

Universidade Federal de Santa Catarina
Curso de Pós-Graduação em Matemática e
Computação Científica

Métodos Iterativos e Multigrid

Anderson Adaime de Borba

Orientador: Prof. Dr. Mario C. Zambaldi

Florianópolis
Fevereiro de 2004

Universidade Federal de Santa Catarina
Curso de Pós-Graduação em Matemática e
Computação Científica

Métodos Iterativos e Multigrid

Dissertação apresentada ao Curso de Pós-Graduação em Matemática e Computação Científica, do Centro de Ciências Físicas e Matemáticas da Universidade Federal de Santa Catarina, para a obtenção do grau de Mestre em Matemática, com Área de Concentração em Matemática Aplicada.

Anderson Adaime de Borba
Florianópolis
Fevereiro de 2004

Métodos iterativos e Multigrid

por

Anderson Adaime de Borba

Esta Dissertação foi julgada para a obtenção do Título de “Mestre”,
área de Concentração em Matemática Aplicada, e aprovada em sua forma
final pelo Curso de Pós-Graduação em Matemática e
Computação Científica.

Prof. Dr. Igor E. Mozolevski
Coordenador

Comissão Examinadora

Prof. Dr. Mário C. Zambaldi (UFSC - Orientador)

Prof. Dr. Marcondes (UFCE)

Prof. Dr. Igor E. Mozolevski (UFSC)

Prof. Dr. Jauber C. de Oliveira (UFSC)

Florianópolis, fevereiro de 2002.

A Deus.
À minha família, por iluminar o meu caminho
quando o crepúsculo se apresentava.
À minha namorada Marcia, pela força, dedicação,
paciência e amor.
E principalmente ao meu pai e grande amigo
Claudio Borba.

Agradecimentos

Primeiramente agradeço à minha família que me acompanhou em todos os caminhos, tanto nas estradas tranqüilas, quanto nas turbulentas.

À Marcia companheira de percurso mais recente, porém de muita importância para tranqüilizar estradas turbulentas.

À minha mãe e também à minha irmã pelo carinho e compreensão.

Aos amigos feito em cada vila que o caminho passou agradeço os sorrisos, a convivência harmônica, e a força dada em percursos difíceis para seguir em frente enfrentando a estrada juntos.

Peço desculpa por não nominar estes amigos, mas tenho certeza que cada um deles entende o momento que unimos nossas força e continuamos o caminho.

Aos professores que ajudaram a pavimentar a estrada tornando-a menos turbulenta, principalmente Rui Charrão e Fermín Bazán que foi um bom companheiro quando estava cansado de seguir.

À Elisa secretária da PG, sempre amável e competente.

Ao CNPq pelo suporte financeiro.

E agradeço especialmente a Mario Zambaldi que de forma muito generosa motivou este caminho tornando mais tranqüilo, para que eu alcançasse este destino.

Sumário

Lista de Figuras	vii
Lista de Tabelas	viii
Resumo	x
Abstract	xi
1 Introdução	1
2 Métodos Iterativos Para Sistemas Lineares	7
2.1 Matriz de Iteração	7
2.2 Resultado Gerais Sobre a Convergência dos Métodos Iterativos	15
2.3 Partição Regular	21
2.4 Matrizes Diagonais Dominantes	23
2.5 Propriedade A e Ordenação Consistente	32
3 Métodos Multigrid	38
3.1 Introdução aos Métodos Multigrid	38
3.1.1 Características dos Métodos Iterativos	38
3.1.2 Multigrid	49
3.2 Elementos dos Métodos Multigrid	52
3.2.1 Nomenclatura Usada Neste Capítulo	52
3.2.2 Operadores de Transferência entre Malhas	54

3.2.3	Introdução ao Ciclo de Duas Malhas	57
3.2.4	O Ciclo Multigrid	65
3.2.5	Estrutura dos Métodos Multigrid	67
4	Resultados Numéricos	70
4.1	Resultados Numéricos para os Métodos Iterativos do Capítulo 2	70
4.2	Resultados Numéricos para os Métodos Multigrid	78
5	Conclusões e Futuras Investigações	84
	Referências Bibliográficas	86

Lista de Figuras

1.1	Ordenação Forward	5
1.2	(a): Ordenação Backward. (b): Ordenação Red Black - Red(R), Black(B).	6
3.1	Estimativa inicial, $n = 64$	40
3.2	(a): JOR com parâmetro $\frac{2}{3}$. (b): Gauss Seidel. (c): Gauss Seidel Red Black.	41
3.3	Estimativa inicial $k_1=1$ $k_2=6$ $k_3=32$, $n=64$	41
3.4	JOR com parâmetro $\frac{2}{3}$ e estimativa inicial $\frac{v_1+v_6+v_{32}}{3}$	42
3.5	(a): JOR com parâmetro $\frac{2}{3}$ e $k = 3$. (b): JOR com parâmetro $\frac{2}{3}$ e $k = 16$. (c): JOR com parâmetro $\frac{2}{3}$ e $\frac{v_2+v_{16}}{2}$	45
3.6	(a): JOR com parâmetro $\frac{2}{3}$. (b): Gauss Seidel,	46
3.7	(a): Estimativa inicial, $k_1 = 1$ e $k_2 = 1$. (b): Gauss Seidel com 5 iterações, $k_1 = 1$ e $k_2 = 1$.(c): Gauss Seidel com 10 iterações, $k_1 = 1$ e $k_2 = 1$	47
3.8	(a): Estimativa inicial, $k_1 = 10$ $k_2 = 10$. (b): Gauss Seidel com 5 iterações, $k_1 = 10$ e $k_2 = 10$. (c): Gauss Seidel com 10 iterações, $k_1 = 10$ e $k_2 = 10$. .	48
3.9	(a): Estimativa inicial $k_1 = 30$ $k_2 = 30$. (b): Gauss Seidel com 5 iterações, $k_1 = 30$ e $k_2 = 30$. (c): Gauss Seidel com 10 iterações, $k_1 = 30$ e $k_2 = 30$. .	49
3.10	(a): Estimativa inicial $\frac{v_{16}+v_{40}}{2}$. (b): Uma iteração na malha fina. (c): Três iterações na malha fina. (d): Uma iteração na malha grossa. (e): Três iterações na malha grossa. (f): Três iterações na malha fina.	51
3.11	(a): Malha grossa usual. (b): Malha semi-grossa em x.	54
3.12	(a): Malha red black. (b): Malha $(3h, 3h)$	55
3.13	Retrição injetiva da malha fina para a grossa	56
3.14	Ciclo V, Ciclo W e FMG	66

Lista de Tabelas

4.1	Problema Modelo I - $tol = 10^{-10}$	72
4.2	Problema Modelo II - $tol = 10^{-8}$	72
4.3	Problema Modelo III - $tol = 10^{-8}$	73
4.4	Problema Modelo IV - $tol = 10^{-8}$	73
4.5	Problema Modelo V - $tol = 10^{-7}$	74
4.6	Problema Modelo VI - $tol = 10^{-7}$	74
4.7	Problema Modelo VII - $tol = 10^{-7}$	75
4.8	Problema Modelo VIII - $tol = 10^{-7}$	75
4.9	Problema Modelo IX - $tol = 10^{-10}$	76
4.10	Problema Modelo X - $tol = 10^{-7}$	76
4.11	Problema I - Razão = $\frac{RBGS}{RBSOR}$	77
4.12	Problema V - Razão = $\frac{RBGS}{RBSOR}$	78
4.13	Problema X - Razão = $\frac{RBGS}{RBSOR}$	78
4.14	Problema Modelo I - $tol = 10^{-10}$	79
4.15	Problema Modelo II - $tol = 10^{-8}$	79
4.16	Problema Modelo III - $tol = 10^{-8}$	79
4.17	Problema Modelo IV - $tol = 10^{-8}$	80
4.18	Problema Modelo V - $tol = 10^{-7}$	80
4.19	Problema Modelo VI - $tol = 10^{-7}$	80
4.20	Problema Modelo VII - $tol = 10^{-7}$	80
4.21	Problema Modelo VIII - $tol = 10^{-7}$	81
4.22	Problema Modelo IX - $tol = 10^{-10}$	81

4.23	Problema Modelo X - $tol = 10^{-7}$	81
4.24	Problema I - Razão = $\frac{RBSOR}{Mgrid}$	82
4.25	Problema V - Razão = $\frac{RBSOR}{Mgrid}$	82
4.26	Problema X - Razão = $\frac{RBSOR}{Mgrid}$	83

Resumo

Este trabalho apresenta uma análise dos métodos iterativos estacionário clássicos e os principais elementos dos métodos Multigrid. Esses métodos são aplicados a problemas de valor de fronteira para Equações Diferenciais Parciais de segunda ordem. Especial atenção é dada aos resultados numéricos que comparam os métodos iterativos e os métodos Multigrid.

Abstract

This work presents an analysis of classic steady-state iterative methods and the main Multigrid methods elements. These methods are applied in boundary value problems to Partial Differential Equations of second order. Special importance is given to numeric results that compare iterative methods to Multigrid methods.

Capítulo 1

Introdução

Nas duas últimas décadas os métodos Multigrid tem sido estudados e implementados para muitos problemas relevantes em computação científica. Sua eficiência tem sido largamente comprovada em computação de grande porte em que sistemas de milhares de variáveis precisam ser resolvidos. Como as idéias Multigrid são baseadas em ações sobre malhas dos domínios do modelo em questão, sua maior aplicabilidade tem sido em sistemas provenientes de discretização de equações diferenciais parciais (EDP). Outra idéia central é que são baseados em propriedades dos métodos iterativos lineares quando aplicados na resolução de modelos com diferentes malhas do domínio.

A escolha desse importante tema para este trabalho foi precedida de um estudo sistematico dos métodos iterativos para resolução de sistemas lineares. Depois disso tivemos dois caminhos a seguir: o estudo dos preconditionadores para métodos não estacionários (Gradientes Conjugados e generalizações) ou Métodos Multigrid. Como estávamos mais interessados em métodos numéricos para resolução de EDPs optamos pelo segundo.

Os primeiros estudos investigando os métodos Multigrid foram feitos por Fedorenko [9, 10] (1961 – 1964) e Bakhvalov [1] (1966). Tanto Fedorenko quanto Bakhvalov investigaram a convergência dos problemas de valor de fronteira de segunda ordem. Fedorenko usou a equação de Poisson nos seus artigos, enquanto Bakhvalov trabalhou com a equação da convecção difusão iniciando o estudo da convergência para as equações elípticas. Em (1980) Hackbusch [13] desenvolveu os elementos fundamentais dos métodos Multigrid, trabalhando com investigações teóricas e aplicações práticas da ordenação red black para os métodos de relaxação. No artigo [17] (1977), Nicolaidis discutiu a conexão dos métodos Multigrid com os métodos de elementos finitos, até então as discussões dos métodos Multigrid eram tratadas em conexão com o esquema de diferenças finitas, como nos artigos de Fedorenko [9, 10], e Bakhvalov [1]. Um artigo clássico e de grande importância para os

métodos Multigrid é o trabalho de Brandt [4] (1977) que reúne idéias dos trabalhos do autor de (1970) até (1977). Neste artigo Brandt investiga os métodos Multigrid lineares, os métodos Multigrid não lineares, técnicas adaptativas aplicadas aos métodos Multigrid, malhas não uniformes, idéias dos métodos Multigrid aplicados nos métodos de elementos finitos, experimentos numéricos para problemas elípticos e problemas de fluxo transônicos.

Na teoria deste trabalho e na construção dos métodos Multigrid foram usados os seguintes livros textos: Briggs [5], Trottenberg [19], Demmel [6], e Wesseling [21]. Os experimentos numéricos são inspirados no artigo de Zhang [25], que consiste num resumo dos trabalhos realizados pelo autor na década de noventa, principalmente os artigos de Gupta, Kouatchou, Zhang [11, 12], respectivamente de Janeiro e Outubro de (1996). Os problemas modelos são descritos em seguida.

Problemas Modelos

Os seguintes problemas modelos serão definidos e usados em todo o trabalho. Estes problemas modelos são encontrados em artigos recentes tratando dos métodos Multigrid.

PROBLEMA MODELO I

$$\begin{cases} -\Delta u = 0, & 0 < x, y < 1 \\ u(x, y) = 0, & \forall x, y \in \partial\Omega \end{cases} \quad (1.1)$$

PROBLEMA MODELO II

$$\begin{cases} -\Delta u = 2(y^2 - y) + 2(x^2 - x), & 0 < x, y < 1 \\ u(x, y) = (x^2 - x)(y^2 - y), & \forall x, y \in \partial\Omega \end{cases} \quad (1.2)$$

PROBLEMA MODELO III

$$\begin{cases} -\Delta u = 0, & 0 < x, y < 1 \\ u(x, y) = 1 + x^2 - y^2, & \forall x, y \in \partial\Omega \end{cases} \quad (1.3)$$

PROBLEMA MODELO IV

$$\begin{cases} -\Delta u = x^2(1-x^2)(2-12y^2) - y^2(1-y^2)(2-12x^2), & 0 < x, y < 1 \\ u(x, y) = x^2y^2(1-x^2)(1-y^2), & \forall x, y \in \partial\Omega \end{cases} \quad (1.4)$$

PROBLEMA MODELO V

$$\begin{cases} -\Delta u = (x^2 + y^2)\exp(xy), & 0 < x, y < 1 \\ u(x, y) = \exp(xy), & \forall x, y \in \partial\Omega \end{cases} \quad (1.5)$$

PROBLEMA MODELO VI

$$\begin{cases} -\Delta u = 52\cos(4x + 6y), & 0 < x, y < 1 \\ u(x, y) = \cos(4x + 6y), & \forall x, y \in \partial\Omega \end{cases} \quad (1.6)$$

PROBLEMA MODELO VII

$$\begin{cases} -\epsilon\Delta u + du_x = x(1-x)\text{sen}(ly), & 0 < x, y < 1 \\ u(x, y) = (ax^2 + bx + c)\text{sen}(ly), & \forall x, y \in \partial\Omega \end{cases} \quad (1.7)$$

PROBLEMA MODELO VIII

$$\begin{cases} \Delta u + \frac{\sigma}{\epsilon}u_x + \frac{\tau}{\epsilon}u_y = 0, & 0 < x, y < 1 \\ u(x, y) = \frac{1}{\exp(a)-1}(\exp(ax) - 1 + \exp(ay) - 1), & \forall x, y \in \partial\Omega \end{cases} \quad (1.8)$$

PROBLEMA MODELO IX

$$\begin{cases} \Delta u + \cos(\alpha)u_x + \sin(\alpha)u_y = 0, & 0 < x, y < 1 \\ u(x, y) = 0, & \forall x, y \in \partial\Omega \end{cases} \quad (1.9)$$

PROBLEMA MODELO X

$$\begin{cases} \Delta u + \frac{p}{\epsilon}u_x + \frac{q}{\epsilon}u_y = f(x, y), & 0 < x, y < 1 \\ u(x, y) = g(x, y), & \forall x, y \in \partial\Omega \end{cases} \quad (1.10)$$

onde

$$g(x, y) = \sin(\pi x) + \sin(6\pi x) + \sin(\pi y) + \sin(6\pi y)$$

$$f(x, y) = f_1(x, y) + f_2(x, y)$$

$$f_1(x, y) = \pi^2 \sin(\pi x) + 36\pi^2 \sin(6\pi x) + \pi^2 \sin(\pi y) + 36\pi^2 \sin(6\pi y)$$

$$f_2(x, y) = \frac{p}{\epsilon}(\pi \cos(\pi x) + 6\pi \cos(6\pi x)) + \frac{q}{\epsilon}(\pi \cos(\pi y) + 6\pi \cos(6\pi y))$$

Os problemas modelos (1.1) até (1.6) são equações da difusão, enquanto, os problemas modelos (1.7) até (1.10) são equações da convecção difusão. Podendo ser escrito da seguinte forma:

$$-\epsilon \Delta u + au_x + bu_y = f \quad (1.11)$$

$$\begin{cases} -\epsilon \Delta u + au_x + bu_y = f, & 0 < x, y < 1 \\ u(x, y) = g(x, y), & \forall x, y \in \partial\Omega \end{cases} \quad (1.12)$$

Os parâmetros a , b e ϵ são constantes fixas.

Discretização e ordenação

A discretização da equação da convecção difusão foi efetuada por diferenças finitas com aproximação de segunda ordem, gerando a seguinte equação de diferenças:

$$a_1 u_{i+1,j} + a_2 u_{i-1,j} + a_3 u_{i,j} + a_4 u_{i,j+1} + a_5 u_{i,j-1} = f_{i,j}. \quad (1.13)$$

em que, a_1, a_2, a_3, a_4 , e a_5 , dependem de a, b, c e ϵ que são os coeficientes da equação da convecção difusão e h é um parâmetro proveniente da discretização de EDP.

Podemos ter a representação “stencil”, da equação de diferenças (1.13):

$$\Delta_h u_h = \frac{1}{h^2} \begin{bmatrix} 0 & a_4 & 0 \\ a_1 & a_3 & a_2 \\ 0 & a_5 & 0 \end{bmatrix}_h u_h(x_i, y_j) = f_h,$$

O parametro h é o tamanho da malha empregada.

Depois de particionado o domínio temos que usar algum tipo de ordenação na malha. Neste trabalho usaremos as seguintes ordenações:

- Ordenação Lexográfica (“Forward”) - Figura 1.1.
- Ordenação “Backward” - Figura 1.2-(a).
- Ordenação “red black” (Tabuleiro de Xadrez) - Figura 1.2-(b).

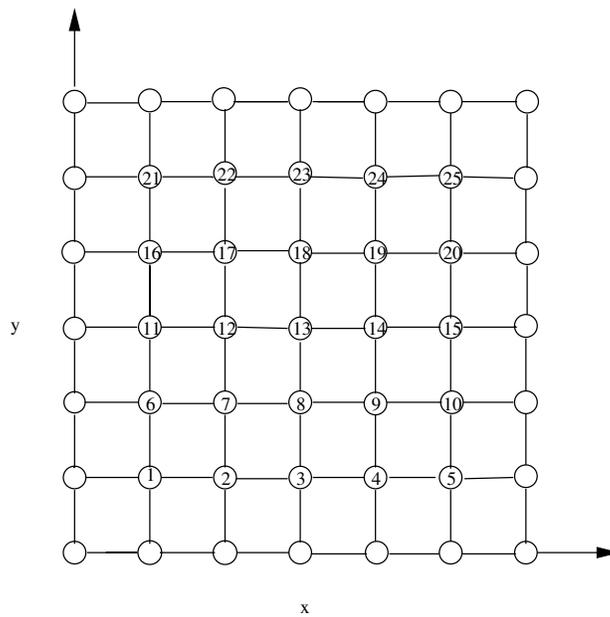


Figura 1.1: Ordenação Forward

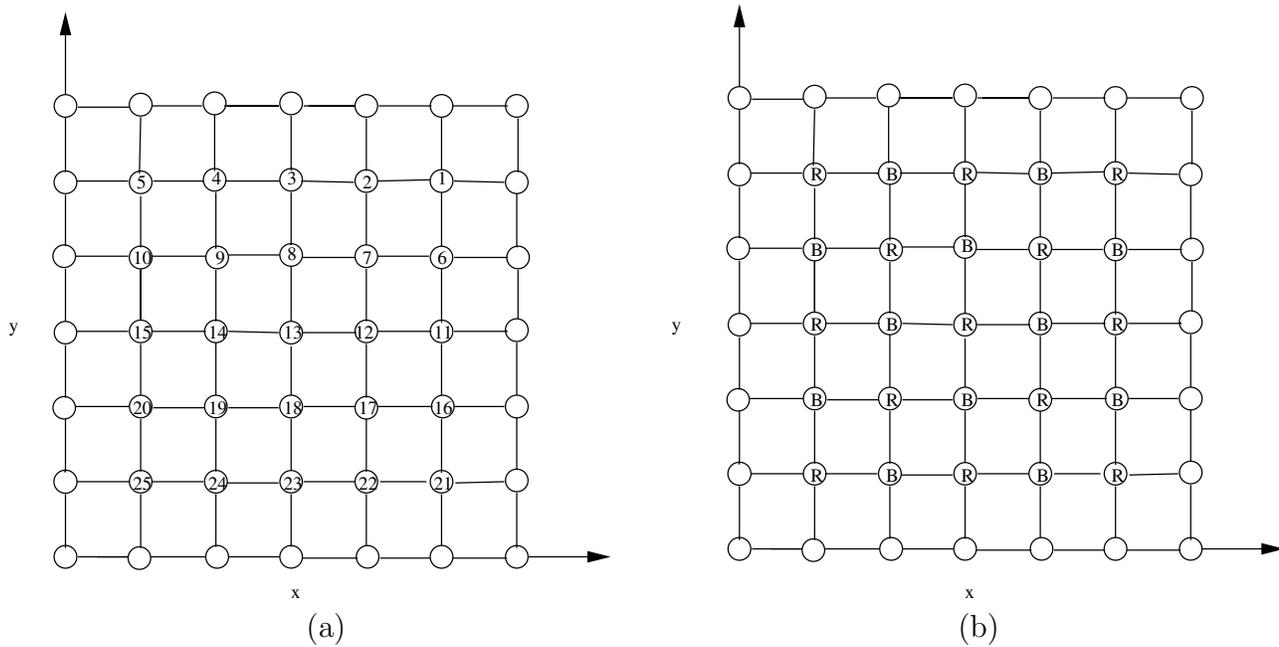


Figura 1.2: (a): Ordenação Backward. (b): Ordenação Red Black - Red(R), Black(B).

Este trabalho tem como objetivo investigar:

- No Capítulo 2 será investigado a construção dos métodos iterativos clássicos e suas principais propriedades de convergência. No decorrer do trabalho quando tivermos nos referindo aos métodos iterativos, serão aqueles que vamos definir neste capítulo.
- A primeira parte do Capítulo 3 mostra algumas propriedades importantes para os métodos iterativos que são fundamentais para o desenvolvimento dos métodos Multigrid, o que é feito na segunda parte deste capítulo. Finalizando este capítulo será demonstrado um importante resultado sobre a construção da matriz de iteração para os métodos Multigrid.
- No capítulo 4 serão mostrados os resultados numéricos para os métodos iterativos e para os métodos Multigrid.
- No capítulo 5 teremos as conclusões e serão citados alguns tópicos para investigações futuras.

Capítulo 2

Métodos Iterativos Para Sistemas Lineares

O objetivo deste capítulo é inicialmente mostrar como podem ser construídos os métodos iterativos clássicos, usados para resolver o sistema linear proveniente da discretização das equações diferenciais.

Depois de construídos os métodos iterativos, podemos questionar sobre a convergência destes métodos. Algumas destas questões serão respondidas posteriormente neste capítulo, onde mostraremos resultados gerais sobre a convergência dos métodos iterativos.

Em cada um dos métodos descritos, seguiremos a rotina, que constrói a seqüência de iteração do método.

2.1 Matriz de Iteração

Particionando $A = M - N$ tal que M é inversível, teremos:

$$Ax = b$$

$$(M - N)x = b$$

$$Mx - Nx = b$$

$$Mx = Nx + b$$

$$x = M^{-1}Nx + M^{-1}b$$

Definindo a matriz de iteração $G = M^{-1}N$ e $f = M^{-1}b$. A seqüência de iteração é

definida de um modo geral da seguinte maneira:

$$x_{k+1} = Gx_k + f$$

O que diferencia os métodos é a maneira como são definidas M e N para gerar a matriz de iteração G e o termo independente f . Mostraremos agora como construímos os métodos iterativos clássicos.

Para o restante deste capítulo vamos usar a seguinte nomenclatura:

D - Matriz Diagonal.

L - Matriz Estritamente Triangular Inferior.

U - Matriz Estritamente Triangular Superior.

