

Unesp - Universidade Estadual Paulista

Departamento de Matemática e Computação - DMC

PósMAC

Relatório 3º e 4º Trabalhos Práticos

Aluno: Gilmar Francisco de Oliveira Santos

Professora Responsável: Analice Costacurta Brandi

Julho

2020

Sumário

1	Resumo	1
2	Equações Diferenciais Parciais	2
2.1	Ordem e grau de uma Equação Diferencial Parcial	2
2.2	Equações Elípticas	3
2.2.1	Equações Elípticas com Contorno de Dirichlet	4
2.2.2	Condição de Fronteira de Neumann: Domínio Retangular	8
3	Métodos Iterativos para Resolução de Sistemas de Equações	9
3.1	Método de Jacobi	11
3.2	Método de Gauss-Seidel	11
3.3	Método de SOR (Sucessive Over Relaxation Method)	13
3.3.1	Escolha do ω ótimo	13
3.4	Convergência de Métodos Iterativos	15
4	Problemas Práticos	19
4.1	Problema 1	19
4.2	Problema 2	40
5	Conclusão	47
	Bibliografia	48

1 Resumo

Trabalho prático visando o exercício de métodos numéricos para solução de Equações Diferenciais Parciais(EDPs) do tipo Elípticas.

A análise numérica e computacional é uma área da Matemática na qual se encontram vários e diversificados métodos que se constituem em ferramentas para se determinar, exata ou aproximadamente, soluções numéricas de problemas modelados matematicamente. Em um campo de aplicação mais amplo, a busca de soluções aproximadas tem sido muito importante na resolução destes problemas quando não existe a possibilidade de resolvê-los por técnicas exatas.

Neste trabalho a linguagem Python3 é utilizada na implementação dos métodos numéricos. Python é uma linguagem de programação interpretada, orientada à objeto, de alto nível e de semântica dinâmica. Possui tipagem dinâmica, é rápida, e sua simplicidade a torna atraente e produtiva.

Possui uma infinidade de bibliotecas e hoje é uma das linguagens mais utilizadas no mundo. Para computação científica Python possui ótimas bibliotecas, as utilizadas neste trabalho serão: NumPy(processamento rápido de matrizes) e Matplotlib(plotagem de gráficos).

2 Equações Diferenciais Parciais

Uma equação diferencial parcial (EDP) é uma equação envolvendo duas ou mais variáveis independentes x, y, z, t, \dots e derivadas parciais de uma função (variável dependente) $u = u(x, y, z, t, \dots)$. De maneira mais precisa, uma EDP em n variáveis independentes é uma equação que apresenta o seguinte formato:

$$F(x_1, \dots, x_n, u, \frac{\partial u}{\partial x_1}, \frac{\partial^2 u}{\partial x_1^2}, \dots, \frac{\partial^2 u}{\partial x_1 x_n}, \dots, \frac{\partial^n u}{\partial x^n}) = 0 \quad (1)$$

onde $(x_1, \dots, x_n) \in \Omega$, sendo Ω um subconjunto aberto de R^n , F é uma função dada e $u(x)$ é a função que se quer determinar.

2.1 Ordem e grau de uma Equação Diferencial Parcial

A ordem de uma equação diferencial parcial é a ordem da mais alta derivada que ocorre na equação e o grau é o expoente da derivada mais alta quando a equação está escrita em uma forma semelhante a uma função polinomial em que as potências fazem o papel das derivadas da ordem respectiva.

Uma diferença importante entre EDOs e EDPs é a informação suplementar necessária para a unicidade de solução. Por exemplo, na solução de uma EDO linear, tem-se uma ou mais constantes arbitrárias: podemos determinar estas constantes impondo condições iniciais, isto é, fixando os valores da solução e de suas derivadas até certa ordem em um determinado ponto; podemos também obter unicidade no caso de intervalos finitos, impondo condições nos extremos, as chamadas condições de contorno. A situação para as EDPs é fundamentalmente diferente: mesmo no caso linear, a solução geral (quando é possível determiná-la), envolve funções arbitrárias das variáveis independentes, de modo que existe um grau de generalidade muito maior com relação à forma da solução. O espaço das variáveis independentes é, neste caso, multidimensional: procuramos soluções definidas em um aberto $\Omega \subseteq R^n$.

Exemplos de Equações Diferenciais Parciais

1. Equação do Calor: $\frac{\partial u}{\partial t} = \alpha^2 \frac{\partial^2 u}{\partial x^2}$
2. Equação do Calor: $\frac{\partial u}{\partial t} = \alpha^2 (\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})$
3. Equação da Onda: $\frac{\partial^2 u}{\partial t^2} = \alpha^2 \frac{\partial^2 u}{\partial x^2}$
4. Equação da Onda: $\frac{\partial^2 u}{\partial t^2} = \alpha^2 (\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})$

5. Equação de Laplace : $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$
6. Equação de Laplace : $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 0$
7. $\frac{\partial u}{\partial x} = x + y$
8. $\frac{\partial^3 u}{\partial x^3} + 2y \frac{\partial^2 u}{\partial x^2} + x \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} + \left(\frac{\partial u}{\partial x}\right)^2 = \text{sen}(xy)$

No exemplo anterior, as equações dos itens 1, 2, 3, 4 e 5 são de segunda ordem, a do item 6 é de primeira ordem e a do item 7 é de terceira ordem.

2.2 Equações Elípticas

Representam problemas de equilíbrio que não dependem, em geral, do tempo. Exemplos: problemas de vibração de membranas, problemas de difusão, etc. Problemas de valor no contorno, em duas ou três dimensões, são geralmente descritas por equações elípticas. A discretização de uma equação estacionária fornece um sistema de equações que pode ser resolvido por meio de métodos diretos ou iterativos.

Os conceitos de métodos explícitos e implícitos no tempo não são aplicáveis às equações elípticas devido à ausência da integração temporal. Sempre é necessário resolver um sistema de equações lineares e, como a matriz de coeficientes do sistema não é, em geral, tridiagonal, a solução é obtida por métodos iterativos.

Se R uma região do plano e ∂R a sua fronteira. A equação

$$a(x, y) \frac{\partial^2 u}{\partial x \partial y} + 2b(x, y) \frac{\partial^2 u}{\partial x \partial y} + c(x, y) \frac{\partial^2 u}{\partial y^2} = d(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}) \quad (2)$$

é elíptica se $b^2 - 4ac < 0, \forall (x, y) \in R$. São muitos os problemas práticos que são governados pela equação (1). Com $a = c = -1$ e $b = 0$. Se $d = 0$, temos a equação de Laplace:

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 0 \quad (3)$$

Se $d = f(x, y)$, então temos a equação de Poisson:

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x, y) \quad (4)$$

Como (2) é uma equação linear, se u e v são soluções de (2) então $w = \alpha u + \beta v$ também será solução de (2), isso implica que a equação (2) tem infinitas soluções.

Para que (2) tenha solução única é preciso especificar condições de contorno de u de tal forma que a equação (2) tenha solução única (as condições de contorno não podem ser arbitrárias).

Em geral, a equação (2) é resolvida sujeita a 3 tipos de condições de contorno: Dirichlet, Neumann e Robbin (ou mista).

Para resolver (2) vamos utilizar o método de diferenças finitas que consiste em substituir as derivadas parciais em (2) por aproximações por diferenças finitas.

2.2.1 Equações Elípticas com Contorno de Dirichlet

Para ilustrar o método numérico, vamos considerar a equação de Poisson,

$$-\Delta u = -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f(x, y) \quad (5)$$

definido num retângulo $R = \{0 < x < a, 0 < y < b\}$, com condição de Dirichlet: $u(x, y) = g(x, y)$ na fronteira ∂R .

