H^T} \rightarrow P^T A^h P = (P^T A^h P)^T$$

$$P^T A^h P = P^T A^{h^T} P$$

$$A^h = A^{h^T}$$

Portanto, para $R = P^T$, se A^h for simétrica, A^H será simétrica, tirando proveito das vantagens de economia de memória na sua representação, além de poder contar com suavizadores mais eficientes em sistemas de equações lineares simétricos.

4.6. Algumas Variações do Método MG

Nesta seção são apresentados: o método Multigrid Geométrico (GMG); o método AMG e o método WAMG, para o qual este trabalho é focado.

4.6.1. Multigrid Geométrico (GMG)

No método GMG, todo o grid grosso corresponde a uma malha menos refinada obtida do malhador do MEF, sendo que todos os nós, existentes no grid grosso, devem existir nos grids mais refinados.

A matriz P é obtida de forma que o componente $\Delta x_{i \in H}^h$ (componente de incremento da solução do nó i encontrado no grid grosso) seja o próprio componente x_i^H (componente da solução deste nó i no grid grosso) e, o componente $\Delta x_{j \notin H}^h$ (componente de incremento da solução do nó j do grid refinado não existente no grid grosso) seja uma média dos componentes $x_{k \in \text{Viz}(j), \in H}^h$ (componentes da solução dos nós vizinhos ao nó j encontrados no grid grosso), o que corresponde a transformar o erro do grid grosso em componentes de erro de baixa frequência do grid refinado.

O inconveniente deste método é de depender dos dados obtidos no processo de refinamento da malha de discretização.

4.6.2. Multigrid Algébrico (AMG)

Visando dispensar a dependência dos dados obtidos no processo de refinamento da malha de discretização, foi desenvolvido o método Multigrid Algébrico.

Como foi apresentado no capítulo 3, a partir da matriz A , é possível construir um grafo correspondente, impondo a existência de um nó i para cada incógnita i e a existência de uma conexão c_{ij} para cada elemento a_{ij} não nulo.

No método AMG, é obtido um grafo a partir da matriz A^h . A partir de um algoritmo de seleção de nós é determinado quais nós existentes no grid refinado pertencerão ao grid grosso.

Com estes elementos é possível determinar a matriz P da mesma forma que é determinado no método GMG.

O algoritmo de seleção dos nós classifica as conexões do grafo do grid refinado em conexões fortes e conexões fracas. A força de uma conexão c_{ij} corresponde ao módulo do elemento a_{ij} . Uma conexão c_{ij} é classificada forte quando a relação entre a sua força e a força da conexão mais forte do nó i for maior que o parâmetro α , conhecido por **Strength Threshold**. A existência de uma conexão forte c_{ij} implica que o nó j é representante do nó i . Um nó pode ser eliminado se existir um representante selecionado no grid grosso.

O inconveniente deste método é que a escolha inadequada do parâmetro α resulta na matriz A^H com ordem e densidade indesejáveis.

4.6.3. Wavelet based Algebraic Multigrid (WAMG)

Visando eliminar o parâmetro **Strength Threshold** e obter uma matriz A^H com ordem e densidade convenientes, foi desenvolvido o método WAMG [1].

No método WAMG, a matriz P é obtida de forma que seus vetores colunas sejam wavelets de baixa frequência, obtendo uma matriz A^H com ordem correspondente à metade da ordem da matriz A^h .

Uma característica dos vetores wavelets é que seus elementos não nulos encontram-se em seqüência.

4.7. Conclusões

A grande diferença entre os métodos AMG e WAMG é que no AMG, o operador P depende da vizinhança dos nós no grafo obtido a partir da matriz e no método WAMG, a matriz P independe desta vizinhança.

Se a matriz do sistema de equações for de banda, os nós associados aos elementos não nulos de cada wavelet tenderão a ser vizinhos, fazendo com que o método WAMG tenda a ter o mesmo resultado que o método AMG.

Capítulo 5

METODOLOGIA DE ANÁLISE E PROPOSIÇÃO DE SUAVIZADORES

Neste capítulo, são apresentados: um modelo matemático de análise do método MG baseado em subespaços; a sugestão de um critério de obtenção de operadores P e R ótimos; uma análise da garantia de presença da característica de suavização nos métodos iterativos estacionários; indicadores propostos para serem confrontados com os resultados experimentais; e propostos de alguns suavizadores com características para-otimizadas.

5.1. Análise de Subespaços do Método Multigrid

Neste modelo proposto de análise do método Multigrid, os operadores P e R definem subespaços. Esta análise tem como propósito verificar o comportamento dos componentes de erro e resíduo relativos a estes subespaços.

Para simplificar a análise, vamos considerar a existência de apenas 2 níveis, grid refinado e grid grosso, verificando o comportamento destes componentes de erro e resíduo no grid refinado após a correção em grid grosso, admitindo solução exata no grid grosso.

5.1.1. Matrizes associadas à correção em grid grosso com solução exata

As matrizes iteração M , iteração de erro S e iteração de resíduo Z , atribuídas aos métodos iterativos estacionários, também podem ser atribuídas à correção em grid grosso com solução exata, permitindo tratar esta correção em grid grosso com solução exata como uma iteração do método iterativo estacionário.

5.1.1.1 Matriz iteração M_{MG}

Utilizando a expressão $\Delta x_{k+1} = M_k r_k$ (5) no método Multigrid, teremos:

$$\Delta x_{k+1}^h = M_{MG} r_k^h$$

Substituindo $\Delta x_{k+1}^h = P A^{H^{-1}} R r_k^h$ (99), teremos:

$$PA^{H^{-1}}Rr_k^h = M_{MG}r_k^h$$

$$M_{MG} = PA^{H^{-1}}R \quad (101)$$

5.1.1.2 Matriz iteração de erro S_{MG}

A matriz iteração de erro S_{MG} é obtida a partir de $S = I - MA$ (8):

$$S_{MG} = (I - M_{MG}A^h)$$

Substituindo $M_{MG} = PA^{H^{-1}}R$ (101), teremos:

$$S_{MG} = (I - PA^{H^{-1}}RA^h) \quad (102)$$

5.1.1.3 Matriz iteração de resíduo Z_{MG}

A matriz iteração de resíduo Z_{MG} é obtida a partir de $Z = I - AM$ (10):

$$Z_{MG} = (I - A^hM_{MG})$$

Substituindo $M_{MG} = PA^{H^{-1}}R$ (101), teremos:

$$Z_{MG} = (I - A^hPA^{H^{-1}}R) \quad (103)$$

5.1.2. Subespaços associados aos Operadores P e R

Vamos definir $\Omega(P)$ como o subespaço correspondente à combinação linear dos vetores coluna da matriz P e $\Omega(\bar{P})$ como subespaço complementar e ortogonal a $\Omega(P)$:

A matriz $\bar{P}$, aqui denominada de complemento de P , possui as seguintes propriedades:

$$P^{T*}\bar{P} = [0] \quad (104)$$

$$P(P^{T*}P)^{-1}P^{T*} + \bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*} = I^h \quad (105)$$

O erro pode ser decomposto em dois componentes: o componente $e_{\Omega(P)}^h$ projeção em $\Omega(P)$ e o componente $e_{\Omega(\bar{P})}^h$ ortogonal a $\Omega(P)$:

$$e^h = e_{\Omega(P)}^h + e_{\Omega(\bar{P})}^h \quad (106)$$

$$\|e^h\|^2 = \|e_{\Omega(P)}^h\|^2 + \|e_{\Omega(\bar{P})}^h\|^2 \quad (107)$$

$$e_{\Omega(P)}^h = P(P^{T*}P)^{-1}P^{T*}e^h \quad (108)$$

$$e_{\Omega(\bar{P})}^h = [I - P(P^{T*}P)^{-1}P^{T*}]e^h \quad (109)$$

Ou ainda:

$$e_{\Omega(P)}^h = P(P^{T*}P)^{-1}P^{T*}e_{\Omega(P)}^h \quad (110)$$

$$e_{\Omega(\bar{P})}^h = [I - P(P^{T*}P)^{-1}P^{T*}]e_{\Omega(\bar{P})}^h \quad (111)$$

Podemos verificar que estes componentes são ortogonais entre si:

$$\langle e_{\Omega(P)}^h, e_{\Omega(\bar{P})}^h \rangle = (e_{\Omega(P)}^h)^{T*} e_{\Omega(\bar{P})}^h$$

Substituindo $e_{\Omega(P)}^h = P(P^{T*}P)^{-1}P^{T*}e_{\Omega(P)}^h$ (110) e $e_{\Omega(\bar{P})}^h = [I - P(P^{T*}P)^{-1}P^{T*}]e_{\Omega(P)}^h$

(111), teremos:

$$\begin{aligned} \langle e_{\Omega(P)}^h, e_{\Omega(\bar{P})}^h \rangle &= [P(P^{T*}P)^{-1}P^{T*}e^h]^{T*} [I - P(P^{T*}P)^{-1}P^{T*}]e^h \\ \langle e_{\Omega(P)}^h, e_{\Omega(\bar{P})}^h \rangle &= [e^{hT*}P(P^{T*}P)^{-1}P^{T*}][I - P(P^{T*}P)^{-1}P^{T*}]e^h \\ \langle e_{\Omega(P)}^h, e_{\Omega(\bar{P})}^h \rangle &= e^{hT*}P(P^{T*}P)^{-1}P^{T*}e^h - e^{hT*}P(P^{T*}P)^{-1}P^{T*}P(P^{T*}P)^{-1}P^{T*}e^h \\ \langle e_{\Omega(P)}^h, e_{\Omega(\bar{P})}^h \rangle &= e^{hT*}P(P^{T*}P)^{-1}P^{T*}e^h - e^{hT*}P(P^{T*}P)^{-1}P^{T*}e^h \\ \langle e_{\Omega(P)}^h, e_{\Omega(\bar{P})}^h \rangle &= 0 \end{aligned} \quad (112)$$

Fig. 5.1 - Representação da decomposição do erro em componentes pertencentes e ortogonais a $\Omega(P)$

Vamos definir $\Omega(R^{T*})$ como subespaço correspondente à combinação linear dos vetores coluna da matriz R^{T*} e $\Omega(\bar{R}^{T*})$ como subespaço ortogonal a $\Omega(R^{T*})$.

A matriz $\bar{R}$, aqui denominada de complemento de R , possui as seguintes propriedades:

$$R\bar{R}^{T*} = [0] \quad (113)$$

$$R^{T*}(RR^{T*})^{-1}R + \bar{R}^{T*}(\bar{R}\bar{R}^{T*})^{-1}\bar{R} = I^h \quad (114)$$

O resíduo pode ser decomposto em dois componentes: o componente e_h^h projeção em $\Omega(R^{T*})$ e o componente e_h^h ortogonal a $\Omega(R^{T*})$:

$$r^h = r_H^h + r_h^h \quad (115)$$

$$\|r^h\|^2 = \|r_H^h\|^2 + \|e_h^h\|^2 \quad (116)$$

$$r_H^h = R^{T*}(RR^{T*})^{-1}Rr^h \quad (117)$$

$$r_h^h = [I - R^{T*}(RR^{T*})^{-1}R]r^h \quad (118)$$

Ou ainda:

$$r_H^h = R^{T*}(RR^{T*})^{-1}Rr_H^h \quad (119)$$

$$r_h^h = [I - R^{T*}(RR^{T*})^{-1}R]r_h^h \quad (120)$$

Podemos verificar que estes componentes são ortogonais entre si:

$$\langle r_H^h, r_h^h \rangle = (r_H^h)^{T*} r_h^h$$

Substituindo $r_H^h = R^{T*}(RR^{T*})^{-1}Rr_H^h$ (119) e $r_h^h = [I - R^{T*}(RR^{T*})^{-1}R]r_h^h$ (120),

teremos:

$$\begin{aligned} \langle r_H^h, r_h^h \rangle &= [R^{T*}(RR^{T*})^{-1}Rr_H^h]^{T*} [I - R^{T*}(RR^{T*})^{-1}R]r_h^h \\ \langle r_H^h, e_h^h \rangle &= [r_H^h]^{T*} R^{T*}(RR^{T*})^{-1}R[I - R^{T*}(RR^{T*})^{-1}R]r_h^h \\ \langle r_H^h, e_h^h \rangle &= r_H^h]^{T*} R^{T*}(RR^{T*})^{-1}Rr_h^h - r_H^h]^{T*} R^{T*}(RR^{T*})^{-1}RR^{T*}(RR^{T*})^{-1}Rr_h^h \\ \langle r_H^h, e_h^h \rangle &= r_H^h]^{T*} R^{T*}(RR^{T*})^{-1}Rr_h^h - r_H^h]^{T*} R^{T*}(RR^{T*})^{-1}Rr_h^h \\ \langle r_H^h, e_h^h \rangle &= 0 \end{aligned} \quad (121)$$

Se substituirmos $r^H = Rr^h$ (91) em $r^h = r_H^h + r_h^h$ (115), teremos:

$$r^H = R(r_H^h + r_h^h)$$

Substituindo $r_H^h = R^{T*}(RR^{T*})^{-1}Rr^h$ (117) e $r_h^h = [I - R^{T*}(RR^{T*})^{-1}R]r^h$ (118),

teremos:

$$\begin{aligned}
 r^H &= RR^{T*} (RR^{T*})^{-1} Rr_H^h + R[I - R^{T*} (RR^{T*})^{-1} R]r_h^h \\
 r^H &= Rr_H^h + [R - RR^{T*} (RR^{T*})^{-1} R]r_h^h \\
 r^H &= Rr_H^h + [R - R]r_h^h \\
 r^H &= Rr_H^h
 \end{aligned} \tag{122}$$

Podemos verificar que a transformação do componente r_h^h para o grid grosso é nula:

$$\begin{aligned}
 Rr_h^h &= R[I - R^{T*} (RR^{T*})^{-1} R]r_h^h \\
 Rr_h^h &= [R - RR^{T*} (RR^{T*})^{-1} R]r_h^h \\
 Rr_h^h &= [R - R]r_h^h \\
 Rr_h^h &= 0
 \end{aligned} \tag{123}$$

Podemos notar que apenas o componente r_H^h , pertencente a $\Omega(R^{T*})$, é “transferido” (transformado) para o grid grosso.

Fig. 5.2- Representação da decomposição do resíduo em componentes pertencentes e ortogonais a $\Omega(R^{T*})$

5.1.3. Análise de casos

A partir da análise de casos particulares é possível identificar propriedades isoladamente.

No caso geral, são apresentadas as formulações utilizadas em todos os casos particulares.