Método de Jacobi

No método de Jacobi, a partição é:

$$M = D \quad \text{e} \quad N = (L + U)$$

então

$$A = D - L - U$$

portanto

$$Ax = b$$

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

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

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

$$x = D^{-1}(L + U)x + D^{-1}b.$$

Para gerar o processo iterativo usamos:

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

sendo

$$G_j = D^{-1}(L + U) = M^{-1}N$$

e

$$f_j = D^{-1}b = M^{-1}b$$

com isso escreve-se a seguinte seqüência de iteração:

$$x_{k+1} = G_j x_k + f_j$$

onde G_j é a matriz de iteração do Jacobi.

Método JOR

No método JOR a partição é:

$$M = D \quad \text{e} \quad N = ((1 - \omega)D + \omega(L + U))$$

então

$$A = D - L - U$$

portanto

$$Ax = b$$

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

multiplicando ω e adicionando Dx temos

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

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

$$Dx = -\omega Dx + \omega Lx + \omega Ux + Dx + \omega b$$

$$Dx = \omega(L + U)x + (1 - \omega)Dx + \omega b$$

$$x = \omega D^{-1}(L + U)x + (1 - \omega)x + \omega D^{-1}b$$

sendo $G_j = D^{-1}(L + U)$ temos

$$x = \omega G_j x + (1 - \omega)x + \omega D^{-1}b,$$

logo

$$x = ((1 - \omega)I + \omega G_j)x + \omega D^{-1}b$$

para gerar o processo iterativo usamos:

$$x_{k+1} = G_{j\omega} x_k + f_{j\omega}$$

onde

$$G_{j\omega} = ((1 - \omega)I + \omega G_j) = M^{-1}N$$

e

$$f_{j\omega} = \omega D^{-1}b = M^{-1}b$$

com isso escreve-se a seqüência de iteração na seguinte forma:

$$x_{k+1} = G_{j\omega}x_k + f_{j\omega}$$

Método de Gauss Seidel

Gauss Seidel Forward

No método de Gauss Seidel forward a partição é:

$$M = (D - L) \quad \text{e} \quad N = U$$

então

$$A = D - L - U$$

portanto

$$Ax = b$$

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

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

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

$$x = (D - L)^{-1}Ux + (D - L)^{-1}b$$

para gerar o processo iterativo usamos:

$$x_{k+1} = (D - L)^{-1}Ux_k + (D - L)^{-1}b$$

sendo

$$G_{gsf} = (D - L)^{-1}U = M^{-1}N$$

e

$$f_{gsf} = (D - L)^{-1}b = M^{-1}b$$

escreve-se a seqüência de iteração na seguinte forma:

$$x_{k+1} = G_{gsf}x_k + f_{gsf}$$

Gauss Seidel Backward

No método de Gauss Seidel backward a partição é:

$$M = (D - U) \quad \text{e} \quad N = L$$

então

$$A = D - L - U$$

portanto

$$Ax = b$$

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

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

$$(D - U)x = b + Lx$$

$$x = (D - U)^{-1}Lx + (D - U)^{-1}b$$

para gerar o processo iterativo usamos:

$$x_{k+1} = (D - U)^{-1}Lx_k + (D - U)^{-1}b$$

sendo

$$G_{gsb} = (D - U)^{-1}L = M^{-1}N$$

e

$$f_{gsb} = (D - U)^{-1}b = M^{-1}b$$

escreve-se a seqüência de iteração na seguinte forma:

$$x_{k+1} = G_{gsb}x_k + f_{gsb}$$

Método da Sobre Relaxação Sucessiva(SOR)

SOR Forward

No método SOR forward a partição é:

$$M = (D - \omega L) \quad \text{e} \quad N = ((1 - \omega)D + \omega U)$$

então

$$A = D - L - U$$

$$Ax = b$$

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

multiplicando ω e adicionando Dx temos:

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

$$Dx + \omega Dx - \omega Lx - \omega Ux = Dx + \omega b$$

$$Dx - \omega Lx = Dx - \omega Dx + \omega Ux + \omega b$$

$$x = (D - \omega L)^{-1}((1 - \omega)D + \omega U)x + (D - \omega L)^{-1}\omega b$$

sendo

$$G_{sf} = (D - \omega L)^{-1}((1 - \omega)D + \omega U) = M^{-1}N$$

e

$$f_{sf} = (D - \omega L)^{-1}\omega b = M^{-1}b$$

logo

$$x = G_{sf}x + f_{sf}$$

com isso escrevemos a seqüência de iteração na seguinte forma:

$$x_{k+1} = G_{sf}x_k + f_{sf}$$

SOR Backward

No método SOR backward a partição é:

$$M = (D - \omega U) \quad \text{e} \quad N = ((1 - \omega)D + \omega L)$$

então

$$A = D - L - U$$

portanto

$$Ax = b$$

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

multiplicando ω e adicionando Dx temos:

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

$$Dx + \omega Dx - \omega Lx - \omega Ux = Dx + \omega b$$

$$Dx - \omega Ux = Dx - \omega Dx + \omega Lx + \omega b$$

$$x = (D - \omega U)^{-1}((1 - \omega)D + \omega L)x + (D - \omega U)^{-1}\omega b$$

sendo

$$G_{sb} = (D - \omega U)^{-1}((1 - \omega)D + \omega L) = M^{-1}N$$

e

$$f_{sb} = (D - \omega U)^{-1}\omega b = M^{-1}b$$

logo

$$x = G_{sb}x + f_{sb}$$

com isso escrevemos a seqüência de iteração na seguinte forma:

$$x_{k+1} = G_{sb}x_k + f_{sb}$$

Método da Sobre Relaxação Simétrica Sucessiva(SSOR)

Este esquema consiste em um passo SOR forward da seguinte forma:

$$G_{sf} = (D - \omega L)^{-1}((1 - \omega)D + \omega U)$$

e

$$f_{sf} = (D - \omega L)^{-1}\omega b$$

$$x_{k+\frac{1}{2}} = G_{sf}x_k + f_{sf}$$

e depois um passo SOR backward

$$G_{sb} = (D - \omega U)^{-1}((1 - \omega)D + \omega L)$$

e

$$f_{sb} = (D - \omega U)^{-1}\omega b$$

logo

$$x_{k+1} = G_{sb}x_{k+\frac{1}{2}} + f_{sb}$$

Podemos gerar a seguinte seqüência de iteração:

$$x_{k+1} = G_{sb}x_{k+\frac{1}{2}} + f_{sb}$$

$$x_{k+1} = G_{sb}(G_{sf}x_k + f_{sf}) + f_{sb}$$

$$x_{k+1} = G_{sb}G_{sf}x_k + G_{sb}f_{sf} + f_{sb}$$

com isso teremos

$$G_{ss} = G_{sb}G_{sf}$$

e

$$f_{ss} = G_{sb}f_{sf} + f_{sb}$$

isto é

$$G_{ss} = (D - \omega U)^{-1}((1 - \omega)D + \omega L)(D - \omega L)^{-1}((1 - \omega)D + \omega U)$$

$$G_{ss} = (D - \omega U)^{-1}(D - \omega L)^{-1}((1 - \omega)D + \omega L)((1 - \omega)D + \omega U)$$

$$f_{ss} = G_{sb}f_{sf} + f_{sb}$$

$$f_{ss} = (D - \omega U)^{-1}(D - \omega L)^{-1}((1 - \omega)D + \omega L)\omega b + (D - \omega U)^{-1}\omega b$$

$$f_{ss} = (D - \omega U)^{-1}\omega b((D - \omega L)^{-1}((1 - \omega)D + \omega L) + I)$$

$$f_{ss} = (D - \omega U)^{-1}\omega b((D - \omega L)^{-1}(D - \omega D + \omega L) + I)$$

$$f_{ss} = (D - \omega U)^{-1}\omega b((D - \omega L)^{-1}(D + D - D - \omega D + \omega L) + I)$$

$$f_{ss} = (D - \omega U)^{-1}\omega b((D - \omega L)^{-1}((2 - \omega)D + (\omega L - D) + I))$$

$$f_{ss} = (D - \omega U)^{-1} \omega b ((D - \omega L)^{-1} ((2 - \omega)D - (D - \omega L) + I))$$

$$f_{ss} = (D - \omega U)^{-1} \omega b ((D - \omega L)^{-1} (2 - \omega)D - I + I)$$

$$f_{ss} = (D - \omega U)^{-1} \omega b ((D - \omega L)^{-1} (2 - \omega)D)$$

$$f_{ss} = \omega(2 - \omega)(D - \omega U)^{-1} D (D - \omega L)^{-1} b$$

A seqüência de iteração é gerada por:

$$x_{k+1} = G_{ss} x_k + f_{ss}$$

2.2 Resultado Gerais Sobre a Convergência dos Métodos Iterativos

Esta seção tem como objetivo principal responder às seguintes questões:

- (i)- Se a iteração converge será que o limite é solução do sistema linear?
- (ii)- Sobre quais condições esta iteração converge?
- (iii)- Se converge com que velocidade isso acontece?

Para estudar a convergência dos métodos iterativos consideremos a seqüência do processo iterativo.

$$x_{k+1} = Gx_k + f \text{ sendo } G = M^{-1}N \text{ e } f = M^{-1}b .$$

Para responder ao item (i), vamos considerar que a seqüência converge, isto é

$$\lim_{k \rightarrow \infty} x_{k+1} = \lim_{k \rightarrow \infty} x_k = x$$

seja

$$x_{k+1} = Gx_k + f$$

$$\lim_{k \rightarrow \infty} x_{k+1} = \lim_{k \rightarrow \infty} Gx_k + f = G \lim_{k \rightarrow \infty} x_k + f = Gx + f$$

portanto

$$x = Gx + f \implies x - Gx = f \implies (I - G)x = f$$

como $G = M^{-1}N$ e $f = M^{-1}b$, podemos escrever

$$(I - M^{-1}N)x = M^{-1}b \implies x - M^{-1}Nx = M^{-1}b \implies x = M^{-1}Nx + M^{-1}b$$

$$\implies Mx = Nx + b \implies Mx - Nx = b$$

$$\implies (M - N)x = b \implies Ax = b$$

Logo, se a seqüência converge para x , o limite x é solução de $Ax = b$, isto responde o item (i). Para responder aos outros itens é preciso demonstrar alguns teoremas.

Para tanto, temos que fazer a seguinte observação:

tomando

$$x_{k+1} = Gx_k + f \tag{2.1}$$

e

$$x = Gx + f \tag{2.2}$$

e subtraindo (2.2) de (2.1)

$$x_{k+1} - x = G(x_k - x)$$

$$x_{k+1} - x = G(Gx_{k-1} + f - Gx - f) \implies x_{k+1} - x = G^2(x_{k-1} - x)$$

$$x_{k+1} - x = G^2(Gx_{k-2} + f - Gx - f) \implies x_{k+1} - x = G^3(x_{k-2} - x)$$

$$\vdots$$

$$x_{k+1} - x = G^k(Gx_0 + f - Gx - f) \implies x_{k+1} - x = G^{k+1}(x_0 - x)$$

$$x_{k+1} - x = G^{k+1}(x_0 - x) \quad (2.3)$$

Teorema 1. *A seqüência G^k converge para zero se e somente se $\rho(G) < 1$.*

Demonstração: (\implies)

Se

$$\lim_{k \rightarrow \infty} G^k = 0 \implies \lim_{k \rightarrow \infty} \|G^k\| = 0,$$

considere agora o autovalor λ de maior módulo e seu respectivo autovetor u_1 , tal que u_1 é unitário,

$$|\lambda^k| = |\lambda^k| \|u_1\| = \|\lambda^k u_1\| = \|G^k u_1\| \leq \|G^k\| \|u_1\| = \|G^k\|$$

$$\lim_{k \rightarrow \infty} |\lambda^k| \leq \lim_{k \rightarrow \infty} \|G^k\| = 0 \text{ por hipótese, portanto } \rho(G) < 1$$

(\impliedby)

se $\rho(G) < 1$, fatorando G usando a decomposição de Jordan, temos

$$G = X J X^{-1}$$

com isso

$$G^k = X J^k X^{-1}$$

para mostrar que $\lim_{k \rightarrow \infty} G^k = 0$, é suficiente mostrar

$$\lim_{k \rightarrow \infty} J^k = 0$$

isto é,

$$\lim_{k \rightarrow \infty} \|J\|^k = 0.$$

Sabemos que a matriz de Jordan é formada por blocos, deste modo basta mostrar que cada bloco de Jordan J_i converge para zero.

Cada bloco de Jordan é da seguinte forma $J_i = \lambda_i I + E_i$, onde I e E_i são respectivamente a matriz identidade e uma matriz nilpotente de índice l_i , isto é, $E_i^{l_i} = 0$. Então:

$$(J_i)^k = (\lambda_i I + E_i)^k = \sum_{j=1}^k \frac{k!}{j!(k-j)!} \lambda_i^{k-j} E_i^j = \sum_{j=1}^{l_i-1} \frac{k!}{j!(k-j)!} \lambda_i^{k-j} E_i^j$$

pois $k > l_i \implies E_i^k = 0$

$$\| J_i^k \| = \left\| \sum_{j=1}^{l_i-1} \frac{k!}{j!(k-j)!} \lambda_i^{k-j} E_i^j \right\| \leq \sum_{j=1}^{l_i-1} \frac{k!}{j!(k-j)!} |\lambda_i|^{k-j} \| E_i \|^j \rightarrow 0$$

pois $\rho(G) < 1$, e $\lim_{k \rightarrow \infty} |\lambda|^k = 0$ por hipótese, sendo que temos uma soma finita em que cada termo vai para zero, portanto:

$$(\forall i) \quad \lim_{k \rightarrow \infty} \| J_i^k \| = 0 \quad \implies \lim_{k \rightarrow \infty} \| J^k \| = 0 \quad \implies \lim_{k \rightarrow \infty} \| G \|^k = 0$$

logo,

$$\lim_{k \rightarrow \infty} G^k = 0$$

□

Teorema 2. A série $\sum_{k=0}^{\infty} G^k$ converge se e somente se $\rho(G) < 1$. Sobre as mesmas condições $I - G$ é não singular e o limite da série é $(I - G)^{-1}$.

Demonstração: (\implies)

se $\sum_{k=0}^{\infty} G^k$ converge $\implies \lim_{k \rightarrow \infty} \| G \|^k = 0 \implies$, pelo Teorema 1, $\rho(G) < 1$

(\impliedby)

se $\rho(G) < 1$

$$I - G^{k+1} = (I - G)(I + G^2 + G^3 + \dots + G^k)$$

novamente usando o Teorema 1 temos $\lim_{k \rightarrow \infty} G^{k+1} = 0$, dessa forma

$I = (I - G) \sum_{k=0}^{\infty} G^k$, podemos afirmar que:

$$(I - G)^{-1} = \sum_{k=0}^{\infty} G^k$$

portanto a série que converge para o limite $(I - G)^{-1}$ e $(I - G)$ é não singular.

□

Teorema 3. *Seja G uma matriz quadrada tal que $\rho(G) < 1$, então $(I - G)$ é não singular e a iteração $x_{k+1} = Gx_k + b$ converge, sendo $x_{(0)}$ a estimativa inicial arbitrária. Inversamente se a iteração $x_{k+1} = Gx_k + b$ converge, com estimativa inicial $x_{(0)}$ arbitrária então $\rho(G) < 1$.*

Demonstração: (\implies)

Pelo Teorema 2, se $\rho(G) < 1$ então $(I - G)$ é não singular

Da equação (2.3) temos:

$$x_{k+1} - x = G^{k+1}(x_0 - x)$$

logo,

$$\|x_{k+1} - x\| = \|G^k(x_0 - x)\| \leq \|G^k\| \|x_0 - x\| \leq \|G\|^k \|x_0 - x\|$$

por hipótese e pelo Teorema 1, temos:

$$\lim_{k \rightarrow \infty} \|G\|^k = 0$$

então

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x\| = 0$$

$$\lim_{k \rightarrow \infty} \|x_{k+1}\| = \|x\| \quad \forall x_0$$

portanto a seqüência x_k

$$x_{k+1} = Gx_k + b$$

converge para qualquer x_0 .

(\impliedby)

Se

$$x_{k+1} = Gx_k + b$$

converge para todo x_0 , dessa forma

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x\| = 0$$

isto é,

$$\lim_{k \rightarrow \infty} x_{k+1} = x$$

logo

$$\lim_{k \rightarrow \infty} G^{k+1}(x_0 - x) = 0 \implies \lim_{k \rightarrow \infty} G^{k+1}v = 0,$$

onde $v = x_0 - x$ é arbitrário $\implies \lim_{k \rightarrow \infty} G^{k+1} = 0$ então pelo Teorema 1, $\rho(G) < 1$.

□

Corolário 1. *Seja G uma matriz quadrada tal que $\|G\| < 1$ então $(I - G)$ é não singular e a iteração $x_{k+1} = Gx_k + b$ converge.*

Com esses teoremas respondemos ao item (ii), isto é, sobre quais condições o processo iterativo converge.

Agora, vamos responder o item (iii), qual é a velocidade de convergência do processo iterativo?

Considerando a equação (2.3),

$$x_k - x = G^k(x_0 - x),$$

definindo $d_k = x_k - x$ e $d_0 = x_0 - x$, podemos escrever $d_k = G^k d_0$. Usando a decomposição de Jordan, temos que $G = XJX^{-1}$. Portanto $G^k = XJ^kX^{-1}$.

Reescrevendo:

$$d_k = G^k d_0 = XJ^kX^{-1}d_0 = \frac{\lambda^k}{\lambda^k} XJ^kX^{-1}d_0 = \lambda^k X(J/\lambda)^k X^{-1}d_0,$$

podemos assumir que existe um único autovalor $\lambda_i \neq 0$ de G sendo i fixo tal que $1 \leq i \leq n$, n é a dimensão da matriz G e λ_i autovalor de maior módulo. Assim,

$$\lim_{k \rightarrow \infty} \left\| \left(\frac{J}{\lambda_i} \right)^k \right\| = 0$$

para todos os blocos, exceto o bloco associado com o λ_i , pois

$$\left| \frac{\lambda_k}{\lambda_i} \right|^k < 1.$$

Logo:

$$\lim_{k \rightarrow \infty} \left| \frac{\lambda_k}{\lambda_i} \right|^k = 0.$$

Seja o bloco de Jordan de dimensão p da forma $J_{\lambda_i} = \lambda_i I + E$, onde E é nilpotente de ordem p , isto é, $E^p = 0$

$$(J_{\lambda_i})^k = (\lambda_i I + E)^k = \lambda_i^k (I + \lambda^{-1} E)^k = \lambda_i^k \sum_{i=0}^n \frac{k!}{i!(k-i)!} \lambda^{-i} E^i$$

$$(J_{\lambda_i})^k = \lambda_i^k \sum_{i=0}^{p-1} \frac{k!}{i!(k-i)!} \lambda^{-i} E^i.$$

O termo dominante é o último pois $p - 1$ é o maior expoente de λ na soma finita acima. Podemos fazer uma aproximação de $J_{\lambda_i}^k$, da seguinte forma:

$$J_{\lambda_i}^k \approx \frac{(p-1)!}{(k-p+1)!} \lambda^{k-p+1} E^{p-1}$$

Isto implica que

$$\|d_k\| \approx c(E, d_0, X) |\lambda|^{k-p+1}$$

onde $c(E, d_0, X)$ é uma constante que depende de E , d_0 e X . Podemos afirmar que a velocidade de decaimento de $\|d_k\|$ depende do maior autovalor em módulo da matriz A .

2.3 Partição Regular

Definição 1. Uma matriz A é não negativa se $A \geq 0$, isto é, $(\forall i, j) a_{i,j} \geq 0$

Definição 2. Uma matriz A é uma M -matriz se satisfaz as seguintes propriedades:

- 1- $(\forall i) a_{i,i} > 0$
- 2- $(\forall i, j) a_{i,j} \leq 0$ se $i \neq j$
- 3- A é não singular
- 4- $A^{-1} \geq 0$

Definição 3. Seja A, M, N três matrizes tal que $A = M - N$. O par de matrizes M, N é uma partição regular de A se M é não singular e M^{-1} e N são não negativas

Teorema 4. Seja G uma matriz não negativa e $\rho(G) < 1$ então $(I - G)$ é não singular e $(I - G)^{-1}$ é não negativa.

Demonstração: Por hipótese $\rho(G) < 1$, sabemos pelo Teorema 2 que $(I - G)$ é não singular e que

$$(I - G)^{-1} = \sum_{k=0}^{\infty} G^k.$$

Como $G \geq 0$, então $G^k \geq 0$. Assim $(I - G)^{-1} \geq 0$.

□

Agora será demonstrado uma generalização do Teorema 2.

Teorema 5. *Seja M, N uma partição regular de A , então $\rho(M^{-1}N) < 1$, se e somente se A é não singular e A^{-1} é não negativa.*

Demonstração: (\implies)

Se $G = M^{-1}N$, $\rho(G) < 1$ então $(I - G)$ é não singular e $(I - G)^{-1}$ é não negativa pelo Teorema 4, com isso $A = M - N = M(I - M^{-1}N) = M(I - G)$, e $AA^{-1} = M(I - G)(M(I - G))^{-1} = M(I - G)(I - G)^{-1}M^{-1} = MIM^{-1} = I$, logo A é não singular. Sendo $A = M(I - G)$ e $A^{-1} = (I - G)^{-1}M^{-1}$ então A^{-1} é não negativa pois $(I - G)^{-1}$ e M^{-1} são não negativas.

(\impliedby)

Se A é não singular e A^{-1} é não negativa e ainda M, N é uma partição regular, isto é, M é não singular e M^{-1} e N são não negativas. E ainda,

$$A^{-1}N = (I - M^{-1}N)^{-1}M^{-1}N = (I - G)^{-1}G$$

sabemos que:

$$Gx = \rho(G)x$$

e

$$(I - G)^{-1}x = \frac{1}{1 - \rho(G)}x$$

então

$$A^{-1}Nx = (I - G)^{-1}Gx = (I - G)^{-1}\rho(G)x = \frac{\rho(G)}{1 - \rho(G)}x$$

logo,

$$A^{-1}Nx = \frac{\rho(G)}{1 - \rho(G)}x.$$

Considerando o Teorema de Peron-Frobenius, que afirma: "se uma matriz é não negativa temos a garantia de que o autovetor x associado ao seu respectivo autovalor, é não negativo". Portanto podemos afirmar que x é não negativo, e o fato de $A^{-1}N$ ser não negativo, Temos:

$$\frac{\rho(G)}{1 - \rho(G)} \geq 0$$

desse modo:

$$\rho(G) \leq 0 \text{ e } 1 - \rho(G) < 0 \implies \rho(G) > 1 \text{ o que é inconsistente}$$

ou

$$\rho(G) \geq 0 \text{ e } 1 - \rho(G) > 0 \implies \rho(G) < 1, \text{ portanto}$$

$$\rho(G) < 1, \text{ isto é, } \rho(M^{-1}N) < 1.$$

□

Corolário 2. *Se M, N é uma partição regular de A , e se M é uma M -matriz, então $x_{k+1} = M^{-1}Nx_k + M^{-1}b$ sempre converge.*

2.4 Matrizes Diagonais Dominantes

Definição 4. *A é uma matriz irredutível, se não existe uma matriz P de permutação tal que:*

$$PAP^T = \begin{bmatrix} A_{1,1} & A_{1,2} \\ 0 & A_{2,2} \end{bmatrix},$$

onde $A_{i,j}$ são matrizes em blocos.

Definição 5. *Uma matriz A é*

- *Fracamente diagonal dominante se*

$$|a_{j,j}| \geq \sum_{\substack{i=1 \\ i \neq j}}^n |a_{i,j}|, \quad j = 1, \dots, n$$

- *Estritamente diagonal dominante se*

$$|a_{j,j}| > \sum_{\substack{i=1 \\ i \neq j}}^n |a_{i,j}|, \quad j = 1, \dots, n$$

- *Irredutível diagonal dominante se A é irredutível e*

$$|a_{j,j}| \geq \sum_{\substack{i=1 \\ i \neq j}}^n |a_{i,j}|, \quad j = 1, \dots, n$$

com desigualdade estrita para algum j .