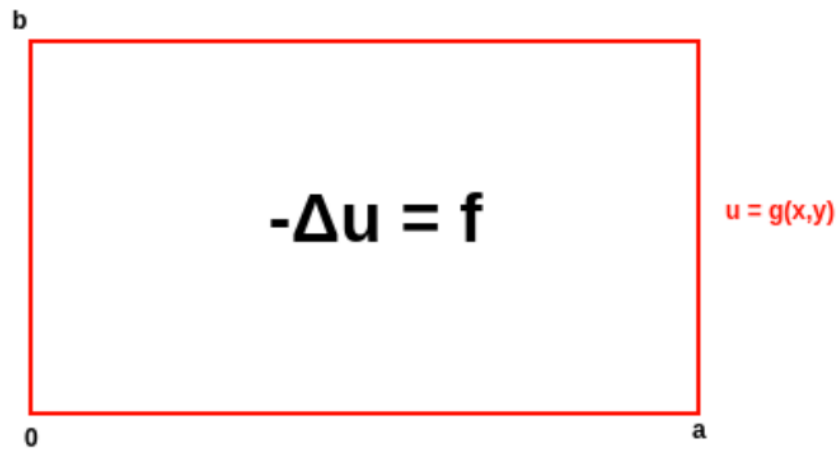


Figura 1: Região do problema

Definimos uma malha:

$$\begin{aligned} x_i &= ih, \quad i = 0, 1, \dots, N. \quad h = \frac{a}{N}. \\ y_j &= jk, \quad j = 0, 1, \dots, M. \quad k = \frac{b}{M}. \end{aligned}$$

Considere a seguinte malha discretizada

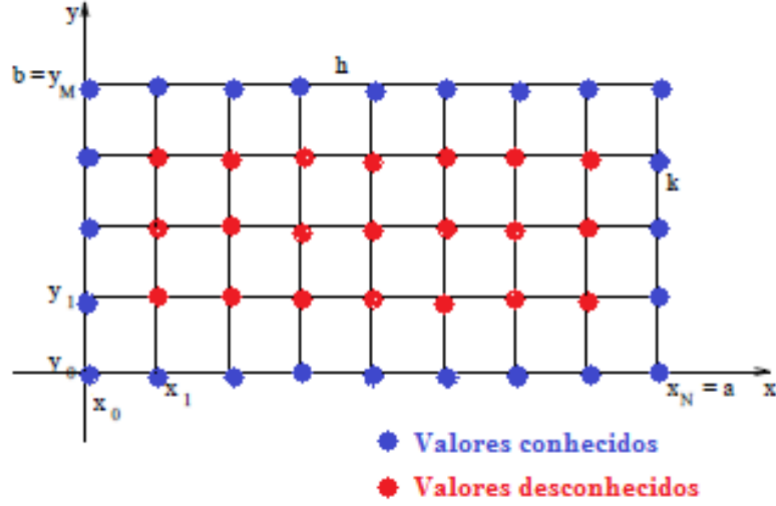


Figura 2: Geometria discretizada

Dessa forma, pode-se descrever os pontos no interior da malha e os pontos da fronteira da seguinte forma

$$R_\delta = \{(x_i, y_j) | x_i = ih, y_j = jk, 1 \leq i \leq N-1, 1 \leq j \leq M-1\}$$

$$R_\delta = \{(x_i, y_0), (x_i, y_M), \quad i = 0, 1, \dots, N, \\ (x_0, y_j), (x_N, y_j), \quad j = 0, 1, \dots, M\}.$$

A equação (5) é válida para qualquer ponto de R_δ , ou seja,

$$-\left(\frac{\partial^2 u}{\partial x^2}(x_i, y_j) + \frac{\partial^2 u}{\partial y^2}(x_i, y_j)\right) = f(x_i, y_j) \quad (6)$$

para $i = 1, 2, \dots, N-1$ e $j = 1, 2, \dots, M-1$. As aproximações para as derivadas por diferenças finitas são da- das por

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u(x_i + h, y_j) - 2u(x_i, y_j) + u(x_i - h, y_j)}{h^2} \quad (7)$$

$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) = \frac{u(x_i, y_j + k) - 2u(x_i, y_j) + u(x_i, y_j - k)}{k^2} \quad (8)$$

Substituindo as equações (7) e (8) em (6), obtemos

$$-\left(\frac{u(x_i + h, y_j) - 2u(x_i, y_j) + u(x_i - h, y_j)}{h^2} + \frac{u(x_i, y_j + k) - 2u(x_i, y_j) + u(x_i, y_j - k)}{k^2}\right) \approx f(x_i, y_j) \quad (9)$$

Observamos que a equação (9) não representa uma equação porque o primeiro membro é uma aproximação para o segundo membro. Para transformarmos (9) em uma equação trocamos o sinal de aproximado pelo sinal de igualdade e denotamos o valor de $u(x_i, y_j)$ por $u_{i,j}$ e a solução aproximada por $U_{i,j}$. Neste caso, obtemos a equação de diferenças:

$$-\left(\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{k^2}\right) = f(x_i, y_j) \quad (10)$$

Para pontos adjacentes à fronteira, a equação (10) envolverá pontos na fronteira que são obtidos diretamente da condição de fronteira de Dirichlet, ou seja, $U_{i,j} = g(x_i, y_j), \forall (x_i, y_j) \in \partial R_\delta$.

Para simplificar a notação, vamos introduzir o operador

$$-\Delta_\delta U_{i,j} = -\left(\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{k^2}\right) \quad (11)$$

Com essa notação, a equação (10) juntamente com a condição de contorno, pode ser escrita como:

$$-\Delta_\delta U_{i,j} = f(x_i, y_j), (x_i, y_j) \in R_\delta \quad (12)$$

$$U_{i,j} = g(x_i, y_j), (x_i, y_j) \in \partial R_\delta \quad (13)$$

A equação (12) quando aplicada nos pontos interiores R_δ , resulta num sistema linear simétrico com $(N-1) * (M-1)$ equações. Espera-se que resolvendo esse sistema linear, os valores $U_{i,j}$ encontrados seja uma boa aproximação para $u_{i,j}$. De fato, veremos que a solução obtida por (12) converge para $u_{i,j}$ se a malha for refinada.

O sistema linear proveniente (12) matricial $AU = r$, tal que:

$$\underbrace{\begin{bmatrix} a & b & & & c \\ b & a & b & & & \ddots & 0 \\ & \ddots & \ddots & \ddots & & 0 & \ddots \\ & & \ddots & \ddots & \ddots & & c \\ c & & & \ddots & \ddots & \ddots & \\ & \ddots & 0 & \ddots & \ddots & \ddots & \\ 0 & \ddots & & & b & a & b \\ & & c & & a & b \end{bmatrix}}_A \underbrace{\begin{bmatrix} U_{1,1} \\ \vdots \\ U_{N-1,1} \\ U_{1,2} \\ U_{2,2} \\ \vdots \\ U_{N-1,2} \\ \vdots \\ U_{N-1,M-1} \end{bmatrix}}_U = \underbrace{\begin{bmatrix} f_{1,1} + \frac{g(x_0,y_1)}{h^2} + \frac{g(x_1,y_0)}{k^2} \\ f_{2,1} + \frac{g(x_2,y_0)}{h^2} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_{N-1,1} + \frac{g(x_N,y_1)}{h^2} + \frac{g(x_{N-1},y_0)}{k^2} \end{bmatrix}}_r$$

onde $a = \frac{2}{h^2} + \frac{2}{k^2}$, $b = \frac{-1}{h^2}$ e $c = \frac{-1}{k^2}$.