5.1.3.1 Caso Geral

A partir de $e_{k+1} = S e_k$, teremos:

$$e_{k+1}^h = S_{MG} e_k^h$$

Substituindo $e^h = e_{\Omega(P)}^h + e_{\Omega(\bar{P})}^h$ (106), teremos:

$$e_{k+1}^h = S_{MG} (e_{k\Omega(P)}^h + e_{k\Omega(\bar{P})}^h)$$

Substituindo $e_{\Omega(P)}^h = P(P^{T*}P)^{-1}P^{T*}e^h$ (108), teremos:

$$e_{k+1}^h = S_{MG} (P(P^{T*}P)^{-1}P^{T*}e_{k\Omega(P)}^h + e_{k\Omega(\bar{P})}^h)$$

Substituindo $S_{MG} = (I - PA^{H^{-1}}RA^h)$ (102), teremos:

$$\begin{aligned} (6.2) \quad e_{k+1}^h &= (I - PA^{H^{-1}}RA^h)(P(P^{T*}P)^{-1}P^{T*}e_{k\Omega(P)}^h + e_{k\Omega(\bar{P})}^h) \\ e_{k+1}^h &= (I - PA^{H^{-1}}RA^h)P(P^{T*}P)^{-1}P^{T*}e_{k\Omega(P)}^h + (I - PA^{H^{-1}}RA^h)e_{k\Omega(\bar{P})}^h \\ e_{k+1}^h &= P(P^{T*}P)^{-1}P^{T*}e_{k\Omega(P)}^h - PA^{H^{-1}}RA^hP(P^{T*}P)^{-1}P^{T*}e_{k\Omega(P)}^h + (I - PA^{H^{-1}}RA^h)e_{k\Omega(\bar{P})}^h \\ e_{k+1}^h &= P(P^{T*}P)^{-1}P^{T*}e_{k\Omega(P)}^h - PA^{H^{-1}}A^H(P^{T*}P)^{-1}P^{T*}e_{k\Omega(P)}^h + (I - PA^{H^{-1}}RA^h)e_{k\Omega(\bar{P})}^h \end{aligned}$$

Se A^H não for singular, resulta em:

$$\begin{aligned} e_{k+1}^h &= P(P^{T*}P)^{-1}P^{T*}e_{k\Omega(P)}^h - P(P^{T*}P)^{-1}P^{T*}e_{k\Omega(P)}^h + (I - PA^{H^{-1}}RA^h)e_{k\Omega(\bar{P})}^h \\ e_{k+1}^h &= (I - PA^{H^{-1}}RA^h)e_{k\Omega(\bar{P})}^h \end{aligned} \quad (124)$$

O componente $e_{k+1\Omega(P)}^h$, obtida a partir de $e_{\Omega(P)}^h = P(P^{T*}P)^{-1}P^{T*}e^h$ (108), resultará

em:

$$e_{k+1\Omega(P)}^h = P(P^{T*}P)^{-1}P^{T*}e_{k+1}^h$$

Substituindo $e_{k+1}^h = (I - PA^{H^{-1}}RA^h)e_{k\Omega(\bar{P})}^h$ (124), teremos:

$$\begin{aligned} e_{k+1\Omega(P)}^h &= P(P^{T*}P)^{-1}P^{T*}[I - PA^{H^{-1}}RA^h]e_{k\Omega(\bar{P})}^h \\ e_{k+1\Omega(P)}^h &= [P(P^{T*}P)^{-1}P^{T*} - P(P^{T*}P)^{-1}P^{T*}PA^{H^{-1}}RA^h]e_{k\Omega(\bar{P})}^h \\ e_{k+1\Omega(P)}^h &= [P(P^{T*}P)^{-1}P^{T*} - PA^{H^{-1}}RA^h]e_{k\Omega(\bar{P})}^h \\ e_{k+1\Omega(P)}^h &= -PA^{H^{-1}}RA^he_{k\Omega(\bar{P})}^h \end{aligned} \quad (125)$$

O componente $e_{k+1\Omega(\bar{P})}^h$, obtida a partir de $e^h = e_{\Omega(P)}^h + e_{\Omega(\bar{P})}^h$ (106), resultará em:

$$e_{k+1\Omega(\bar{P})}^h = e_{k+1}^h - e_{k+1\Omega(P)}^h$$

Substituindo $e_{\Omega(\bar{P})}^h = [I - P(P^{T*}P)^{-1}P^{T*}]e^h$ (109), teremos:

$$e_{k+1\Omega(\bar{P})}^h = (I - PA^{H^{-1}}RA^h)e_{k\Omega(\bar{P})}^h - e_{k+1\Omega(P)}^h$$

Substituindo $e_{k+1\Omega(P)}^h = -PA^{H^{-1}}RA^he_{k\Omega(\bar{P})}^h$ (125), teremos:

$$e_{k+1\Omega(\bar{P})}^h = (I - PA^{H^{-1}}RA^h)e_{k\Omega(\bar{P})}^h + PA^{H^{-1}}RA^he_{k\Omega(\bar{P})}^h$$

$$e_{k+1\Omega(\bar{P})}^h = e_{k\Omega(\bar{P})}^h \quad (126)$$

Fig. 5.3 - Representação do erro antes e depois da resolução em grid grosso

Se decomposermos o resíduo da iteração k , teremos:

$$r_{k+1}^h = Z_{MG} r_k^h$$

Substituindo $r^h = r_H^h + r_h^h$ (115), teremos:

$$r_{k+1}^h = Z_{MG}(r_{kH}^h + r_{kh}^h)$$

Substituindo $Z_{MG} = (I - A^h PA^{H^{-1}}R)$ (103), teremos:

$$r_{k+1}^h = Z_{MG} r_{kH}^h + (I - A^h PA^{H^{-1}}R) r_{kh}^h$$

Substituindo $Rr_h^h = 0$ (123), teremos:

$$r_{k+1}^h = Z_{MG} r_{kH}^h + r_{kh}^h \quad (127)$$

Substituindo $Z_{MG} = (I - A^h PA^{H^{-1}}R)$ (103), teremos:

$$r_{k+1}^h = (I - A^h PA^{H^{-1}}R) r_{kH}^h + r_{kh}^h \quad (128)$$

O componente r_{k+1H}^h , obtido a partir de $r_H^h = R^{T*}(RR^{T*})^{-1}Rr^h$ (117), resultará em:

$$r_{k+1H}^h = R^{T*} (RR^{T*})^{-1} R r_{k+1}^h$$

Substituindo $r_{k+1}^h = (I - A^h PA^{H-1} R) r_{kH}^h + r_{kh}^h$ (128), teremos:

$$r_{k+1H}^h = R^{T*} (RR^{T*})^{-1} R (I - A^h PA^{H-1} R) r_{kH}^h + R^{T*} (RR^{T*})^{-1} R r_{kh}^h$$

Substituindo $R r_h^h = 0$ (123), teremos:

$$r_{k+1H}^h = R^{T*} (RR^{T*})^{-1} R (I - A^h PA^{H-1} R) r_H^h$$

$$r_{k+1H}^h = [R^{T*} (RR^{T*})^{-1} R - R^{T*} (RR^{T*})^{-1} R A^h PA^{H-1} R] r_H^h$$

Substituindo $A^H = R A^h P$ (95), teremos:

$$r_{k+1H}^h = [R^{T*} (RR^{T*})^{-1} R - R^{T*} (RR^{T*})^{-1} A^H A^{H-1} R] r_H^h$$

$$r_{k+1H}^h = [R^{T*} (RR^{T*})^{-1} R - R^{T*} (RR^{T*})^{-1} R] r_H^h$$

$$r_{k+1H}^h = 0 \quad (129)$$

Podemos concluir que o componente de resíduo r_{k+1H}^h , pertencente a $\Omega(R^{T*})$, será nulo após a correção em grid grosso com solução exata.

O componente r_{k+1h}^h , obtido a partir de $r^h = r_H^h + r_h^h$ (115), resultará em:

$$r_{k+1h}^h = r_{k+1}^h - r_{k+1H}^h$$

Substituindo $r_{k+1H}^h = 0$ (129), teremos:

$$r_{k+1h}^h = r_{k+1}^h$$

Substituindo $r_{k+1}^h = (I - A^h PA^{H-1} R) r_{kH}^h + r_{kh}^h$ (128), teremos:

$$r_{k+1h}^h = (I - A^h PA^{H-1} R) r_{kH}^h + r_{kh}^h \quad (130)$$

Fig. 5.4 - Representação do resíduo antes e depois da resolução em grid grosso

5.1.3.2 Caso o erro pertencer a $\Omega(P)$

Podemos verificar que, se na iteração k só existir o componente de erro pertencente a $\Omega(P)$, ou seja, o componente $e_{k \Omega(\bar{P})}^h$ for nulo e a matriz A^H não for singular, após a correção em grid grosso com solução exata, o erro e_{k+1}^h da iteração $k+1$ será nulo, obtendo a solução exata:

$$e_{k+1}^h = (I - PA^{H^{-1}}RA^h)e_{k \Omega(\bar{P})}^h$$

Para $e_{k \Omega(\bar{P})}^h = 0 \Rightarrow e_{k+1}^h = 0$ (131)

Fig. 5.5 - Representação do erro antes e depois da resolução em grid grosso, para $e_{k \Omega(\bar{P})}^h = 0$.

5.1.3.3 Caso o erro seja ortogonal a $\Omega(P)$

No caso em que o componente $e_{k \Omega(P)}^h$ seja nulo, após a correção em grid grosso com solução exata, o componente ortogonal a $\Omega(P)$ permanecerá inalterado e aparecerá um componente de erro pertencente a $\Omega(P)$, caracterizando uma divergência.

Para verificação, vamos utilizar $\|e^h\|^2 = \|e_{\Omega(P)}^h\|^2 + \|e_{\Omega(\bar{P})}^h\|^2$ (107):

$$e_{k\Omega(P)}^h = 0 \Rightarrow \|e_k^h\|^2 = \|e_{k\Omega(\bar{P})}^h\|^2 \quad (132)$$

$$\|e_{k+1}^h\|^2 = \|e_{k+1\Omega(P)}^h\|^2 + \|e_{k+1\Omega(\bar{P})}^h\|^2$$

Substituindo $e_{k+1\Omega(P)}^h = -PA^{H^{-1}}RA^he_{k\Omega(\bar{P})}^h$ (125) e $e_{k+1\Omega(\bar{P})}^h = e_{k\Omega(\bar{P})}^h$ (126), teremos:

$$\|e_{k+1}^h\|^2 = \|-PA^{H^{-1}}RA^he_{k\Omega(\bar{P})}^h\|^2 + \|e_{k\Omega(\bar{P})}^h\|^2$$

Como neste caso $e_{k\Omega(\bar{P})}^h = e_k^h$:

$$\|e_{k+1}^h\|^2 = \|-PA^{H^{-1}}RA^he_{k\Omega(\bar{P})}^h\|^2 + \|e_k^h\|^2$$

$$\|e_{k+1}^h\| \geq \|e_k^h\|$$

Fig. 5.6 - Representação do erro antes e depois da resolução em grid grosso, para $e_{k\Omega(P)}^h = 0$.

Para que o componente de erro $e_{k+1\Omega(P)}^h$ seja nulo após a correção em grid grosso, o componente de resíduo r_{kH}^h pertencente a $\Omega(R^{T*})$ antes da correção em grid grosso deve ser nulo:

$$R^{T*}(RR^{T*})^{-1}RAP(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*} = [0]$$

Portanto, a norma Frobenius da matriz projeção da matriz A nos subespaços $\Omega(\bar{P})$ e $\Omega(R^{T*})$ serve como indicador desta característica indesejável:

$$Proj_{Hh} = \frac{\|R^{T*}(RR^{T*})^{-1}RAP(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}\|_F^2}{\|A\|_F^2} \quad (133)$$

Vamos denominar que quando o valor deste indicador for baixo, o operador Restrição terá boa compatibilidade com o operador Prolongamento.

Antes da correção em grid grosso, é inesperada a situação, na qual, o componente de erro ortogonal a $\Omega(P)$ seja maior que o componente de erro pertencente a $\Omega(P)$, pois após a pré-suavização, o componente de erro ortogonal a $\Omega(P)$ tende a ser menor que o componente de erro pertencente a $\Omega(P)$.

Supõe-se que quanto maior o valor do indicador sugerido, maior deverá ser a quantidade de passos na pré-suavização.

O ângulo θ , representado na fig. 5.6, representa um ângulo limite da convergência do erro na correção em grid grosso, seu valor depende do componente de erro $e_{k\Omega(P)}^h$, mas vamos sugerir como estimativa da média de sua tangente:

$$tg(\theta) = \frac{\|R^{T*}(RR^{T*})^{-1}RAP(P^{T*}P)^{-1}P^{T*}\|_F}{\|R^{T*}(RR^{T*})^{-1}R\bar{A}\bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}\|_F} \quad (134)$$

Supõe-se que o valor desta estimativa também possa servir como indicador de compatibilidade entre o operador Restrição e o operador Prolongamento.

5.1.3.4 Caso o resíduo pertencer a $\Omega(R^{T*})$

Podemos verificar que, se na iteração k só existir o componente de resíduo pertencente a $\Omega(R^{T*})$, ou seja, o componente $r_{k h}^h$ for nulo, após a resolução em grid grosso, o resíduo r_{k+1}^h da iteração $k+1$ só terá o componente ortogonal a $\Omega(R^{T*})$:

Para o componente pertencente a $\Omega(R^{T*})$, fica valendo $r_{k+1 H}^h = 0$ (129).

$$r_{k+1 H}^h = 0$$

Substituindo $r_{k h}^h = 0$ em $r_{k+1 h}^h = (I - A^h P A^{H-1} R) r_{k H}^h + r_{k h}^h$ (130), teremos:

$$r_{k h}^h = 0 \Rightarrow r_{k+1 h}^h = (I - A^h P A^{H-1} R) r_{k H}^h \quad (135)$$

Fig. 5.7 - Representação do resíduo antes e depois da resolução em grid grosso, para $r_{kh}^h = 0$.

5.1.3.5 Caso o resíduo seja ortogonal a $\Omega(R^{T*})$

No caso em que só existir o componente de resíduo ortogonal a $\Omega(R^{T*})$ na iteração k , ou seja, o componente r_{kH}^h seja nulo, a transformação do resíduo para o grid grosso resultará no resíduo do grid grosso r_k^H nulo, não alterando a solução na resolução em grid grosso:

Para o componente pertencente a $\Omega(R^{T*})$, fica valendo $r_{k+1H}^h = 0$ (129).

$$r_{k+1H}^h = 0$$

Substituindo $r_{k+1H}^h = 0$ em $r_{k+1h}^h = (I - A^h P A^{H-1} R) r_{kH}^h + r_{kh}^h$ (130), teremos:

$$\text{Para } r_{kH}^h = 0 \Rightarrow r_{k+1h}^h = r_{kh}^h \quad (136)$$

Fig. 5.8 - Representação do resíduo antes e depois da resolução em grid grosso, para $r_{kH}^h = 0$.

A situação na qual, o componente de resíduo ortogonal a $\Omega(R^{T*})$ é inesperada, pois após a pré-suavização, o componente de resíduo pertencente a $\Omega(R^{T*})$ tende a ser maior ao componente de resíduo ortogonal a $\Omega(R^{T*})$.

5.1.4. Base de Transformação Ortonormal

Se os vetores coluna da matriz P forem ortonormais, vale as seguintes propriedades:

$$\overline{P}^{T*} \overline{P} = I \quad (137)$$

$$\|\overline{P}Q\|_F = \|Q\|_F \quad (138)$$

Onde Q é qualquer matriz

Se as matrizes P , $\overline{P}$, R^{T*} e $\overline{R}^{T*}$ corresponderem a bases ortonormais, valem:

$$P(P^{T*}P)^{-1}P^{T*} = PP^{T*} \quad (139)$$

$$\overline{P}(\overline{P}^{T*}\overline{P})^{-1}\overline{P}^{T*} = \overline{P}\overline{P}^{T*} \quad (140)$$

$$R^{T*}(RR^{T*})^{-1}R = R^{T*}R \quad (141)$$

$$\overline{R}^{T*}(\overline{R}\overline{R}^{T*})^{-1}\overline{R} = \overline{R}^{T*}\overline{R} \quad (142)$$

No método WAMG, os operadores P e R^{T*} possuem vetores colunas ortonormais, simplificando a determinação de projeções de erro e resíduo.

As matrizes P e $\overline{P}$ com filtro Haar (Filtro baseado em Wavelets) correspondem a:

$$P = \begin{bmatrix} \frac{\sqrt{2}}{2} & & & \\ & \frac{\sqrt{2}}{2} & & \\ & & \ddots & \\ & & & \frac{\sqrt{2}}{2} \end{bmatrix} \quad \overline{P} = \begin{bmatrix} \frac{\sqrt{2}}{2} & & & \\ & \frac{\sqrt{2}}{2} & & \\ & & \ddots & \\ & & & \frac{\sqrt{2}}{2} \\ & & & & 1 \end{bmatrix}$$

5.1.5. Operadores P e R ideais

De certa forma, podemos afirmar que no método Multigrid V-Cycle, o suavizador deve relaxar os componentes de erro ortogonais a $\Omega(P)$ na pré-suavização; a resolução em grid grosso deve relaxar os componentes de erro

pertencentes a $\Omega(P)$ e os componentes de resíduo pertencentes a $\Omega(R^{T*})$ e, finalmente, o suavizador deve relaxar os componentes de resíduo ortogonais a $\Omega(R^{T*})$ na pós-suavização.