Teorema 6 (Gershgorin). Cada autovalor λ da matriz A é localizado em um dos discos fechados no plano complexo centrado em $a_{i,i}$ e tendo raio $\rho_i = \sum_{\substack{i=1 \\ i \neq j}}^n |a_{i,j}|$ ou seja

$$(\forall) \lambda \in \sigma(A), \exists i \text{ tal que } |\lambda - a_{i,i}| \leq \sum_{\substack{i=1 \\ j \neq i}}^n |a_{i,j}|.$$

Demonstração: Seja $\frac{x}{\|x\|}$ o autovetor associado com o maior autovalor em módulo, e seja m o índice do vetor x , tal que $|x_m| = 1$ e $|x_i| < 1$ se $i \neq m$

$$Ax = \lambda x \implies Ax - \lambda x = 0 \implies (A - \lambda I)x = 0$$

seja m fixado arbitrariamente:

$$(a_{m,m} - \lambda)x_m + \sum_{\substack{i=1 \\ i \neq m}}^n a_{m,i}x_i = 0 \implies (a_{m,m} - \lambda)x_m = -\sum_{\substack{i=1 \\ i \neq m}}^n a_{m,i}x_i$$

$$\begin{aligned} |(a_{m,m} - \lambda)| &= |(a_{m,m} - \lambda)||x_m| = |(a_{m,m} - \lambda)x_m| = \\ &= \left| -\sum_{\substack{i=1 \\ i \neq m}}^n a_{m,i}x_i \right| \leq \sum_{\substack{i=1 \\ i \neq m}}^n |a_{m,i}||x_i| \leq \sum_{\substack{i=1 \\ i \neq m}}^n |a_{m,i}|. \end{aligned}$$

Portanto

$$\forall \lambda \in \sigma(A), \exists i \text{ tal que } |(a_{m,m} - \lambda)| \leq \sum_{\substack{i=1 \\ i \neq m}}^n |a_{m,i}|$$

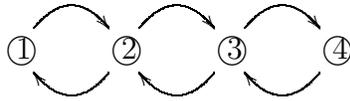
□

Definição 6. Um grafo dirigido é uma coleção finita de nós conectados por uma coleção finita de eixos dirigidos. Um caminho em um grafo dirigido é uma seqüência de nós n_1, \dots, n_m com um eixo indo de n_i para n_{i+1} .

Definição 7. Um grafo dirigido de A , $G(A)$, é um grafo com nós $1, 2, \dots, n$ e um eixo indo do nó i para o nó j se e somente se $a_{i,j} \neq 0$

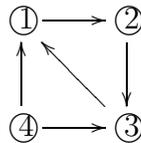
Exemplo 1 :

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix},$$



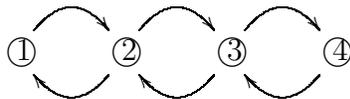
Exemplo 2 :

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 1 \end{bmatrix},$$



Definição 8. Um grafo dirigido de A , $G(A)$ é chamado de fortemente conectado, se existe um caminho de todo nó i para todo nó j .

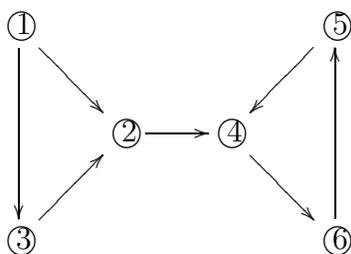
Exemplo 3 : Considerando o grafo da matriz do exemplo 1,



podemos afirmar que este grafo é fortemente conectado.

Exemplo 4 :

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$



O grafo anterior não é fortemente conectado.

Teorema 7. *A é irredutível se e somente se $G(A)$ é fortemente conectado.*

Demonstração: (\implies)

Vamos supor que $G(A)$ não é fortemente conectado. Nesse caso, renumeramos as linhas e colunas de maneira que os nós fortemente conectados venham primeiro. Logo existe uma matriz P de permutação, tal que PAP^T é bloco triangular, portanto A é irredutível.

(\impliedby)

Consideremos que:

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix},$$

é redutível, com isso podemos escrever

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ 0 & A_{2,2} \end{bmatrix},$$

porém, claramente, os blocos $A_{1,1}$ e $A_{2,2}$ não são conectados, logo $G(A)$ não é fortemente conectado.

□

A matriz do Exemplo 4 é redutível, pois sabemos que A não é fortemente conectado.

Teorema 8. *Seja A uma matriz irredutível e assumamos que os autovalores λ de A estão na fronteira da união dos n discos de Gershgorin, então λ está na fronteira de todos os discos.*

Demonstração: Seja x um autovetor associado a λ com $|x_m| = 1$ e $|x_i| \leq 1$ para $i \neq m$. Supondo agora que λ pertença ao disco m , isto é,

$\forall \lambda \in \sigma(A), \exists m$ tal que:

$$|\lambda - a_{m,m}| \leq \sum_{\substack{j=1 \\ j \neq m}}^n |a_{m,j}|.$$

Definindo:

$$B = \bigcup_{i=1}^n D(\lambda, \rho_i)$$

temos, por hipótese, que $\lambda \in \partial B$, sendo assim, λ não pode estar no interior do disco m , portanto:

$$|\lambda - a_{m,m}| = \sum_{\substack{i=1 \\ i \neq m}}^n |a_{m,i}| |x_i| = \sum_{\substack{i=1 \\ i \neq m}}^n |a_{m,i}| = \rho_m.$$

Se A é irredutível então $G(A)$ é fortemente conectado, logo existe um caminho $m, m_1, m_2, \dots, m_k = j$. Ainda temos, por definição, que $a_{m,m_i} \neq 0$. Pela igualdade acima, temos que $|x_j| = 1$ para todo $x_j \neq 0$, logo $|x_{m_1}| = 1$.

Repetindo o argumento, e trocando m por m_1 , obtemos a seguinte igualdade:

$$|\lambda - a_{m_1,m_1}| = \sum_{\substack{j=1 \\ j \neq m_1}}^n |a_{m_1,j}| |x_j| = \sum_{\substack{j=1 \\ j \neq m_1}}^n |a_{m_1,j}| = \rho_{m_1}$$

A continuidade deste argumento mostra que:

$$|\lambda - a_{m_i,m_i}| = \rho_{m_i}$$

para $i = 1, \dots, k$. Portanto λ pertence a fronteira de todos os discos. □

Teorema 9. *Se uma matriz A é estritamente diagonal dominante ou irredutível diagonal dominante, então A é não singular.*

Demonstração: Suponhamos que A é estritamente diagonal dominante e que A é singular, isto é, $\lambda = 0$ é autovalor de A . Então, para cada i temos, pelo Teorema de Gershgorin, que $|a_{i,i}| \leq \rho_i$, porém, por hipótese, A é estritamente diagonal dominante com isso $\lambda = 0$ não é autovalor de A , assim A é não singular.

Considerando agora A irredutível diagonal dominante e supondo A singular, logo $\lambda = 0$ é autovalor de A temos, pelo Teorema de Gershgorin, que $|a_{i,i}| \leq \rho_i$; por A ser irredutível

$|a_{i,i}| \geq \rho_i$ então $|\lambda - a_{i,i}| = \rho_i$, logo $\lambda = 0$ pertence à fronteira da união dos discos de Gershgorin, portanto pertence à fronteira de todos os discos de Geshgorin, dessa forma:

$$|\lambda - a_{i,i}| = |a_{i,i}| = \sum_{\substack{j=1 \\ i \neq j}}^n |a_{i,j}|, \forall j = 1, \dots, n \text{ porém isso contradiz o fato de ser irredutível}$$

diagonal dominante. □

Teorema 10. *Se A é estritamente diagonal dominante ou é irredutível diagonal dominante, então a iteração de Jacobi e Gauss Seidel converge para qualquer x_0 dado como estimativa inicial.*

Demonstração: Primeiro temos que provar o resultado para matrizes estritamente diagonal dominante. Seja λ o autovalor dominante de $G_j = D^{-1}(L+U)$ e $G_{gs} = (D-L)^{-1}U$. Considerando $|x_m| = 1$ e $|x_i| \leq 1 \forall i \neq m$ e pelo Teorema de Gershgorin temos:

$$|\lambda| \leq \left(\sum_{\substack{j=1 \\ j \neq i}}^n |a_{i,j}| \right) \frac{1}{|a_{i,i}|}, \text{ mas } |a_{i,i}| > \left(\sum_{\substack{j=1 \\ j \neq i}}^n |a_{i,j}| \right) \frac{1}{|a_{i,i}|}, \text{ logo } |\lambda| < 1 \text{ então } \rho(A) < 1, \text{ com}$$

isso Jacobi converge para toda estimativa inicial.

Agora vamos considerar

$$G_{gs}x = \lambda x \implies (D-L)^{-1}Ux = \lambda x \implies Ux = \lambda(D-L)x = \lambda(Dx - Lx)$$

de forma que:

$$\sum_{j>m}^n a_{m,j}x_j = \lambda \left[a_{m,m}x_m - \sum_{j<m}^n a_{m,j}x_j \right], \quad \lambda = \frac{\sum_{j>m}^n a_{m,j}x_j}{a_{m,m}x_m - \sum_{j<m}^n a_{m,j}x_j}.$$

Portanto:

$$|\lambda| \leq \frac{\sum_{j>m}^n |a_{m,j}|}{|a_{m,m}| - \sum_{j<m}^n |a_{m,j}|}$$

Definindo $d = |a_{m,m}|$ e $\sigma_2 = \sum_{j>m}^n |a_{m,j}|$ e $\sigma_1 = \sum_{j<m}^n |a_{m,j}|$ teremos:

$$|\lambda| \leq \frac{\sigma_2}{d-\sigma_1} = \frac{\sigma_2}{d-\sigma_1+\sigma_2-\sigma_2} = \frac{\sigma_2}{\sigma_2+(d-\sigma_1-\sigma_2)},$$

como

$$|a_{m,m}| > \sum_{\substack{j=1 \\ j \neq m}}^n |a_{m,j}| \implies (d - \sigma_1 - \sigma_2) > 0 \text{ então } |\lambda| < 1, \text{ portanto a itera\c{c}\~{a}o de Gauss}$$

Seidel converge.

Agora vamos considerar a matriz A irredutível diagonal dominante, e $M^{-1}N = G$. Sabemos pela afirmação acima que $\rho(M^{-1}N) = \rho(G) \leq 1$, ou seja, necessitamos que $\rho(G) < 1$, portanto, assumamos que $|\lambda| = 1$ é um autovalor de $M^{-1}N$. Assim a matriz $M^{-1}N - \lambda I$ é singular, sendo $M^{-1}(N - \lambda M) \implies A_1 = N - \lambda M$. A matriz A_1 também é singular. Como, $\lambda = 1$ ou $\lambda = -1$ temos:

$$A_1 = N - M = -A \text{ e } A_1 = N + M = A$$

Logo A_1 é irredutível e singular, contradizendo o Teorema 9,

Portanto $\rho(M^{-1}N) < 1$.

□

Observação 1. Para a seqüência deste capítulo vamos usar a notação

$$G_{sf} = G_{sor} = (D - \omega L)^{-1}((1 - \omega)D + \omega U)$$

para a matriz de iteração do método SOR forward.

Teorema 11. $\rho(G_{sor}) \geq |\omega - 1|$, portanto a convergência ocorre se $0 < \omega < 2$.

Demonstração: Vamos definir a função $\phi(\lambda) = \det(\lambda I - G_{sor})$. Substituindo G_{sor} , nesta equação e usando o seguinte fato:

$$1 = \det(\lambda I - G_{sor}) = \det(I - \omega L) \det(\lambda I - G_{sor}) = \det((I - \omega L)(\lambda I - G_{sor}))$$

temos:

$$\begin{aligned} \phi(\lambda) &= \det(\lambda I - G_{sor}) = \det((I - \omega L)(\lambda I - G_{sor})) \\ &= \det(\lambda I - G_{sor} - \omega \lambda L + \omega L G_{sor}) = \det(\lambda I - \omega \lambda L + (\omega L - I)G_{sor}) \\ &= \det(\lambda I - \omega \lambda L - (I - \omega L)G_{sor}) = \det(\lambda I - \omega \lambda L - (I - \omega L)(I - \omega L)^{-1} \\ &\quad ((1 - \omega)I + \omega U)) = \det(\lambda I - \omega \lambda L - ((1 - \omega)I + \omega U)) \\ &= \det(\lambda I - \omega \lambda L - (I - \omega I + \omega U)) = \det(\lambda I - \omega \lambda L - I + \omega I - \omega U) \end{aligned}$$

$$\begin{aligned}
&= \det((\lambda - 1 + \omega)I - \omega(\lambda L + U)) \implies \\
&\phi(\lambda) = \det((\lambda - 1 + \omega)I - \omega(\lambda L + U)) \\
&\phi(\lambda) = \lambda^2 + \text{tr}(A)\lambda + \det(A) \implies \phi(0) = \det(A).
\end{aligned}$$

Portanto:

$$\begin{aligned}
\phi(0) = \det((\omega - 1)I) = (\omega - 1)^n \implies (\omega - 1)(\omega - 1) \dots (\omega - 1) = \lambda_1 \lambda_2 \dots \lambda_n \\
\implies \max_{0 \leq i \leq n} |\lambda_i| \geq |\omega - 1| \implies \rho(G_{sor}) \geq |\omega - 1|
\end{aligned}$$

□

Teorema 12. *Se A é uma matriz simétrica positiva definida então*

$$\rho(G_{sor}) < 1,$$

e o SOR converge, $\forall 0 < \omega < 2$.

Demonstração: Considere $G_{sor} = P$ e $A = D - L - U$ e $M = \omega^{-1}(D - \omega L)$. Vamos provar este teorema em dois passos:

- (i)– Se $Q = A^{-1}(2M - A)$ então $\rho(Q) > 1$
- (ii)– Se $R = (Q - I)(Q + I)^{-1}$ então $|\lambda_i(R)| < 1 \forall i$

(i)

$$Qx = \lambda x$$

$$A^{-1}(2M - A)x = \lambda x \implies (2M - A)x = \lambda Ax$$

$$x^*(2M - A)x = \lambda x^* Ax \tag{2.4}$$

$$x^*(2M^* - A)x = \bar{\lambda} x^* Ax \tag{2.5}$$

somando a equação (2.4) com a equação (2.5) e sendo $A = A^*$, temos:

$$x^*(2M - A + 2M^* - A)x = 2 * Re(\lambda)x^*Ax$$

$$x^*(2(M + M^*) - 2A)x = 2 * Re(\lambda)x^*Ax$$

$$2x^*((M + M^*) - A)x = 2 * Re(\lambda)x^*Ax$$

$$Re(\lambda) = \frac{x^*((M + M^*) - A)x}{x^*Ax}$$

$$M + M^* - A = \omega^{-1}(D - \omega L) + \omega^{-1}(D - \omega L) - (D - L - U) =$$

$$\omega^{-1}D - L + \omega^{-1}D - U - D + L + U = \left(\frac{2}{\omega} - 1\right)D$$

$$Re(\lambda) = \frac{x^*((M + M^*) - A)x}{x^*Ax} = \left(\frac{2}{\omega} - 1\right) \frac{x^*Dx}{x^*Ax} > 0$$

pois, por hipótese $\left(\frac{2}{\omega} - 1\right) > 0$ e $a_{i,i} > 0$ e A positiva definida. Portanto $Re(\lambda) > 0$

(ii)

$$Q = A^{-1}(2M - A) \implies Q - I = 2(A^{-1}M - I)$$

$$Q + I = 2A^{-1}M \implies (Q + I)^{-1} = \frac{1}{2}M^{-1}A$$

$$\begin{aligned} (Q - I)(Q + I)^{-1} &= 2(A^{-1}M - I)\frac{1}{2}M^{-1}A = A^{-1}MM^{-1}A - M^{-1}A = I - M^{-1}A = \\ &= I - \omega(D - \omega L)^{-1}(D - L - U) = (D - \omega L)^{-1}[(D - \omega L) - \omega(D - L - U)] = \\ &= (D - \omega L)^{-1}[D - \omega L - \omega D + \omega L + \omega U] = (D - \omega L)^{-1}[(1 - \omega)D + \omega U] = P \end{aligned}$$

logo:

$$P = (Q - I)(Q + I)^{-1}.$$

Se

$$f(R) = R$$

e

$$f(Q) = (Q - I)(Q + I)^{-1}.$$

então

$$f(R) = f(Q)$$

$$f(R)x = f(Q)x$$

$$f(\lambda(R))x = f(\lambda(Q))x,$$

isto é,

$$\lambda(R) = f(\lambda(R)) = f(\lambda(Q)) = \frac{(\lambda(Q) - 1)}{(\lambda(Q) + 1)^{-1}}$$

portanto:

$$\lambda(R) = \frac{(\lambda(Q) - 1)}{(\lambda(Q) + 1)}.$$

Sabendo que

$$|\lambda(R)| = |\lambda(R)\overline{\lambda(R)}|^{\frac{1}{2}} = \left| \left(\frac{\lambda(Q) - 1}{\lambda(Q) + 1} \right) \overline{\left(\frac{\lambda(Q) - 1}{\lambda(Q) + 1} \right)} \right|^{\frac{1}{2}}$$

$$|\lambda(R)| = \left| \left(\frac{\lambda(Q) - 1}{\lambda(Q) + 1} \right) \left(\frac{\overline{\lambda(Q)} - 1}{\overline{\lambda(Q)} + 1} \right) \right|^{\frac{1}{2}}$$

$$|\lambda(R)| = \left| \frac{\lambda(Q)\overline{\lambda(Q)} - \lambda(Q) - \overline{\lambda(Q)} + 1}{\lambda(Q)\overline{\lambda(Q)} + \lambda(Q) + \overline{\lambda(Q)} + 1} \right|^{\frac{1}{2}}$$

$$|\lambda(R)| = \left| \frac{Re(\lambda(Q))^2 + Im(\lambda(Q))^2 - 2Re(\lambda(Q)) + 1}{Re(\lambda(Q))^2 + Im(\lambda(Q))^2 + 2Re(\lambda(Q)) + 1} \right|^{\frac{1}{2}}$$

$$|\lambda(R)| = \left| \frac{(Re(\lambda(Q)) - 1)^2 + Im(\lambda(Q))^2}{(Re(\lambda(Q)) + 1)^2 + Im(\lambda(Q))^2} \right|^{\frac{1}{2}} < 1$$

temos

$$|\lambda(R)| < 1,$$

isto é,

$$|\lambda(\rho_{sor})| < 1$$

□

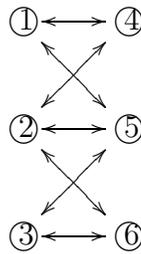
2.5 Propriedade A e Ordenação Consistente

Definição 9. *Uma matriz tem propriedade A se os vértices adjacentes de um grafo podem ser particionados em dois conjuntos S_1 e S_2 , e os eixos dos gráficos ligam o conjunto S_1*

com o conjunto S_2

Exemplo 5 :

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix},$$



Esta matriz tem a propriedade A , pois há duas partições com os vértices adjacentes que são $S_1 = \{1, 2, 3\}$ e $S_2 = \{4, 5, 6\}$.

Definição 10. Uma matriz T tem a propriedade A se existe uma permutação P tal que:

$$PAP^T = \begin{bmatrix} T_{1,1} & T_{1,2} \\ T_{2,1} & T_{2,2} \end{bmatrix},$$

onde $T_{1,1}$ e $T_{2,2}$ são diagonais.

Definição 11. Seja $G_j(\alpha) = \alpha L + \frac{1}{\alpha}U$ então $G_j(1) = L + U = G_j$ onde G_j é a matriz de iteração de Jacobi.

Proposição 1 Seja B uma matriz com a seguinte estrutura

$$B = \begin{bmatrix} 0 & B_{1,2} \\ B_{2,1} & 0 \end{bmatrix},$$

e seja L e U matrizes triangulares superior e inferior de B respectivamente, então:

- (i)– Se μ é autovalor de B então $-\mu$ também o é.
- (ii)– Os autovalores da matriz $B(\alpha) = \alpha L + \frac{1}{\alpha}U, \alpha \neq 0$. são independentes de α

Demonstração: Primeiramente, vamos provar (i). Seja $Bx = \mu x \implies (B - \mu I)x = 0$, assim

$$\begin{bmatrix} -\mu I & B_{1,2} \\ B_{2,1} & -\mu I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Podemos reescrever este sistema linear da seguinte forma:

$$\begin{bmatrix} \mu I & B_{1,2} \\ B_{2,1} & \mu I \end{bmatrix} \begin{bmatrix} x_1 \\ -x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

portanto $(B - (-\mu)I)x = 0 \implies -\mu$ é autovalor

(ii)

$$B(\alpha) = \alpha L + \frac{1}{2}U =$$

$$\begin{bmatrix} 0 & \frac{1}{\alpha}U \\ \alpha L & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} 0 & U \\ L & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \frac{1}{\alpha}I \end{bmatrix},$$

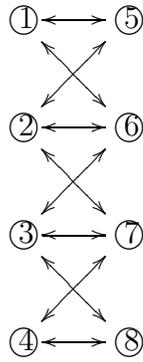
logo $B(\alpha) = XBX^{-1}$, portanto os autovalores de $B(\alpha)$ e B são iguais. Com isso os autovalores não dependem de α .

Definição 12. Uma matriz A é dita consistentemente ordenada se os vértices adjacentes de seu grafo podem ser particionados em p conjuntos S_1, S_2, \dots, S_p com a propriedade de que cada dois vértices i e j no grafo pertença a duas partições consecutivas S_k e $S_{k'}$, sendo:

$$\begin{cases} k' = k - 1 & \text{se } j < i \\ k' = k + 1 & \text{se } j > i \end{cases},$$

Exemplo 6 :

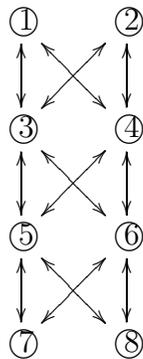
$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix},$$



Se $S_1 = \{1, 2, 3, 4\}$ e $S_2 = \{5, 6, 7, 8\}$ são dois conjuntos em que cada elemento do conjunto S_1 está ligado a outro elemento do conjunto S_2 , por meio do seu grafo, e se S_1 e S_2 , são duas partições consecutivas, então A é consistentemente ordenada e tem a propriedade A .

Exemplo 7

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix},$$



Os vértices dos grafos podem ser particionados em quatro conjuntos $S_1 = \{1, 2\}$, $S_2 = \{3, 4\}$, $S_3 = \{5, 6\}$, $S_4 = \{7, 8\}$. Podemos afirmar que a matriz A é consistentemente ordenada, mas não tem a propriedade A .

Definição 13. Seja A uma matriz tal que $A = D - L - U$ e $G_j(\alpha) = \alpha D^{-1}L + \frac{1}{\alpha}D^{-1}U$.

Se $G_j(\alpha)$ tem autovalores independentes de α então A é consistentemente ordenada.

Teorema 13. Se A é consistentemente ordenada e $\omega \neq 0$ então:

(i)– Os autovalores de G_j aparecem aos pares positivos e negativos.

(ii)– Se μ é um autovalor de G_j e $(\lambda + \omega - 1)^2 = \lambda\omega^2\mu^2$, então λ é autovalor de $G_{sor}(\omega)$.

(iii)– Se $\lambda \neq 0$ é um autovalor de $G_{sor}(\omega)$ então μ em $(\lambda + \omega - 1)^2 = \lambda\omega^2\mu^2$ é autovalor de G_j .

Demonstração: (i)

Seja A consistentemente ordenada; $P(\alpha)$ não depende de α , $G_j(\alpha) = \alpha L + \frac{1}{\alpha}U \implies G_j(1) = L + U$ e $G_j(-1) = -L - U = -(L + U) = -G_j(1)$, logo os autovalores aparecem aos pares.