Erro na Aproximação

Quando aproximamos a equação de Poisson pela equação (10), cometemos um erro de truncamento local

$$\tau_{i,j} = -\Delta_\delta u_{i,j} - f(x_i, y_j) \quad (14)$$

Lema

e a solução $u(x, y)$ é diferenciável até ordem 4 com derivada limitada então o erro de truncamento local satisfaz:

$$|\tau_{i,j}| \leq c_1 h^2 + c_2 k^2$$

onde c_1 e c_2 são constantes independentes de h e k .

Erro Global: O erro global, denotado por $e_{i,j}$ é definido por

$$e_{i,j} = u_{i,j} - U_{i,j}$$

onde $u_{i,j}$ denota a solução exata e $U_{i,j}$ a solução aproximada.

Teorema 1 (Princípio do Máximo)

i. Se $V(x, y)$ é uma função discreta definida em $R_\delta \cup \partial R_\delta$ e satisfaz

$$\Delta_\delta(x, y) \geq 0, \forall (x, y) \in R_\delta, \text{ então}$$

$$\max_{(x,y) \in R_\delta} V(x, y) \leq \max_{(x,y) \in \partial R_\delta} V(x, y)$$

ii. Alternativamente, se $V(x, y)$ satisfaz $\Delta_\delta(x, y) \leq 0, \forall (x, y) \in R_\delta$ então

$$\min_{(x,y) \in R_\delta} V(x, y) \geq \min_{(x,y) \in \partial R_\delta} V(x, y)$$

Teorema 2 Seja $V(x, y)$ uma função discreta definida em R_δ e ∂R_δ . Então

$$\max_{(x,y) \in R_\delta} |V(x, y)| \leq \max_{(x,y) \in \partial R_\delta} |V(x, y)| + \frac{a^2}{2} \max_{(x,y) \in R_\delta} |\Delta_\delta V(x, y)|$$

para $a = \frac{2}{h^2} + \frac{2}{k^2}$

Obtenção da estimativa para o erro global: — Fazendo $V(x, y) = e(x, y)$ no Teorema 2, temos

$$\pm e_{i,j} \leq \max_{\partial R_\delta} |e_{i,j}| + \frac{a^2}{2} \max_{R_\delta} |\Delta_\delta e_{i,j}|$$

Porém como $U_{i,j} = u_{i,j} - g_{i,j}$ em ∂R_δ , temos que $e_{i,j} = 0$ em ∂R_δ . Logo,

$$e_{i,j} \leq \frac{a^2}{2} \max_{R_\delta} |\Delta_\delta e_{i,j}|$$

Agora $\Delta_\delta e_{i,j} = \Delta_\delta(U_{i,j} - u_{i,j}) = \Delta_\delta U_{i,j} - \Delta_\delta u_{i,j} = \Delta_\delta U_{i,j} - f_{i,j} = \tau_{i,j}$ Logo, pelo Lema anterior,

$$e_{i,j} \leq \frac{a^2}{2} \max_{R_\delta} |\tau_{i,j}| \leq \frac{a^2}{2} (c_1 h^2 + c_2 k^2)$$

Isso mostra que, quando $h, k \rightarrow 0 \Rightarrow |e_{i,j}| \rightarrow 0$, isso implica que o método converge para a solução exata. Como consequência do Teorema 2, pode-se mostrar que o sistema de equações $AU = B$ tem solução única (A é não singular).

2.2.2 Condição de Fronteira de Neumann: Domínio Retangular

Quando usamos a condição de Dirichlet, o valor de u é conhecido na fronteira. A condição de Neumann envolve o valor da derivada de u e, portanto, é necessário determinar u nos pontos da fronteira. Para isso, aplica-se a equação de Poisson na fronteira.

Considerando o ponto C da figura anterior, obtemos a equação de Poisson discreta,

$$-\left[\frac{U_O - 2U_C + U_L}{h^2} + \frac{U_N - 2U_C + U_S}{k^2} \right] = f_C \quad (15)$$

O ponto fictício U_O é obtido em função dos pontos interiores pela aplicação da condição de Neumann

$$\frac{d^2 u}{dx^2}[C] = f_1 y_C \Rightarrow \frac{U_L - U_O}{2h} = f_1(y_C) \Rightarrow U_O = U_L - Sh f_1(y_C) \quad (16)$$

Substituindo a equação (16) na equação (15), obtemos:

$$\frac{2U_L - 2U_C}{h^2} + \frac{U_N - 2U_C + U_S}{k^2} = -f_C + \frac{2}{h} f_1(y_C) \quad (17)$$

Observando a equação (17), vemos que o efeito da condição de Neumann na equação discretizada é a modificação do termo independente e de alguns elementos da matriz. As modificações da matriz são as mais importantes visto que essas modificações podem destruir propriedades importantes da matriz tais como simetria e dominância da diagonal.

3 Métodos Iterativos para Resolução de Sistemas de Equações

Muitos sistemas lineares são demasiadamente grandes para serem resolvidos por métodos diretos. Por outro lado, em geral, A é uma matriz esparsa e a aplicação de métodos diretos ao sistema linear $Ax = b$ gera “Fill-in” (processo que transforma elementos nulos em não nulos).

Um método numérico começa com uma aproximação inicial que é sucessivamente modificada de acordo com algum processo. Para ser útil, a iteração deve convergir e a convergência deve ser rápida.

Para resolvermos o sistema não singular:

$$Au = v, \quad u = A^{-1}v$$

Iterativamente, precisamos que a sequência $u^{(k)}$ convirja para $A^{-1}v$, ou seja, $u^{(k)} \rightarrow A^{-1}v$ quando $k \rightarrow \infty$. Se $u^{(k)}$ é uma função de $A, u, u^{(k-1)}, \dots, u^{(k-g)}$, dizemos que g é o grau de iteração. Usualmente, fazemos $g = 1, 2, 3, \dots$.

Logo, com $g = 1$, temos:

$$u^{(k)} = F_k(A, v, u^{(k-1)})$$

Se F_k é independente de k , o método é chamado “estacionário” e, se F_k for linear em relação a $u^{(k-1)}$ então o método é linear.

$$u^{(k)} = G_k u^{(k-1) + r_k} \quad (18)$$

onde G_k é uma matriz que depende de A e r_k um vetor. Para que o método seja constante devemos ter:

$$u^{(k-1)} = A^{-1}v, \quad u^{(k)} = A^{-1}v \quad (19)$$

Substituindo as expressões (19) na equação (18), obtemos:

$$A^{-1}v = G_k A^{-1}v + r_k, \quad (20)$$

$$IA^{-1}v - G_k A^{-1}v = r_k, \quad (21)$$

$$r_k = (I - G_k)A^{-1}v \quad (22)$$

em que a equação (22) é a Condição de Consistência.

Fazendo $M_k = (I - G_k)A^{-1}$ odemos escrever o método iterativo (18) como:

$$u^{(k)} = G_k u^{(k-1)} + M_k v \quad (23)$$

Podemos analisar a convergência do método iterativo (23), examinando o erro e_k dado por:

$$\begin{aligned} e_k &= u^{(k)} - A^{-1}v = G_k u^{(k-1)} + M_k v - A^{-1}v \\ &= G_k u^{(k-1)} + M_k v - (G_k A^{-1}v + (I - G_k)A^{-1}v), \\ &= G_k u^{(k-1)} + M_k v - (G_k A^{-1}v - G_k A^{-1}v) - M_k v, \\ &= G_k u^{(k-1)} - (G_k A^{-1}v - G_k A^{-1}v), \\ &= G_k (u^{(k-1)} - A^{-1}v), \\ &= G_k e_{k-1} \end{aligned} \quad (24)$$

Aplicando a equação (24) repetidamente, temos:

$$\begin{aligned} e_1 &= G_1 e_0 \\ e_2 &= G_2 e_1 = G_2 G_1 e_0 \\ &\vdots \\ e_k &= G_k e_{k-1} = G_k G_{k-1} \dots G_1 e_0 \end{aligned}$$

Fazendo $H_k = G_k G_{k-1} \dots G_1$, podemos escrever:

$$e_k = H_k e_0$$

Logo, a convergência do método, para um erro inicial e_0 , dependendo da condição

$$H_k e_0 = G_k G_{k-1} \dots G_1 e_0 \rightarrow 0, \text{ quando } k \rightarrow \infty$$

Como o método é estacionário e linear

$$G + k = G_{k-1} = \dots = G_2 = G_1 = G$$

Portanto, $H_k = G^k$ e a convergência do método depende da análise de $G^k e_0$.