Para se obter a melhor taxa de convergência por iteração no método Multigrid, o ideal seria que o subespaço $\Omega(\bar{P})$ agrupasse os erros para os quais o método iterativo (suavizador do Multigrid) tenha melhor convergência, e o subespaço $\Omega(\bar{R}^{T*})$ agrupasse os resíduos para os quais o método iterativo tenha melhor convergência.

Baseado neste princípio, todos os métodos iterativos, para os quais existem um subespaço que agrupe os erros de melhor convergência e outro que agrupe os resíduos de melhor convergência, são candidatos a método iterativo a ser utilizado no método Multigrid.

Vamos representar o subespaço ortogonal ao subespaço que contenham os erros de melhor convergência como $\Omega(P_{Ideais})$ e o subespaço ortogonal ao subespaço que contenham os resíduos de melhor convergência de $\Omega(R_{Ideais}^{T*})$.

Embora existam infinitos operadores P e R ideais, é improvável a viabilidade na determinação de um par destes operadores ideais. Vamos considerar que P define um subespaço satisfatório se o subespaço $\Omega(P)$ for próximo do subespaço $\Omega(P_{Ideais})$ e R define um subespaço satisfatório se o subespaço $\Omega(R^{T*})$ for próximo do subespaço $\Omega(R_{Ideais}^{T*})$.

Devido ao fato de se valer do princípio da superposição (propriedade distributiva do erro e do resíduo), conclui-se que os métodos iterativos estacionários garantem a existência de um subespaço que agrupe os erros de melhor convergência e outro que agrupe os resíduos de melhor convergência. Para se obter a melhor taxa de convergência por iteração no método MG suavizado pelo método iterativo estacionário, o ideal seria que:

- a combinação linear dos vetores coluna da matriz P definisse um subespaço ortogonal ao subespaço $\Omega(V_{\lambda Mins}^S)$ (-obtido pela combinação linear dos autovetores da matriz iteração de erro S do suavizador com menores módulos de autovalor), e;

- a combinação linear dos vetores coluna da Matriz R^{T*} definisse um subespaço ortogonal ao subespaço $\Omega(V_{\lambda Mins}^Z)$ (-obtido pela combinação linear dos autovetores da matriz iteração de resíduo Z do suavizador com menores módulos de autovalor).

$$\Omega(P_{Ideais}) = \perp \Omega(V_{\lambda Mins}^S) \quad (143)$$

$$\Omega(R_{Ideais}^{T*}) = \perp \Omega(V_{\lambda Mins}^Z) \quad (144)$$

Segundo esta linha de raciocínio, os operadores P e R , que resultam na melhor convergência por iteração no método Multigrid, dependem respectivamente das matrizes S e Z do método iterativo estacionário utilizado.

5.1.6. Operadores P e R ótimos

Para que os operadores P e R sejam ótimos, ou seja, o método Multigrid encontre a solução desejada no menor tempo possível, a obtenção destes operadores deve demandar baixo custo computacional, suas matrizes devem ser esparsas e devem definir subespaços satisfatórios.

5.1.7. Sugestão de Obtenção de Operadores P e R Para-Ótimos

Esta sugestão consiste em obter a matriz P , de forma que o subespaço $\Omega(\bar{P})$ tende a conter os componentes de erro de melhor relaxação na pré-suavização e em obter a matriz R , de forma a reduzir o limite de convergência de erro na correção em grid grosso.

Após K passos de suavização na pré-suavização, o componente de erro $e_{K \Omega(\bar{P})}$ ortogonal a $\Omega(P)$ corresponde a:

$$\begin{aligned} e_{K \Omega(\bar{P})} &= \bar{P}(\bar{P}^{T*} \bar{P})^{-1} \bar{P}^{T*} e_K \\ e_{K \Omega(\bar{P})} &= \bar{P}(\bar{P}^{T*} \bar{P})^{-1} \bar{P}^{T*} S_{eq} e_0 \\ e_{K \Omega(\bar{P})} &= \bar{P}(\bar{P}^{T*} \bar{P})^{-1} \bar{P}^{T*} S^K e_0 \end{aligned}$$

A matriz P ótima é obtida minimizando o componente de erro $e_{K \Omega(\bar{P})}$:

$$\min(\|\bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}S^K\|_F) \rightarrow P_{Ot} \quad (145)$$

A matriz R é obtida minimizando o limite de convergência de erro na correção em grid grosso, minimizando o indicador (133):

$$\min(\|R^{T*}(RR^{T*})^{-1}RA\bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}\|_F) \rightarrow R_{Ot} \quad (146)$$

Para que seja viável a determinação destas matrizes P e R ótimas, é sugerido que tanto a matriz P quanto a matriz R^{T*} tenham no máximo um elemento não nulo por linha e que os vetores colunas sejam ortonormais, tendo como exemplo:

$$P = \begin{bmatrix} \sin \theta_1 & & & & \\ & \sin \theta_2 & & & \\ \cos \theta_1 & & & & \\ & & \dots & & \\ & & & \sin \theta_{J=N/2} & \\ & & & \cos \theta_{J=N/2} & \\ & & & & \cos \theta_2 \end{bmatrix} \quad \bar{P} = \begin{bmatrix} \cos \theta_1 & & & & \\ & \cos \theta_2 & & & \\ -\sin \theta_1 & & & & \\ & & \dots & & \\ & & & \cos \theta_{J=N/2} & \\ & & & -\sin \theta_{J=N/2} & \\ & & & & -\sin \theta_2 \\ & & & & & 1 \end{bmatrix}$$

Neste caso, a equação (145) resulta em:

$$\min(\|\bar{P}^{T*}S^K\|_F) \rightarrow P_{Ot}$$

A equação (146) resulta em:

$$\min(\|RA\bar{P}\|_F) \rightarrow R_{Ot} \quad (147)$$

Os elementos não nulos do vetor coluna das matrizes P_{Ot} e $\bar{R}_{Ot}$ tendem a corresponder a nós vizinhos ou a nós que possuem vinhos em comuns.

Supõe-se que o algoritmo, que efetue estas tentativas de otimização tenha complexidades de tempo e espaço aceitáveis, mas sensível à densidade das matrizes S^K e $A\bar{P}$.

5.2. Análise da Característica de Suavização dos MIEs

A característica de suavização dos métodos iterativos estacionários é observada experimentalmente, não se encontrando teoremas que atestem esta

característica para todos os casos. Para por a garantia de existência desta característica a prova, vamos utilizar dois sistemas de equações: $Ax = b$ e $A^{\wedge}x^{\wedge} = b^{\wedge}$, sendo que o sistema de equações $A^{\wedge}x^{\wedge} = b^{\wedge}$ é obtido a partir do sistema de equações $Ax = b$ através da seguinte transformação:

$$Ax = b$$

$$ATT^{-1}x = b$$

$$TATT^{-1}x = Tb$$

$$[TAT][T^{-1}x] = [Tb] \equiv A^{\wedge}x^{\wedge} = b^{\wedge}$$

Onde:

$$A^{\wedge} = TAT \quad (148)$$

$$x^{\wedge} = T^{-1}x \quad (149)$$

$$b^{\wedge} = Tb \quad (150)$$

A matriz transformação T corresponde a:

$$T = T^{-1} = \begin{bmatrix} 1 & & & & & \\ & -1 & & & & \\ & & 1 & & & \\ & & & -1 & & \\ & & & & \dots & \\ & & & & & 1 \\ & & & & & & -1 \end{bmatrix} \quad (151)$$

Para esta matriz T , valem as seguintes igualdades:

$$D^{\wedge} = TDT$$

$$L^{\wedge} = TLT$$

$$U^{\wedge} = TUT$$

Onde D é a matriz diagonal principal da matriz A , $D^{\wedge}$ é a matriz diagonal principal da matriz $A^{\wedge}$, L é a matriz triangular inferior da matriz A , $L^{\wedge}$ é a matriz triangular inferior da matriz $A^{\wedge}$, U é a matriz triangular superior da matriz A e $U^{\wedge}$ é a matriz triangular superior da matriz $A^{\wedge}$.

Se for adotado o filtro Haar de baixa frequência como critério de definição do subespaço dos componentes de baixa frequência, o subespaço dos componentes de baixa frequência será o mesmo para os dois sistemas de equações e se a ordem dos sistemas de equações for par, valem as seguintes igualdades:

$$\overline{P}_{Haar} = TP_{Haar}$$

$$P_{Haar} = T\overline{P}_{Haar}$$

Portanto, para este critério de definição do subespaço dos componentes de baixa frequência, a matriz T transforma os componentes de alta frequência em componentes de baixa frequência e os componentes de baixa frequência em componentes de alta frequência, mantendo o módulo.

A partir de $S_{DJ} = (1 - \omega)I + \omega D^{-1}(L + U)$ (33), teremos:

$$S_{DJ}^* = (1 - \omega)I + \omega D^{-1}(L^* + U^*)$$

$$S_{DJ}^* = (1 - \omega)TIT + \omega(TDT)^{-1}(TLT + TUT)$$

$$S_{DJ}^* = (1 - \omega)TIT + \omega TD^{-1}TT(L + U)T$$

$$S_{DJ}^* = (1 - \omega)TIT + \omega TD^{-1}(L + U)T$$

$$S_{DJ}^* = T[(1 - \omega)I + \omega D^{-1}(L + U)]T$$

$$S_{DJ}^* = TS_{DJ}T$$

Da mesma forma, a partir de $Z_{DJ} = (1 - \omega)I + \omega(L + U)D^{-1}$ (34), vale:

$$Z_{DJ}^* = TZ_{DJ}T$$

A partir de $S_{FSOR} = -(\frac{1}{\omega}D - L)^{-1}[(1 - \frac{1}{\omega})D - U]$ (64), teremos:

$$S_{FSOR}^* = -(\frac{1}{\omega}D^* - L^*)^{-1}[(1 - \frac{1}{\omega})D^* - U^*]$$

$$S_{FSOR}^* = -(\frac{1}{\omega}TDT - TLT)^{-1}[(1 - \frac{1}{\omega})TDT - TUT]$$

$$S_{FSOR}^* = -T(\frac{1}{\omega}D - L)^{-1}TT[(1 - \frac{1}{\omega})D - U]T$$

$$S_{FSOR}^* = -T(\frac{1}{\omega}D - L)^{-1}[(1 - \frac{1}{\omega})D - U]T$$

$$S_{FSOR}^* = TS_{FSOR}T$$

Da mesma forma, a partir de $Z_{FSOR} = -[(1 - \frac{1}{\omega})D - U](\frac{1}{\omega}D - L)^{-1}$ (65), vale:

$$Z_{FSOR}^* = TZ_{FSOR}T$$

Podemos verificar que para todos os métodos iterativos estacionários citados, valem as seguintes igualdades:

$$S^{\wedge} = TST$$

$$Z^{\wedge} = TZT$$

Como $e = S^k_0 e$, teremos:

$$e^{\wedge}_k = (S^{\wedge})^k e^{\wedge}_0 = (TST)^k e^{\wedge}_0$$

$$e^{\wedge}_k = TS^k T e^{\wedge}_0$$

Para $e = \bar{x}$, teremos:

$$e^{\wedge}_k = TS^k T \bar{x}$$

$$e^{\wedge}_k = TS^k TT^{-1} \bar{x}$$

$$e^{\wedge}_k = TS^k \bar{x}$$

$$e^{\wedge}_k = TS^k e_0$$

$$e^{\wedge}_k = T e_k$$

O mesmo desenvolvimento pode ser repetido para o resíduo, resultando em:

$$r^{\wedge}_k = T r_k$$

Se para o sistema de equações $Ax = b$, um destes métodos iterativos estacionários apresentar boa relaxação para os componentes de alta freqüência e péssima relaxação para os componentes de baixa freqüência, este mesmo método iterativo apresentará boa relaxação para os componentes de baixa freqüência e péssima relaxação para os componentes de alta freqüência para o sistema de equações $A^{\wedge}x^{\wedge} = b^{\wedge}$, implicando que este método iterativo não apresenta a característica de suavização para o sistema de equações $A^{\wedge}x^{\wedge} = b^{\wedge}$, quando utilizado o filtro Haar de baixa freqüência como definição do subespaço dos componentes de baixa freqüência.

O subespaço dos componentes de baixa frequência não tem uma definição rígida, pois a mesma depende de critério adotado, sendo que nos métodos GMG e AMG clássicos, este subespaço depende da vizinhança dos nós no grid e no método WAMG, depende do filtro escolhido.

O fato da matriz T garantir a mesma vizinhança de nós entre o grafo obtido a partir da matriz A e o grafo obtido a partir da matriz A' , mantendo as mesmas forças de conexão, faz do operador P tender ser o mesmo nos dois sistemas de equações por critério de vizinhança. O subespaço dos componentes de baixa frequência definido pelo Filtro Haar tende a coincidir com o subespaço de baixa frequência definido pelo critério de vizinhança, sendo razoável supor que a prova baseada no Filtro Haar sirva também para o operador P obtido por critério de vizinhança.

Esta prova põe em dúvida a estratégia do método MG em tirar proveito da característica de suavização dos métodos iterativos estacionários.

No caso do Filtro Haar, é possível afirmar que para todo sistema de equações $Ax=b$, para o qual um dos métodos iterativos estacionários que satisfazem a condição $S'=TST$ apresentar ótima característica de suavização, existe outro sistema de equações $A'x'=b'$, para o qual este método iterativo estacionário apresentará péssima característica de suavização.

5.3. Suavizadores Propostos

As características do método iterativo estacionário que o torna bom suavizador no método MG são a característica de convergência do MIE como solver e a característica de suavização.

Através da tentativa de otimização baseada na minimização da norma Frobenius pelo método dos mínimos quadrados, são propostos suavizadores com mesmo grau de liberdade de otimização, mas diferenciados pela característica otimizada, visando comparar suavizador com melhor característica de convergência ao suavizador com melhor característica de suavização.

5.3.1. Características para-otimizadas

Além de apresentar boa convergência quando solver, quando suavizador do método MG, o método iterativo estacionário deve minimizar o componente de erro

de alta freqüência na pré-suavização e apresentar boa convergência para resíduos com predominância de componentes de alta freqüência na pós-suavização.

5.3.1.1 Minimizando o componente $e_{K\Omega(\bar{P})}$

Após K iterações, o componente de erro $e_{K\Omega(\bar{P})}$ ortogonal a $\Omega(P)$ corresponde

a:

$$e_{K\Omega(\bar{P})} = \bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}e_K$$

$$e_{K\Omega(\bar{P})} = \bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}S_{eq}e_0$$

$$e_{K\Omega(\bar{P})} = \bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}S^Ke_0$$

Se a matriz $\bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}S^K$ for nula, o componente de erro $e_{K\Omega(\bar{P})}$ será nulo, portanto, a minimização da norma Fobenius de $\bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}S^K$ tenderá a minimizar o componente de erro $e_{K\Omega(\bar{P})}$:

$$\min(\|\bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}S^K\|_F)$$

5.3.1.2 Limitando a Divergência do componente de erro $e_{K\Omega(P)}$

Ao minimizar o componente de erro $e_{K\Omega(\bar{P})}$, o componente $e_{K\Omega(P)}$ tenderá a divergência. Se ocorrer divergência do componente de erro pertencente a $\Omega(P)$, a convergência do método MG poderá ficar comprometida. Para contornar este problema, pode se aumentar as iterações nos níveis mais grosseiros utilizando o W-Cycle e/ou limitar a divergência deste componente.

Após K iterações, o erro e_K corresponde a:

$$e_K = S_{eq}e_0$$

$$e_K = S^Ke_0$$

Se a matriz S^K for nula, o componente de erro e_K será nulo, portanto, a minimização da norma Fobenius de S^K tenderá a minimizar o erro e_K :

$$\min(\|S^K\|_F)$$

Esta tentativa de otimização corresponde a melhorar a característica de convergência do MIE como solver.

5.3.1.3 Melhorando a Convergência do erro

Na pós-suavização, deseja-se minimizar o erro, para tanto, propõe-se o seguinte critério:

$$\min(\|S^K\|_F)$$

5.3.1.4 Melhorando a Convergência de resíduo na Pós-suavização

Após a correção em grid grosso, o componente de resíduo $r_{0\Omega(\bar{R}^{T*})}$ tende a ser maior que o componente $r_{0\Omega(\bar{R}^{T*})}$. Este critério visa melhorar a relaxação do resíduo para esta situação.