(ii)

Seja $G_j x = \mu x$ e $(\lambda + \omega - 1)^2 = \lambda\omega^2\mu^2$. Se $\lambda = 0$, temos $(\omega - 1)^2 = 0 \implies \omega = 1$ $G_{sor}(1) = (I - L)^{-1}U = G_{gs} \implies G_{gs}x - \lambda x = 0 \implies (G_{gs} - \lambda I)x = 0 \implies ((I - L)^{-1}U - \lambda I)x = 0 \implies ((I - L)(I - L)^{-1}U - \lambda(I - L))x = 0 \implies (U - \lambda(I - L))x = 0 \implies \lambda = 0$ é autovalor de G_{gs} .

Se $\lambda \neq 0$ teremos:

$$\begin{aligned} 0 &= \det(\lambda I - G_{sor}(\omega)) = \det((I - \omega L)(\lambda I - G_{sor}(\omega))) = \\ &= \det((I - \omega L)(\lambda I - (I - \omega L)^{-1}((1 - \omega)I + \omega U))) = \det(\lambda(I - \omega L) - ((1 - \omega)I + \omega U)) \\ &= \det(\lambda I - \lambda\omega L - (1 - \omega)I - \omega U) = \det((\lambda + \omega - 1)I - \lambda\omega L - \omega U) \\ 0 &= \det\left(\sqrt[2]{\lambda}\omega \left(\left(\frac{\lambda + \omega - 1}{\sqrt[2]{\lambda}\omega}\right)I - \sqrt[2]{\lambda}L - \frac{1}{\sqrt[2]{\lambda}}U\right)\right) \\ 0 &= \det\left(\sqrt[2]{\lambda}\omega \left(\left(\frac{\lambda + \omega - 1}{\sqrt[2]{\lambda}\omega}\right)I - \left(\sqrt[2]{\lambda}L + \frac{1}{\sqrt[2]{\lambda}}U\right)\right)\right) \\ 0 &= \det\left(\sqrt[2]{\lambda}\omega \left(\left(\frac{\lambda + \omega - 1}{\sqrt[2]{\lambda}\omega}\right)I - (L + U)\right)\right). \end{aligned}$$

Sendo $\mu = \frac{\lambda + \omega - 1}{\sqrt[2]{\lambda}}$,

$$0 = \det(\mu I - (L + U)) = \det(\mu I - G_j)$$

então μ é autovalor e G_j , logo λ é autovalor de $G_{sor}(\omega)$.

A prova de (iii) é análoga à prova de (ii).

□

Corolário 3. *Se A é consistentemente ordenada, então $\rho(G_{gs}) = \rho(G_j)^2$. Isto afirma que Gauss Seidel é duas vezes mais rápido que o Jacobi*

Capítulo 3

Métodos Multigrid

Este capítulo é dividido em duas partes, inicialmente na seção 3.1 é descrito algumas propriedades dos métodos iterativos e como serão aproveitadas pelos métodos Multigrid.

Posteriormente, é definido formalmente os elementos dos métodos Multigrid e a construção da matriz de iteração que gera o processo iterativo.

3.1 Introdução aos Métodos Multigrid

O interesse desta seção é descrever algumas características básicas dos métodos iterativos, por meio de experimentos numéricos.

Na subseção 3.1.1, serão definidos dois tipos de equações diferenciais parciais, utilizadas para realizar os experimentos numéricos, em que são observadas algumas características dos métodos iterativos. Para finalizarmos na subseção 3.1.2, serão descritas algumas propriedades introdutórias dos métodos Multigrid que servem de motivação para a próxima seção.

3.1.1 Características dos Métodos Iterativos

Historicamente, os primeiros artigos citados como precursores dos métodos Multigrid são os de Fedorenko [9, 10] e Bakhvalov [1]. Nos artigos de Fedorenko a equação de Poisson é usada como problema de estudo, enquanto Bakhvalov usou a equação da convecção difusão. Para o estudo das características dos métodos iterativos serão usados os seguintes problemas:

$$\begin{cases} -u''(x) = f(x), & 0 < x < 1 \\ u(0) = u(1) = 0 \end{cases} \quad (3.1)$$

$$\begin{cases} \frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = f(x,y), & 0 < x, y < 1 \\ u(x,y) = 0, & \forall x, y \in \partial\Omega \end{cases} \quad (3.2)$$

Para o problema (3.1) considera-se $f(x) = 0$, conseqüentemente, a solução exata é $u(x) = 0$ para todo o domínio. No problema (3.2) o termo fonte será $f(x, y) = 0$ e a solução exata é $u(x, y) = 0$ para todo o domínio.

Discretizando os problemas modelo (3.1) e (3.2), usando diferenças finitas, elementos finitos ou volumes finitos, vamos gerar um sistema linear, $Au = b$. Para verificar algumas propriedades dos métodos iterativos a fim de resolver o sistema linear, usaremos as estimativas iniciais $v_i^k = \sin(\frac{ik\pi}{n}), 1 \leq i \leq n$, para o problema (3.1), e $v_{i,j}^{k_1,k_2} = \sin(\frac{ik_1\pi}{n}) \sin(\frac{jk_2\pi}{n}), 1 \leq i, j \leq n$, para o problema (3.2), onde k, k_1 , e k_2 definem o tipo de estimativa inicial no sentido das seguintes definições:

Definição 1 (no contexto do problema (3.1)) . Para $k \in \{1, \dots, n-1\}$, a estimativa inicial v_i tem baixa frequência, se $k < \frac{n}{2}$ e alta frequência se $\frac{n}{2} \leq k < n$.

Definição 2 (no contexto do problema (3.2)) . Para $k_1, k_2 \in \{1, \dots, n-1\}$, a estimativa inicial $v_{i,j}$ tem baixa frequência, se $\max(k_1, k_2) < \frac{n}{2}$ e alta frequência se $\frac{n}{2} \leq \max(k_1, k_2) < n$.

Para os testes com o problema (3.1), consideram-se as estimativas iniciais $v_i^k = \sin(\frac{ik\pi}{n}), 1 \leq i \leq n$, com $k = 1, k = 3$ e $k = 6$, mostradas na Figura 3.1. Essa estimativa tem baixa frequência pela Definição 1.

Para melhor entender a idéia dos métodos Multigrid, são feitos alguns experimentos numéricos com o problema (3.1) e (3.2).

Experimentos Numéricos para o Problema (3.1)

Nos experimentos numéricos com o problema (3.1) são usadas as estimativas mostradas na Figura 3.1. As figuras a seguir mostram propriedades dos métodos iterativos JOR, Gauss Seidel com ordenação clássica e Gauss Seidel com ordenação red black, descrito no Capítulo 2.

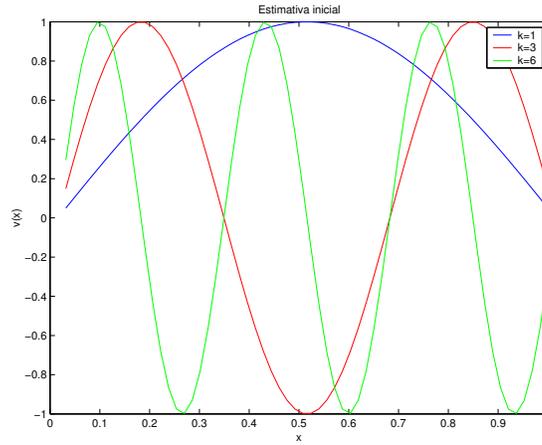


Figura 3.1: Estimativa inicial, $n = 64$

Nas figuras que seguem está ilustrado o comportamento da norma do erro:

$$\|e\|_{\infty} = \max_{1 \leq i \leq n} |e_i| = \max_{1 \leq i \leq n} |u_i - v_i|$$

com o número de iteração, para cada método.

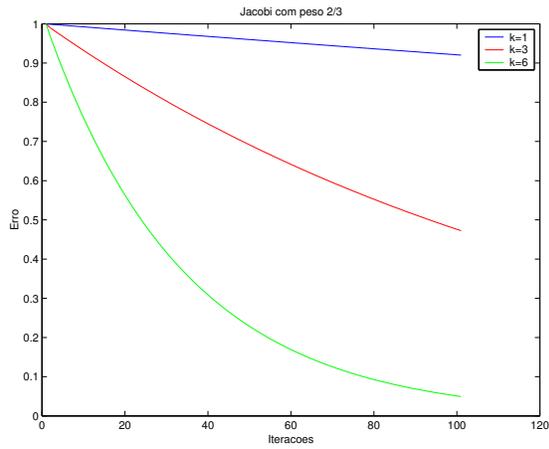
Considerando os gráficos, podemos observar que a medida que aumenta a frequência da estimativa inicial, em qualquer um dos métodos escolhidos, o erro decai com maior velocidade.

Observando as Figuras 3.2-(b) e 3.2-(c), nota-se que em Gauss Seidel e Gauss Seidel red black o erro decai com maior velocidade nas primeiras iterações, em relação ao JOR na Figura 3.2-(a). Isso mostra que devemos ter a preocupação com o tipo de método usado.

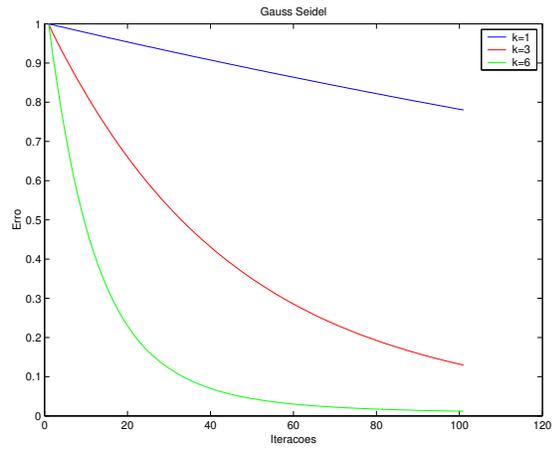
Observando as Figuras 3.2-(b) e 3.2-(c), nota-se pouca diferença no decaimento do erro. Porém, conforme artigo de Zhang [25], o esquema red black é bastante recomendado por ser altamente paralelizável.

Sabemos que o método SOR normalmente converge com mais velocidade que Gauss Seidel. Nos métodos Multigrid ocorre a mudança de malha, fazendo com que o parâmetro de relaxação do SOR seja diferente para cada malha, dificultando assim o uso do SOR. O mesmo artigo citado anteriormente mostra uma maneira de contornar o problema.

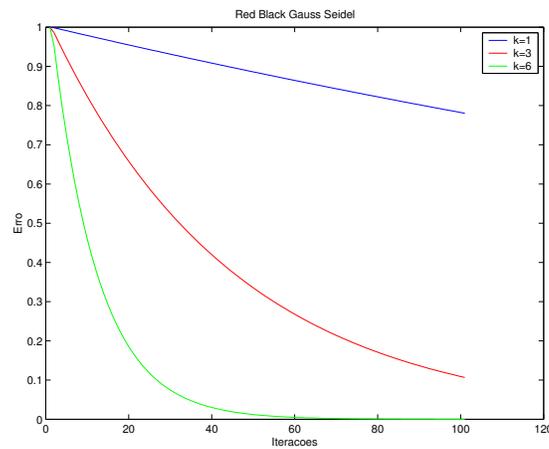
A Figura 3.3 mostra a estimativa inicial $v_i^{k_1, k_2, k_3} = \sin\left(\frac{k_1 i \pi}{n}\right) + \sin\left(\frac{k_2 i \pi}{n}\right) + \sin\left(\frac{k_3 i \pi}{n}\right)$, com $k_1 = 1$, $k_2 = 6$ e $k_3 = 32$; com isso podemos afirmar que $v_i^{k_1, k_2, k_3}$ tem alta frequência, pela Definição 1.



(a)



(b)



(c)

Figura 3.2: (a): JOR com parâmetro $\frac{2}{3}$. (b): Gauss Seidel. (c): Gauss Seidel Red Black.

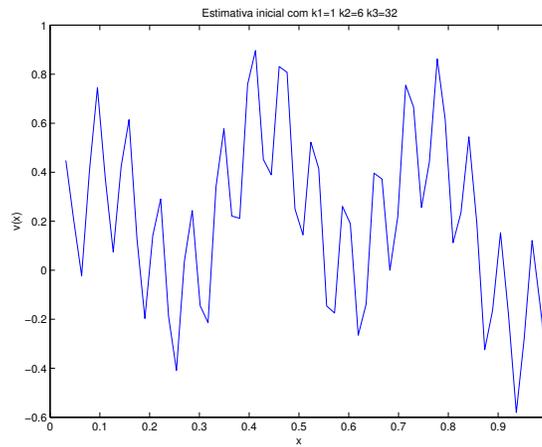


Figura 3.3: Estimativa inicial k1=1 k2=6 k3=32, n=64

A Figura 3.4 mostra o gráfico do *erro* com o número de iterações do JOR para o problema (3.1), com $n = 64$.

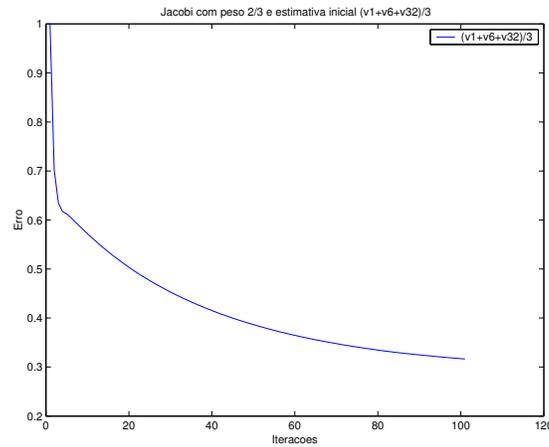


Figura 3.4: JOR com parâmetro $\frac{2}{3}$ e estimativa inicial $\frac{v_1+v_6+v_{32}}{3}$

Podemos observar que o erro decresce rapidamente nas primeiras iterações, depois disso decresce lentamente. O decréscimo inicial corresponde à rápida eliminação dos termos de alta frequência na estimativa inicial e o decréscimo lento deve-se à existência de baixa frequência na estimativa inicial.

A velocidade de decaimento dos métodos é maior quando as estimativas iniciais são de alta frequência, assim como a velocidade de decaimento é prejudicada quando as estimativas iniciais são de baixa frequência.

Pode-se explicar essa afirmação da seguinte forma: o nosso problema é resolver $Au = b$. Gostaríamos de ter uma solução única; então, A deve ser não singular. Portanto, seus autovetores formam uma base linearmente independente em \mathbb{R}^{n-2} , isto é, podemos escrever o erro inicial (e^o) como combinação linear dos autovetores de A , logo

$$e^o = \sum_{k=1}^{n-2} (c_k w_k),$$

em que w_k são os autovetores de A e $c_k \in \mathbb{R}$, $k \in \mathbb{N}$. Conhecemos a seguinte equação de diferenças para o erro:

$$e^m = G_{j\omega}^m e^o,$$

em que

$$G_{j\omega} = (1 - \omega)I + \omega G_j;$$

G_j é a matriz de iteração de Jacobi.

Então,

$$\begin{aligned}
 e^m &= G_{j\omega}^m e^o = G_{j\omega}^m \sum_{k=1}^{n-2} (c_k w_k) = \\
 &= \sum_{k=1}^{n-2} (c_k G_{j\omega}^m w_k) = \sum_{k=1}^{n-2} (c_k (\lambda_k(G_{j\omega}))^m w_k)
 \end{aligned}$$

$$\lambda_k(G_{j\omega}) = 1 - 2\omega \left(\sin\left(\frac{k\pi}{2n}\right) \right)^2,$$

em que $\lambda_k(G_{j\omega}) = \lambda_k$ é autovalor de $G_{j\omega}$.

Chegamos ao resultado acima da seguinte forma:

$$\begin{aligned}
 G_{j\omega} &= (1 - \omega)I + \omega G_j \\
 G_{j\omega} &= (1 - \omega)I + \omega D^{-1}(L + U) \\
 A &= D - L - U \implies D - A = L + U \\
 G_{j\omega} &= (1 - \omega)I + \omega(D^{-1}D - D^{-1}A) \\
 G_{j\omega} &= I - \omega I + \omega I - \omega D^{-1}A \\
 G_{j\omega} &= I - \omega D^{-1}A.
 \end{aligned}$$

Para encontrar os autovalores de $G_{j\omega}$, procedemos da seguinte forma:

$$\begin{aligned}
 G_{j\omega}x &= \lambda(G_{j\omega})x \\
 (I - \omega D^{-1}A)x &= \lambda(G_{j\omega})x \\
 x - \omega D^{-1}Ax &= \lambda(G_{j\omega})x,
 \end{aligned}$$

em que $\lambda(A)$ é autovalor de A e $\lambda(D)$ é autovalor de D :

$$\begin{aligned}
 x - \omega D^{-1}\lambda(A)x &= \lambda(G_{j\omega})x \\
 x - \omega \lambda(A)D^{-1}x &= \lambda(G_{j\omega})x \\
 x - \omega \frac{\lambda(A)}{\lambda(D)}x &= \lambda(G_{j\omega})x \\
 \left(1 - \omega \frac{\lambda(A)}{\lambda(D)}\right)x &= \lambda(G_{j\omega})x,
 \end{aligned}$$

portanto,

$$\lambda(G_{j\omega}) = \left(1 - \omega \frac{\lambda(A)}{\lambda(D)}\right),$$

Estamos procurando os autovalores de uma matriz tridiagonal com elementos dispostos em linha da seguinte maneira: $(1 \quad -2 \quad 1)$, proveniente da discretização do problema (3.1). Os seus autovalores são conhecidos, D é uma matriz diagonal ($d_{ii} = a_{ii}$), assim $\lambda(A) = 4 \sin(\frac{k\pi}{2n})$ e $\lambda(D) = 2$. Portanto,

$$\lambda(G_{j\omega}) = \left(1 - \omega \frac{\lambda(A)}{\lambda(D)}\right).$$

Substituindo os valores acima, tem-se $\lambda_k(G_{j\omega}) = 1 - 2\omega(\sin(\frac{k\pi}{2n}))^2$. Podemos concluir que: se $k = 1$, temos que $|\lambda_k(G_{j\omega})| \simeq 1$, com isso $(\lambda_1(G_{j\omega}))^k \rightarrow 1$ se $k \rightarrow \infty$.

se $k = 64$, temos que $|\lambda_k(G_{j\omega})| \simeq \frac{1}{3}$, com isso $(\lambda_{64}(G_{j\omega}))^k \rightarrow 0$ se $k \rightarrow \infty$.

Portanto, os termos de alta freqüência convergem para zero mais rapidamente que os termos de baixa freqüência, o que comprova as observações anteriores.

Outro teste interessante é aplicar o JOR em três situações. As Figuras 3.5-(a), 3.5-(b) e 3.5-(c), mostram as estimativas iniciais para o problema (3.1): v_i^3 , v_i^{16} e $\frac{v_i^2 + v_i^{16}}{2}$, assim como o resultado para as iterações 1 e 10, respectivamente.

Nota-se que o método tem dificuldade para eliminar estimativas de baixa freqüência, pois na Figura 3.5-(a), a estimativa inicial decresce muito pouco após 10 iterações. O mesmo comportamento aparece na Figura 3.5-(c). Depois do mesmo número de iterações, a estimativa de baixa freqüência persiste.

Por outro lado, a Figura 3.5-(b) mostra muito bem o decaimento da estimativa inicial de alta freqüência. O mesmo é observado na Figura 3.5-(a), pois a estimativa de alta freqüência decresce rapidamente depois de 10 iterações, praticamente restando a influência da estimativa de baixa freqüência.

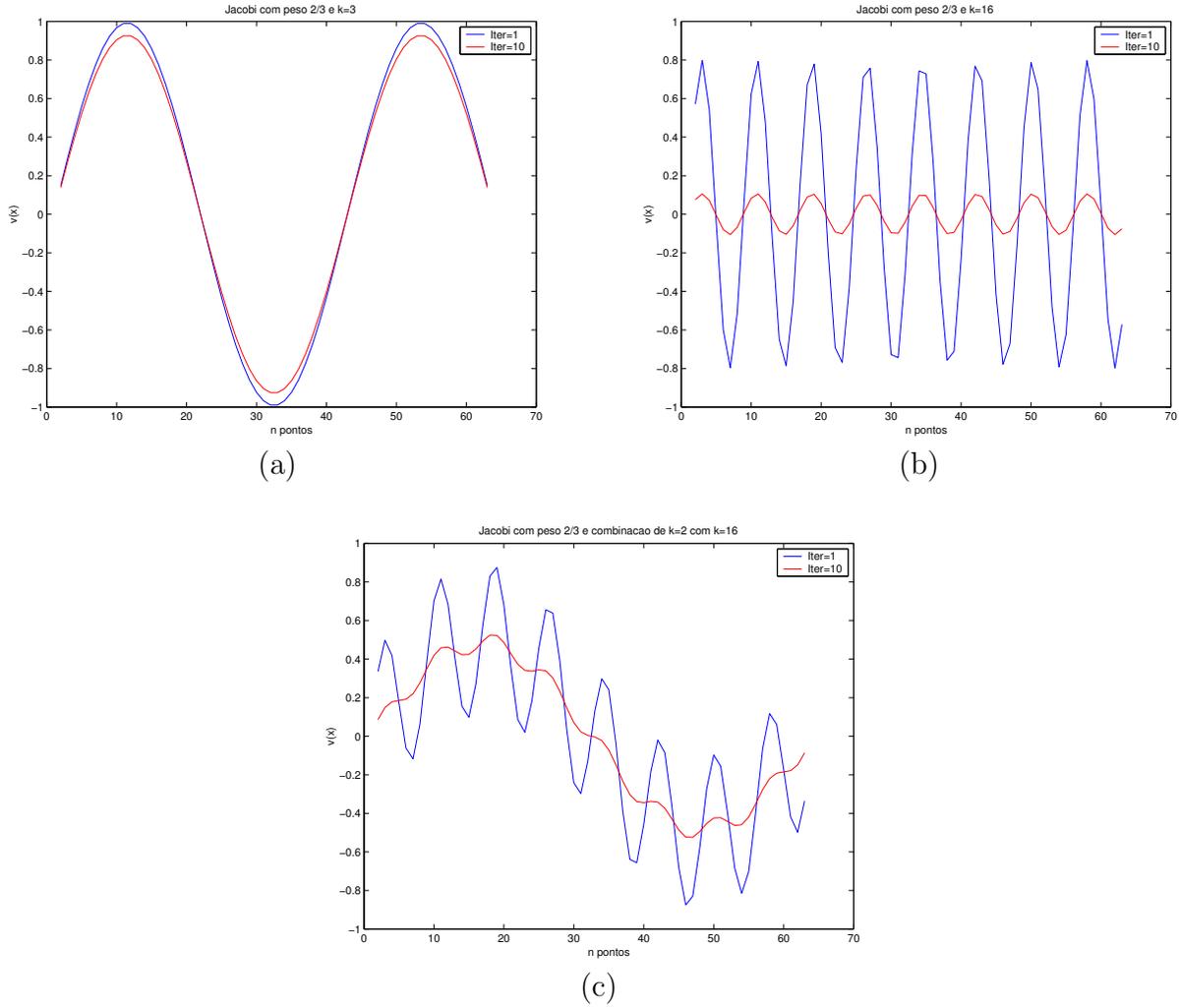


Figura 3.5: (a): JOR com parâmetro $\frac{2}{3}$ e $k = 3$. (b): JOR com parâmetro $\frac{2}{3}$ e $k = 16$. (c): JOR com parâmetro $\frac{2}{3}$ e $\frac{v_2 + v_{16}}{2}$.

Experimentos Numéricos para o Problema (3.2)

Vamos gerar os gráficos para o problema (3.2) calculando a norma infinito do erro com o número de iterações e usando a estimativa inicial $v_{i,j}^{k_1,k_2} = \sin\left(\frac{ik_1\pi}{n}\right) \sin\left(\frac{jk_2\pi}{n}\right)$, $1 \leq i, j \leq n$.

A Figura 3.6-(a) mostra o resultado do JOR com parâmetro $\frac{2}{3}$ e com a malha quadrada com $n = 64$ partição em cada direção. A figura mostra ainda o desempenho do JOR com diferentes estimativas iniciais, $v_{i,j}^{1,1}$, $v_{i,j}^{10,10}$ e $v_{i,j}^{30,30}$, respectivamente, sendo que as duas primeiras estimativas iniciais são de baixa frequência e a última é de alta frequência.