Para obter métodos Iterativos separamos A em três matrizes, onde $A = L + D + S$, sendo que L é uma matriz estritamente triangular inferior, D é diagonal e S é estritamente triangular superior.

3.1 Método de Jacobi

É muito utilizado pela sua simplicidade de implementação, portanto sua convergência pode ser lenta comparada com outros métodos. Assumimos $a_{ii} \neq 0, \forall i$.

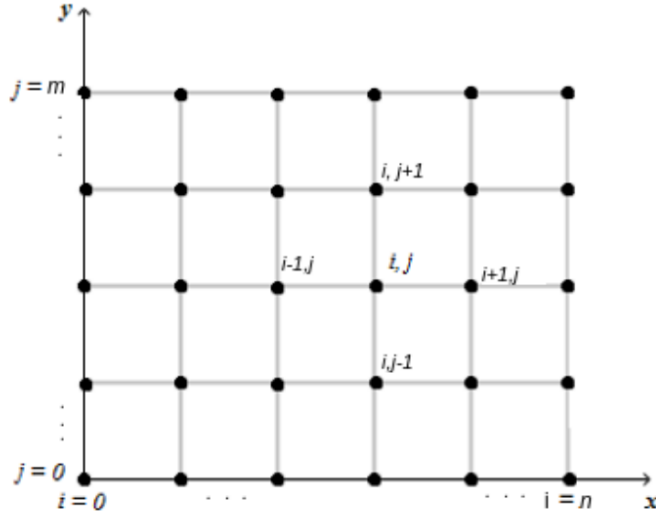


Figura 3: Pontos usados no método de Jacobi

$$\begin{aligned} Au = v &\Leftrightarrow (L + D + S)u = v \Leftrightarrow Du = -(L + S)u + v \Leftrightarrow \\ &\Leftrightarrow u = -D^{-1}(L + S)u + D^{-1}v \end{aligned}$$

Logo, dado $u^{(0)}$, o método de Jacobi consiste em calcular aproximações $u^{(k)}$ por meio da equação:

$$u^{(k)} = -D^{-1}(L + S)u^{(k-1)} + D^{-1}v \quad (25)$$

a equação (25) consiste no método de Jacobi em sua forma matricial. Note que esse é um método iterativo análogo ao definido pela equação (23), onde $G_k = -D^{-1}(L + S)$ e $M_k = D^{-1}$. A matriz de iteração do método de Jacobi é dada por:

$$G_J = -D^{-1}(L + S)$$

3.2 Método de Gauss-Seidel

Considere o método de Jacobi aplicado ao sistema $Au = v$, ou seja, o calculado pela equação (4.7) e suponhamos que a sequência de vetores $u^{(k)}$ seja convergente. Então,

podemos observar o seguinte na forma vetorial:

$$\begin{aligned} G_J &= -D^{-1}(L + S) = -D^{-1}L - D^{-1}S, \\ Iu^{(k)} &= -D^{-1}Lu^{(k)} - D^{-1}Su^{(k-1)} + t_i, \\ (I + D^{-1}L)u^{(k)} &= -D^{-1}Su^{(k-1)} + t_i, \end{aligned}$$

onde G_J é a matriz de iteração do método de Jacobi. Agora, multiplicando por D , temos

$$\begin{aligned} (D_L)u^{(k)} &= -Su^{(k-1)} + Dt_i, \\ u^{(k)} &= -(L + D)^{-1}Su^{(k-1)} + (L + D)^{-1}Dt_i, \end{aligned}$$

Logo, o método de Gauss-Seidel é um método iterativo estacionário linear que tem como matriz de iteração $G_S = -(L + D)^{-1}S$. A matriz $(L + D)^{-1}$ existe sempre que A tenha elementos diagonais não nulos.

3.3 Método de SOR (Sucessive Over Relaxation Method)

Os métodos de Jacobi e Gauss-Seidel são atrativos, porque são fáceis de implementar computacionalmente. Porém, a convergência desses métodos depende diretamente do raio espectral $\rho(G)$, da matriz de iteração G desses métodos. Se $\rho(G) \approx 1$, a convergência desses métodos torna-se muito lenta. O método SOR consiste em uma modificação do método de Gauss-Seidel de modo a minimizar o valor do raio espectral G . Existem três teoremas que ajudam a determinar qual valor de ω permite obter uma convergência ótima do sistema.

Em notação vetorial, podemos escrever

$$\begin{aligned} Iu^{(k)} &= (1 - \omega)Iu^{(k-1)} + \omega [-D^{-1}Lu^{(k)} - D^{-1}Su^{(k-1)} + D^{-1}v] \\ (I + D^{-1}L)u^{(k)} &= [(1 - \omega)I - \omega D^{-1}S] u^{(k-1)} + \omega D^{-1}v \end{aligned}$$

Multiplicando por D , obtemos:

$$\begin{aligned} (D + \omega L)u^{(k)} &= [(1 - \omega)D - \omega S] u^{(k-1)} + \omega v \\ u^{(k)} &= (D + \omega L)^{-1} [(1 - \omega)D - \omega S] u^{(k-1)} + \omega(D + \omega L)^{-1}v \end{aligned}$$

que é a forma vetorial do método. Logo, a matriz de iteração é dada por:

$$G_{SOR} = (D + \omega L)^{-1} [(1 - \omega)D - \omega S]$$

Se fizermos $\omega = 1$ em (4.14), então a matriz G_{SOR} reduz-se a matriz do método de Gauss-Seidel. Espera-se que com uma escolha apropriada de ω , torne o método mais rápido, ou seja, espera-se que $\rho(G_{SOR}) \leq \rho(G_S)$. Porém, a escolha do parâmetro ω não é tão simples e a sua escolha é feita por experimentação ou é feita empiricamente baseando-se em propriedades do problema a ser resolvido.

3.3.1 Escolha do ω ótimo

Onde o termo ω , parâmetro de relaxamento, pode acelerar a convergência para a solução do sistema, sendo que ω pode ser calculado como segue. Existem três teoremas que ajudam a determinar qual valor de ω permite obter uma convergência ótima do sistema: **Teorema 1 (Kahan - condição necessária)** - Se $a_{ii} \neq 0$ para todos os valores de i , então $\rho(T\omega) \geq |\omega - 1|$. Isso implica que o Método SOR somente converge para $0 < \omega < 2$.

Teorema 2 (Ostrowski-Reich - condição suficiente) - Se A é uma matriz positiva definida e $0 < \omega < 2$, então o Método SOR converge para qualquer aproximação inicial de x_i .

Teorema 3 - Se A é uma matriz positiva definida e tridiagonal, então $\rho(T_g) = [\rho(T_j)]^2 < 1$ e a opção ótima para ω é dada por $\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}}$. Onde $T_j = D^{-1}(L + S)$. Obtemos então $\rho(T_j)$ por $\det(T_j - \lambda I) = 0$.