Após K iterações, o resíduo r_K corresponde a:

$$r_K = Z^K r_0$$

Como inicialmente o resíduo tem como predominância o componente $r_{0\Omega(\bar{R}^{T*})}$, podemos considerar que $r_0 \approx r_{0\Omega(\bar{R}^{T*})}$, resultando em:

$$r_K \cong Z^K r_{0\Omega(\bar{R}^{T*})}$$

$$r_K \cong Z^K \bar{R}^{T*} (\bar{R} \bar{R}^{T*})^{-1} \bar{R} r_0$$

Se a matriz $Z^K \bar{R}^{T*} (\bar{R} \bar{R}^{T*})^{-1} \bar{R}$ for nula, o resíduo r_K será nulo, portanto, a minimização da norma Fobenius de $Z^K \bar{R}^{T*} (\bar{R} \bar{R}^{T*})^{-1} \bar{R}$ tenderá a minimizar o resíduo

$$r_K :$$

$$\min(\|Z^K \bar{R}^{T*} (\bar{R} \bar{R}^{T*})^{-1} \bar{R}\|_F)$$

5.3.1.5 Limitando a Divergência do componente de resíduo $r_{0\Omega(\bar{R}^{T*})}$

Ao maximizar a relaxação do resíduo utilizando o critério anterior, se o componente $r_{0\Omega(\bar{R}^{T*})}$ não for predominante, poderá ocorrer divergência, podendo comprometer a convergência do método MG. Para contornar este problema, pode-se aumentar as iterações nos níveis mais grosseiros utilizando o W-Cycle, e/ou limitar esta divergência com o seguinte critério:

$$\min(\|Z^K\|_F)$$

5.3.2. Combinando os critérios

Para combinar os critérios, propõe-se a seguinte ponderação:

$$\min(\alpha_1 \|\bar{P} (\bar{P}^{T*} \bar{P})^{-1} \bar{P}^{T*} S^K\|_F^2 + \alpha_2 \|S^K\|_F^2 + \alpha_3 \|Z^K \bar{R}^{T*} (\bar{R} \bar{R}^{T*})^{-1} \bar{R}\|_F^2 + \alpha_4 \|Z^K\|_F^2)$$

Onde α_1 , α_2 , α_3 e α_4 são pesos de ponderação.

Na pré-suavização, deve-se atribuir valores significativos para α_1 , forçando o método iterativo estacionário resultar no menor componente de erro $e_{K\Omega(\bar{P})}$ e atribuir valores razoáveis para α_2 , evitando a divergência do componente de erro $e_{\Omega(P)}$.

Na pós-suavização, deve-se atribuir valores significativos para α_2 , forçando o método iterativo estacionário ter boa convergência de erro, atribuir valores significativos para α_3 , forçando o método iterativo estacionário ter boa convergência do resíduo, quando este tiver predominância do componente $r_{\Omega(\bar{R}^{T*})}$ e atribuir valores razoáveis para α_4 , evitando a divergência do resíduo, caso não haja predominância do componente $r_{\Omega(\bar{R}^{T*})}$.

5.3.3. WAMG com Filtro Haar em Sistemas de Equações Simétricos

Pelo fato dos vetores coluna da matriz P serem ortonormais e $R = P^T$, valem as seguintes igualdades:

$$P(P^{T*}P)^{-1}P^{T*} = R^{T*}(RR^{T*})^{-1}R = PP^T$$

$$\bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*} = \bar{R}^{T*}(\bar{R}\bar{R}^{T*})^{-1}\bar{R} = \bar{P}\bar{P}^T$$

$$\|\bar{P}\bar{P}^T S^K\|_F = \|\bar{P}^T S^K\|_F$$

No caso em que a matriz A seja simétrica, se for imposto que a matriz iteração M do suavizador seja também simétrica, vale a seguinte igualdade:

$$\|S^K\|_F = \|Z^K\|_F$$

Vamos denominar o suavizador que possui esta característica de simetria de suavizador simétrico.

O critério de tentativa de otimização composto resulta em:

$$\min[(\alpha_1 + \alpha_3)\|\bar{P}^T S^K\|_F^2 + (\alpha_2 + \alpha_4)\|S^K\|_F^2]$$

Ou

$$\min(\beta_1\|\bar{P}^T S^K\|_F^2 + \beta_2\|S^K\|_F^2)$$

Onde β_1 e β_2 são pesos de ponderação.

Na pré-suavização, o peso β_1 força o componente de erro $e_{K\Omega(\bar{P})}$ ser menor que o componente $e_{K\Omega(P)}$ e o peso β_2 evita a divergência do componente de erro $e_{\Omega(P)}$.

Na pós-suavização, o peso β_1 força o método iterativo estacionário ter boa convergência de resíduo, quando este tiver predominância do componente $r_{\Omega(\bar{R}^{T*})}$ e o peso β_2 força o método iterativo estacionário ter boa convergência de erro e evita a divergência do resíduo, caso não haja predominância do componente $r_{\Omega(\bar{R}^{T*})}$.

Aumentar um peso ou outro corresponde em melhorar a característica de suavização ou a característica de convergência.

5.3.4. Suavizadores com Características Para-Otimizadas

Experimentalmente, são testados três tipos de suavizadores simétricos para-otimizados para o método WAMG com filtro Haar para sistemas de equações com matriz A simétrica.

Para que os suavizadores para-otimizados possam ser comparados, são testados apenas suavizadores para-otimizados com o grau de liberdade de otimização igual a N.

5.3.4.1 Suavizadores com $\beta_1 = 0$

Neste caso, visa-se otimizar a característica de convergência de erro e resíduo do MIE:

Ao anular o peso β_1 , o critério de tentativa de otimização se resume em:

$$\min(\|S^K\|_F)$$

Esta tentativa de otimização vale tanto para sistemas de equações simétricos como assimétricos.

Os métodos SPAI seguem este critério de tentativa de otimização para um passo de suavização.

Na comparação experimental de suavizadores para-otimizados simétricos com grau de liberdade de otimização igual a N, está sendo utilizado o SPAI-0.

Aplicando este critério para o método Damped Jacobi com um passo de suavização, teremos:

$$\begin{aligned} \min_{\omega} \|S_{DJ}\|_F &\rightarrow \omega_{DJ\acute{O}timo(K=1, \beta_1=0)} \\ \|S\|_F^2 &= \sum_i \left(\sum_{j < i} \left| \omega \frac{a_{ij}}{a_{ii}} \right|^2 + |1 - \omega|^2 \right) \\ \|S\|_F^2 &= \sum_i \left(\sum_{j < i} \left| \omega \frac{a_{ij}}{a_{ii}} \right|^2 + 1 - \omega - \omega^* + |\omega|^2 \right) \\ \|S\|_F^2 &= \sum_i \left(\sum_j \left| \omega \right|^2 \left| \frac{a_{ij}}{a_{ii}} \right|^2 + 1 - \omega - \omega^* \right) \\ \|S\|_F^2 &= |\omega|^2 \sum_i \sum_j \left| \frac{a_{ij}}{a_{ii}} \right|^2 + \sum_i (1 - \omega - \omega^*) \end{aligned} \tag{152}$$

$$\begin{aligned}
 \frac{\partial \|S\|_F^2}{\partial \omega} = 0 &\Rightarrow \omega^* \sum_i \sum_j \left| \frac{a_{ij}}{a_{ii}} \right|^2 + \omega \frac{\partial \omega^*}{\partial \omega} \sum_i \sum_j \left| \frac{a_{ij}}{a_{ii}} \right|^2 - N - N \frac{\partial \omega^*}{\partial \omega} = 0 \\
 \omega^* \sum_i \sum_j \left| \frac{a_{ij}}{a_{ii}} \right|^2 - N &= - \frac{\partial \omega^*}{\partial \omega} (\omega \sum_i \sum_j \left| \frac{a_{ij}}{a_{ii}} \right|^2 - N) \\
 \omega \sum_i \sum_j \left| \frac{a_{ij}}{a_{ii}} \right|^2 - N &= 0 \\
 \omega_{DJ\acute{O}timo(K=1, \beta_1=0)} &= \frac{N}{\sum_{i=1}^N \sum_{j=1}^N \left| \frac{a_{ij}}{a_{ii}} \right|^2} \quad (153)
 \end{aligned}$$

Se $\omega_{DJ\acute{O}timo(K=1, \beta_1=0)}$ for igual a um, implica que a matriz A é diagonal e só será igual a 0 se existir elementos nulos na diagonal principal. Este coeficiente serve como indicador de dominância da diagonal principal.

A norma Frobenius da matriz S para $\omega_{DJ\acute{O}timo(K=1, \beta_1=0)}$ resulta em:

De $\|S\|_F^2 = |\omega|^2 \sum_i \sum_j \left| \frac{a_{ij}}{a_{ii}} \right|^2 + \sum_i (+1 - \omega - \omega^*)$ (152), temos

$$\begin{aligned}
 \|S(\omega_{DJ\acute{O}timo(K=1, \beta_1=0)})\|_F^2 &= \omega_{DJ\acute{O}timo(K=1, \beta_1=0)} \frac{N}{\sum_{i=1}^N \sum_{j=1}^N \left| \frac{a_{ij}}{a_{ii}} \right|^2} \sum_i \sum_j \left| \frac{a_{ij}}{a_{ii}} \right|^2 + N - 2N\omega_{DJ\acute{O}timo(K=1, \beta_1=0)} \\
 \|S(\omega_{DJ\acute{O}timo(K=1, \beta_1=0)})\|_F^2 &= N\omega_{DJ\acute{O}timo(K=1, \beta_1=0)} + N - 2N\omega_{DJ\acute{O}timo(K=1, \beta_1=0)} \\
 \|S(\omega_{DJ\acute{O}timo(K=1, \beta_1=0)})\|_F^2 &= N(1 - \omega_{DJ\acute{O}timo(K=1, \beta_1=0)}) \\
 \|S(\omega_{DJ\acute{O}timo(K=1, \beta_1=0)})\|_F &= \sqrt{N(1 - \omega_{DJ\acute{O}timo(K=1, \beta_1=0)})}
 \end{aligned}$$

5.3.4.2 Suavizadores com $\beta_2 = 0$

Neste caso, visa-se otimizar a característica de suavização.

Ao anular o peso β_2 , o critério de tentativa de otimização resulta em:

$$\min(\|\bar{P}^T S^K\|_F)$$

O suavizador com esta tentativa de otimização pode ser recomendado para MG W-Cycle.

Na comparação experimental de suavizadores para-otimizados simétricos com grau de liberdade de otimização igual a N , é utilizado o suavizador nomeado de OS2WAMG (Optimized Smoother to WAMG). Este suavizador possui a matriz iteração M com apenas os elementos da diagonal principal não nulos e é para-otimizado para um passo de suavização.

Os elementos não nulos de M são obtidos 2 a 2, variando J de 1 a N , resolvendo o seguinte sistema de equações:

$$\begin{bmatrix} \sum_{i=1}^N |a_{i,2J-1}|^2 & -\sum_{i=1}^N a_{i,2J-1}^* a_{i,2J} \\ -\sum_{i=1}^N a_{i,2J}^* a_{i,2J-1} & \sum_{i=1}^N |a_{i,2J}|^2 \end{bmatrix} \begin{bmatrix} m_{2J-1,2J-1} \\ m_{2J,2J} \end{bmatrix} = \begin{bmatrix} a_{2J-1,2J-1}^* - a_{2J,2J-1}^* \\ a_{2J,2J}^* - a_{2J-1,2J}^* \end{bmatrix}$$

Aplicando este critério para o método Damped Jacobi com um passo de suavização, teremos:

$$\begin{aligned} \min_{\omega} \left\| \bar{P}^T S_{DJ} \right\|_F &\rightarrow \omega_{DJ\acute{O}timo(K=1, \beta_2=0)} \\ \omega_{DJ\acute{O}timo(K=1, \beta_2=0)} &= \frac{N - \sum_{J=1}^{N/2} \left(\frac{a_{2J,2J-1}^*}{a_{2J,2J}^*} + \frac{a_{2J-1,2J}^*}{a_{2J-1,2J-1}^*} \right)}{\sum_{J=1}^{N/2} \sum_{j=1}^N \left| \frac{a_{2J-1,j}}{a_{2J-1,2J-1}} - \frac{a_{2J,j}}{a_{2J,2J}} \right|^2} \end{aligned} \quad (154)$$

5.3.4.3 Suavizadores com $\beta_1 = \beta_2$

Neste caso, visa-se otimizar tanto a convergência quanto a característica de suavização.

Ao igualar $\beta_1 = \beta_2$, o critério de tentativa de otimização resulta em:

$$\min \left(\left\| \bar{P}^T S^K \right\|_F^2 + \left\| S^K \right\|_F^2 \right)$$

Na comparação experimental de suavizadores para-otimizados simétricos com grau de liberdade de otimização igual a N , é utilizado o suavizador nomeado de OS2WAMG3. Este suavizador possui a matriz iteração M com apenas os elementos da diagonal principal não nulos e é para-otimizado para um passo de suavização.

Os elementos não nulos de M são obtidos 2 a 2, variando J de 1 a N , resolvendo o seguinte sistema de equações:

$$\begin{bmatrix} 3\sum_{i=1}^N |a_{i,2J-1}|^2 & -\sum_{i=1}^N a_{i,2J-1}^* a_{i,2J} \\ -\sum_{i=1}^N a_{i,2J}^* a_{i,2J-1} & 3\sum_{i=1}^N |a_{i,2J}|^2 \end{bmatrix} \begin{bmatrix} m_{2J-1,2J-1} \\ m_{2J,2J} \end{bmatrix} = \begin{bmatrix} 3a_{2J-1,2J-1}^* - a_{2J,2J-1}^* \\ 3a_{2J,2J}^* - a_{2J-1,2J}^* \end{bmatrix}$$

5.4. Indicadores de Auxílio à Decisão

Um dos objetivos deste trabalho é determinar indicadores, obtidos com baixo custo computacional, que auxiliam na configuração do método WAMG.

5.4.1. Dominância da Diagonal

Grande parte dos métodos iterativos tem sua convergência sensível à dominância da diagonal principal. Um indicador que quantifique a dominância da diagonal principal pode auxiliar na escolha entre um método de menor custo computacional por iteração e um método menos sensível a dominância da diagonal principal.

No caso extremo em que existem elementos nulos na diagonal principal, os métodos iterativos que efetuam divisão pelo valor dos elementos da diagonal principal, tendo como exemplo Jacobi, Gauss Seidel e SOR, resultam em divisão por 0, caracterizando uma sensibilidade à dominância da diagonal principal, o que não acontece com o método SPAl(ϵ). No caso extremo em que o indicador for 1, nos métodos iterativos como Jacobi, Gauss Seidel e SPAl, a solução é obtida em uma única iteração.

Um critério usual para qualificar se a matriz A é diagonal dominante é verificar para todo elemento da diagonal principal, se seu valor absoluto é maior que o valor absoluto de qualquer outro elemento da respectiva linha:

$$|a_{ii}| > |a_{ij}| \text{ para qualquer } i \text{ e qualquer } j \quad (155)$$

Outro critério usual para qualificar se uma matriz A é diagonal dominante é se para todo elemento da diagonal principal, seu valor absoluto for maior que a soma do valor absoluto dos demais elementos da respectiva linha:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \text{ para qualquer } i \quad (156)$$

Conforme apresentado na seção [3.2.4.1], o seguinte indicador serve para quantificar a dominância da diagonal principal:

$$ddp_{DJ} = \omega_{DJ \acute{O}rimo(K=1, \beta_1=0)} = \frac{N}{\sum_{i=1}^N \sum_{j=1}^N \frac{|a_{ij}|^2}{|a_{ii}|}}$$

O valor deste indicador proposto varia entre 0 a 1, sendo 0 para o caso extremo em que existe algum elemento nulo na diagonal principal e 1 no caso extremo em que a matriz for diagonal.