A Figura 3.6-(b) mostra o resultado de Gauss Seidel para o problema modelo (3.2) com as seguintes estimativas iniciais: $v_{i,j}^{1,1}$, $v_{i,j}^{10,10}$ e $v_{i,j}^{30,30}$.

Podemos afirmar, com isso, que acontece o mesmo comportamento do problema modelo(3.1),

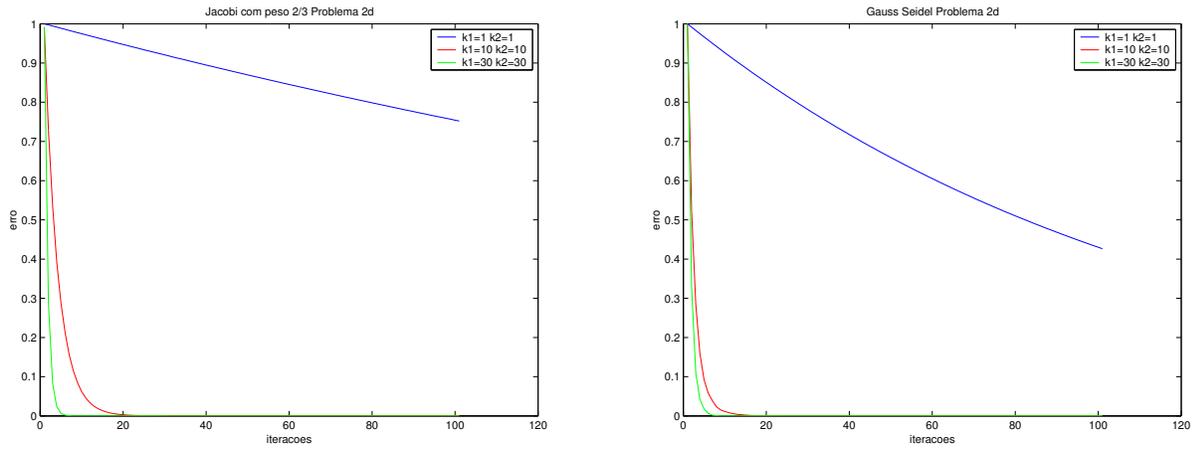


Figura 3.6: (a): JOR com parâmetro $\frac{2}{3}$. (b): Gauss Seidel,

ou seja, quanto mais alta frequência da estimativa inicial, mais rápido o método iterativo converge.

As próximas figuras mostram o comportamento da solução numérica, variando o número de iteração, dadas várias estimativas iniciais.

Estimativa Inicial com $\mathbf{k}_1 = 1$ e $\mathbf{k}_2 = 1$ ($\mathbf{v}_{i,j}^{1,1}$)

As Figuras 3.7-(a), 3.7-(b) e 3.7-(c) têm como estimativa inicial $v_{i,j}^{1,1} = \text{sen}(\frac{i\pi}{n})\text{sen}(\frac{j\pi}{n})$ e mostram as iterações 0,5, e 10, respectivamente.

A estimativa inicial dada para gerar as Figuras 3.7-(a), 3.7-(b) e 3.7-(c), tem baixa frequência pela Definição 2. Como esperado, a velocidade de decaimento da estimativa inicial é muito pequena, mostrando a deficiência dos métodos iterativos para avançar em direção à solução.

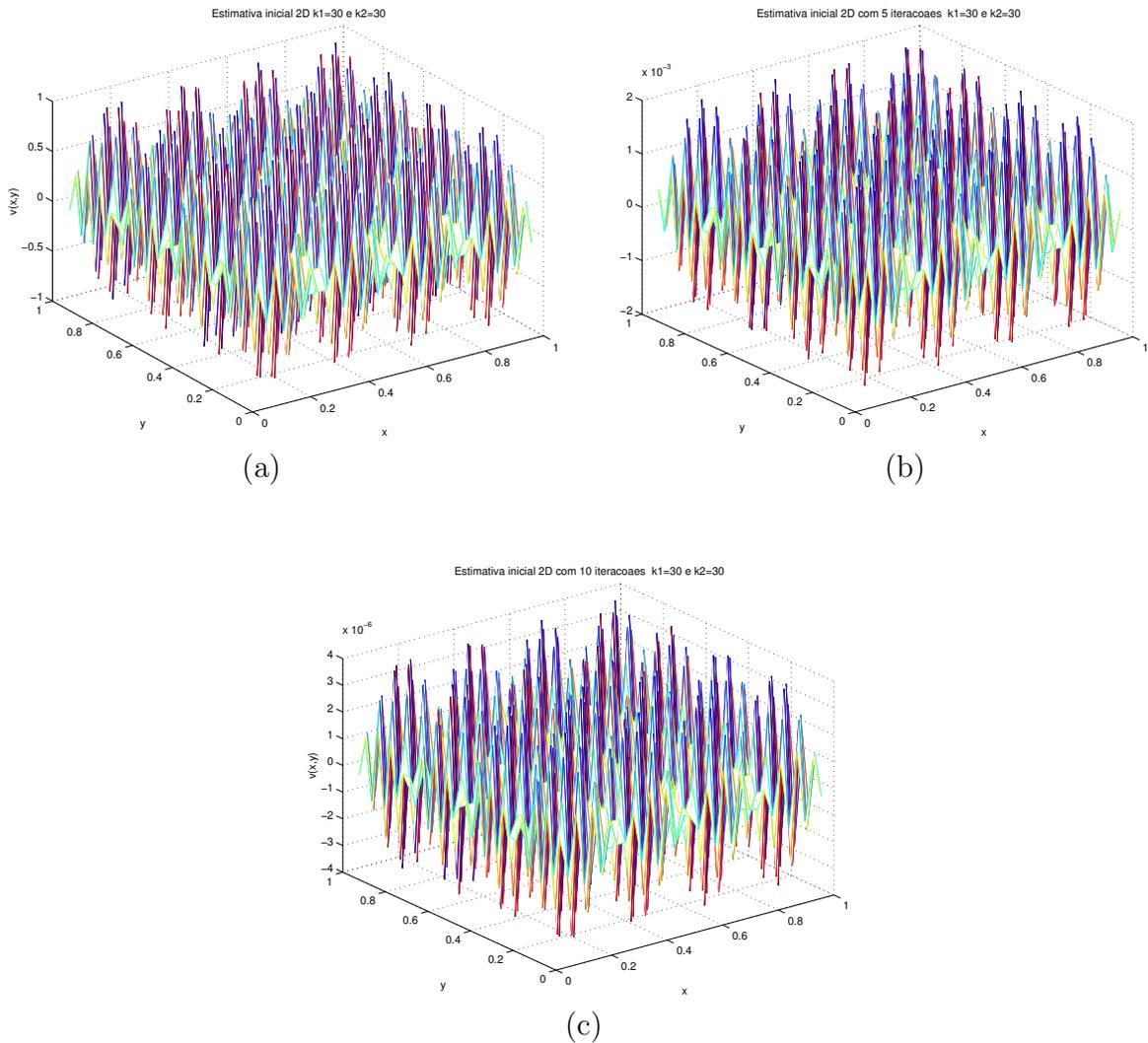


Figura 3.9: (a): Estimativa inicial $k_1 = 30$ $k_2 = 30$. (b): Gauss Seidel com 5 iterações, $k_1 = 30$ e $k_2 = 30$. (c): Gauss Seidel com 10 iterações, $k_1 = 30$ e $k_2 = 30$.

tivos e como estas podem ser exploradas pelos métodos Multigrid.

3.1.2 Multigrid

Basicamente, a idéia dos métodos Multigrid consiste em iterar poucas vezes numa malha fina. Estas poucas vezes significam o suficiente para eliminar as altas frequências da estimativa inicial. A partir deste ponto, quando temos somente as componentes de baixa frequência da estimativa inicial, o método numérico converge com menor velocidade. Nesta etapa, usa-se uma malha mais grosseira.

Para transferir informações entre malhas, usamos os seguintes operadores: o de restrição, para passar de uma malha mais fina para uma mais grossa, e o operador de inter-

polação, que passa informações da malha grossa para a malha fina. Eles serão descritos com mais detalhes na próxima seção.

A idéia principal dos métodos Multigrid é usar esses operadores para transformar a estimativa inicial cuja alta frequência tenha sido eliminada, com o auxílio de métodos iterativos, novamente em estimativa de alta frequência, porém na malha grossa. Dessa forma, podemos aproveitar as características dos métodos iterativos que é eliminar com maior velocidade de decaimento as estimativas de alta frequência.

Analisando o próximo exemplo, temos uma idéia de como o operador de restrição atua para aumentar a frequência da estimativa inicial.

Considere a dimensão da malha $n = 12$, sendo $k = 4$ fixo, e a estimativa inicial $v_i^4 = \text{sen}(\frac{4i\pi}{12})$, $1 < i < 12$, logo $k < \frac{n}{2}$. Esta estimativa tem baixa frequência pela Definição 1. E, como foi visto anteriormente, a convergência dos métodos iterativos fica prejudicada.

Aplicando o operador de restrição, a malha grosseira fica com a dimensão $n = 6$, então $v_i^4 = \text{sen}(\frac{4i\pi}{6})$, $1 < i < 6$, logo $k > \frac{n}{2}$. Assim a estimativa passa e ter alta frequência pela Definição 1. Agora aplicando um método iterativo, conseguimos obter um bom desempenho, pois está mantida a alta frequência.

Para se ter uma idéia dos métodos Multigrid, vamos acompanhar as próximas figuras.

A Figura 3.10-(a) mostra a estimativa inicial $\frac{v_{16}+v_{40}}{2}$ de alta frequência, para o problema (3.1), com $n = 64$.

Em 3.10-(b), temos $v(x)$ após uma iteração do JOR na malha fina.

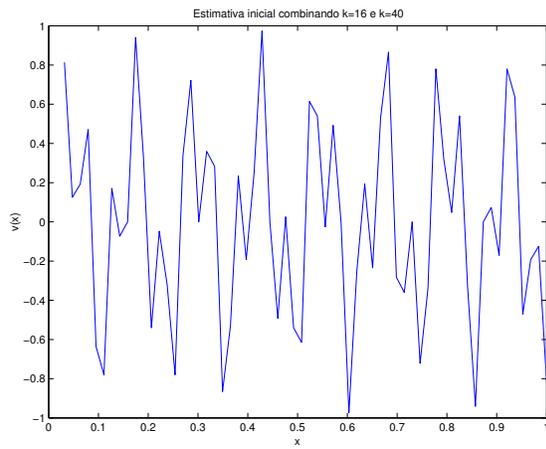
Em 3.10-(c), temos $v(x)$ após três iteração do JOR na malha fina.

Em 3.10-(d), temos $v(x)$ na malha fina, após uma iteração do JOR na malha grossa.

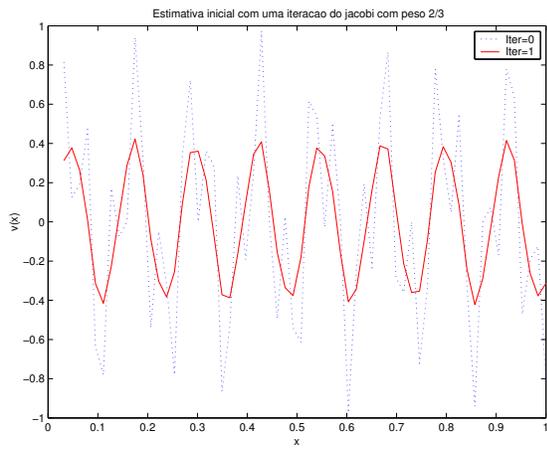
Em 3.10-(e), temos $v(x)$ na malha fina, após três iteração do JOR na malha grossa.

Em (3.10)-(f), temos $v(x)$ na malha fina, após três iteração do JOR na malha fina, depois da correção na malha grossa.

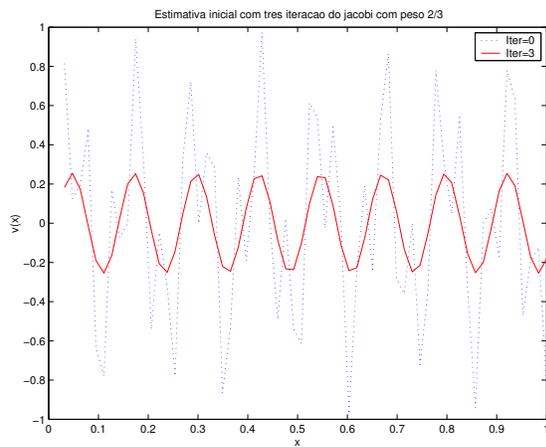
Desse modo, mantemos os métodos numéricos trabalhando sempre com bom desempenho porque, quando se faz a troca de malha, mantemos $v(x)$ sempre com alta frequência.



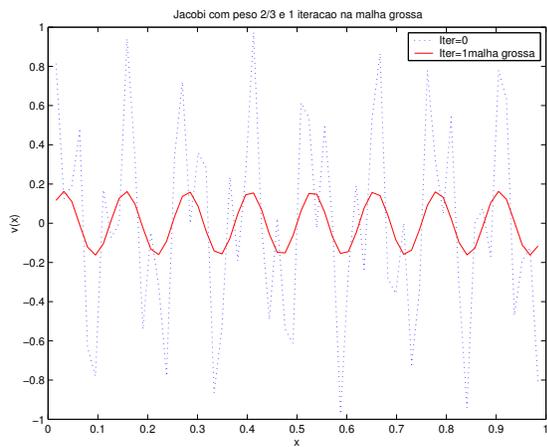
(a)



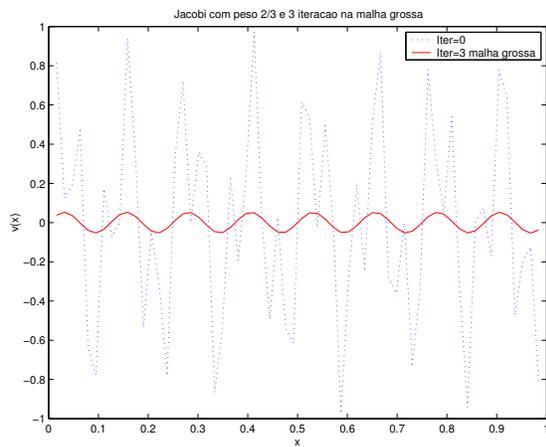
(b)



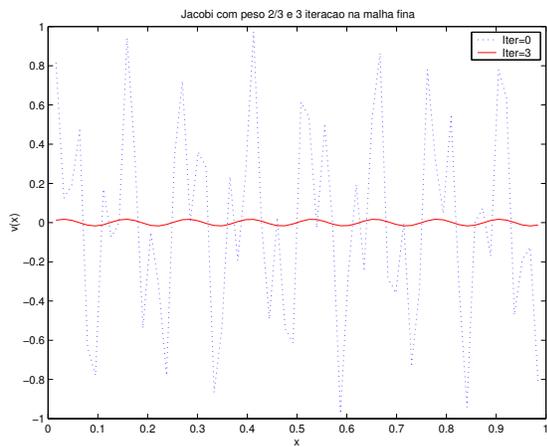
(c)



(d)



(e)



(f)

Figura 3.10: (a): Estimativa inicial $\frac{v_{16}+v_{40}}{2}$. (b): Uma iteraçã na malha fina. (c): Três iterações na malha fina. (d): Uma iteraçã na malha grossa. (e): Três iterações na malha grossa. (f): Três iterações na malha fina.

3.2 Elementos dos Métodos Multigrid

Nesta seção serão definidos os elementos dos métodos Multigrid, começando por uma representação mais formal dos problemas de valor de fronteira contínuo e o problema de valor de fronteira discreto ($Au = b$).

Na subseção 3.2.2, serão definidos os operadores de restrição e de interpolação, que são importantes para fazer a transferência de informação entre as malhas.

Na subseção 3.2.3, será descrito o ciclo de duas malhas, mostrando de forma construtiva a matriz de iteração da correção em malha grosseira com e sem suavização (relaxação).

O tipo de ciclo que é usado nos métodos Multigrid está definido na subseção 3.2.4, que mostra algumas maneiras de fazer o processo de correção do ciclo de duas malhas. Para finalizar esta seção, será mostrado o Teorema 14, que constrói a matriz para gerar a equação de iteração nos métodos Multigrid.

3.2.1 Nomenclatura Usada Neste Capítulo

Neste capítulo consideramos o seguinte problema de valor de fronteira:

$$\begin{cases} L^\Omega u(x) = f^\Omega(x), & x \in \Omega \subset \mathbb{R}^n \\ L^\Gamma u(x) = f^\Gamma(x), & x \in \Gamma := \partial\Omega \end{cases} \quad (3.3)$$

Se a dimensão de Ω é n , um elemento de Ω é representado por $x = (x_1, x_2, \dots, x_n)^T$, L^Ω é um operador diferencial em Ω e L^Γ é um operador diferencial em $\partial\Omega$; além disso, f^Ω e f^Γ são funções em Ω e $\partial\Omega$, respectivamente. Desta forma, definimos o problema contínuo (EDP).

Uma das maneiras de resolver as Equações Diferenciais Parciais é aplicando técnicas de discretização, comentadas anteriormente. Nessas técnicas, precisamos particionar o domínio em que a EDP está definida. Se o domínio for bidimensional usamos um parâmetro $h = (h_x, h_y)$ de discretização que mostra o espaçamento da malha na direção x e y .

Dado o conjunto $G_h := \{(x, y) : x = x_i = ih_x, y = y_j = jh_y; i, j \in \mathbb{N}\}$, definimos o domínio e a fronteira do problema como $\Omega_h = \Omega \cap G_h$ e $\Gamma_h = \Gamma \cap G_h$, respectivamente. E seja u_h definida em $\Omega_h \cup \Gamma_h$ uma aproximação de u no problema contínuo, e ainda as funções discretas $f_h^{\Omega_h}$ e $f_h^{\Gamma_h}$ análogas às funções contínuas f^Ω e f^Γ no problema contínuo (3.3).

Definindo $g(\Omega_h)$ como o espaço linear das funções discretas em Ω_h e denotando $L_h^{\Omega_h}$

como o operador diferencial discreto definido em $g(\Omega_h)$, o operador $L_h^{\Omega_h}$ pode ser representado da seguinte maneira:

$$\begin{aligned} L_h^{\Omega_h} : g(\Omega_h) &\rightarrow g(\Omega_h) \\ u_h(x_i, y_j) &\mapsto L_h^{\Omega_h} u_h(x_i, y_j) = f_h^{\Omega_h} \end{aligned} ,$$

da mesma forma definimos L_h^Γ , o operador discreto na fronteira

$$\begin{aligned} L_h^\Gamma : g(\Gamma_h) &\rightarrow g(\Gamma_h) \\ u_h(x_i, y_j) &\mapsto L_h^\Gamma u_h(x_i, y_j) = f_h^\Gamma \end{aligned} .$$

Assim, o problema discreto análogo a (3.3) é denotado por

$$\begin{cases} L_h^{\Omega_h} u_h(x, y) = f_h^{\Omega_h}(x, y), & (x, y) \in \Omega \subset \mathbb{R}^2 \\ L_h^\Gamma u_h(x, y) = f_h^\Gamma(x, y), & (x, y) \in \Gamma := \partial\Omega \end{cases} . \quad (3.4)$$

Notação “stencil” para a Equação de Poisson

Considere

$$\begin{cases} -\Delta u_h(x, y) = f_h^\Omega(x, y), & (x, y) \in \Omega \subset (R)^2 \\ u_h(x, y) = f_h^\Gamma(x, y), & (x, y) \in \Gamma := \partial\Omega \end{cases} . \quad (3.5)$$

O problema discreto de Poisson, onde $L_h = \Delta_h$. Para a notação stencil de $-\Delta_h$, considere $h = h_x = h_y$ (partição homogênea). Como $Lu = -u_{xx} - u_{yy}$, temos o seguinte problema discreto correspondente:

$$L_h u_h = \frac{1}{h^2} (4u(x, y) - u(x + h, y) - u(x - h, y) - u(x, y + h) - u(x, y - h))$$

$$L_h u_h = \frac{1}{h^2} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}_h u_h(x_i, y_j) = f_h,$$

com isso temos

$$\Delta_h = \frac{1}{h^2} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}_h ,$$

Este tipo de notação ajuda a representação stencil dos operadores de restrição e interpolação.

3.2.2 Operadores de Transferência entre Malhas

Nesta subseção, serão descrito algumas possíveis escolhas para a transferência de informações entre as malhas. Nos problemas estudados teremos sempre a mesma relação entre a malha grosseira e a malha fina, dada da seguinte forma: $\Omega_H = \frac{1}{2}\Omega_h$.

Veremos a aplicação de alguns tipos de operadores de restrição. Para isso vamos usar as Figuras 3.11 e 3.12 para melhor entender o processo:

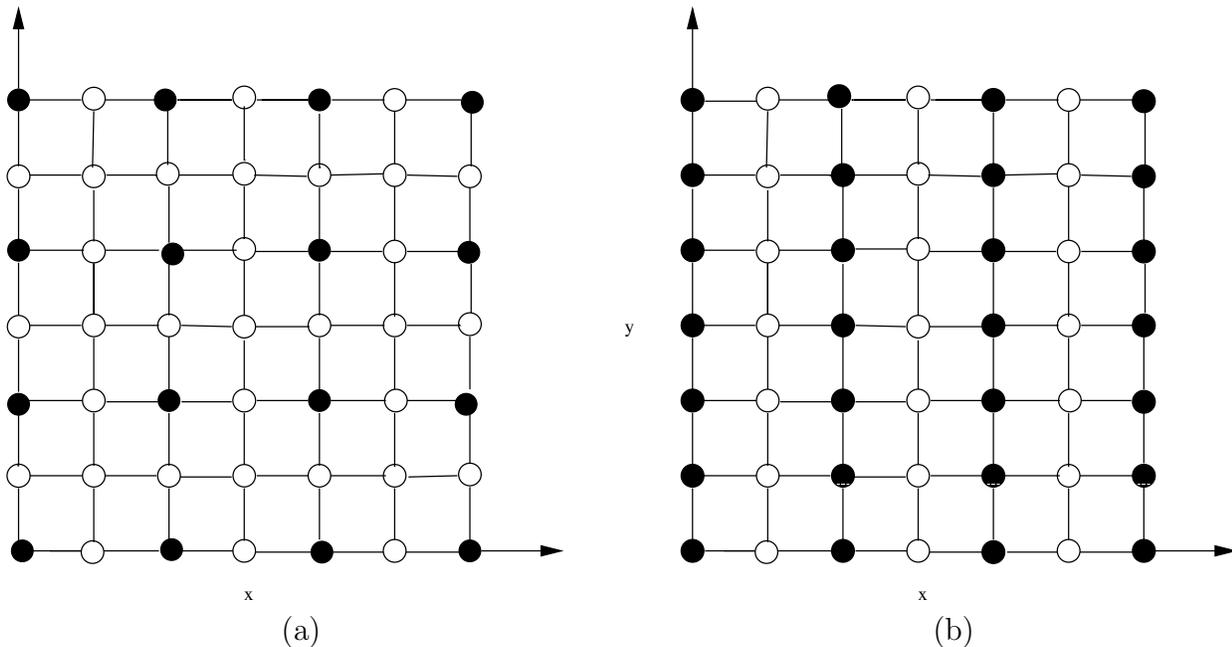


Figura 3.11: (a): Malha grossa usual. (b): Malha semi-grossa em x .