Se a matriz A não for simétrica devemos então calcular ω_{otimo} como segue:

$$\omega_{otimo} = 1 + \left(\frac{\rho(T_j)}{1 + \sqrt{1 - \rho(T_j)^2}} \right)^2$$

Exemplo:

Vamos analisar o seguinte sistema de equações:

$$9x_1 + 4x_2 = 20$$

$$4x_1 + 9x_2 - x_3 = 12$$

$$-x_2 + 9x_3 = 51$$

Que pode ser expresso na forma matricial

$$\begin{bmatrix} 9 & 4 & 0 \\ 4 & 9 & -1 \\ 0 & -1 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 20 \\ 12 \\ 51 \end{bmatrix}$$

Devemos obter $\rho(T_j)$ para obter ω . Então calculamos $T_j = D^{-1}(L + S)$

$$T_j = \begin{bmatrix} \frac{1}{9} & 0 & 0 \\ 0 & \frac{1}{9} & 0 \\ 0 & 0 & \frac{1}{9} \end{bmatrix} \left(\begin{bmatrix} 0 & 0 & 0 \\ -4 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -4 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \right)$$

Então ficamos com:

$$T_j = \begin{bmatrix} \frac{1}{9} & 0 & 0 \\ 0 & \frac{1}{9} & 0 \\ 0 & 0 & \frac{1}{9} \end{bmatrix} \begin{bmatrix} 0 & -4 & 0 \\ -4 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \frac{-4}{9} & 0 \\ \frac{-4}{9} & 0 & \frac{1}{9} \\ 0 & \frac{1}{9} & 0 \end{bmatrix}$$

Podemos obter então $\rho(T_j)$ por $\det(T_j - \lambda I) = 0$.

$$\det(T_j - \lambda I)_j = \det \left(\begin{bmatrix} -\lambda & \frac{-4}{9} & 0 \\ \frac{-4}{9} & -\lambda & \frac{1}{9} \\ 0 & \frac{1}{9} & -\lambda \end{bmatrix} \right) = \lambda^3 + \frac{17}{81}\lambda = -\lambda(\lambda^2 - \frac{17}{81} = 0)$$

Temos então que $\rho(T_j) = \sqrt{\frac{17}{81}}$ que pode ser aplicado diretamente na equação:

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}} = \frac{2}{1 + \sqrt{1 - \left[\sqrt{\frac{17}{81}}\right]^2}} = 1.058823529$$

Aplicando esse parâmetro no método SOR temos os seguintes resultados para uma tolerância de 10^{-10} :

Método	Iterações
Jacobi	31
Gauss-Seidel	18
SOR	12

Na figura a seguir temos uma comparação do número de iterações para diferentes valores de ω no intervalo $0 < \omega < 2$:

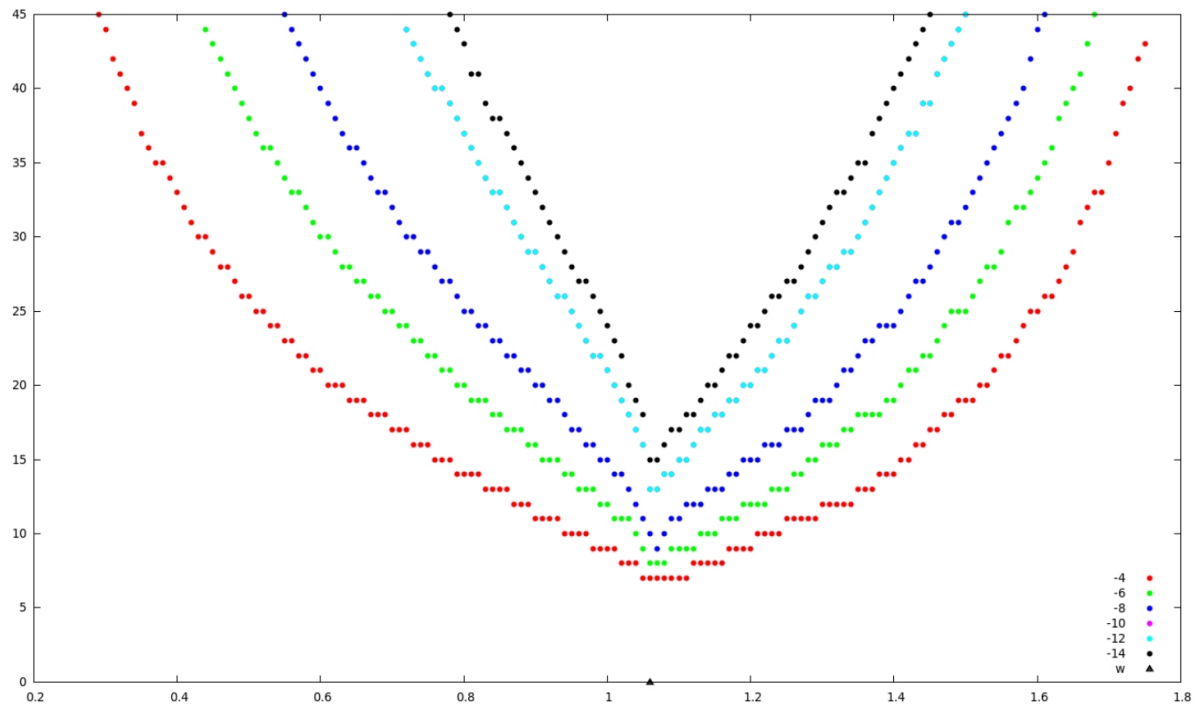


Figura 4: Número de iterações k para valores do parâmetro ω em diferentes ordens de grandeza da tolerância

3.4 Convergência de Métodos Iterativos

Seja A uma matriz complexa $n \times n$. O raio espectral de A , denotado por $\rho(A)$, é definido por $\rho(A) = \max |\lambda_i|$, tal que λ_i é o autovalor de A . A norma espectral de uma

matriz A é dada por

$$\|A\| = [\rho(A^*A)]^{\frac{1}{2}}$$

onde A^* é a transposta conjugada. O raio espectral e a norma espectral estão relacionados da seguinte forma:

1. Para qualquer matriz A tem-se $\|A\| \geq \rho(A)$
2. Se A for hermitiana (isto é, $A^* = A$), então $\|A\| = \rho(A)$
3. Se $\rho(A) < 1$ e n for grande, então $\|A^n\| \approx \rho(A)^n$

Definição

Uma matriz $A_{n \times n}$ é uma matriz convergente, se

$$\lim_{n \rightarrow \infty} A^n = 0$$

onde 0 é a matriz nula.

Teorema

A é uma matriz convergente se, e somente se, $\rho(A) < 1$.

Consideremos o método iterativo linear estacionário:

$$u^k = G_k u^{(k-1)} + r$$

A solução u desse problema satisfaz $u = Gu + r$. Pode-se mostrar que o erro $e_k = u - u^{(k)}$ satisfaz a equação

$$e_k = G^k e_0 \tag{26}$$

Disso, vemos que, se G for uma matriz convergente, isto é, $\rho(A) < 1$, então $\lim_{k \rightarrow \infty} G^k = 0$, então $e_k \rightarrow 0$ quando $k \rightarrow \infty$, então o método iterativo é convergente.