O método SPAI-0 é um método sensível à diagonal principal. Vamos utilizar um indicador relacionado à norma Frobenius da matriz iteração de erro $\|S_{SPAI-0}\|$ que varie de 0 a 1:

$$ddp_{SPAI-0} = 1 - \frac{\|S_{SPAI-0}\|^2}{N}$$

Substituindo $\|S_{SPAI-0}\|^2 = N - \sum_{i=1}^N \frac{|a_{ii}|^2}{\sum_{j=1}^N \|a_{ij}\|^2}$ (82), teremos:

$$\begin{aligned} ddp_{SPAI-0} &= 1 - \frac{N - \sum_{i=1}^N \frac{|a_{ii}|^2}{\sum_{j=1}^N \|a_{ij}\|^2}}{N} \\ ddp_{SPAI-0} &= \frac{1}{N} \sum_{i=1}^N \frac{|a_{ii}|^2}{\sum_{j=1}^N \|a_{ij}\|^2} \end{aligned} \quad (157)$$

A complexidade de tempo fechada na determinação destes indicadores é $O(Nm)$.

5.4.2. Densidade da Matriz

A densidade da matriz, que pode ser expressa em número médio de elementos não nulos por linha m , pode ser utilizada como indicador que permite escolher entre um método com melhor taxa de convergência por iteração e um método com complexidade de tempo de iteração menos sensível a densidade da matriz A .

A determinação da quantidade média de elementos não nulos por linha necessita apenas de uma operação.

Além da média aritmética, a média quadrática m_2 e a média cúbica m_3 de elementos não nulos por linha também podem ser utilizados como indicadores, cuja determinação tem como complexidade de tempo fechada $O(N)$.

5.4.3. Largura de Banda

A largura de banda de uma matriz pode ser quantificada de várias formas.

Vamos definir $BW(i)$ como a largura de banda da linha i :

$$BW(i) = |i - j|_{Max} \text{ para } a_{ij} \neq 0 \quad (158)$$

Um indicador de largura de banda da matriz pode ser a largura de banda da linha que tenha a maior largura de banda:

$$Ind_{BW} = BW_{Max} \quad (159)$$

5.4.4. Projeções da Matriz A

Conforme descrito nas seções 5.1.3.3 e 5.1.3.5, a norma Frobenius de projeções da matriz A servem como indicadores:

$$Proj_{hh} = \frac{\|\bar{R}^{T*} (\bar{R}\bar{R}^{T*})^{-1} \bar{R}\bar{A}\bar{P} (\bar{P}^{T*}\bar{P})^{-1} \bar{P}^{T*}\|_F^2}{\|A\|_F^2}$$

$$Proj_{HH} = \frac{\|R^{T*} (RR^{T*})^{-1} RAP (P^{T*}P)^{-1} P^{T*}\|_F^2}{\|A\|_F^2}$$

$$Proj_{hH} = \frac{\|\bar{R}^{T*} (\bar{R}\bar{R}^{T*})^{-1} \bar{R}\bar{A}\bar{P} (P^{T*}P)^{-1} P^{T*}\|_F^2}{\|A\|_F^2}$$

$$\text{Pr } oj_{Hh} = \frac{\|R^{T*}(RR^{T*})^{-1}RA\bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}\|_F^2}{\|A\|_F^2}$$

5.4.5. Limite de Convergência do Erro na Correção em Grid Grosso

Conforme apresentado na seção 5.1.3.3, serve como indicador de limite de convergência do componente de erro pertencente a $\Omega(P)$ em grid grosso:

$$\cot g(\theta) = \frac{\|R^{T*}(RR^{T*})^{-1}RA\bar{P}(\bar{P}^{T*}\bar{P})^{-1}\bar{P}^{T*}\|_F}{\|R^{T*}(RR^{T*})^{-1}RAP(P^{T*}P)^{-1}P^{T*}\|_F}$$

5.5. Conclusões

O uso de operadores P e R que transformam apenas os componentes de baixa frequência tem a vantagem de exigir baixo custo computacional na obtenção de P e R esparsos, sendo um transposto do outro. A desvantagem desta estratégia é que a aplicação do método MG é supostamente recomendado apenas para sistemas de equações oriundos do MEF aplicado em equações diferenciais elípticas.

Segundo o modelo proposto, supõe-se que o método Multigrid pode ser mais abrangente, qualquer método iterativo, estacionário ou não, com viabilidade de obtenção das matrizes P e R esparsas que definem subespaços satisfatórios, pode ser utilizado em conjunto com o método Multigrid.

Capítulo 6

APLICAÇÕES E RESULTADOS

A parte experimental deste trabalho consiste em testes do método WAMG V-Cycle com Filtro Haar, variando o sistema de equações, o suavizador, o coeficiente de relaxação para os suavizadores Damped Jacobi e SOR e as configurações de pré e pós-suavização.

Como suavizadores do método WAMG, são testados os métodos iterativos estacionários Damped Jacobi (DJ), Forward SOR (FSOR), Backward SOR (BSOR), Red Black SOR (RBSOR), Symmetric SOR (SSOR), SPAI-0 (SPAI0), OS2WAMG e OS2WAMG3 e os métodos iterativos não estacionários CG, ICCG e o BiCGSTAB. O coeficiente de relaxação para o método Damped Jacobi é variado de 0,1 a 1,4 com variação de 0,1 e para os métodos SOR é variado de 0,8 a 1,6 com variação de 0,2.

Os passos de pré-suavização v_1 e pós-suavização v_2 são variados de 0 a 5, totalizando 36 configurações por suavizador. A configuração do método Multigrid WAMG é representada por WAMG(Suavizador, v_1, v_2), tendo como exemplo WAMG(SSOR(0.8), 1, 2) que corresponde ao método WAMG suavizado pelo método Symmetric SOR com coeficiente de relaxação $\omega = 0,8$, com uma iteração na pré-suavização e duas iterações na pós-suavização.

Para cada método iterativo testado como suavizador do método WAMG, é testado também este método como solver, totalizando 37 resoluções por método iterativo testado.

Geralmente, para os métodos iterativos estacionários e para os métodos MG, a tangente da curva logarítmica do erro e do resíduo tende a um limite de taxa de convergência, que coincide com a pior taxa de convergência e geralmente, independe dos coeficientes independentes do sistema de equações (vetor b).

Para comparações, é adotada a estimativa desta taxa de convergência limítrofe, estimada pela taxa de convergência de fim de curva representada no Fig. 6.1:

Fig. 6.1

Como esta curva pode conter oscilações, optou-se por determinar esta taxa de convergência a partir da tangente da reta obtida pela interpolação linear dos últimos cinco pontos da curva.

A confiabilidade desta estimativa de taxa de convergência limite depende do critério de parada, como módulo do resíduo relativo limite e tempo máximo de relaxamento. Para viabilizar uma amostragem significativa de configurações testadas, não foram adotados critérios de parada que garantisse a máxima confiabilidade desta estimativa.

No caso da matriz dos coeficientes dependentes ser singular, existem infinitas soluções, não existindo um único vetor erro para cada solução aproximada. Para que a estimativa da pior taxa de convergência seja efetuada a partir da curva do erro, está sendo considerado que as matrizes testadas são não singulares e como não foi possível determinar a singularidade das matrizes testadas, esta consideração resulta em incertezas.

Para as matrizes assimétricas, são testados apenas o método Damped Jacobi, os métodos SOR, o método SPAI-0 e o método BiCGSTAB e para as matrizes simétricas, são testados o método Damped Jacobi, os métodos SOR, o método SPAI-0, o método BiCGSTAB, o método dos Gradientes Conjugados (CG), o método ICCG e os métodos propostos OS2WAMG e OS2WAMG3.

Para cada valor do coeficiente de relaxação dos métodos Damped Jacobi e SOR testado, o mesmo valor é utilizado em todos os níveis do método WAMG.

6.1. Matrizes testadas

Este estudo visa analisar o método WAMG aplicado a um sistema de equações qualquer, sem levar em consideração a sua origem, portanto, não está sendo considerada a classe do problema da origem do sistema de equações.

O método WAMG é recomendado para sistemas de equações de ordem elevada, mas para viabilizar os testes em uma variedade de configurações, foi dada a preferência a matrizes de ordem reduzida.

Os critérios de escolha de matrizes foram: obtenção de uma amostragem de matrizes para um mesmo problema; uma matriz obtida pela transformação citada na seção 5.2; matrizes mal condicionadas; matrizes que não sejam de banda e uma matriz de banda obtida a partir de uma das matrizes que não são de banda.

Matriz	Ordem	eñn	eñn/linha
Poisson961	961	14.641	15,23
Poisson3721	3.721	58.081	15,61
Poisson10201	10.201	160.801	15,76
Poisson961_T	961	14.641	15,24
nos7	729	4.617	6,33
bcsstk14	1.806	63.454	35,14
Eletroima	4.910	33.670	6,86
MotorRelutancia	4.197	28.511	6,79
MotorRelutanciaBand	4.197	28.511	6,79

Tab. 1 – Matrizes testadas.

6.1.1. Matrizes Poisson961, Poisson3721 e Poisson10201

Estas matrizes reais assimétricas foram geradas para um problema de Poisson [29] com ordens diferentes.

Em conjunto com estas matrizes, foram geradas suas malhas com elementos quadriláteros, simplificando a “plotagem” dos vetores erro e resíduo na malha.

Para avaliar as incertezas dos resultados, foram feitos três testes diferentes para a matriz Poisson961: teste Poisson961, teste Poisson961_18 e teste Poisson961_2CF.

No teste Poisson961, o erro inicial tem predominância de componentes de baixa frequência e o critério de parada consiste no resíduo relativo de 10^{-9} e no tempo máximo de relaxação de 15 segundos.

No teste Poisson961_18, o erro inicial é a mesmo do teste Poisson961 e o critério de parada consiste no resíduo relativo de 10^{-18} e no tempo máximo de relaxação de 30 segundos.

No teste Poisson961_2CF, o erro inicial é composto por um componente de frequência $N/2$ e um componente de frequência 0 e o critério de parada consiste no resíduo relativo de 10^{-18} e no tempo máximo de relaxação de 60 segundos.

Para a matriz Poisson3721, foi feito o teste Poisson3721, cujo erro inicial corresponde ao mesmo do teste Poisson961 e o critério de parada consiste no resíduo relativo de 10^{-9} e no tempo máximo de relaxação de 210 segundos.

Para a matriz Poisson10201, foi feito o teste Poisson10201, cujo erro inicial corresponde ao mesmo do teste Poisson961 e o critério de parada consiste no resíduo relativo de 10^{-9} e no tempo máximo de relaxação de 1.550 segundos.

Fig. 6.2 – Ilustração dos elementos não nulos para a) Poisson961, b) Poisson3721 e c) Poisson10201.

6.1.2. Matrizes transformada Poisson961_T

A matriz Poisson961_T é obtida pela transformação citada na seção 5.2 da matriz Poisson961.

No teste Poisson961_T, o erro inicial é obtido pela transformação do erro inicial do teste Poisson961 e o critério de parada consiste no resíduo relativo de 10^{-9} e no tempo máximo de relaxação de 15 segundos.

6.1.3. Matrizes nos7 e bcsstk14

As matrizes reais simétricas positivo definida nos7 e bcsstk14 foram obtidas no site do Matrix Market [28] site National Institute of Standards and Technology, “Matrix Market”. Available from: <<http://math.nist.gov/MatrixMarket/>>..

Estas matrizes foram escolhidas por serem mal condicionadas, sendo que a matriz nos7 possui número de condição $4,1 \times 10^9$ e a matriz bcsstk14 possui número de condição $1,3 \times 10^{10}$.

Fig. 6.3 – Ilustração dos elementos não nulos para a) nos7 e b) bcsstk14.

Para a matriz nos7, foi feito o teste nos7, cujo erro inicial é composto por um componente de frequência $N/2$ e outro componente de frequência 0 e, o critério de parada consiste no resíduo relativo de 10^{-9} e no tempo máximo de relaxação de 10 segundos.

Para a matriz bcsstk14, foram feitos os testes bcsstk14, cujo erro inicial é composta por um componente de frequência $N/2$ e outro componente de frequência 0 e, o critério de parada consiste no resíduo relativo de 10^{-9} e no tempo máximo de relaxação de 60 segundos e o teste bcsstk14_240 com critério de parada no tempo máximo de 240 segundos.

6.1.4. Matrizes Eletroima e MotorRelutancia

As matrizes reais simétricas Eletroima e MotorRelutancia foram obtidas do laboratório LMAG-EPUSP.

Fig. 6.4 – Ilustração dos elementos não nulos para a) Eletroima e b) MotorRelutancia.

Para a matriz Eletroima, foi feito o teste Eletroima, cujo erro inicial é composto por um componente de frequência $N/2$ e outro componente de frequência 0 e, o critério de parada consiste no resíduo relativo de 10^{-9} e no tempo máximo de relaxação de 370 segundos.

Para a matriz MotorRelutancia, foi feito o teste MotorRelutancia, cujo erro inicial é composto por um componente de frequência $N/2$ e outro componente de frequência 0 e, o critério de parada consiste no resíduo relativo de 10^{-9} e no tempo máximo de relaxação de 148 segundos.

6.1.5. Matriz MotorRelutanciaBand

A matriz real simétrica MotorRelutanciaBand foi obtida a partir da matriz MotorRelutancia, reordenando as linhas e colunas, obtendo uma matriz de banda.

A partir do grafo associado à matriz, são determinados os nós mais distantes. Para um nó extremo é fixado o valor 0, para outro nó extremo é fixado o valor 1 e para os demais nós, seu atributo de ordenação é iniciado com a média de 0 e 1 ponderada pelas suas distâncias aos nós extremos. A cada iteração do método de ordenação, a cada um dos nós é re-atribuído o valor médio entre os valores do atributo dos nós vizinhos com maior valor e com menor valor. Após algumas

iterações, os nós são ordenados pelo valor do atributo de ordenação, obtendo a matriz de banda.

Fig. 6.5 – Ilustração dos elementos não nulos da matriz MotorRelutanciaBand.

Para a matriz MotorRelutanciaBand, foi feito o teste MotorRelutanciaBand, cujo erro inicial é composto por um componente de frequência $N/2$ e outro componente de frequência 0 e, o critério de parada consiste no resíduo relativo de 10^{-9} e no tempo máximo de relaxação de 136 segundos.

6.2. Resultados Obtidos

Os resultados obtidos são apresentados em gráficos, tabelas e ilustrações.

Para simplificar a referência a conjuntos de resultados na análise, estes resultados estão organizados em seções e discriminados por números e letras.

6.2.1. Indicadores das Matrizes

Para cada matriz testada, são traçados 8 gráficos tendo o nível do grid como abscissa:

- Gráfico a – Ordem da Matriz $N \times \text{Nível}$.
- Gráfico b – Quantidade de Elementos não nulos $\times \text{Nível}$.
- Gráfico c – Elementos não Nulos por Linha $\times \text{Nível}$.
- Gráfico d – Largura de Banda $\times \text{Nível}$.
- Gráfico e – Projeções nos Subespaços descritas na seção 5.4.4 em função do nível.
- Gráfico f – Teta $\times \text{Nível}$. Indicador proposto na seção 5.4.5.
- Gráfico g – Dominância da Diagonal Principal $\times \text{Nível}$. Contem as curvas dos indicadores apresentados na seção 5.4.1.

6.2.1.1 Matriz Poisson961

6.2.1.2 Matriz Poisson3721

6.2.1.3 Matriz Poisson10201

6.2.1.4 Matriz Poisson961_T

6.2.1.5 Matriz nos7

6.2.1.6 Matriz bcsstk14

6.2.1.7 Matriz Eletroima

6.2.1.8 Matriz MotorRelutancia

a

b

c

d

e

f

g

6.2.1.9 Matriz MotorRelutanciaBand

6.2.2. Distribuição de elementos não nulos

6.2.2.1 Matriz Poisson961

Nível 0

Nível 1

Nível 2

Nível 3

Nível 4

Nível 5

Nível 6

6.2.2.2 Matriz MotorRelutancia

Nível 0

Nível 1

Nível 2

Nível 3

Nível 4

Nível 5

Nível 6

Nível 7

Nível 8

6.2.3. Comparação entre os Métodos Iterativos

Na comparação entre os métodos iterativos, como solver e como suavizador, são traçados gráficos com curvas de erro e resíduo.