- A malha grosseira, mais usada na bibliografia, está representada na Figura 3.11-(a).
- Podem-se encontrar operadores duplos em alguma direção; isto é chamado de malha semi-grosseira. Por exemplo, a malha x semigrosseira $H = (2h_x, h_y)$, normalmente usada em problemas anisotrópicos, está representada na 3.11-(b).
- Outra maneira de transferir informações é usar os pontos red black quando a ordenação da malha fina for do tipo red black, explicada no Capítulo 1. Este processo está mostrado na 3.12-(a).
- Tem-se ainda a malha grosseira $H = (3h_x, 3h_y)$, que algumas vezes pode ser usada e

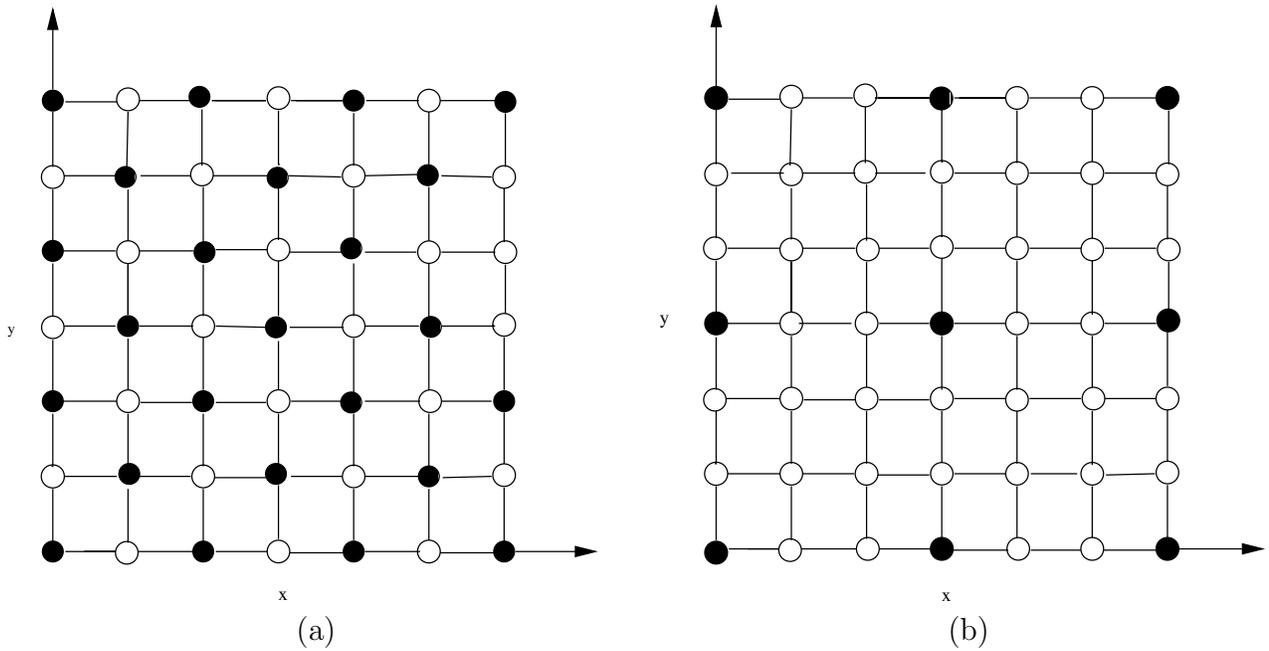


Figura 3.12: (a): Malha red black. (b): Malha $(3h, 3h)$.

está representada na 3.12-(b).

A escolha do operador de restrição ou de interpolação, para transferência de informação entre as malhas, será descrita abaixo. No restante do trabalho será considerada a seguinte relação entre a malha fina e a malha grosseira: $H = 2h$.

Operador de Restrição

O operador de restrição é representado por

$$\begin{aligned} I_h^{2h} : g(\Omega_h) &\rightarrow g(\Omega_{2h}) \\ u(x_i, y_j)_h &\mapsto I_h^{2h}u(x_i, y_j)_h \end{aligned} .$$

O primeiro tipo de operador de restrição é o operador injetivo, mostrado na figura 3.13

Outro tipo de operador frequentemente usado é o "Full Weighting" (FW), que pode ser representado com a seguinte notação stencil:

$$A = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^{2h} .$$

Podemos aplicar este operador no resíduo ou em qualquer função discreta pertencente a $g(\Omega_h)$; por exemplo $d_h(x, y)$ calculado na malha fina, o que resulta no resíduo $d_{2h}(x, y)$ na malha grosseira. Podemos usar o seguinte algoritmo para obter d_{2h} :

$$d_{2h}(x, y) = I_h^{2h} d_h(x, y) = \frac{1}{16} [4d_h(x, y) + 2d_h(x + h, y) + 2d_h(x - h, y) + 2d_h(x, y + h) + 2d_h(x, y - h) + d_h(x + h, y + h) + d_h(x - h, y + h) + d_h(x + h, y - h) + d_h(x - h, y - h)],$$

isto é, $d_{2h}(x, y)$ é obtido com a média de nove pontos vizinhos.

Um terceiro tipo de operador de restrição é "Half Weighting" (HW), que pode ser representado pela seguinte notação matricial:

$$A = \frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}_h^{2h}.$$

Podemos aplicar este operador no resíduo $d_h(x, y)$, calculado na malha fina, e tem-se o

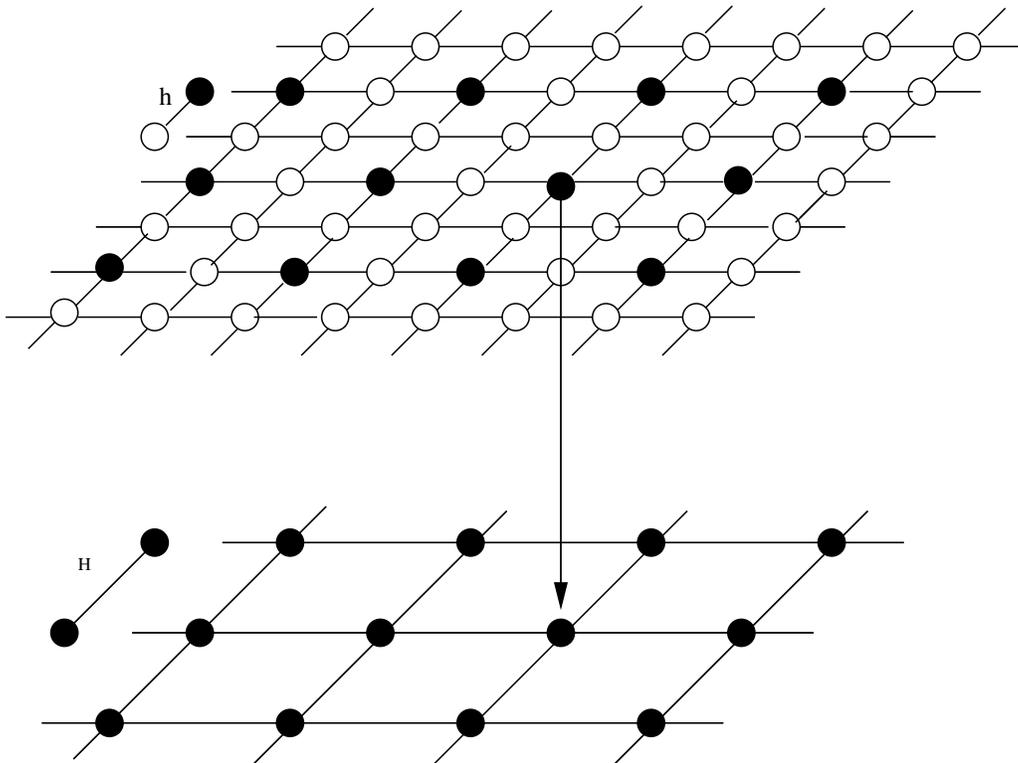


Figura 3.13: Retrição injetiva da malha fina para a grossa

Vamos assumir que exista L_h^{-1} e denotando u_h como sendo a solução discreta do problema $L_h u_h = f_h$. Considere que u_h^m é a solução aproximada do problema $L_h u_h = f_h$ na iteração m e, ainda, v_h^m o erro na iteração m . Para facilitar a notação usamos L_h em vez de L_h^Ω , isso tudo com relação à malha fina.

Aproximação Usando a Equação Residual

Para cada aproximação u_h^m da solução u_h , do problema $L_h u_h = f_h$, definimos as seguintes equações:

Equação do erro:

$$v_h^m := u_h - u_h^m \text{ ou } u_h = u_h^m + v_h^m \text{ (Correção na malha } h)$$

Equação do resíduo:

$$d_h^m := f_h - L_h u_h^m$$

Além disso, existe uma equivalência entre $L_h u_h = f_h$ e $L_h v_h^m = d_h^m$, isto é

$$L_h u_h = f_h \Leftrightarrow L_h v_h^m = d_h^m,$$

pois

$$L_h u_h = f_h \Leftrightarrow L_h u_h - L_h u_h^m = f_h - L_h u_h^m$$

$$L_h(u_h - u_h^m) = f_h - L_h u_h^m.$$

Portanto,

$$L_h v_h^m = d_h^m.$$

Supondo que o operador L_h é aproximado por um operador \widehat{L}_h , tal que \widehat{L}_h^{-1} exista. A motivação desta aproximação será mostrada no final desta seção.

Com isso podemos encontrar a equação de iteração por meio do seguinte processo:

Tem-se assim que \widehat{v}_h^m é solução de uma nova aproximação $\widehat{L}_h \widehat{v}_h^m = d_h^m$ e, ainda, temos a equação de correção $u_h^{m+1} = u_h^m + \widehat{v}_h^m$.

$$u_h^m \longrightarrow d_h^m = f_h - L_h u_h^m \longrightarrow \widehat{L}_h \widehat{v}_h^m = d_h^m \longrightarrow u_h^{m+1} = u_h^m + \widehat{v}_h^m.$$

Portanto, partindo da correção na malha h :

$$u_h^{m+1} = u_h^m + \widehat{v}_h^m \implies u_h^{m+1} = u_h^m + \widehat{L}_h^{-1} d_h^m \implies u_h^{m+1} = u_h^m + \widehat{L}_h^{-1} (f_h - L_h u_h^m)$$

$$u_h^{m+1} = u_h^m + \widehat{L}_h^{-1} f_h - \widehat{L}_h^{-1} L_h u_h^m \implies u_h^{m+1} = (I - \widehat{L}_h^{-1} L_h) u_h^m + \widehat{L}_h^{-1} f_h$$

Definindo:

$M_h = (I - \widehat{L}_h^{-1} L_h)$, matriz de iteraç o e $S_h = \widehat{L}_h^{-1} f_h$, o vetor independente na equa o de itera o.

Assim, a equa o de itera o fica:

$$u_h^{m+1} = M_h u_h^m + S_h.$$

O erro no processo iterativo   $\widehat{v}_h^m = u_h^{m+1} - u_h^m$. Usando a equa o de itera o para a m e $m + 1$, temos:

$$u_h^{m+1} = M_h u_h^m + S_h \tag{3.6}$$

$$u_h^m = M_h u_h^{m-1} + S_h. \tag{3.7}$$

Subtraindo (3.7) de (3.6) temos:

$$u_h^{m+1} - u_h^m = M_h u_h^m - M_h u_h^{m-1} = M_h (u_h^m - u_h^{m-1}).$$

Portanto, substituindo a defini o do erro na equa o acima temos:

$$v_h^{m+1} = M_h v_h^m,$$

logo

$$v_h^{m+1} = (I - \widehat{L}_h^{-1} L_h) v_h^m. \tag{3.8}$$

Esta   a equa o de itera o para o erro. Abaixo ser  encontrada uma equa o para o res duo:

$$\begin{aligned} v_h^{m+1} &= M_h v_h^m \\ L_h v_h^{m+1} &= L_h M_h L_h^{-1} L_h v_h^m \\ L_h v_h^{m+1} &= L_h (I - \widehat{L}_h^{-1} L_h) L_h^{-1} L_h v_h^m \\ d_h^{m+1} &= (L_h L_h^{-1} - L_h \widehat{L}_h^{-1} L_h L_h^{-1}) d_h^m \\ d_h^{m+1} &= (I - L_h \widehat{L}_h^{-1}) d_h^m. \end{aligned} \tag{3.9}$$

Se tivermos uma estimativa inicial $u^0 = 0$ para a equa o de itera o, teremos o seguinte processo:

$$\begin{aligned}
u_h^1 &= \widehat{L}_h^{-1} f_h \\
u_h^2 &= (M_h + I) \widehat{L}_h^{-1} f_h \\
u_h^3 &= (M_h^2 + M_h + I) \widehat{L}_h^{-1} f_h \\
u_h^4 &= (M_h^3 + M_h^2 + M_h + I) \widehat{L}_h^{-1} f_h \\
&\vdots \\
u_h^m &= (M_h^{m-1} + \dots + M_h^3 + M_h^2 + M_h + I) \widehat{L}_h^{-1} f_h.
\end{aligned} \tag{3.10}$$

Substituindo a seguinte igualdade:

$$(M_h^{m-1} + \dots + M_h^3 + M_h^2 + M_h + I)(I - M_h) = (I - M_h^m)$$

em (3.10), temos:

$$\begin{aligned}
u_h^m &= (I - M_h^m)(I - M_h)^{-1} \widehat{L}_h^{-1} f_h \\
u_h^m &= (I - M_h^m)(I - I + \widehat{L}_h^{-1} L_h)^{-1} \widehat{L}_h^{-1} f_h \\
u_h^m &= (I - M_h^m)(\widehat{L}_h^{-1} L_h)^{-1} \widehat{L}_h^{-1} f_h \\
u_h^m &= (I - M_h^m) L_h^{-1} \widehat{L}_h \widehat{L}_h^{-1} f_h.
\end{aligned}$$

Portanto,

$$u_h^m = (I - M_h^m) L_h^{-1} f_h. \tag{3.11}$$

Raio Espectral

Com o auxílio das igualdades (3.8) e (3.9), podemos estimar limitantes para o erro e para o resíduo da seguinte forma

$$\begin{aligned}
\|v_h^{m+1}\| &= \| (I - \widehat{L}_h^{-1} L_h) v_h^m \| \leq \| (I - \widehat{L}_h^{-1} L_h) \| \|v_h^m\| \\
\|d_h^{m+1}\| &= \| (I - L_h \widehat{L}_h^{-1}) d_h^m \| \leq \| (I - L_h \widehat{L}_h^{-1}) \| \|d_h^m\|.
\end{aligned}$$

Essas desigualdades fornecem uma estimativa para duas propriedades importantes, o erro e o resíduo. O Lema 1, afirma que a velocidade de decaimento do erro e do resíduo é a mesma. Porém, tanto o erro como o resíduo têm que respeitar a ordem de aproximação do método de discretização.

Lema 1.

$$\rho((I - \widehat{L}_h^{-1}L_h)) = \rho((I - L_h\widehat{L}_h^{-1})).$$

Demonstração:

$$M_h = (I - \widehat{L}_h^{-1}L_h) = L_h^{-1}(I - L_h\widehat{L}_h^{-1})L_h$$

$$\det(M_h) = \det(L_h^{-1}(I - L_h\widehat{L}_h^{-1})L_h) = \det(L_h^{-1}) \det((I - L_h\widehat{L}_h^{-1})) \det(L_h),$$

logo,

$$\det(I - \widehat{L}_h^{-1}L_h) = \det(I - L_h\widehat{L}_h^{-1}),$$

portanto,

$$\rho((I - \widehat{L}_h^{-1}L_h)) = \rho((I - L_h\widehat{L}_h^{-1})),$$

□

A motivação para introduzirmos o operador \widehat{L}_h^{-1} , inversível que aproxima L_h^{-1} , deve-se ao fato de os métodos iterativos clássicos como Jacobi ou Gauss Seidel, quando aplicados ao problema (3.4), poderem ser interpretados como uma aproximação para resolver a equação residual. Veremos como fica este processo para o método de Jacobi:

$$x = D^{-1}(L + U)x + D^{-1}b$$

$$A = D - L - U \implies D - A = L + U$$

$$x = D^{-1}(D - A)x + D^{-1}b$$

$$x = D^{-1}Dx - D^{-1}Ax + D^{-1}b$$

$$x = x - D^{-1}Ax + D^{-1}b$$

$$x = x + D^{-1}(b - Ax)$$

$$x_{m+1} = x_m + D^{-1}d_m.$$

Neste caso $\widehat{L}_h = D$.

Mais genericamente, $M_h = (I - \widehat{L}_h^{-1}L_h)$

$$u_h^{m+1} = M_h u_h^m + S_h$$

$$u_h^{m+1} = (I - \widehat{L}_h^{-1}L_h)u_h^m + \widehat{L}_h^{-1}f_h$$

$$\begin{aligned}
u_h^{m+1} &= u_h^m - \widehat{L}_h^{-1} L_h u_h^m + \widehat{L}_h^{-1} f_h \\
u_h^{m+1} &= u_h^m + \widehat{L}_h^{-1} (f_h - L_h u_h^m) \\
u_h^{m+1} &= u_h^m + \widehat{L}_h^{-1} d_h^m.
\end{aligned}$$

Isso ajuda a entender a introdução do \widehat{L}_h .

Correção em Malha Grossa

Nesta subseção, daremos uma primeira idéia sobre a correção em malha grosseira. Considere os principais elementos dos métodos Multigrid: o operador diferencial discreto, o operador de restrição e o de interpolação. Considere por facilidade de notação que $H = 2h$.

Operador discreto na malha grossa:

$$L_H : g(\Omega_H) \longrightarrow g(\Omega_H).$$

Operador de restrição:

$$I_h^H : g(\Omega_h) \longrightarrow g(\Omega_H).$$

Operador de interpolação:

$$I_H^h : g(\Omega_H) \longrightarrow g(\Omega_h).$$

Os operadores de restrição e interpolação atuam nas funções discretas da seguinte maneira:

$$\begin{aligned}
d_H^m &= I_h^H d_h^m \\
\widehat{v}_h^m &= I_H^h \widehat{v}_H^m.
\end{aligned}$$

Agora vejamos como é o procedimento para montar a matriz de iteração para a correção na malha grosseira.

- Seja u_h^m , a estimativa inicial para um método iterativo
- Calcule o resíduo $d_h^m = f_h - L_h u_h^m$
- Restrinja o resíduo $d_H^m = I_h^H d_h^m$

- Resolva em Ω_H e equação residual $L_H \widehat{v}_H^h = d_H^m$
- Interpole a correção $\widehat{v}_h^m = I_H^h \widehat{v}_H^m$
- Faça a correção $u_h^{m+1} = u_h^m + \widehat{v}_h^m$

Com esse processo, podemos montar a equação de iteração do ciclo de duas malhas:

$$u_h^{m+1} = u_h^m + \widehat{v}_h^m = u_h^m + I_H^h \widehat{v}_H^m = u_h^m + I_H^h L_H^{-1} d_H^m$$

$$u_h^{m+1} = u_h^m + I_H^h L_H^{-1} I_h^H d_h^m$$

$$u_h^{m+1} = u_h^m + I_H^h L_H^{-1} I_h^H (f_h - L_h u_h^m)$$

$$u_h^{m+1} = u_h^m + I_H^h L_H^{-1} I_h^H f_h - I_H^h L_H^{-1} I_h^H L_h u_h^m$$

$$u_h^{m+1} = (I - I_H^h L_H^{-1} I_h^H L_h) u_h^m + I_H^h L_H^{-1} I_h^H f_h.$$

Definindo $M_h = I - I_H^h L_H^{-1} I_h^H L_h$ e $S_h = I_H^h L_H^{-1} I_h^H f_h$,

podemos escrever a seqüência de iteração da seguinte forma:

$$u_h^{m+1} = M_h u_h^m + S_h.$$

De acordo com o Capítulo 2, seria muito bom que o raio espectral da matriz M_h fosse menor que 1. Infelizmente isso não acontece, o que inviabiliza o uso da correção de malha grosseira, isto é,

$$\rho(I - I_H^h L_H^{-1} I_h^H L_h) \geq 1.$$

Para o operador I_h^H , temos $n = \dim(N(I_h^H)) + \frac{n}{2}$, logo $\dim(N(I_h^H)) = \frac{n}{2}$.

Com isso, I_h^H não é injetivo, portanto $I_H^h L_H^{-1} I_h^H L_h v_h = 0$ para algum $v_h \neq 0$, logo $I_H^h L_H^{-1} I_h^H L_h$ não é invertível.

Esse argumento inviabiliza o uso da correção na malha grosseira, pois não podemos garantir a convergência do método iterativo. Para contornar esse problema, usa-se a correção na malha grosseira com suavização feita por algum método iterativo do capítulo

anterior.

Correção em Malha Grosseira com Suavização

Como não tivemos condição de usar a correção em malha grosseira, temos de fazer uma modificação nesse processo para que o método convirja. Assim montamos a equação de iteração do ciclo de duas malhas com suavização

Sendo conhecido u_h^m , a estimativa inicial para um método iterativo e consideramos o algoritmo abaixo.

(1) Pré-suavização

- Suavizar u_h^m com algum método do Capítulo 2, ν_1 vezes, $\bar{u}_h^m = relax(u_h^m, L_h, f_h)$

(2) Correção em malha grosseira

- Calcule o resíduo $\bar{d}_h^m := f_h - L_h \bar{u}_h^m$
- Restrinja o resíduo $\bar{d}_H^m = I_h^H \bar{d}_h^m$
- Resolva em Ω_H e equação residual $L_H \hat{v}_H^m = \bar{d}_H^m$
- Interpole a correção $\hat{v}_h^m = I_H^h \hat{v}_H^m$
- Correção $\bar{u}_h^{m+1} = \bar{u}_h^m + \hat{v}_h^m$

(3) Pós-suavização

- Suavizar \bar{u}_h^{m+1} com algum método do Capítulo 2, ν_2 vezes, $\bar{u}_h^{m+1} = relax(\bar{u}_h^{m+1}, L_h, f_h)$

Com este algoritmo podemos encontrar a equação de iteração:

$$u_h^{m+1} = S^{\nu_2} \bar{u}_h^{m+1} = S^{\nu_2} (\bar{u}_h^m + \hat{v}_h^m) = S^{\nu_2} \bar{u}_h^m + S^{\nu_2} \hat{v}_h^m$$

$$u_h^{m+1} = S^{\nu_2} \bar{u}_h^m + S^{\nu_2} I_H^h \hat{v}_H^m = S^{\nu_2} \bar{u}_h^m + S^{\nu_2} I_H^h \hat{L}_H^{-1} \bar{d}_H^m$$

$$u_h^{m+1} = S^{\nu_2} \bar{u}_h^m + S^{\nu_2} I_H^h L_H^{-1} I_h^H \bar{d}_h^m = S^{\nu_2} \bar{u}_h^m + S^{\nu_2} I_H^h L_H^{-1} I_h^H (f_h - L_h \bar{u}_h^m)$$

$$u_h^{m+1} = S^{\nu_2} \bar{u}_h^m + S^{\nu_2} I_H^h L_H^{-1} I_h^H (f_h - L_h \bar{u}_h^m)$$

$$u_h^{m+1} = S^{\nu_2} \bar{u}_h^m - S^{\nu_2} I_H^h L_H^{-1} I_h^H L_h \bar{u}_h^m + S^{\nu_2} I_H^h L_H^{-1} I_h^H f_h$$

$$u_h^{m+1} = (S^{\nu_2} - S^{\nu_2} I_H^h L_H^{-1} I_h^H L_h) \bar{u}_h^m + S^{\nu_2} I_H^h L_H^{-1} I_h^H f_h$$

$$u_h^{m+1} = S^{\nu_2} (I - I_H^h L_H^{-1} I_h^H L_h) \bar{u}_h^m + S^{\nu_2} I_H^h L_H^{-1} I_h^H f_h$$

$$u_h^{m+1} = S^{\nu_2} (I - I_H^h L_H^{-1} I_h^H L_h) S^{\nu_1} u_h^m + S^{\nu_2} I_H^h L_H^{-1} I_h^H f_h$$

Definindo $K_h = I - I_H^h L_H^{-1} I_h^H L_h$ e $M_h = S^{\nu_2} K_h S^{\nu_1}$, teremos a equação de iteração

$$u_h^{m+1} = M_h u_h^m + S_h,$$

onde $S_h = S^{\nu_2} I_H^h L_H^{-1} I_h^H f_h$.