Por outro lado, seja $\|\cdot\|$ uma norma de vetor e consideremos a norma de matriz associada a norma do vetor. Então, por (26) temos $\|e_k\| = \|G^k e_0\| \leq \|G^k\| \cdot \|e_0\|$. Logo

$$\|e_k\| \leq \|G^k\| \cdot \|e_0\| \Rightarrow \frac{\|e_k\|}{\|e_0\|} \leq \|G^k\| \tag{27}$$

Como $\|G^k\|$ é uma norma consistente de matrizes, podemos escrever

$$\begin{aligned} \|G^k\| &= \|G^k \cdot G^{k-1}\| \leq \|G\| \cdot \|G^{k-1}\| \\ &\leq \|G\| \cdot \|G\| \cdot \|G^{k-2}\| \\ &\vdots \\ \|G^k\| &\leq \|G\|^k \end{aligned} \tag{28}$$

Substituindo (28) em (27), obtemos

$$\|e_k\| \leq \|G^k\| \cdot \|e_0\| \Rightarrow \frac{\|e_k\|}{\|e_0\|} \leq \|G\|^k \quad (29)$$

De (29) vemos que se $\|G\| < 1$ então $\|G\|^k \rightarrow 0$, quando $k \rightarrow \infty \Rightarrow \|G\|^k \|e_0\| \rightarrow 0 \rightarrow \|e_k\| \rightarrow 0$, quando $k \rightarrow \infty$. E isso significa que o método é convergente.

Logo, temos que se $\|G\| < 1$ então o método iterativo $u^{(k)} = Gu^{(k-1)} + r$ converge para $u \in R^n$.

Estimativa de erro na iterada $u^{(k)}$

Da equação $e_k = Ge_{k-1}$, temos

$$e_k = u - u^{(k)} = G(u - u^{(k-1)}),$$

$$G(u - u^{(k-1)} + u^{(k)} - u^{(k-1)}) = G(u - u^{(k)}) + G(u^{(k)} - u^{(k-1)}).$$

Tomando uma norma do vetor, vem

$$\|e_k\| = \|Ge_k + G(u^{(k)} - u^{(k-1)})\|,$$

$$\|e_k\| \leq \|Ge_k\| + \|G(u^{(k)} - u^{(k-1)})\|,$$

$$\|e_k\| \leq \|G\| \cdot \|e_k\| + \|G(u^{(k)} - u^{(k-1)})\|,$$

$$\|e_k\|(1 - \|G\|) \leq \|G\| \cdot \|u^{(k)} - u^{(k-1)}\|,$$

Estimativa para $\rho(G)$: Suponha k grande ($k \gg 1$). Temos que

$$\|e_k\| \leq \|G^k\| \cdot \|e_0\| \approx [\rho(G)]^k \|e_0\|$$

$$\therefore \|e_k\| \leq [\rho(G)]^k \|e_0\|$$

e $\|e_{k+1}\| \leq [\rho(G)]^{k+1} \|e_0\|$. Logo,

$$\frac{\|e_{k+1}\|}{\|e_k\|} \leq \rho(G) \quad (30)$$

A equação (30) mostra que para valores grandes de k , o fator $\frac{\|e_{k+1}\|}{\|e_k\|}$ se aproxima de $\rho(G)$. Isso mostra que o erro decresce de um fator $\rho(G)$ a cada iteração, isso implica que quanto menor for $\rho(G)$, mais rápida é a convergência.

Definição

A taxa de convergência do método iterativo $u^{(k)} = Gu^{(k-1)} + r$ é dado por

$$R(G) = -\log_{10} \rho(G) \quad (31)$$

Por (31), temos

$$\lambda = \frac{\|e_k\|}{\|e_0\|} \leq [\rho(G)]^k$$

Tomando o logaritmo, vem

$$\log(\lambda) \leq k \log[\rho(G)] \Rightarrow -\log(\lambda) \geq -k \log[\rho(G)] \Rightarrow k \leq \frac{\log(\lambda)}{\log[\rho(G)]}$$

Em geral, tomamos

$$k \approx \frac{\log(\lambda)}{R(G)}$$

Como $\|e_k\| = \lambda \|e_0\|$, λ pode ser visto como o fator de redução do erro inicial $\|e_0\|$, obtido após k -iterações. Logo, a anterior pode ser interpretada como uma estimativa para o número necessário de iterações para reduzir o erro inicial de um fator λ .

Método de Jacobi: O raio espectral de G_J é dado por

$$\rho(G_J) = \frac{\cosh + \cosh}{2} = \cosh \approx 1 - \frac{h^2}{2}, \quad h \rightarrow 0$$

e a taxa de convergência é dada por

$$R(G_J) = -\log(\cosh(h)) \approx -\log\left(1 - \frac{h^2}{2}\right) \approx \frac{h^2}{2} + O(h^4)$$

Portanto, a taxa de convergência do método de Jacobi é aproximadamente $\frac{h^2}{2}$, o que é bastante lenta para valores pequenos de h .

Método de Gauss-Seidel: O raio espectral de G_S é dado por

$$\rho(G_S) = \cos^2 h \approx 1 - h^2,$$

e a taxa de convergência é dada por

$$R(G_S) = -\log(\cos^2(h)) = -2 \log(\cos(h)) \approx h^2$$

donde vemos que o método de Gauss-Seidel é duas vezes mais rápido que o método de Jacobi. Porém, converge lentamente se $h \ll 1$.

Método SOR: A taxa de convergência para o método SOR pode ser expressa como:

$$R(G_{SOR}) = \begin{cases} \frac{1}{4} \left(\omega \rho(G_J) + \sqrt{\omega^2 \rho(G_J)^2 - 4(\omega - 1)} \right)^2, & 0 < \omega \leq \omega_{otimo} \\ \omega - 1, & \omega_{otimo} < \omega < 2 \end{cases}$$

4 Problemas Práticos

4.1 Problema 1

Considere a equação de condução de calor:

$$\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} = 0 \quad (32)$$

em que θ é a temperatura. Considerando um domínio quadrado de lados unitários, resolva o problema bidimensional de condução (32), com condições de contorno de Dirichlet contendo descontinuidades na temperatura prescrita, sem e com condição de Neumann prescrita em uma das bordas.

a) No primeiro caso, simule a equação no estado estacionário com condição de contorno tipo Dirichlet, como mostrado na Figura:

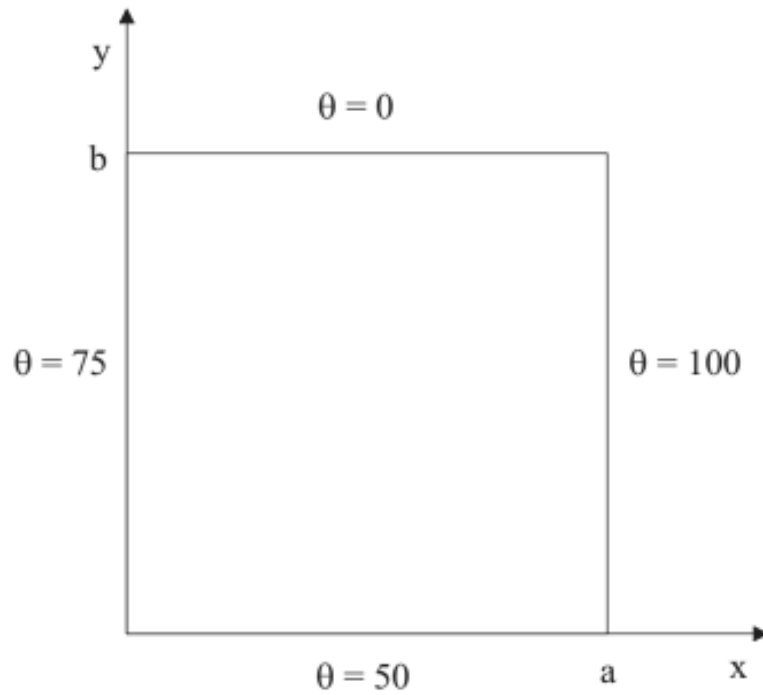


Figura 5: Geometria do problema de condução bidimensional com condição de Dirichlet.