Para ilustrar a influência da ordenação SOR, tanto como solver quanto como suavizador WAMG, são traçados gráficos para o erro e para o resíduo, com uma curva para o FSOR, outra para o BSOR, outra para o RBSOR e outra para o SSOR.

Como os suavizadores SPAI-0, OS2WAMG e OS2WAMG3 são para-otimizados para um passo de suavização, são ilustrados gráficos comparando os suavizadores WAMG testados para um passo de pré e um passo de pós suavização.

- Gráfico a - Curvas de Erro em função do tempo para os métodos iterativos como solver.
- Gráfico b - Curvas de Resíduo em função do tempo para os métodos iterativos como solver.
- Gráfico c - Curvas de Erro em função do tempo para os métodos iterativos como suavizador WAMG.
- Gráfico d - Curvas de Resíduo em função do tempo para os métodos iterativos como suavizador WAMG.
- Gráfico e - Curvas de Erro em função do tempo para os métodos SOR como solver.
- Gráfico f - Curvas de Resíduo em função do tempo para os métodos SOR como solver.
- Gráfico g - Curvas de Erro em função do tempo para os métodos SOR como suavizadores WAMG.
- Gráfico h - Curvas de Resíduo em função do tempo para os métodos SOR como suavizadores WAMG.
- Gráfico i - Curvas de Erro em função do tempo para os métodos iterativos como suavizadores WAMG configurado com 1 passo de pré e 1 passo de pós-suavização.
- Gráfico j - Curvas de Resíduo em função do tempo para os métodos iterativos como suavizadores WAMG configurado para 1 passo de pré e 1 passo de pós-suavização.

6.2.3.1 Teste Poisson961

a

b

c

d

e

f

g

h

i

j

6.2.3.2 Teste Poisson961_18

a

b

c

d

e

f

g

h

i

j

6.2.3.3 Teste Poisson961_2CF

a

b

c

d

e

f

g

h

i

j

6.2.3.4 Teste Poisson3721

a

b

c

d

e

f

g

h

i

j

6.2.3.5 Teste Poisson10201

a

b

c

d

e

f

g

h

i

j

6.2.3.7 Teste nos7

a

b

c

d

e

f

g

h

i

j

6.2.3.8 Teste bcsstk14

a

b

c

d

e

f

g

h

i

j

6.2.3.9 Teste bcsstk14_240

a

b

c

d

e

f

g

h

i

j

6.2.3.10 Teste Eletroíma

a

b

c

d

e

f

g

h

i

j

6.2.3.11 Teste MotorRelutancia

a

b

c

d

e

f

g

h

i

j

6.2.3.12 Teste MotorRelutanciaBand

a

b

c

d

e

f

g

h

i

j

6.2.4. Ranque das Configurações do Método WAMG

Para cada configuração do método WAMG (suavizador, coeficiente de relaxação, passos de pré-suavização e passos de pós-suavização), são obtidas por amostra de testes: a taxa relativa média de convergência de erro, o desvio padrão da taxa relativa de convergência de erro e a pior taxa relativa de convergência de erro. A taxa relativa de convergência corresponde à relação entre a taxa de convergência absoluta da respectiva configuração e a taxa de convergência absoluta da configuração que apresentou o melhor resultado no respectivo teste. Os testes foram selecionados para a amostragem do problema de Poisson, para a amostragem completa e para a amostragem completa não viciada.

Para cada amostra são reproduzidas três tabelas:

- a) Ranque classificado pela média da taxa relativa de convergência de erro.
- b) Ranque classificado pela pior taxa relativa de convergência de erro estimada. Esta estimativa corresponde à subtração da média pelo quádruplo do desvio padrão da taxa relativa de convergência de erro. Para uma distribuição normal, esta diferença representa o valor mínimo possível com a confiabilidade próxima a 100%.
- c) Ranque classificado pela menor taxa relativa de convergência de erro observada.

Para cada amostra e cada suavizador do método WAMG, é apresentado o gráfico de distribuição, considerando distribuição normal, referente ao ranque classificado pela média da taxa relativa de convergência de erro.

6.2.4.1 Amostra completa

Suavizador	v_1	v_2	Med(txConvErrRel)	dp(txConvErrRel)	Med-4*dp	Min(txConvErrRel)
BSOR($\omega=1,6$)	4	1	0,632719304	0,292518469	-0,5373546	0,136498582
RBSOR($\omega=1,6$)	5	2	0,616223417	0,231929458	-0,3114944	0,098214364
SSOR($\omega=1,6$)	5	0	0,576344645	0,22993482	-0,3433946	0,034841649
FSOR($\omega=1,6$)	2	5	0,535724779	0,279154458	-0,5808931	0,019815395
SPAI0	0	4	0,208505226	0,128845672	-0,3068775	0,053822365
CG	4	2	0,198387117	0,367395724	-1,2711958	2,9442E-07
OS2WAMG3	0	2	0,188535742	0,212823621	-0,6627587	0,042533936
OS2WAMG	0	1	0,183824293	0,227880002	-0,7276957	0,03352493
DJ($\omega=0,7$)	4	0	0,183629983	0,108163225	-0,2490229	0,067921842
ICCG	5	4	0,132759143	0,063387157	-0,1207895	0,071043797
BiCGSTAB	0	2	0,110515579	0,269858465	-0,9689183	0

Tabela a - Ranque classificado pela média da taxa relativa de convergência de erro

Suavizador	v_1	v_2	Med(txConvErrRel)	Dp(txConvErrRel)	Med-4*dp	Min(txConvErrRel)
SSOR($\omega=1,6$)	1	3	0,44924023	0,106972246	0,02135124	0,215715489
ICCG	2	5	0,10937394	0,020468394	0,02750036	0,076070354
CG	1	0	0	0	0	0
BSOR($\omega=1,0$)	0	4	0,328169126	0,111436736	-0,1175778	0,142078163
DJ($\omega=0,4$)	5	5	0,124607423	0,063481157	-0,1293172	0,052135125
RBSOR($\omega=1,4$)	4	1	0,510980754	0,164323079	-0,1463116	0,242075751
OS2WAMG3	5	5	0,141868971	0,07358182	-0,1524583	0,063081364
SPAI0	5	5	0,148758248	0,087957079	-0,2030701	0,062773866
FSOR($\omega=1,2$)	0	4	0,349169507	0,155073796	-0,2711257	0,145222399
BiCGSTAB	5	4	0,044065397	0,080576342	-0,27824	1,65307E-05
OS2WAMG	2	1	0,166267781	0,149759918	-0,4327719	0,046158391

Tabela b - Ranque classificado pela pior taxa relativa de convergência de erro estimada

Suavizador	v_1	v_2	Med(txConvErrRel)	dp(txConvErrRel)	Med-4*dp	Min(txConvErrRel)
RBSOR($\omega=1,6$)	0	2	0,50729643	0,19843964	-0,28646215	0,283772116
BSOR($\omega=1,6$)	1	5	0,6300917	0,21712025	-0,23838932	0,271685755
SSOR($\omega=1,6$)	3	4	0,444752907	0,1305772	-0,07755588	0,243620501
FSOR($\omega=1,4$)	2	3	0,518912677	0,20494641	-0,30087295	0,235734448
ICCG	5	3	0,130078009	0,05499245	-0,08989179	0,080779994
DJ($\omega=0,7$)	0	5	0,182330911	0,10028867	-0,21882376	0,071870634
OS2WAMG3	5	5	0,141868971	0,07358182	-0,15245831	0,063081364
SPAI0	5	5	0,148758248	0,08795708	-0,20307007	0,062773866
OS2WAMG	2	1	0,166267781	0,14975992	-0,43277189	0,046158391
BiCGSTAB	5	2	0,046805558	0,08865318	-0,30780716	5,30024E-05
CG	2	5	0,173235929	0,28179962	-0,95396253	6,83759E-07

Tabela c - Ranque classificado pela pior taxa de convergência relativa de erro observada

Gráfico a - gráfico de distribuição, considerando distribuição normal, referente ao ranque classificado pela média da taxa relativa de convergência de erro.

6.2.4.2 Amostragem do problema de Poisson

Nesta amostragem, foram selecionados os testes do problema de Poisson com erro inicial que contenha apenas componentes de baixa frequência: teste Poisson961_18, teste Poisson3721 e teste Poisson10201.

Suavizador	v_1	v_2	Med(txConvErrRel)	dp(txConvErrRel)	Med-4*dp	Min(txConvErrRel)
SSOR($\omega=1,2$)	5	5	0,785432774	0,36991561	-0,69422969	0,358292566
RBSOR($\omega=1,6$)	5	3	0,635223758	0,31638008	-0,63029657	0,43551292
FSOR($\omega=1,6$)	0	5	0,62724673	0,39194124	-0,94051825	0,349142457
BSOR($\omega=1,4$)	0	3	0,600600041	0,1073375	0,171250023	0,498507498
DJ($\omega=1,1$)	0	3	0,228186892	0,03127698	0,103078959	0,192203768
SPAI0	0	4	0,187605366	0,02340904	0,093969217	0,160575043
BiCGSTAB	5	1	0,000135759	0,00012004	-0,00034441	4,90283E-05

Tabela a - Ranque classificado pela média da taxa relativa de convergência de erro

Suavizador	v_1	v_2	Med(txConvErrRel)	dp(txConvErrRel)	Med-4*dp	Min(txConvErrRel)
SSOR($\omega=1,4$)	0	3	0,469947681	0,02710775	0,361516688	0,452373081
BSOR($\omega=1,6$)	2	5	0,479029491	0,05545524	0,25720855	0,431509075
RBSOR($\omega=1,6$)	0	2	0,411627427	0,05212452	0,203129365	0,351467919
DJ($\omega=1,3$)	2	4	0,186452269	0,00958594	0,148108493	0,180626866
SPAI0	0	4	0,187605366	0,02340904	0,093969217	0,160575043
FSOR(1,0)	5	5	0,306563991	0,07340399	0,012948034	0,242385678
BiCGSTAB	0	5	1,64205E-05	3,0333E-06	4,28741E-06	1,39279E-05

Tabela b - Ranque classificado pela pior taxa relativa de convergência de erro estimada

Suavizador	v_1	v_2	Med(txConvErrRel)	dp(txConvErrRel)	Med-4*dp	Min(txConvErrRel)
BSOR($\omega=1,4$)	0	3	0,600600041	0,1073375	0,171250023	0,498507498
SSOR($\omega=1,6$)	5	0	0,532683445	0,07158052	0,246361385	0,484375724
RBSOR($\omega=1,4$)	0	3	0,583223999	0,13341109	0,049579645	0,465667878
FSOR($\omega=1,4$)	3	5	0,500632648	0,17091415	-0,18302395	0,367070479
DJ($\omega=1,2$)	0	4	0,218510533	0,01961928	0,140033425	0,198636639
SPAI0	0	4	0,187605366	0,02340904	0,093969217	0,160575043
BiCGSTAB	5	2	0,000133554	7,3871E-05	-0,00016193	5,30024E-05

Tabela c - Ranque classificado pela pior taxa relativa de convergência de erro observada

Gráfico a - gráfico de distribuição, considerando distribuição normal, referente ao ranque classificado pela média da taxa relativa de convergência de erro.

6.2.4.3 Amostragem completa não viciada

Nesta amostragem, foram desconsiderados os testes Poisson961, Poisson961_18, Poisson3721, bcsstk14 e MotorRelutanciaBand, mantendo os testes Poisson961_2CF, Poisson10201, Poisson961_T, nos7, bcsstk14_240, Eletroima e MotorRelutancia.

Suavizador	v_1	v_2	Med(txConvErrRel)	dp(txConvErrRel)	Med-4*dp	Min(txConvErrRel)
BSOR($\omega=1,6$)	1	5	0,640388373	0,22340252	-0,25322172	0,271685755
RBSOR($\omega=1,6$)	5	1	0,615114372	0,23302665	-0,31699223	0,320970521
SSOR($\omega=1,6$)	5	0	0,551490216	0,29969342	-0,64728346	0,034841649
FSOR($\omega=1,6$)	4	3	0,549975073	0,26151949	-0,4961029	0,178352234
CG	4	2	0,223394157	0,43683449	-1,5239438	2,9442E-07
SPAI0	5	0	0,181593155	0,10443016	-0,23612748	0,102691228
DJ($\omega=0,7$)	5	0	0,16797795	0,08721525	-0,18088304	0,096885292
ICCG	5	4	0,147657261	0,06910342	-0,12875641	0,071043797
OS2WAMG3	5	0	0,147233228	0,05571051	-0,07560881	0,103143083
BiCGSTAB	0	2	0,144959843	0,34327549	-1,22814211	0
OS2WAMG	1	2	0,129568394	0,06237839	-0,11994517	0,050730119

Tabela a - Ranque classificado pela média da taxa relativa de convergência de erro

Suavizador	v_1	v_2	Med(txConvErrRel)	dp(txConvErrRel)	Med-4*dp	Min(txConvErrRel)
ICCG	2	5	0,118146089	0,0150698	0,057866886	0,105946868
OS2WAMG3	1	3	0,14605611	0,04735592	-0,04336758	0,100701791
OS2WAMG	0	2	0,119712257	0,04954328	-0,07846085	0,066798475
SSOR($\omega=1,6$)	1	3	0,463998099	0,13583887	-0,0793574	0,215715489
DJ($\omega=0,4$)	5	5	0,121895465	0,05903724	-0,11425351	0,059725668
BSOR($\omega=1,0$)	0	2	0,26706715	0,10658199	-0,15926081	0,147525967
SPAI0	1	0	0,135027736	0,08771655	-0,21583848	0,049808532
RBSOR($\omega=1,6$)	2	2	0,560035326	0,20235758	-0,24939498	0,257102486
BiCGSTAB	5	4	0,045112408	0,07903935	-0,27104498	6,85215E-05
FSOR($\omega=0,8$)	3	0	0,194614212	0,11715107	-0,27399006	0,065873382
CG	2	1	0,057611869	0,11831349	-0,41564209	0

Tabela b - Ranque classificado pela pior taxa de convergência relativa de erro estimada

Suavizador	v_1	v_2	Med(txConvErrRel)	dp(txConvErrRel)	Med-4*dp	Min(txConvErrRel)
RBSOR($\omega=1,6$)	5	1	0,615114372	0,23302665	-0,31699223	0,320970521
BSOR($\omega=1,6$)	1	5	0,640388373	0,22340252	-0,25322172	0,271685755
SSOR($\omega=1,6$)	3	4	0,415340122	0,1434523	-0,15846907	0,243620501
FSOR($\omega=1,4$)	2	3	0,467936826	0,22948058	-0,44998551	0,235734448
OS2WAMG3	1	5	0,142399794	0,05251302	-0,06765227	0,109211878
ICCG	2	5	0,118146089	0,0150698	0,057866886	0,105946868
SPAI0	3	2	0,175321956	0,10967676	-0,26338507	0,103232199
DJ($\omega=0,7$)	5	0	0,16797795	0,08721525	-0,18088304	0,096885292
OS2WAMG	0	2	0,119712257	0,04954328	-0,07846085	0,066798475
BiCGSTAB	5	4	0,045112408	0,07903935	-0,27104498	6,85215E-05
CG	2	5	0,174588714	0,32477303	-1,12450342	6,83759E-07

Tabela c - Ranque classificado pela pior taxa de convergência relativa de erro observada

Gráfico a - gráfico de distribuição, considerando distribuição normal, referente ao ranque classificado pela média da taxa relativa de convergência de erro.