Chega-se, dessa forma, a um processo iterativo, sendo válidos os teoremas do Capítulo 2 para garantir à sua convergência.

3.2.4 O Ciclo Multigrid

Em seções anteriores descrevemos a correção numa malha grosseira, um elemento do ciclo de duas malhas que serve de base para os métodos Multigrid, pois de maneira simplificada, os métodos Multigrid são uma seqüência de correções em malha grosseira com suavização.

A idéia do Multigrid partiu de observações da boa convergência da correção em malha grosseira com suavização. Como estamos fazendo a correção em malha grosseira, surgiu a necessidade de resolvermos o sistema linear nesta malha, $L_H \hat{v}_H = d_H$.

Podemos observar que a equação $L_H \hat{v}_H = d_H$ é da mesma forma que a equação $L_h u_h = f_h$. Aplicando recursivamente a correção em malha grosseira até onde se deseja, por exemplo fazer somente duas correções em malha grosseira, encontrando uma aproximação por meio de um método iterativo, ou mesmo obtendo a solução por meio de um método direto. Se necessário podemos descer até a malha mais grossa onde, para resolvermos o problema, é necessário apenas uma operação de divisão.

A Figura (3.14) ilustra alguns tipos de ciclos que podem ser utilizados no controle da correção da malha grosseira. O ciclo localizado na parte superior da figura será chamado de ciclo V e terá um parâmetro associado ($\gamma = 1$); o ciclo localizado na parte intermediária da figura será chamado de ciclo W e terá um parâmetro associado ($\gamma = 2$); enquanto o

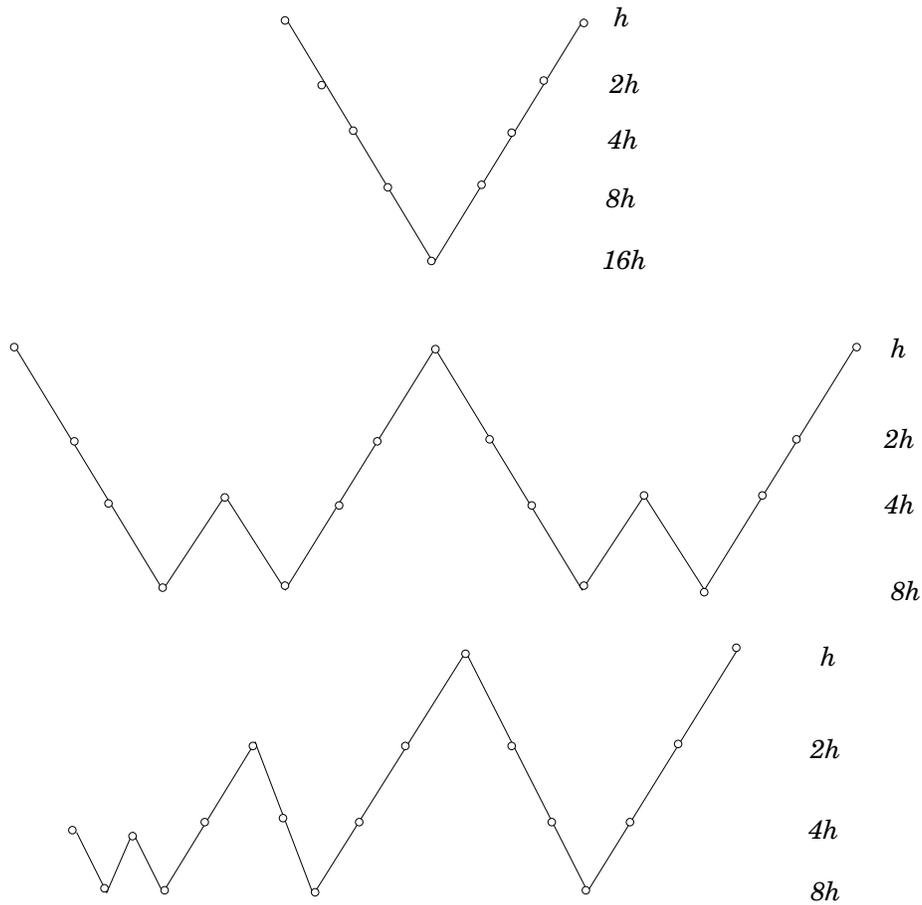


Figura 3.14: Ciclo V, Ciclo W e FMG

ciclo localizado na parte inferior da figura será chamado de ciclo FMG “Full Multigrid”.

Para controlar a aplicação sucessiva da correção em malha grosseira, usamos os ciclos ilustrados na Figura (3.14), formando assim a estrutura dos métodos Multigrid.

Quando voltamos ao mesmo nível em que iniciamos a correção da malha grosseira, completamos um ciclo e damos por finalizada uma iteração. Então, se para resolver um sistema linear usando os métodos Multigrid é necessário 10 iterações, isso corresponde a ter feito 10 ciclos, ou seja 10 descidas e subidas.

Observação 2. *É importante esclarecer que nos métodos Multigrid, o número de vezes que é aplicado um esquema de relaxação varia de acordo com os parâmetros ν_1 e ν_2 , portanto em uma iteração dos métodos Multigrid o número de vezes que usamos o esquema de relaxação é $(\nu_1 + \nu_2) * l$, tal que l é o número de correção em malha grosseira.*

3.2.5 Estrutura dos Métodos Multigrid

No Multigrid é usada uma seqüência de malhas representadas abaixo, tal que h_k é o tamanho da partição da malha, $k = l, l-1, \dots, 0$. A seqüência de malhas é representada por:

$$\Omega_{h_l}, \Omega_{h_{l-1}}, \dots, \Omega_{h_0}$$

A notação Ω_{h_0} representa a malha mais grossa da seqüência com tamanho da partição h_0 , e Ω_{h_l} representa a malha mais fina com tamanho da partição h_l .

Podemos representar Ω_{h_k} por Ω_k com $k = l, l-1, \dots, 0$, simplesmente para facilitar a notação. Redefinimos os operadores elementares do Multigrid da seguinte maneira:

$$\begin{aligned} L_k : g(\Omega_k) &\longrightarrow g(\Omega_k) & I_k^{k-1} : g(\Omega_k) &\longrightarrow g(\Omega_{k-1}) \\ I_{k-1}^k : g(\Omega_{k-1}) &\longrightarrow g(\Omega_k) & S_k : g(\Omega_k) &\longrightarrow g(\Omega_k) \end{aligned}$$

O operador S_k^ν representa o número de vezes que a relaxação (Jacobi, Gauss Seidel) é aplicada em uma função discreta com domínio Ω_k .

Tendo definido os seus principais elementos, podemos gerar uma equação de iteração para os métodos Multigrid usando o algoritmo a seguir.

Usando o operador L_k com $k = l, l-1, \dots, 0$ e $I_k^{k-1}, I_{k-1}^k, S_k$ com $k = l, l-1, \dots, 1$. Usando ν_1 e ν_2 como parâmetros de pré e pós-relaxação, γ é uma constante fixada que representa o tipo de ciclo.

O objetivo é encontrar u_k^{m+1} , conhecido u_k^m para iniciar o processo iterativo.

CICLO MULTIGRID

$$u_k^{m+1} = \text{mgciclo}(k, \gamma, u_k^m, f_k, \nu_1, \nu_2)$$

(1) Pré-relaxação

- Obtenha $\bar{u}_k^m = \text{relax}(u_k^m, L_k, f_k), \nu_1$ vezes

(2) Correção em grid grosseiro

- Calcule o resíduo $\bar{d}_k^m = f_k - \bar{u}_k^m$
- Restrinja o resíduo $\bar{d}_k^{m-1} = I_k^{k-1} \bar{d}_k^m$

- Se $k = 1$, resolva o sistema linear $L_{k-1}\bar{v}_{k-1}^m = \bar{d}_k^{m-1}$

se não faça $\hat{v}_{k-1}^m = \text{mgciclo}(k, \gamma, u_k^m, f_k, \nu_1, \nu_2)$

- Interpole a correção $\hat{v}_k^m = I_{k-1}^k \hat{v}_{k-1}^m$
- Obtenha a correção $\bar{u}_k^{m+1} = \bar{u}_k^m + \hat{v}_k^m$

(3) Pós-relaxação

- Obtenha $\bar{u}_k^{m+1} = \text{relax}(\bar{u}_k^{m+1}, L_k, f_k), \nu_2$ vezes

Para encontrarmos qual é a matriz de iteração, considere o teorema abaixo.

Teorema 14 . A matriz de iteração M_l é dada pela seguinte recursão:

$$M_0 = 0$$

$$M_k = S_k^{\nu_2} (I_k - I_{k-1}^k (I_{k-1} - M_{k-1}^\gamma) L_{k-1}^{-1} I_k^{k-1} L_k) S_k^{\nu_1} \quad (k = 1, \dots, l)$$

Demonstração: A demonstração será feita por indução sobre k . Note que I_k representa a matriz identidade.

Verificar se é válido para $k = 1$

$$M_1 = S_1^{\nu_2} (I_1 - I_0^1 (I_0 - M_0^\gamma) L_0^{-1} I_1^0 L_1) S_1^{\nu_1}$$

$M_0 = 0$, por hipótese,

$$M_1 = S_1^{\nu_2} (I_1 - I_0^1 I_0 L_0^{-1} I_1^0 L_1) S_1^{\nu_1}$$

$$M_1 = S_1^{\nu_2} (I_1 - I_0^1 L_0^{-1} I_1^0 L_1) S_1^{\nu_1}$$

M_1 é equivalente ao operador na malha grosseira para $k = 1$:

$$M_1^0 = S_1^{\nu_2} (I_1 - I_0^1 L_0^{-1} I_1^0 L_1) S_1^{\nu_1}.$$

Portanto, é válido para $k = 1$.

Suponha válido para $k - 1$, ou seja, a matriz de iteração de $k - 1$ é

$$M_{k-1} = S_{k-1}^{\nu_2} (I_{k-1} - I_{k-2}^{k-1} (I_{k-2} - M_{k-2}^\gamma) L_{k-2}^{-1} I_{k-1}^{k-2} L_{k-1}) S_{k-1}^{\nu_1}.$$

Agora temos que fazer a correção em malha grosseira para k . Acompanhando o processo iterativo, teremos:

$$u_k^{m+1} = S_{k-1}^{\nu_2} \bar{u}^{m+1} = S_{k-1}^{\nu_2} (\bar{u}_k^m + \hat{v}_k^m) = S_{k-1}^{\nu_2} (\bar{u}_k^m + I_{k-1}^k \hat{v}_{k-1}^m).$$

Para a correção na malha grosseira temos que M_{k-1} é válido por hipótese. Como a partir deste passo o processo iterativo sempre parte de uma estimativa inicial nula, para resolver a equação $L_k \hat{u}_k^m = \bar{d}_k^m$, podemos usar a equação (3.11):

$$\hat{v}_{k-1}^m = (I_{k-1} - M_{k-1}^\gamma) L_{k-1}^{-1} d_{k-1}^m.$$

Portanto,

$$\begin{aligned} u_k^{m+1} &= S_{k-1}^{\nu_2} (\bar{u}_k^m + I_{k-1}^k (I_{k-1} - M_{k-1}^\gamma) L_{k-1}^{-1} d_{k-1}^m) = \\ &= S_{k-1}^{\nu_2} (\bar{u}_k^m + I_{k-1}^k (I_{k-1} - M_{k-1}^\gamma) L_{k-1}^{-1} I_k^{k-1} d_k^m) = \\ &= S_{k-1}^{\nu_2} (\bar{u}_k^m + I_{k-1}^k I_{k-1} L_{k-1}^{-1} I_k^{k-1} d_k^m - I_{k-1}^k M_{k-1}^\gamma L_{k-1}^{-1} I_k^{k-1} d_k^m) = \\ &= S_{k-1}^{\nu_2} (\bar{u}_k^m + I_{k-1}^k I_{k-1} L_{k-1}^{-1} I_k^{k-1} (f_k - L_k \bar{u}_k^m) - I_{k-1}^k M_{k-1}^\gamma L_{k-1}^{-1} I_k^{k-1} (f_k - L_k \bar{u}_k^m)) \\ \\ u_k^{m+1} &= S_{k-1}^{\nu_2} (\bar{u}_k^m - I_{k-1}^k I_{k-1} L_{k-1}^{-1} I_k^{k-1} L_k \bar{u}_k^m) - I_{k-1}^k M_{k-1}^\gamma L_{k-1}^{-1} I_k^{k-1} L_k \bar{u}_k^m) + \\ &\quad + (I_{k-1}^k I_{k-1} L_{k-1}^{-1} I_k^{k-1} - I_{k-1}^k M_{k-1}^\gamma L_{k-1}^{-1} I_k^{k-1}) f_k. \end{aligned}$$

Definindo $S_k = I_{k-1}^k I_{k-1} L_{k-1}^{-1} I_k^{k-1} - I_{k-1}^k M_{k-1}^\gamma L_{k-1}^{-1} I_k^{k-1}$

$$\begin{aligned} u_k^{m+1} &= S_{k-1}^{\nu_2} (I_k - I_{k-1}^k (I_{k-1} - M_{k-1}^\gamma) L_{k-1}^{-1} I_k^{k-1} L_k) \bar{u}_k^m + S_k \\ u_k^{m+1} &= S_{k-1}^{\nu_2} (I_k - I_{k-1}^k (I_{k-1} - M_{k-1}^\gamma) L_{k-1}^{-1} I_k^{k-1} L_k) S_{k-1}^{\nu_1} u_k^m + S_k \end{aligned}$$

$$u_k^{m+1} = M_k u_k^m + S_k.$$

□

Com este teorema geramos a seqüência de iteração para os métodos Multigrid.

Capítulo 4

Resultados Numéricos

Neste capítulo será apresentado os resultados numéricos para Métodos Iterativos e Multigrid, respectivamente nas seções 4.1 e 4.2.

Para implementação dos métodos usamos a linguagem de programação Fortran77 com precisão dupla. Para executar esses programas usamos um microcomputador com processador de 1.0 Ghz e 256 MB de memória RAM.

4.1 Resultados Numéricos para os Métodos Iterativos do Capítulo 2

Nesta seção são apresentadas as experiências computacionais realizadas para os problemas modelos descritos no Capítulo 1. Os Métodos iterativos são aqueles descritos no Capítulo 2.

O critério para fornecer a estimativa inicial para os métodos iterativos consiste em perturbar a solução exata dos problemas modelos por uma função $v(x, y) = \text{sen}(k_1\pi x)\text{sen}(k_2\pi y)$, k_1 e $k_2 \in \mathbb{N}$. Esta perturbação mantém rígidas as condições de contorno exigidas para cada problema modelo.

Os critérios de parada usados nos métodos iterativos foram escolhidos de duas maneiras:

Divergência: é estipulado um número máximo de iteração: 20000. Nas tabelas o símbolo *, indica que o métodos iterativo ultrapassaram esse número máximo de iterações.

Convergência: é dada uma tolerância (tol) que varia de acordo com cada problema modelo. Essa tolerância é comparada com a norma euclidiana do resíduo ($r = f - Au$). Quando o resíduo for menor ou igual a tolerância, uma boa solução aproximada terá sido

obtida.

Na construção das tabelas que seguem, a primeira coluna representa o tipo de método usado seguindo a nomenclatura:

JAC - Método de Jacobi

JOR - Método de Jacobi com sobre relaxação

GSF - Método Gauss Seidel forward

GSB - Método Gauss Seidel backward

SORF - Método da sobre relaxação sucessiva forward

SORB - Método da sobre relaxação sucessiva backward

SSOR - Método SOR simétrico

RBGSF - Método de Gauss Seidel forward com ordenação red black

RBGSB - Método de Gauss Seidel backward com ordenação red black

RBSORF- Método SOR forward com ordenação red black

RBSORB- Método SOR backward com ordenação red black

Nas outras colunas temos o número de vezes que foi particionada a malha na direção x e y do domínio. Usamos o número de partição n : 17, 33, 65 e 129. Portanto, além da coluna definindo o método iterativo usado, temos mais quatro colunas, uma para cada n dado acima. Cada coluna é subdividida em outras duas: uma com o número de iteração (Iter) e outra com o tempo de CPU (CPU), em segundos, necessário para que o método iterativo alcance o critério de convergência.

Como comentado no Capítulo 2, para os métodos JOR, SORF, SORB, SSOR, RB-SORF e RBSORB, existe a necessidade de definir um parâmetro de sobre-relaxação. Foram feitos alguns experimentos numéricos para encontrar esse parâmetro. Dessa forma definimos os seguintes parâmetros:

$$n = 17 - \omega = 1.68$$

$$n = 33 - \omega = 1.86$$

$$n = 65 - \omega = 1.92$$

$$n = 129 - \omega = 1.96$$

Ocorre uma variação de ω para os métodos iterativos SORF, SORB, SSOR, RBSORF e RBSORB, para cada n . Porém esta variação não altera significativamente o número de iterações. Por esse motivo é fixado ω com os valores acima para cada n , em todos os problemas modelos. A única exceção é o JOR, com $\omega = 1.03$ em todos os métodos iterativos.

PM I	n=17		n=33		n=65		n=129	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
JAC	1448	0.200	5963	2.799	*	40.5	*	217.239
JOR	1434	0.170	*	9.579	*	39.090	*	222.870
GSF	725	0.069	2983	1.230	12229	20.240	*	183.580
GSB	725	0.069	2983	1.239	12229	20.099	*	181.519
SORF	81	0.009	209	0.089	385	0.690	809	7.809
SORB	81	0.009	209	0.090	385	0.690	809	7.709
SSOR	108	0.020	226	0.140	461	1.130	950	11.669
RBGSF	733	0.070	3018	1.220	12372	20.369	*	182.359
RBGSB	733	0.069	3018	1.239	12372	20.410	*	177.720
RBSORF	73	0.009	199	0.079	363	0.639	804	7.760
RBSORB	73	0.009	199	0.090	363	0.660	804	7.560

Tabela 4.1: Problema Modelo I - $tol = 10^{-10}$

PM II	n=17		n=33		n=65		n=129	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
JAC	1444	0.259	5949	2.759	*	40.989	*	219.139
JOR	*	2.289	*	9.400	*	38.859	*	227.770
GSF	723	0.069	2976	1.189	12200	20.160	*	184.460
GSB	723	0.070	208	0.090	12200	19.980	*	181.690
SORF	81	0.009	208	0.089	385	0.690	831	8.089
SORB	81	0.009	208	0.090	385	0.700	831	7.960
SSOR	108	0.019	225	0.130	460	1.130	971	11.969
RBGSF	732	0.069	3011	1.190	12344	20.140	*	182.819
RBGSB	733	0.069	3011	1.199	12344	20.299	*	178.859
RBSORF	73	0.009	199	0.079	363	0.639	825	7.560
RBSORB	73	0.010	199	0.090	363	0.639	825	7.560

Tabela 4.2: Problema Modelo II - $tol = 10^{-8}$

PM III	n=17		n=33		n=65		n=129	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
JAC	1448	0.180	5965	2.859	*	41.600	*	218.479
JOR	1434	0.159	*	9.400	*	41.60	*	228.620
GSF	725	0.079	2983	1.200	10318	17.099	*	183.770
GSB	725	0.070	2983	1.189	10318	17.010	*	180.819
SORF	81	0.009	209	0.090	338	0.610	743	7.20
SORB	81	0.009	209	0.089	338	0.610	743	7.130
SSOR	108	0.019	226	0.140	390	0.960	809	9.989
RBGSF	734	0.080	3018	1.179	10461	17.119	*	182.410
RBGSB	734	0.069	3018	1.230	10461	17.169	*	177.309
RBSORF	73	0.009	199	0.079	303	0.529	689	6.660
RBSORB	73	0.009	199	0.079	303	0.539	689	6.430

Tabela 4.3: Problema Modelo III - $tol = 10^{-8}$

PM IV	n=17		n=33		n=65		n=129	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
JAC	1448	0.189	5963	2.750	*	42.479	*	216.599
JOR	*	2.299	*	9.449	*	40.200	*	224.939
GSF	725	0.069	2983	1.190	12229	20.309	*	182.489
GSB	725	0.069	2983	1.189	12229	20.110	*	180.340
SORF	81	0.009	209	0.090	385	0.699	824	7.950
SORB	81	0.009	209	0.089	385	0.699	815	7.850
SSOR	108	0.020	226	0.130	461	1.139	267	11.809
RBGSF	733	0.069	3018	1.189	12372	20.290	*	183.789
RBGSB	733	0.069	3018	1.200	12372	20.339	*	177.929
RBSORF	73	0.009	199	0.090	363	0.649	809	7.889
RBSORB	73	0.009	199	0.090	363	0.649	809	7.680

Tabela 4.4: Problema Modelo IV - $tol = 10^{-8}$

PM V	n=17		n=33		n=65		n=129	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
JAC	1211	0.150	5009	2.440	*	42.330	*	214.070
JOR	*	2.289	*	9.400	*	38.970	*	223.790
GSF	607	0.059	2506	1.010	10318	17.110	*	186.159
GSB	607	0.059	2506	1.010	10318	17.129	*	182.030
SORF	69	0.009	183	0.079	338	0.610	751	7.270
SORB	69	0.009	183	0.079	338	0.610	747	7.189
SSOR	91	0.019	190	0.110	390	0.970	811	10.030
RBGSF	615	0.060	3018	1.189	10461	17.239	*	183.130
RBGSB	615	0.059	2541	1.020	10461	17.279	*	179.260
RBSORF	69	0.009	173	0.079	303	0.550	690	6.799
RBSORB	69	0.009	173	0.069	363	0.649	690	6.799

Tabela 4.5: Problema Modelo V - $tol = 10^{-7}$

PM VI	n=17		n=33		n=65		n=129	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
JAC	1211	0.170	5009	2.409	*	40.840	*	216.199
JOR	*	2.279	*	9.399	*	38.669	*	224.489
GSF	606	0.059	2506	1.000	10318	17.009	*	182.410
GSB	607	0.059	2506	1.000	10318	16.930	*	180.809998
SORF	69	0.009	183	0.079	338	0.610	741	7.230
SORB	69	0.009	190	0.109	338	0.610	741	7.119
SSOR	91	0.019	190	0.110	390	0.960	808	9.979
RBGSF	615	0.060	2541	1.010	10461	17.110	*	190.079
RBGSB	615	0.059	2541	1.009	10461	17.160	*	181.510
RBSORF	69	0.009	173	0.069	303	0.540	689	6.780
RBSORB	69	0.009	173	0.079	303	0.539	689	6.610

Tabela 4.6: Problema Modelo VI - $tol = 10^{-7}$

PM VII	n=17		n=33		n=65		n=129	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
JAC	1203	0.170	4977	2.210	*	40.970	*	217.229
JOR	*	2.289	*	9.409	*	38.750	*	223.029
GSF	602	0.059	2488	0.989	10247	16.830	*	182.739
GSB	604	0.049	2491	0.989	10318	16.930	*	180.289
SORF	67	0.009	184	0.069	340	0.639	768	7.460
SORB	68	0.009	187	0.090	346	0.620	760	7.329
SSOR	90	0.019	189	0.119	387	0.949	814	10.070
RBGSF	611	0.060	2524	0.990	10392	16.929	*	187.269
RBGSB	611	0.059	2524	1.000	10392	16.929	*	179.390
RBSORF	65	0.009	171	0.069	345	0.619	758	7.160
RBSORB	65	0.009	171	0.069	345	0.610	750	7.129

Tabela 4.7: Problema Modelo VII - $tol = 10^{-7}$

PM VIII	n=17		n=33		n=65		n=129	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
JAC	1183	0.140	4895	2.210	*	41.270	*	211.010
JOR	*	2.309	*	9.409	*	38.680	*	222.570
GSF	591	0.059	2445	0.980	10076	16.409	*	182.399
GSB	595	0.059	2453	0.979	350	0.630	*	180.289
SORF	67	0.009	178	0.039	329	0.620	768	7.460
SORB	70	0.009	189	0.079	350	0.630	764	7.349
SSOR	89	0.009	186	0.109	381	0.930	790	9.729
RBGSF	601	0.060	2483	0.989	10223	16.610	*	184.469
RBGSB	601	0.059	2483	1.000	10223	16.639	*	185.030
RBSORF	71	0.009	184	0.079	341	0.610	747	7.359
RBSORB	71	0.009	184	0.090	341	0.610	747	7.229

Tabela 4.8: Problema Modelo VIII - $tol = 10^{-7}$

PM IX	n=17		n=33		n=65		n=129	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
JAC	1431	0.219	5894	2.809	*	41.27	*	212.949
JOR	*	2.279	*	9.320	*	38.790	*	225.059
GSF	718	0.069	2946	1.170	12082	20.150	*	181.990
GSB	715	0.070	2983	1.170	12090	19.840	*	177.809
SORF	84	0.009	206	0.120	385	0.790	801	7.869
SORB	81	0.009	211	0.0999	389	0.699	815	7.729
SSOR	107	0.019	223	0.140	456	1.110	940	11.500
RBGSF	725	0.069	2983	1.170	12227	19.989	*	181.210
RBGSB	725	0.069	2983	1.1700	12227	19.839	*	175.290
RBSORF	83	0.009	213	0.090	401	0.710	860	8.260
RBSORB	83	0.009	213	0.090	401	0.710	860	8.109

Tabela 4.9: Problema Modelo IX - $tol = 10^{-10}$

PM X	n=17		n=33		n=65		n=129	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
JAC	1184	0.150	4896	2.299	*	41.170	*	247.370
JOR	*	2.269	*	9.359	*	40.500	*	241.619
GSF	591	0.059	2445	0.970	10076	16.650	*	199.600
GSB	595	0.059	2453	0.969	10091	17.869	*	194.549
SORF	70	0.019	181	0.079	331	0.700	721	7.349
SORB	72	0.008	185	0.090	348	0.729	767	7.800
SSOR	89	0.019	186	0.110	381	1.069	795	10.289
RBGSF	601	0.060	2483	0.969	10223	19.139	*	195.100
RBGSB	601	0.049	2483	0.980	10223	17.690	*	175.290
RBSORF	71	0.009	184	0.079	341	0.650	756	7.450
RBSORB	71	0.009	184	0.079	341	0.669	756	7.250

Tabela 4.10: Problema Modelo X - $tol = 10^{-7}$

Um fato interessante é a confirmação do Corolário 3 do Capítulo 2. Se A é consistentemente ordenada então $\rho(G_{gs}) = \rho(G_J)^2$. Este resultado afirma que Gauss Seidel é duas vezes mais rápido que Jacobi. Os resultados numéricos, de fato, confirmam a teoria.