Inicialmente, discretize o domínio em uma malha computacional de 5×5 pontos, gerando 25 nós igualmente espaçados ($\delta x = \delta y = 0.25$). Depois, refine a malha e observe a convergência do método. Utilize os métodos de Jacobi, Gauss-Seidel e SOR na resolução do sistema de equações e compare-os analisando os erros.

b) No segundo caso, considere agora, o problema anterior com condição contorno tipo Neumann, como mostrado na Figura a seguir, cuja borda direita está isolada termicamente. Expressa-se esse isolamento através da seguinte equação

$$\left. \frac{\partial \theta}{\partial x} \right|_{x=a} = 0.$$

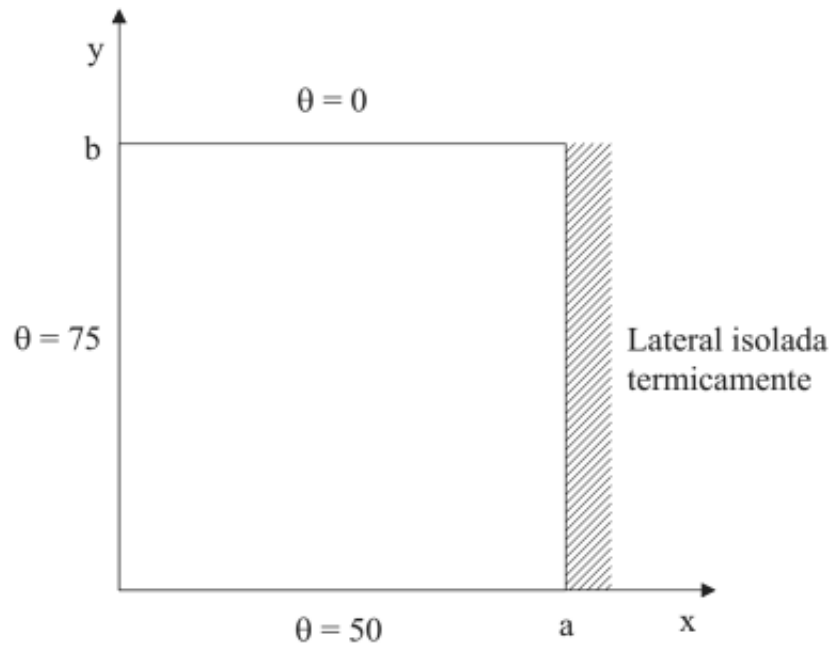


Figura 6: Geometria do problema de condução bidimensional com condição de Neumann.

Solução: a)

Aplicando as diferenças finitas centradas na equação (4) obtemos a equação de diferenças:

$$\begin{aligned}
 (i=1, j=1) \quad & U_{2,1} + U_{0,1} - 4U_{1,1} + U_{1,0} + U_{1,2} = 0 \\
 (i=1, j=2) \quad & U_{2,2} + U_{0,2} - 4U_{1,2} + U_{1,1} + U_{1,3} = 0 \\
 (i=1, j=3) \quad & U_{2,3} + U_{0,3} - 4U_{1,3} + U_{1,2} + U_{1,4} = 0 \\
 (i=2, j=1) \quad & U_{3,1} + U_{1,1} - 4U_{2,1} + U_{2,0} + U_{2,2} = 0 \\
 (i=2, j=2) \quad & U_{3,2} + U_{1,2} - 4U_{2,2} + U_{2,1} + U_{2,3} = 0 \\
 (i=2, j=3) \quad & U_{3,3} + U_{1,3} - 4U_{2,3} + U_{2,2} + U_{2,4} = 0 \\
 (i=3, j=1) \quad & U_{4,1} + U_{2,1} - 4U_{3,1} + U_{3,0} + U_{3,2} = 0 \\
 (i=3, j=2) \quad & U_{4,2} + U_{2,2} - 4U_{3,2} + U_{3,1} + U_{3,3} = 0 \\
 (i=3, j=3) \quad & U_{4,3} + U_{2,3} - 4U_{3,3} + U_{3,2} + U_{3,4} = 0
 \end{aligned}$$

Tendo $h = k = 0.25$, temos que $i, j = 0, 1, 2, 3, 4$, mas os valores na fronteira são conhecidos, então a solução buscada consiste nos índices $i, j = 1, 2, 3$. Pra encontrarmos a matriz A do sistema linear:

$$\begin{aligned}
 (i=1, j=1) \quad & U_{0,1} + U_{2,1} - 4U_{1,1} + U_{0,0} + U_{1,2} = 0 \\
 (i=1, j=2) \quad & U_{0,2} + U_{2,2} - 4U_{1,2} + U_{0,0} + U_{1,3} = 0 \\
 (i=1, j=3) \quad & U_{0,3} + U_{2,3} - 4U_{1,3} + U_{0,0} + U_{1,4} = 0
 \end{aligned}$$

$$\underbrace{\begin{bmatrix} -4 & 1 & & & & \\ 1 & -4 & 1 & & & \\ & 1 & -4 & & 1 & \\ 1 & & & -4 & 1 & 1 \\ & 1 & & 1 & -4 & 1 \\ & & 1 & & 1 & -4 & 1 \\ & & & 1 & & -4 & 1 \\ & & & & 1 & 1 & -4 & 1 \\ & & & & & 1 & 1 & -4 \end{bmatrix}}_A \underbrace{\begin{bmatrix} U_{1,1} \\ U_{1,2} \\ U_{1,3} \\ U_{2,1} \\ U_{2,2} \\ U_{2,3} \\ U_{3,1} \\ U_{3,2} \\ U_{3,3} \end{bmatrix}}_U = \underbrace{\begin{bmatrix} -125 \\ -75 \\ -75 \\ -50 \\ 0 \\ 0 \\ -150 \\ -100 \\ -100 \end{bmatrix}}_r$$

	0	0	0	
75.0	42.8567	37.7226	51.7853	100.0
75.0	58.7047	56.2491	69.4190	100.0
75.0	60.7139	59.1511	69.6424	100.0
	50.0	50.0	50.0	

Tabela 1: Aproximações obtidas pelo Método de Jacobi

	0	0	0	
75.0	42.8569	37.7230	51.7856	100.0
75.0	58.7050	56.2496	69.4194	100.0
75.0	60.7139	59.1514	69.6427	100.0
	50.0	50.0	50.0	

Tabela 2: Aproximações obtidas pelo Método de Gauss-Seidel

	0	0	0	
75.0	42.85	37.72	51.78	100.0
75.0	58.70	56.24	69.41	100.0
75.0	60.71	59.15	69.64	100.0
	50.0	50.0	50.0	

Tabela 3: Aproximações obtidas pelo Método SOR

	0	0	0	
75.0	42.8571	37.7232	51.7857	100.0
75.0	58.7053	56.2500	69.4196	100.0
75.0	60.7141	59.1517	69.6428	100.0
	50.0	50.0	50.0	

Tabela 4: Resultados obtidos por Fortuna

Observando os resultado obtidos por Fortuna e os experimentos realizados neste trabalho, podemos constatar que os métodos obedecem a teoria mostrada na seções iniciais,

gerando resultado condizentes, com valor de erro inferiores a 10^{-2} quando comparados com Fortuna.

Método	Iterações	Tempo(s)
Jacobi	33	0.006008
Gauss-Seidel	18	0.010216
SOR	10	0.003698

Tabela 5: Comparação número de iterações e tempo necessário para o cálculo da solução numérica para $h = k = 0.25$ e tolerância 10^{-3} .