6.2.5. Componentes de Frequência

Para os métodos iterativos como solver, o erro e o resíduo são decompostos em componentes de alta e baixa frequência. Com estas decomposições são traçados os gráficos:

- Gráfico a - Relação entre baixa e alta frequência em função da iteração. São traçadas a curva da relação entre o módulo do componente de erro de baixa frequência e o módulo do componente de erro de alta frequência, a curva da relação entre o módulo do componente de resíduo de baixa frequência e o módulo do componente de resíduo de alta frequência e o valor médio estimado do cotangente de θ descrito na seção 5.1.3.3.
- Gráfico b - Gráfico logarítmico dos componentes de erro e resíduo em função da iteração. São traçadas a curva do módulo do componente de erro de baixa frequência, a curva do módulo componente de erro de alta frequência, a curva do módulo do componente de resíduo de baixa frequência e a curva do módulo componente de resíduo de alta frequência.

Os métodos iterativos estão numerados em:

- 1) Damped Jacobi
- 2) SOR
- 3) SPAI-0
- 4) BiCGSTAB
- 5) CG
- 6) ICCG
- 7) OS2WAMG
- 8) OS2WAMG3

6.2.5.1 Teste Poisson961

1a

1b

2a

2b

3a

3b

4a

4b

1a

1b

6.2.5.2 Teste Poisson961_18

1a

1b

2a

2b

3a

3b

4a

4b

6.2.5.3 Teste Poisson961_2CF

1a

1b

2a

2b

3a

3b

4a

4b

6.2.5.4 Teste Poisson3721

1a

1b

2a

2b

3a

3b

4a

4b

6.2.5.5 Teste Poisson10201

1a

1b

2a

2b

3a

3b

4a

4b

6.2.5.6 Teste Poisson961_T

1a

1b

2a

2b

3a

3b

4a

4b

6.2.5.7 Teste nos7

1a

1b

2a

2b

3a

3b

4a

4b

5a

5b

6a

6b

7a

7b

8a

8b

6.2.5.8 Teste bcsstk14

1a

1b

2a

2b

3a

3b

4a

4b

5a

5b

6a

6b

7a

7b

8a

8b

6.2.5.9 Teste bcsstk14_240

1a

1b

2a

2b

3a

3b

4a

4b

5a

5b

6a

6b

7a

7b

8a

8b

6.2.5.10 Teste Eletroima

1a

1b

2a

2b

3a

3b

4a

4b

5a

5b

6a

6b

7a

7b

8a

8b

6.2.5.11 Teste MotorRelutancia

1a

1b

2a

2b

3a

3b

4a

4b

5a

5b

6a

6b

7a

7b

8a

8b

6.2.5.12 Teste MotorRelutanciaBand

1a

1b

2a

2b

3a

3b

4a

4b

5a

5b

6a

6b

7a

7b

8a

8b

6.2.6. Filtro Haar x Vizinhança

Nos métodos GMG e AMG clássicos, o subespaço dos componentes de baixa frequência depende da vizinhança de nós e no método WAMG, este subespaço dos componentes de baixa frequência depende do filtro adotado.

Para confrontar os subespaços baseados em vizinhança de nós com os subespaços baseados no Filtro Haar, são apresentados:

- Ilustração a - Conjunto de plotagem do erro na malha do MEF, variando a iteração de 0 a 17. Esta ilustração permite visualizar os componentes de frequência baseado em vizinhança dos nós.
- Gráfico b - Gráfico dos componentes de erro por faixa de frequência em função da iteração. Neste gráfico, cada curva corresponde ao módulo do componente de erro pertencendo ao subespaço associado ao respectivo grid. Para decomposição do erro é utilizado Filtro Haar.

6.2.6.1 Teste Poisson961 – DJ(1,3)

Gráfico a

Gráfico b

6.2.6.2 Teste Poisson961_2CF – DJ(1,3)

Gráfico a

Gráfico b

6.2.6.3 Teste Poisson961_2CF – BSOR(1,0)

Gráfico a

Gráfico b

6.2.7. Influência da Configuração de Pré e Pós-suavização

Para cada método iterativo em cada sistema de equações são testadas 36 configurações de pré e pós-suavização, não sendo conveniente a apresentação de todas as curvas de erro ou resíduo em função do tempo em um único gráfico. Para ilustrar a influência destes parâmetros na curva de erro e resíduo em função do tempo, são traçadas apenas as curvas das configurações que resultaram na melhor e na pior taxa de convergência. Para comparações, é traçada em conjunto a curva do método iterativo como solver.

Para cada método iterativo, são traçados os seguintes gráficos:

- Gráfico a - Curvas de Erro em função do tempo para o método iterativo como solver, melhor convergência do método como suavizador WAMG e pior convergência do método como suavizador WAMG. Para os solvers Damped Jacobi e SOR são traçadas as curvas para o coeficiente de relaxação que resultaram na melhor taxa de convergência de erro como solver e como suavizador do método WAMG.
- Gráfico b - Curvas de Resíduo em função do tempo para o método iterativo como solver, melhor convergência do método como suavizador WAMG e pior convergência do método como suavizador WAMG. Para os solvers Damped Jacobi e SOR são traçadas as curvas para o coeficiente de relaxação que resultaram na melhor taxa de convergência de erro como solver e como suavizador do método WAMG.
- Gráfico c - Gráfico 3D da Taxa de Convergência em função dos passos de pré-suavização v_1 e dos passos de pós-suavização v_2 .

Os métodos iterativos estão numerados em:

- 1) Damped Jacobi
- 2) SOR
- 3) SPAI-0
- 4) BiCGSTAB
- 5) CG
- 6) ICCG
- 7) OS2WAMG
- 8) OS2WAMG3

6.2.7.1 Teste Poisson961

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.7.2 Teste Poisson961_18

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.7.3 Teste Poisson961_2CF

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.7.4 Teste Poisson3721

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.7.5 Teste Poisson10201

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.7.6 Teste Poisson961_T

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.7.7 Teste nos7

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

5a

5b

5c

6a

6b

6c

7a

7b

7c

8a

8b

8c

6.2.7.8 Teste bcsstk14

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

5a

5b

5c

6a

6b

6c

7a

7b

7c

8a

8b

8c

6.2.7.9 Teste bcstk14_240

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

5a

5b

5c

6a

6b

6c

7a

7b

7c

8a

8b

8c

6.2.7.10 Teste Eletroima

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

5a

5b

5c

6a

6b

6c

7a

7b

7c

8a

8b

8c

6.2.7.11 Teste MotorRelutancia

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

5a

5b

5c

6a

6b

6c

7a

7b

7c

8a

8b

8c

6.2.7.12 Teste MotorRelutanciaBand

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

5a

5b

5c

6a

6b

6c

7a

7b

7c

8a

8b

8c

6.2.8. Coeficiente de Relaxação do Damped Jacobi

No teste do método Damped Jacobi como suavizador do método WAMG, o coeficiente de relaxação é o mesmo em todos os níveis. Para análise da influência do coeficiente de relaxação para o método Damped Jacobi como solver e como suavizador WAMG, são apresentados os seguintes gráficos:

- Gráfico a – Passos de Pré e Pós-suavização em função do coeficiente de relaxação para as configurações que resultaram na melhor taxa de convergência de erro final.
- Gráfico b – Passos de Pré e Pós-suavização em função do coeficiente de relaxação para as configurações que resultaram na pior taxa de convergência de erro final.
- Gráfico c - Curvas das taxas de convergência de erro e de resíduo finais em função do coeficiente de relaxação para o método Damped Jacobi como Solver e como Suavizador WAMG. As configurações de pré e pós-suavização do método WAMG correspondem às configurações do gráfico a.
- Gráfico d - Curvas de Estimativa de Suavização em função do coeficiente de relaxação, com uma curva para as melhores configurações de pré e pós suavização (gráfico a) e outra curva para as piores configurações de pré e pós suavização (gráfico b). A estimativa de suavização corresponde a diferença entre a taxa de convergência de erro do método WAMG suavizado pelo método Damped Jacobi e a taxa de convergência de erro do método Damped Jacobi como solver.
- Gráfico e - Curva da taxa de convergência de erro final do método WAMG suavizado pelo método Damped Jacobi com um passo de pré e um passo de pós suavização em função do coeficiente de relaxação.
- Gráfico f - Coeficiente de Relaxação Ótimo para Damped Jacobi x Nível em conjunto com o gráfico d. Neste gráfico, são traçadas as curvas dos indicadores extraídos da para-otimização do coeficiente de relaxação do método Damped Jacobi para uma iteração do método, descritos na seção 5.4.1 para serem confrontados com a curva de convergência de erro final do método WAMG suavizado pelo método Damped Jacobi com um passo de pré e um passo de pós-suavização.

6.2.8.1 Teste Poisson961

a

b

c

d

e

f

6.2.8.2 Teste Poisson961_18

a

b

c

d

e

f

6.2.8.3 Teste Poisson961_2CF

a

b

c

d

e

f

6.2.8.4 Teste Poisson3721

a

b

c

d

e

f

6.2.8.5 Teste Poisson10201

a

b

c

d

e

f

6.2.8.6 Teste Poisson961_T

a

b

c

d

e

f

6.2.8.7 Teste nos7

a

b

c

d

e

f

6.2.8.8 Teste bcsstk14

a

b

c

d

e

f

6.2.8.9 Teste bcsstk14_240

a

b

c

d

e

f

6.2.8.10 Teste Eletroima

a

b

c

d

e

f

6.2.8.11 Teste MotorRelutancia

a

b

c

d

e

f

6.2.8.12 Matriz MotorRelutanciaBand

a

b

c

d

e

f

6.2.9. Coeficiente de Relaxação no Método SOR

Para análise da influência do coeficiente de relaxação nos métodos SOR, foram feitos os seguintes gráficos:

- Gráfico a – Passos de Pré e Pós-suavização em função do coeficiente de relaxação. Cada configuração de pré e pós-suavização corresponde à configuração que resultou na melhor convergência de erro para o respectivo coeficiente de relaxação.
- Gráfico b - Curvas das taxas de convergência em função do coeficiente de relaxação. São traçadas a curva de taxa de convergência de erro do método SOR como solver, a curva da taxa de convergência de resíduo do método SOR como solver, a curva da taxa de convergência de erro do método WAMG suavizado pelo método SOR e a curva da taxa de convergência de resíduo do método WAMG suavizado pelo método SOR. As configurações de pré e pós-suavização do método WAMG corresponde às do gráfico a.
- Gráfico c - Curvas de Estimativa de Suavização em função do coeficiente de relaxação, sendo uma curva para as melhores configurações de pré e pós suavização (gráfico a) e outra curva para as piores configurações de pré e pós suavização. A estimativa de suavização corresponde a diferença entre a taxa de convergência de erro do método WAMG suavizado pelo método Damped Jacobi e a taxa de convergência de erro do método Damped Jacobi como solver.

Os métodos SOR estão numerados em:

- 1) FSOR
- 2) BSOR
- 3) RBSOR
- 4) SSOR

6.2.9.1 Teste Poisson961

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.9.2 Teste Poisson961_18

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.9.3 Teste Poisson961_2CF

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.9.4 Teste Poisson3721

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.9.5 Teste Poisson10201

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.9.6 Teste Poisson961_T

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.9.7 Teste nos7

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.9.8 Teste bcsstk14

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.9.9 Teste bcsstk14_240

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.9.10 Teste Eletroímã

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.9.11 Teste MotorRelutancia

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.2.9.12 Teste MotorRelutanciaBand

1a

1b

1c

2a

2b

2c

3a

3b

3c

4a

4b

4c

6.3. Análise dos Resultados

No nível do grid mais refinado do método WAMG, o suavizador tem o compromisso de relaxar a metade dos componentes de erro, o que leva a supor que as características correlacionadas ao grid mais refinado tenham uma influência significativa no comportamento do método WAMG, valendo a comparação entre o método iterativo como solver e o método WAMG suavizado por este método iterativo.

Baseado nesta consideração, uma forma de avaliar a característica de suavização de um método iterativo é comparando a convergência de erro do método WAMG suavizado por este método iterativo com a convergência de erro do respectivo método iterativo como solver.

Partindo do princípio que a convergência do método WAMG depende principalmente da convergência de seu suavizador como solver, da característica de suavização deste suavizador e do custo computacional adicional das transformações entre níveis, supõe-se que a diferença entre a taxa de convergência de erro do método WAMG e a taxa de convergência de erro do suavizador como solver quantifique a característica de suavização do suavizador quando o custo computacional adicional for insignificante. Quando ambos (solver e método WAMG) apresentarem divergência de erro, o custo computacional adicional tende a ser significativo, elevando as incertezas deste indicador da característica de suavização.

A imprecisão na estimativa da pior taxa de convergência de erro, a consideração que permite quantificar a característica de suavização pela diferença de taxas de convergência entre o método WAMG e seu suavizador como solver, a imprecisão nos tempos medidos, a influência do erro inicial no resultado dos testes e a consideração que a matriz dos coeficientes dependentes é singular resultam em incertezas de difícil quantificação. Para avaliar estas incertezas de forma qualitativa, são confrontados os resultados entre os testes Poisson961, Poisson961_18 e Poisson961_2CF e entre os testes bcsstk14 e bcsstk14_240.

6.3.1. Indicadores das Matrizes

Com o engrossamento dos grids, observa-se que a quantidade de elementos não nulos (gráfico b) diminui, a quantidade média de elementos não nulos por linha (gráfico c) tende a diminuir, a largura de banda tende a diminuir (gráfico d) e a dominância da diagonal principal (gráfico g) tende a aumentar.

Quando ao indicador $Pr oj_{Hh}$ (gráfico e), que sinaliza a compatibilidade entre os operadores Prolongamento e Restrição, sinaliza uma compatibilidade aleatória ($Pr oj_{Hh} \approx 0,25$) para as matrizes Poisson961, Poisson961_T, Poisson3721, Poisson10201 e bcsstk14, sinaliza uma melhor compatibilidade ($Pr oj_{Hh} \approx 0,1$) para as matrizes nos7, MotorRelutancia e MotorRelutanciaBand e sinaliza a melhor compatibilidade ($Pr oj_{Hh} \approx 0,05$) para a matriz Eletroima.

Quanto à estimativa do ângulo médio do limite de convergência na correção em grid grosso θ (gráfico f), parece ter uma tendência deste ângulo ser maior que 45° nas matrizes dos grids grosseiros.

A matriz bcsstk14 apresenta valor significativamente baixo para a dominância da diagonal principal (gráfico g). Supõe-se que o suavizador SPAI(ϵ) fosse mais indicado para esta matriz.

6.3.2. Distribuição de elementos não nulos

Podemos observar que o uso de wavelets na obtenção da matriz do grid grosso, a região da matriz que contem elementos não nulos na matriz do grid refinado continua tendo elementos não nulos nas matrizes dos grids grosseiros.

6.3.3. Comparação entre os Métodos Iterativos

Podemos observar que no testes bcsstk14 e bcsstk14_240, o limite de tempo no critério de parada foi insuficiente, pois nenhuma resolução do método WAMG atingiu o critério de parada por resíduo máximo (gráficos d) e nos testes Poisson3721, Poisson10201, MotorRelutancia e MotorRelutanciaBand, podemos observar também que os critérios de parada adotados impediram que o módulo do erro final atingisse valores satisfatórios (gráficos c).

Nas comparações entre os testes Poisson961, Poisson961_18 e Poisson961_2CF e entre os testes bcsstk14 e bcsstk14_240, podemos observar diferenças entre as melhores configurações do coeficiente de relaxação e dos passos

de pré e pós-suavização (gráficos c e g), sinalizando as incertezas na detecção da melhor configuração.

O método WAMG suavizado pelo SOR apresentou melhor resultado em todos os testes efetuados.

O método WAMG apresentou convergência semelhante, quando suavizado pelos suavizadores Damped Jacobi com o melhor coeficiente de relaxação, SPAI-0, OS2WAMG e OS2WAMG3 (gráficos c).

Na configuração de um passo de pré e um passo de pós-suavização (gráfico i), para a qual os suavizadores SPAI-0, OS2WAMG e OS2WAMG3 foram paratimizados, podemos observar que nos testes Eletroímã e MotorRelutanciaBand, o suavizador SOR foi superado por estes três suavizadores.

Nos testes bcsstk14 e bcsstk14_240, o método OS2WAMG apresentou uma significativa divergência como solver (gráficos a), mas não apresentou resultado tão ruim como suavizador do método WAMG (gráficos c), sinalizando que a paratimização da característica de suavização foi eficiente.