O método JOR mostrou resultados ruins para o $\omega = 1.03$. Mesmo com a variação de ω os resultados não melhoram.

Observamos que o desempenho do RBGS e do Gauss Seidel são muito semelhantes, apesar das ilustrações do Capítulo 3 indicarem que o método RBGS teria uma taxa de decaimento do erro melhor que o Gauss Seidel. Entretanto, a vantagem do RBGS em relação ao Gauss Seidel tanto o backward como forward é que RBGS é paralelizável.

Nem as ordenações backward, forward e SOR simétrica mostraram resultados convincentes que justificassem a preferência de uma dessas ordenações em relação às outras.

Podemos constatar a dificuldade de convergência dos métodos iterativos quando o número de partições aumenta. Por exemplo, com $n = 65$, com excessão do SOR, todos os outros métodos ultrapassaram 10000 iterações e quando $n = 129$, o limite de iteração definido como critério de parada é atingido.

Pelos resultados das Tabelas 4.1 à 4.10, nota-se que o melhor desempenho entre os métodos sempre foi daqueles com relaxação sucessiva, exceto o JOR.

Para auxiliar as observações consideramos as Tabelas 4.11, 4.12 e 4.13. Na primeira coluna, é definido o problema modelo, o número de iterações e o tempo de CPU que o método iterativo leva para resolver o sistema linear. A segunda coluna está subdividida em outras duas: uma com o número de iterações e outra com tempo de CPU do RBGS e do RBSOR, respectivamente. Na terceira coluna consta a razão entre o número de iteração do RBGS e o número de iteração do RBSOR. Usaremos o mesmo procedimento para o tempo de CPU.

Os problemas modelos PM I, PM V e PM X foram escolhidos aleatoriamente, enquanto que os métodos RBSOR e RBGS foram escolhidos por apresentarem a ordenação red black que será usada nos métodos Multigrid.

PM I	n=65		Razão
	RBGS	RBSOR	
Iter	12372	163	34
CPU	20.369	0.639	32

Tabela 4.11: Problema I - Razão = $\frac{RBGS}{RBSOR}$

Os resultados confirmam a afirmação feita anteriormante, de que os métodos com

PM V	n=65		Razão
	RBGS	RBSOR	
Iter	10461	303	34
CPU	17.239	0.550	31

Tabela 4.12: Problema V - Razão = $\frac{RBGS}{RBSOR}$

PM X	n=65		Razão
	RBGS	RBSOR	
Iter	10223	341	30
CPU	19.139	0.650	29

Tabela 4.13: Problema X - Razão = $\frac{RBGS}{RBSOR}$

relaxação sucessiva tem o melhor desempenho entre os métodos iterativos. As Tabelas 4.11, 4.12 e 4.13 mostram que RBSOR é aproximadamente 30 vezes mais rápido que o RBGS.

4.2 Resultados Numéricos para os Métodos Multigrid

Nesta seção são apresentadas as experiências computacionais realizadas para os métodos Multigrid descritos no Capítulo 3, que resolvem o sistema linear gerado pelos problemas modelos definidos no Capítulo 1.

Serão definidos os seguintes elementos dos métodos Multigrid usados em todas as experiências numéricas desta seção:

- O operador de restrição é o “Half Weighting”.
- A interpolação usada é a interpolação bilinear.
- O método de relaxação usado é o Gauss Seidel com ordenação red black.
- Os ciclos usados são os ciclos do tipo $V(\nu_1, \nu_2)$, onde ν_1 e ν_2 definem o número de vezes que é feito a pré e a pós relaxações.

Para os testes com os métodos Multigrid, a estimativa inicial e os critérios de parada são os mesmos usados nos testes anteriores exceto o número máximo de iterações, fixado em 500.

Na construção das tabelas, a primeira coluna define o tipo de ciclo e quantas pré e

pós-relaxações devem ser feitas no método Multigrid. Nas outras colunas da tabela é dado o número de vezes que foi particionado a malha na direção x e y do domínio. Para os experimentos com os métodos Multigrid usamos os valores $n = 17, 33, 65, 129$ e 513 .

Cada coluna mostra um valor de n , e é subdividida em outras duas: uma com o número de iterações (Iter) e outra com o tempo de CPU (CPU), em segundos, para a alcançar a convergência.

PM I	n=17		n=33		n=65		n=129		n=513	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
V(0,1)	17	0.009	21	0.039	25	0.189	30	1.060	37	23.400
V(1,0)	39	0.019	64	0.119	99	0.740	156	5.490	424	265.500
V(1,1)	7	0.009	8	0.019	9	0.090	10	0.430	12	9.010
V(1,2)	5	0.009	6	0.019	7	0.079	9	7.900	9	7.900
V(2,1)	5	0.009	6	0.019	7	0.079	7	0.340	9	7.880
V(2,2)	5	0.009	5	0.009	6	0.070	7	0.379	8	8.010

Tabela 4.14: Problema Modelo I - $tol = 10^{-10}$

PM II	n=17		n=33		n=65		n=129		n=513	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
V(0,1)	21	0.019	25	0.029	30	0.230	34	1.210	39	24.449
V(1,0)	33	0.009	54	0.100	84	0.600	133	4.639	369	231.310
V(1,1)	9	0.009	10	0.019	11	0.100	11	0.469	13	9.789
V(1,2)	6	0.009	7	0.019	7	0.070	7	0.329	7	6.150
V(2,1)	6	0.009	7	0.019	8	0.080	8	0.400	9	7.950
V(2,2)	5	0.009	6	0.019	7	0.090	7	0.379	7	7.039

Tabela 4.15: Problema Modelo II - $tol = 10^{-8}$

PM III	n=17		n=33		n=65		n=129		n=513	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
V(0,1)	13	0.009	14	0.019	15	0.119	17	0.600	24	15.069
V(1,0)	33	0.009	54	0.079	84	0.610	123	4.230	500	313.910
V(1,1)	5	0.009	6	0.019	7	0.060	7	0.300	9	6.76
V(1,2)	4	0.009	4	0.009	5	0.049	5	0.239	6	5.239
V(2,1)	4	0.009	5	0.009	5	0.050	5	0.239	6	5.280
V(2,2)	3	0.009	4	0.009	5	0.070	5	0.249	6	6.019

Tabela 4.16: Problema Modelo III - $tol = 10^{-8}$

PM IV	n=17		n=33		n=65		n=129		n=513	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
V(0,1)	15	0.009	16	0.029	18	0.119	21	0.750	28	17.560
V(1,0)	33	0.019	54	0.079	84	0.629	133	4.590	369	233.819
V(1,1)	6	0.009	6	0.009	7	0.070	8	0.319	10	7.500
V(1,2)	4	0.009	5	0.019	5	0.050	6	0.289	7	6.119
V(2,1)	5	0.009	5	0.009	5	0.050	6	0.300	7	6.170
V(2,2)	4	0.009	4	0.009	5	0.070	5	0.290	6	6.030

Tabela 4.17: Problema Modelo IV - $tol = 10^{-8}$

PM V	n=17		n=33		n=65		n=129		n=513	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
V(0,1)	13	0.009	14	0.029	15	0.109	17	0.600	24	15.100
V(1,0)	30	0.009	49	0.090	77	0.590	122	4.289	500	313.830
V(1,1)	4	0.009	5	0.019	6	0.059	7	0.289	9	6.790
V(1,2)	3	0.009	4	0.009	4	0.039	5	0.250	6	5.280
V(2,1)	3	0.009	4	0.009	5	0.049	5	0.239	6	5.299
V(2,2)	3	0.009	3	0.009	4	0.050	5	0.269	6	6.030

Tabela 4.18: Problema Modelo V - $tol = 10^{-7}$

PM VI	n=17		n=33		n=65		n=129		n=513	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
V(0,1)	16	0.009	19	0.019	20	0.159	20	0.710	24	14.979
V(1,0)	30	0.019	49	0.089	77	0.569	122	4.239	341	213.399
V(1,1)	7	0.009	8	0.009	8	0.070	8	0.340	9	6.750
V(1,2)	5	0.009	5	0.009	5	0.050	5	0.249	6	5.280
V(2,1)	5	0.009	6	0.009	6	0.060	6	0.280	6	5.260
V(2,2)	5	0.009	5	0.019	5	0.050	5	0.279	6	6.010

Tabela 4.19: Problema Modelo VI - $tol = 10^{-7}$

PM VII	n=17		n=33		n=65		n=129		n=513	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
V(0,1)	20	0.009	23	0.039	29	0.230	32	1.159	39	24.989
V(1,0)	30	0.019	49	0.100	79	0.589	123	4.379	500	322.549
V(1,1)	9	0.009	10	0.019	10	0.109	11	0.469	11	7.159
V(1,2)	6	0.009	7	0.019	7	0.0799	8	0.409	8	7.190
V(2,1)	7	0.009	8	0.019	8	0.090	9	0.449	10	8.889
V(2,2)	6	0.009	7	0.009	7	0.090	7	0.400	8	8.140

Tabela 4.20: Problema Modelo VII - $tol = 10^{-7}$

PM VIII	n=17		n=33		n=65		n=129		n=513	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
V(0,1)	18	0.009	22	0.050	29	0.189	29	1.029	31	19.289
V(1,0)	30	0.019	31	0.049	31	0.240	31	1.080	31	19.289
V(1,1)	9	0.009	10	0.019	10	0.090	11	0.459	13	9.760
V(1,2)	6	0.009	7	0.019	7	0.070	8	0.399	8	7.000
V(2,1)	7	0.009	8	0.019	8	0.090	9	0.449	10	8.770
V(2,2)	6	0.009	7	0.019	7	0.080	7	0.399	8	8.029

Tabela 4.21: Problema Modelo VIII - $tol = 10^{-7}$

PM IX	n=17		n=33		n=65		n=129		n=513	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
V(0,1)	26	0.009	31	0.050	37	0.280	41	1.440	48	30.159
V(1,0)	39	0.019	64	0.109	99	0.699	156	5.399	426	267.630
V(1,1)	11	0.009	13	0.029	14	0.119	15	0.639	16	12.019
V(1,2)	8	0.009	9	0.019	10	0.109	10	0.480	10	8.789
V(2,1)	9	0.009	10	0.009	8	0.090	11	0.530	12	10.540
V(2,2)	8	0.009	9	0.029	7	0.100	9	0.490	10	10.069

Tabela 4.22: Problema Modelo IX - $tol = 10^{-10}$

PM X	n=17		n=33		n=65		n=129		n=513	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
V(0,1)	25	0.009	27	0.050	34	0.270	35	1.200	36	22.739
V(1,0)	30	0.009	49	0.070	77	0.550	123	4.220	500	311.850
V(1,1)	10	0.009	11	0.019	12	0.109	12	0.479	13	9.739
V(1,2)	7	0.009	8	0.019	8	0.090	8	0.379	8	7.000
V(2,1)	7	0.009	8	0.019	8	0.090	9	0.439	10	8.760
V(2,2)	6	0.009	7	0.019	8	0.090	7	0.379	8	7.980

Tabela 4.23: Problema Modelo X - $tol = 10^{-7}$

A medida que n aumenta, os métodos iterativos tem dificuldade para convergir. Nos métodos Multigrid isso não acontece. Podemos constatar que aumentando a dimensão de n os métodos convergem em um tempo de CPU relativamente baixo e com poucas iterações.

Comparação Entre os Métodos Iterativo e Multigrid

Para observar as vantagens dos métodos Multigrid considere a seguinte tabela: escolhemos os problemas modelos (1.1), (1.5) e o (1.10). A escolha do RBSOR foi feita por ter apresentado bom desempenho nos experimentos computacionais e principalmente porque o red black é a ordenação que foi usada nas experiências numéricas com os métodos Multigrid.

O tipo de método Multigrid escolhido é o ciclo $V(2,2)$, por ter mostrado um bom desempenho nos experimentos numéricos.

PM I	n=129		Razão
	RBSOR	Mgrid	
Iter	804	7	115
CPU	7.760	0.379	20

Tabela 4.24: Problema I - Razão = $\frac{RBSOR}{Mgrid}$

As Tabelas 4.24, 4.25 e 4.26 mostram a superioridade dos métodos Multigrid. Pois para os problemas modelos (1.1), (1.5) e (1.10) o número de iterações dos métodos multigrid é muito menor que o número de iterações do método iterativo de melhor desempenho.

A comparação relevante, que mostra a superioridade dos métodos multigrid, é o tempo que cada método usa de processamento (CPU). Nas Tabela 4.24, 4.25 e 4.26, vemos que para o problema modelo (1.1), (1.5) e (1.10) os tempos de processamento são 20, 25 e 20 vezes menor que RBSOR respectivamente. Entretanto os métodos Multigrid são mais complexos para implementação.

PM V	n=129		Razão
	RBSOR	Mgrid	
Iter	690	5	138
CPU	6.799	0.269	25

Tabela 4.25: Problema V - Razão = $\frac{RBSOR}{Mgrid}$

PM X	n=129		Razão
	RBSOR	Mgrid	
Iter	756	7	108
CPU	7.450	0.379	20

Tabela 4.26: Problema X - Razão = $\frac{RBSOR}{Mgrid}$

Os resultados numéricos mostram que os métodos Multigrid apresentam desempenho muito superior aos métodos iterativos tradicionais para as equações usadas neste trabalho.

Observação 3 *As equações usadas nos resultados numéricos são da forma:*

$$-\epsilon\Delta u + au_x + bu_y = f, \quad (4.1)$$

sendo os parametros usados em cada problema modelo dado abaixo:

$$PM I - \quad \epsilon = -1, \quad a = 0, \quad b = 0$$

$$PM II - \quad \epsilon = -1, \quad a = 0, \quad b = 0$$

$$PM III - \quad \epsilon = -1, \quad a = 0, \quad b = 0$$

$$PM IV - \quad \epsilon = -1, \quad a = 0, \quad b = 0$$

$$PM V - \quad \epsilon = -1, \quad a = 0, \quad b = 0$$

$$PM VI - \quad \epsilon = -1, \quad a = 0, \quad b = 0$$

$$PM VII - \quad \epsilon = -1, \quad a = 1, \quad b = 0$$

$$PM VIII - \quad \epsilon = 1, \quad a = \frac{\sigma}{\epsilon}, \quad b = \frac{\tau}{\epsilon}, \quad \sigma = -1, \quad \tau = -1$$

$$PM IX - \quad \epsilon = 1, \quad a = \cos(\alpha), \quad b = \sin(\alpha), \quad \alpha = \frac{\pi}{4}$$

$$PM X - \quad \epsilon = 1, \quad a = \frac{p}{\epsilon}, \quad b = \frac{q}{\epsilon}, \quad p = 1, \quad q = 1$$

Capítulo 5

Conclusões e Futuras Investigações

Usando os resultados numéricos referentes aos métodos iterativos tradicionais, aqueles com relaxação sucessiva e ordenação red black foram os que mostraram melhores resultados.

No Capítulo 4 vimos a maior eficiência dos métodos Multigrid comparados aos métodos iterativos. Os resultados numéricos confirmam que devemos priorizar o uso dos métodos Multigrid.

Contudo, a nossa experiência com os métodos Multigrid mostrou maior dificuldade em implementar esse método. Enquanto que para os métodos iterativos os programas ocupam poucas linhas de comando, a programação do Multigrid é muito mais elaborada principalmente no que se refere aos operadores de restrição, interpolação, a correção em malha grosseira e ainda o método de relaxação usado pelos métodos Multigrid. Porém os resultados numéricos mostraram que é compensador o uso dos métodos Multigrid.

As dificuldades encontradas na programação foram muito instrutivas. De fato, isso nos possibilita, daqui em diante, a programação mais elaborada de outros métodos numéricos. Nesse mesmo contexto, essa experiência nos será valiosa para futuros trabalhos nos quais, provavelmente, iremos usar softwares “prontos”, agora com uma visão muito crítica possibilitando elaborar programas de interface para melhor aproveitar as potencialidades dos métodos Multigrid.

Futuras Investigações

Para os métodos iterativos devemos investigar mais intensivamente outros tipos de ordenações, como por exemplo as ordenações de quatro cores (four color), ordenações por linhas entre outras.

Para os métodos Multigrid, os tópicos que devem ser investigados para dar maior robustez aos métodos Multigrid são:

- Aumentar a ordem de aproximação do método de discretização aplicado na equação diferencial parcial.

- Investigar o *FMG* (“Full Multigrid”), tanto na solução de sistemas lineares provenientes das equações diferenciais parciais, como numa maneira de encontrar uma estimativa inicial para os métodos iterativos que resolvem equações diferenciais parciais não lineares.

- Acoplar o estudo do Multigrid com os métodos baseados nos subespaços de Krylov.

- Usar malhas não estruturadas e adaptativas, com domínio regular e irregular em conexão com o Multigrid.

- Estudar o método Multigrid (*FAS*), que serve para resolver equações diferenciais parciais não lineares.

- Estudar uma maneira de ter um controle ótimo dos ciclos *V*, *W* e *FMG*, acopladamente, para otimizar os métodos Multigrid.

- Investigar o uso do SOR como método de relaxação usado nos métodos Multigrid.

- Mostrar resultados teóricos sobre a convergência dos métodos Multigrid em equações diferenciais parciais elípticas, parabólicas ou hiperbólicas.

- Estudar a aplicação dos métodos Multigrid para resolver os sistemas lineares provenientes da discretização com elementos finitos.

- Estudar os métodos Multigrid em conexão com os métodos espectrais, e também com os métodos de decomposição de domínio.

O estudo dos métodos Multigrid é uma área bastante próspera da matemática aplicada, existindo questões não resolvidas, principalmente quando tratamos de investigar a convergência desses métodos aplicados à equações diferenciais parciais não elípticas.

Referências Bibliográficas

- [1] Bakhvalov, N.S, *On the convergence of relaxation method with natural constraints on the elliptic operator*. URSS Comp. Math. Phys. **1**(5)(1966), 101-135.
- [2] Bank, R.E, Wagner, C., *Multilevel ILU decomposition*. Numerische Mathematik, **82**(1999),543-576.
- [3] Bramble, J.H, *Multigrid methods*. Longman Scientific and Technical, New York, 1993.
- [4] Brandt, A., *Multi-Level adaptive solutions to boundary-value problems*. Math. Comput., **31**(1977), 333-390.
- [5] Briggs, W.L., *Multigrid tutorial*. SIAM, Philadelphia, 1987.
- [6] Demmel, J.W., *Applied Numerical Linear Algebra*. SIAM , Philadelphia, 1997.
- [7] Dancis, J., *The optimal ω is not best SOR iterative method*. Linear Algebra And Its Applications,**154-156**(1991), 819-845.
- [8] Eiermann, M., Varga, R.S, *Is the optimal ω best the sor iterative methods?* Linear Algebra And Its Applications, **182**(1993), 127-277.
- [9] Fedorenko, R.P, *A relaxation method for solving elliptic difference equation*. URSS Comp. Math. Phys. **1**(5)(1961), 1092-1096.
- [10] Fedorenko, R.P, *The speed of convergence of one iterative process*. URSS Comp. Math. Phys.**4**(3)(1964), 227-235.
- [11] Gupta M.M., Kouatchou J. and Zhang J., *A compact Multigrid solver for convection-difusion equations*. J. Comp. phys., **132** (1997), 123-129.
- [12] Gupta M.M., Kouatchou J. and Zhang J., *Comparision of second- and fourth-order discretizations for Multigrid Poisson solvers*. J. Comp. phys., **132**(1997), 226-232.
- [13] Hackbusch, W., *Convergence of multi-grid iterations applied to difference equations.*, math. Comp., **34**(1980), 425-440.

- [14] Hageman, A.L., Young, D.M., *Applied Iterative Methods.*, Academic Press, san Diego, 1981.
- [15] Jay Kuo, C.C., levy, B.C., *Two-color fourier analysis of the Multigrid method with red-black Gauss Seidel smoothing.* Appl. Math. Comp., **87**(1989), 29-69.
- [16] Karaa S., Zhang J., *Analysis of stationary iterative methods for discrete convection-diffusion equation with a 9-point compact scheme.* J. Comp. Appl.Math., **154**(2003), 447-476.
- [17] Nicolaides R.A, *On l^2 convergence of an algorithm for solving finite element equation.* Math. Comp, **31**(1977), 892-906.
- [18] Saad, Y., *Iterative methods for sparse linear systems.* PWS Publishing Company, Boston, 1995.
- [19] Trottenberg, U., Oosterlee, C., Schuller, A. *Multigrid.*, Academic Press, san Diego, 2001.
- [20] Udawadia, F.E., *Some Convergence results related to JOR iterative method for symmetric, positive-definite matrices.* Appl. Math. comp., **47**(1992), 37-45
- [21] Wesseling, P., *An Introduction to Multigrid Methods.* John Wiley, New York, 1992.
- [22] Woznicki, Z.I., *Estimation of the optimum relaxation factors in partial factorization iterative methods.* SIAM J. Matrix Anal. Appl., **14**(1993), 59-73.
- [23] Yavneh, I., Olvovsky E., *Multigrid smoothing for symmetric nine-point stencil.* Appl. Math. comp., **92**(1998), 229-246.
- [24] Young, D.M., *Iterative solution of large linear systems.* Academic Press, New York, 1971.
- [25] Zhang J., *Acceleration of five-point red-black Gauss Seidel for Poisson equation.* Appl. Math. Comp., **80**(1996), 73-93.
- [26] Zhang J., *Numerical simulation of 2D square driven cavity using fourth-order compact finite difference schemes.*, Comp. Math. Appl., **454**(2003), 43-52.