No gráfico c, ao compararmos o teste MotorRelutancia com o teste MotorRelutanciaBand, podemos observar melhor convergência do método WAMG suavizado por todos os métodos iterativos estacionários no teste MotorRelutanciaBand que no teste MotorRelutancia.

Quanto às configurações de pré e pós-suavização, não foi possível observar padrões que permitam estabelecer critérios de estimativa da melhor configuração, com exceção do suavizador OS2WAMG, cuja melhor configuração tende a ser um passo de pré e um passo de pós-suavização.

6.3.4. Ranque das Configurações do Método WAMG

No geral, podemos observar que os suavizadores SOR apresentaram melhores classificações que os demais. Esta melhor classificação dos métodos SOR pode estar mais relacionada com o fato de apresentarem melhor convergência como solver do que sua característica de suavização.

Na amostragem completa não viciada, os elevados valores de desvio padrão da taxa relativa de convergência de erro sinalizam que não seja recomendado adotar uma destas configurações como padrão.

6.3.5. Componentes de Freqüência

No gráfico a, a característica de suavização pode ser observada quando a relação do módulo do componente de baixa freqüência pelo módulo do componente de alta freqüência tende a estabilizar em um valor elevado.

O método Damped Jacobi com coeficiente de relaxação igual a 1,3 apresentou a característica de suavização de erro no teste Poisson961, mas não apresentou a característica de suavização de erro no teste Poisson961_2CF (gráficos a). Como os dois testes foram realizados para a mesma matriz Poisson961, mas com erros iniciais diferentes, podemos afirmar que, entre os componentes de má convergência (componentes de autovetores da matriz iteração de erro S com módulo elevado de autovalor) do erro inicial, nos componentes com módulo mais significativo, predomina o componente de baixa freqüência no teste Poisson961 e o componente de alta freqüência no teste Poisson961_2CF. Por este fato, a partir do gráfico a, não é possível afirmar que o método iterativo tenha boa característica de suavização, como é o caso do método Damped Jacobi que no teste nos7, com coeficiente de relaxação igual a 1,0, apresentou boa característica de suavização nos resultados das seções 6.3.7 e 6.3.8, mas não apresentou a característica de suavização no gráfico a.

Como esperado, ao compararmos o teste Poisson961 com o teste Poisson961_T, podemos observar que a relação entre o módulo do componente de baixa freqüência e o módulo do componente de alta freqüência se inverteu de um teste para o outro em todos os métodos iterativos, tanto para o erro quanto para o resíduo.

6.3.6. Filtro Haar x Vizinhança

Aparentemente, os subespaços de faixa de freqüências baseados no Filtro Haar (gráfico b) apresentam boa correspondência aos subespaços de faixa de freqüências baseados na vizinhança (gráfico a).

6.3.7. Influência da Configuração de Pré e Pós-suavização

Na comparação entre o método iterativo como solver e o método iterativo como suavizador WAMG (gráfico a), podemos observar que os métodos iterativos estacionários se comportaram melhor como suavizador WAMG e os métodos iterativos não estacionários se comportaram melhor como solver.

Nos testes Poisson961_18, Poisson961_2CF e nos7, podemos observar no gráfico c, que quando a soma dos passos de pré e pós-suavização for ímpar, o suavizador Damped Jacobi se comporta melhor que para a soma par. Uma provável explicação para este comportamento é a existência de pares de componentes de erro, formados por um componente de baixa frequência e outro componente de alta frequência, sendo que a matriz iteração de erro S do método Damped Jacobi pode estar transformando o componente de baixa frequência no seu respectivo componente de alta frequência e vice versa, permitindo que a correção em grid grosso relaxe também os componentes de alta frequência, mas indiretamente.

Ao compararmos o teste Poisson961 ao teste Poisson961_T, podemos observar que o método WAMG suavizado pelos métodos iterativos estacionários Damped Jacobi com coeficiente de relaxação 1,3, SOR com coeficiente de relaxação 1,6 e SPAI-0 apresentou melhor convergência que os respectivos solvers no teste Poisson961, contrário ao observado no teste Poisson961_T.

Podemos observar vários casos em que a configuração de pré e pós-suavização (gráficos a) tenha importância significativa na convergência do método WAMG, mas não foram observados padrões que permitissem estimar a melhor configuração.

6.3.8. Coeficiente de Relaxação do Método Damped Jacobi

Ao compararmos o teste Poisson961 com os testes Poisson961_18 e Poisson961_2CF, podemos observar diferenças entre as curvas dos gráficos a, c, d e e, sinalizando incertezas nestas curvas.

Na comparação entre os testes bcsstk14 e bcsstk14_240, os resultados praticamente coincidiram, sinalizando maior confiabilidade nestes resultados.

A partir dos gráficos a, não foram observados padrões que permitissem estimar a melhor configuração de pré e pós-suavização.

Nos gráficos c, podemos observar que tanto como suavizador do método WAMG, assim como solver, o método Damped Jacobi tende a melhorar a taxa de

convergência de erro e de resíduo com o aumento do coeficiente de relaxação até um ponto de máximo, iniciando uma tendência acentuada de divergência, sendo que o valor ótimo deste coeficiente de relaxação varia caso a caso.

Foi no teste Poisson961_T que o método Damped Jacobi menos apresentou a característica de suavização.

Nos gráficos d, com o aumento do coeficiente de relaxação, podemos observar tanto o aumento quanto a diminuição da característica de suavização.

Os coeficientes de relaxação ótimos estimados para um passo de pré e um passo de pós-suavização parecem ter uma correspondência com o coeficiente de relaxação ótimo obtido experimentalmente (gráficos f), mas não foi possível melhorar a estimativa destes coeficientes ótimos.

6.3.9. Coeficiente de Relaxação nos Métodos SOR

Ao compararmos o teste Poisson961 com os testes Poisson961_18 e Poisson961_2CF e o teste bcsstk14 com o teste bcsstk14_240, podemos observar diferenças entre as curvas dos gráficos a, b e c, sinalizando incertezas nestas curvas.

A partir dos gráficos a, não foram observados padrões que permitissem estimar a melhor configuração de pré e pós-suavização.

Nos gráficos b, podemos observar que em muitos casos, se fossem adotados valores para o coeficiente de relaxação acima de 1,6, teríamos resultados melhores para os métodos SOR como suavizadores do método WAMG, mas não foram observados padrões que permitissem estimar os coeficientes de relaxação ótimos.

Podemos observar que para a matriz Poisson961_T, em contraste com as demais, os métodos SOR não apresentaram boa característica de suavização na média.

Ao confrontarmos o teste Poisson961 com o teste Poisson961_T a partir do gráfico c, podemos observar que no teste Poisson961_T, foram observados casos em que a suavização tende a aumentar com o aumento do coeficiente de relaxação, contrário ao teste Poisson961_T.

Não foram observados padrões que permitissem estimar o melhor entre FSOR, BSOR, RBSOR e SSOR.

CONCLUSÕES

De forma perceptível, mas quase sutil, os resultados do teste MotorRelutanciaBand foram melhores que do teste MotorRelutancia, sendo um indício que o método WAMG apresente melhor resultado para matrizes de banda, pois os nós associados aos elementos não nulos de cada wavelet tendem a ser vizinhos no grafo obtido a partir desta matriz de banda, fazendo com que o subespaço dos componentes de baixa frequência definido por filtro wavelet tende a coincidir com o subespaço dos componentes de baixa frequência definido por critério de vizinhança. A reordenação de linhas e colunas da matriz do grid mais refinado para transformar esta matriz em matriz de banda pode servir como um complemento ao método WAMG.

Embora o subespaço de baixa frequência definido por filtro Haar não coincida exatamente com o subespaço de baixa frequência definido por critério de vizinhança, foi observado experimentalmente que estes subespaços tendem a se coincidir.

Nos testes das matrizes Poisson961_T e bcsstk14, foi observada uma menor presença da característica de suavização em todos os métodos iterativos estacionários testados, que leva a concluir que a característica de suavização depende da matriz dos coeficientes dependentes do sistema de equações. Portanto, o método MG baseado em filtros de frequência (GMG, AMG e WAMG) pode não ser indicado para qualquer sistema de equações.

Em todos os testes realizados, o método WAMG suavizado pelos métodos SOR apresentou melhor convergência que suavizado pelos demais métodos iterativos estacionários testados, mesmo quando a falta da característica de suavização prejudica a convergência do método WAMG. Portanto, os métodos SOR se apresentaram melhor como suavizador do método WAMG mais devido a sua convergência como solver que a sua característica de suavização.

Os elevados valores de desvio padrão da taxa relativa de convergência de erro das configurações do método WAMG obtidos estatisticamente para as amostragens completas sugerem que não exista uma configuração do método WAMG que garanta boa convergência em qualquer sistema de equações.

Pela análise de padrões e pelo confronto de indicadores efetuados neste trabalho, não foi possível elaborar critérios que permitissem estimar a melhor configuração do método WAMG entre os suavizadores testados.

Para os suavizadores para-otimizados com mesmo grau de liberdade de otimização como o suavizador SPAI-0, com sua convergência para-otimizada; o suavizador OS2WAMG, com sua característica de suavização para-otimizada e o suavizador OS2WAMG3, com ambas as características para-otimizadas de forma ponderada, foi observado que o método WAMG apresentou convergência similar para os três suavizadores.

Além das características de suavização e convergência, outra característica do suavizador do método WAMG com Filtro Haar que pode favorecer a convergência do método WAMG é a propriedade de atuar em pares de componentes de erro, onde a matriz iteração de erro transforme um componente de baixa frequência no seu respectivo par (componente de alta frequência) e vice-versa, característica supostamente observada para o método Damped Jacobi nos testes Poisson961_18, Poisson961_2CF e nos7. Provavelmente, esta característica possa ser para-otimizada, permitindo tirar proveitos.

Baseado na análise proposta na seção 5.2 e nos resultados obtidos experimentalmente, pode-se afirmar que a presença de forma plena da propriedade de suavização nos métodos iterativos estacionários não é garantida, tornando questionável a estratégia utilizada nos métodos Multigrid que visa tirar proveito desta propriedade. A estratégia de obtenção dos operadores P e R para-otimizados, proposta na seção 5.1.7, oferece uma alternativa que não visa tirar proveito desta propriedade.

SUGESTÃO PARA TRABALHOS FUTUROS

Para complementar o estudo do método WAMG com filtro Haar, é sugerido o teste de outros suavizadores não testados, como SPAI(ϵ), Fatoração LU Incompleta e etc.

Como uma nova frente de estudo, é sugerida a exploração da estratégia de obtenção dos operadores P e R para-otimizados proposta na seção 5.1.7.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] F. H. Pereira, **O Método Multigrid Algébrico na resolução de sistemas lineares oriundos do Método dos Elementos Finitos**, Tese de Doutorado, LMAG, EPUSP, São Paulo, Brasil (2007).
- [2] F. H. Pereira, S. L. Verardi, S. I. Nabeta, **A wavelet-based algebraic multigrid preconditioner for sparse linear systems**, Applied Mathematics and Computation 182 (2006) 1098–1107.
- [3] R.P.Fedorenko, **The speed of convergence of one iterative process**, USSR Comp. Math. Phys. 4(3), 1964, p.227-235.
- [4] W. Hackbush, **Ein iteratives Verfahren zur schnellen Auflösung elliptischer Randwert probleme**, tech. Rep. 76-12, Institute for Applied Mathematics, University of Cologne, West Germany, 1976.
- [5] P. Wesseling, **A robust and efficient multigrid method. Multigrid Methods**. Lecture Notes in Mathematics 960 (eds W. Hackbusch and U. Trottenberg), p.614–630, Springer, Berlin, 1982.
- [6] Y.Saad, **Iterative Methods for Sparse Linear System**. PWS Publishing, Boston, USA, 1996.
- [7] Braess, Dietrich, **Finite elements. Theory, fast solvers, and applications in solid mechanics**. Transl. 2nd ed. (English) Cambridge University Press, New York, USA, 1992.

- [8] J. L. Hennessy & D. A. Patterson. **Arquitetura de Computadores: uma abordagem quantitativa**. Editora Campus. Rio de Janeiro, RJ. Tradução da Terceira Edição americana, 2003.
- [9] D. A. Patterson & J. L. Hennessy. **Organização e Projeto de Computadores: a interface hardware/software**. Livros Técnicos e Científicos Editora, Inc. Rio de Janeiro, RJ. Tradução da Terceira Edição americana, 2005.
- [10] P.A. Morettin, **Ondas e Ondaletas: Da Análise de Fourier à Análise de Ondaletas**, São Paulo, Edusp, 1999.
- [11] D. Sima D. **Decisive Aspects in the Evolution of Microprocessors**, Proc. IEEE, vol. 92, no.12, 2004, pp. 1895-1926.
- [12] A. Brandt, **Algebraic multigrid theory: The symmetric case**, in Preliminary Proceedings for the International Multigrid Conference, Copper Mountain, Colorado, 1983.
- [13] A. Brandt, S. F. McCormick, and J. W. Ruge, **Algebraic multigrid (AMG) for sparse matrix equations**, in **Sparsity and Its Applications**, D. J. Evans, ed., Cambridge University Press, Cambridge, UK, 1985, pp. 257–284.
- [14] M.J. Grote, T. Huckle, **Parallel preconditioning with sparse approximate inverses**, SIAM J. Sci. Comput. 18 (1997) 838–853.
- [15] O. Broeker, M.J. Grote, C. Mayer and A. Reusken, **Robust Parallel Smoothing for Multigrid Via Sparse Approximate Inverses**, SIAM J. Scient. Comput., 23, Nr. 4, 1395-1416, 2001

- [16] Oliver Bröker, Marcus J. Grote, **Sparse approximate inverse smoothers for geometric and algebraic multigrid**, Applied Numerical Mathematics 41 (2002) 61–80
- [17] P. Vanek, J. Mandel, M. Brezina, **Algebraic multigrid based on smoothed aggregation for second and fourth order problems**, Computing, 56 (1996), pp. 179–196.
- [18] M. Brezina, Boulder, and P. Vanek, **A Black-Box Iterative Solver Based on a Two-Level Schwarz Method**, Computing, 63 (1999), pp. 233–263.
- [19] P. Vanek, M. Brezina, J. Mandel, **Convergence of algebraic multigrid based on smoothed aggregation**. Numer. Math. 88, 559-579 (2001).
- [20] E.H. Cuthill and J.M. Mckee, **Reducing the bandwidth of sparse symmetric matrices**, Proc. 24th National Conference of the Association for Computing Machinery, Brondon Press, New Jersey, (1969) 157-172.
- [21] M. Benzi, D.B. Szyld and A. van Duin, **Orderings for incomplete factorization preconditioning of nonsymmetric problems**. SIAM J. Sci. Comput., 20(5) (1999) 1652-1670.
- [22] M. Benzi, W. Joubert and G. Mateesen, **Numerical experiments with parallel orderings for ILU preconditioners**, Electr. Transact. Numer. Anal., 8 (1999) 88-114.
- [23] J. K. Reid, J. A. Scott, **Ordering Symmetric Sparse Matrices for Small Profile and Wavefront**, International Journal for Numerical Methods in Engineering, 45(12), (1999) pp. 1737-1755.

- [24] Watanabe, K., Igarashi, H., Honma, T., **Convergence of multigrid method for edge-based FEM**, IEEE Transactions on Magnetics, Vol. 39 No.3 (2003), pp.1674-6.
- [25] Jinchao Xu. **An introduction to multigrid convergence theory**. In Iterative methods in scientific computing (Hong Kong, 1995), pages 169-241. Springer, Singapore, 1997
- [26] K. Stüben, **Multigrid Tutorial, Multigrid (MG) and Local Refinement for Elliptic Partial Differential Equations**, Schloss Birlinghoven, Augustin, Germany
- [27] C. Kirsch, **Sparse Approximate Inverse Smoothing for Multigrid**, thesis, ETH Zürich, (2001)
- [28] site National Institute of Standards and Technology, "Matrix Market". Available from: <<http://math.nist.gov/MatrixMarket/>>.
- [29] site <<http://libmesh.sourceforge.net/ex3.php>>