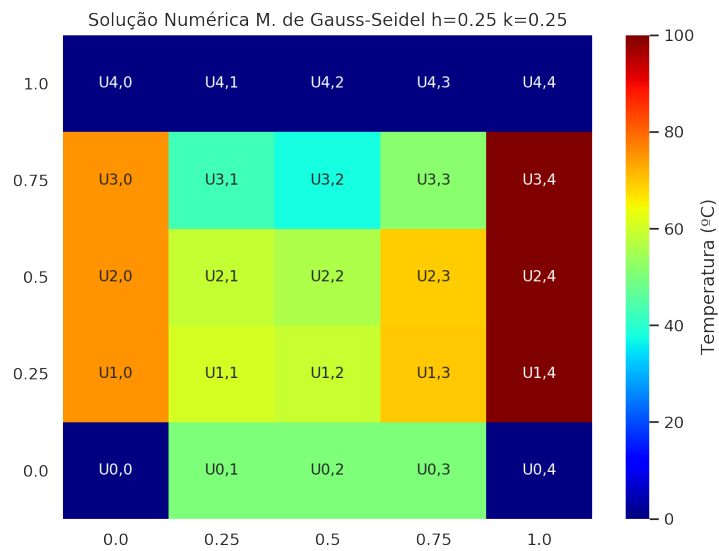


Figura 7: Solução numérica usando Método de Jacobi.

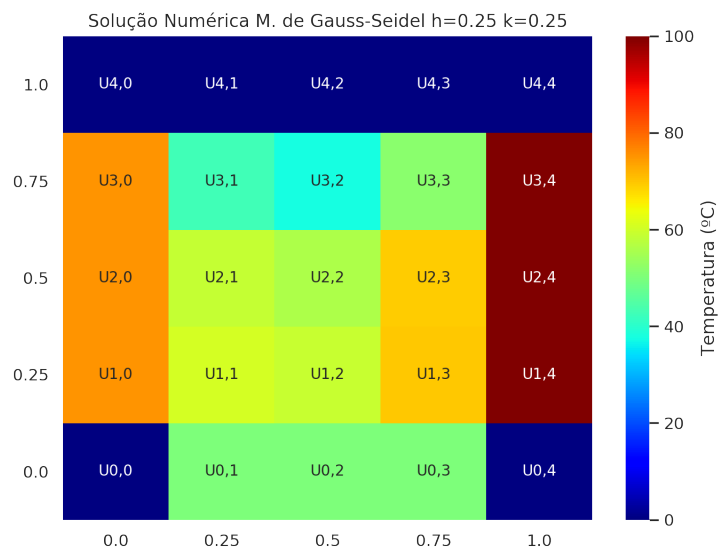


Figura 8: Solução numérica usando Método de Gauss-Seidel.

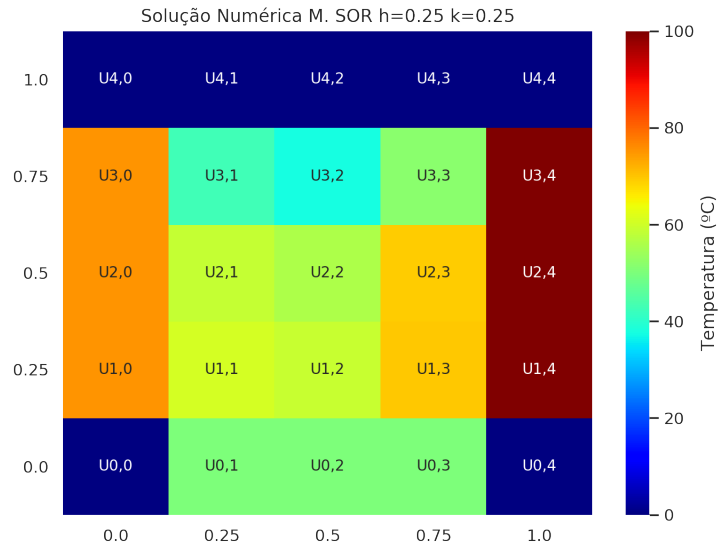


Figura 9: Solução numérica usando Método SOR com $\omega = 1.1715728$.

Caso 2: Refinamento $h = k = 0.05$

Agora faremos um refinamento maior da malha, trabalhando então com $h = k = 0.05$, esse refinamento torna o resultado muito mais fiel a realidade do fenômeno, porém como a matriz A e suas derivadas aumenta bastante de tamanho, o número de iterações realizadas pelos métodos também aumenta muito, o que pode ser observado na tabela 6. Algo notável de se observar é que mesmo trabalhando com menos operações envolvendo matrizes, o método de Jacobi, devido a sua menor velocidade de convergência, quando comparado ao de Gauss-Seidel e SOR (com a escolha apropriada de ω), leva a um maior número de iterações e consequentemente um maior gasto de tempo para convergir.

Método	Iterações	Tempo(s)
Jacobi	754	1.112269
Gauss-Seidel	406	0.460315
SOR	54	0.046512

Tabela 6: Comparação número de iterações e tempo necessário para o cálculo da solução numérica para $h = k = 0.05$ e tolerância 10^{-3} .

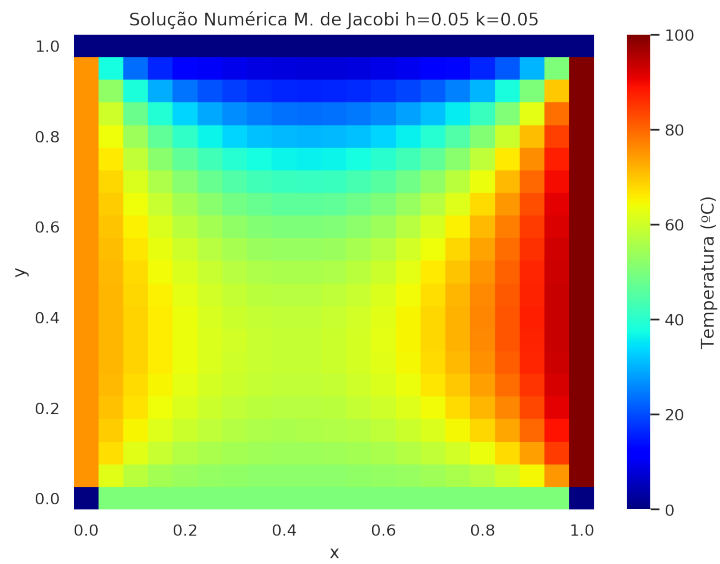


Figura 10: Solução numérica usando Método de Jacobi.

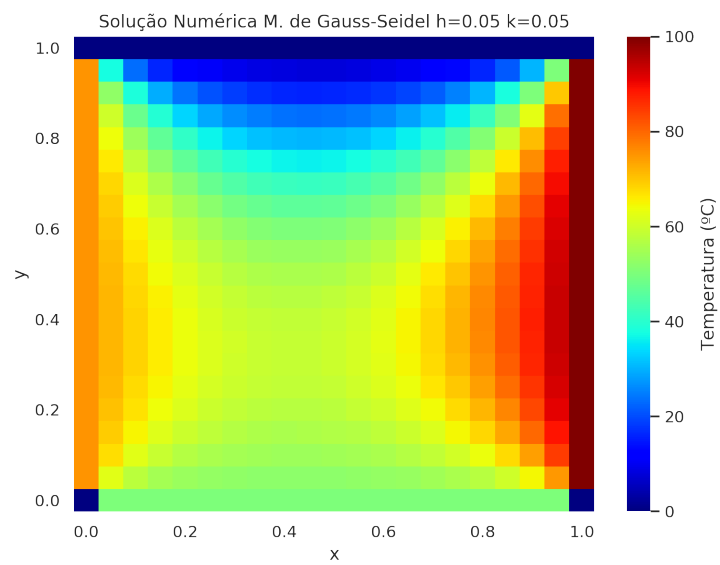


Figura 11: Solução numérica usando Método de Gauss-Seidel.

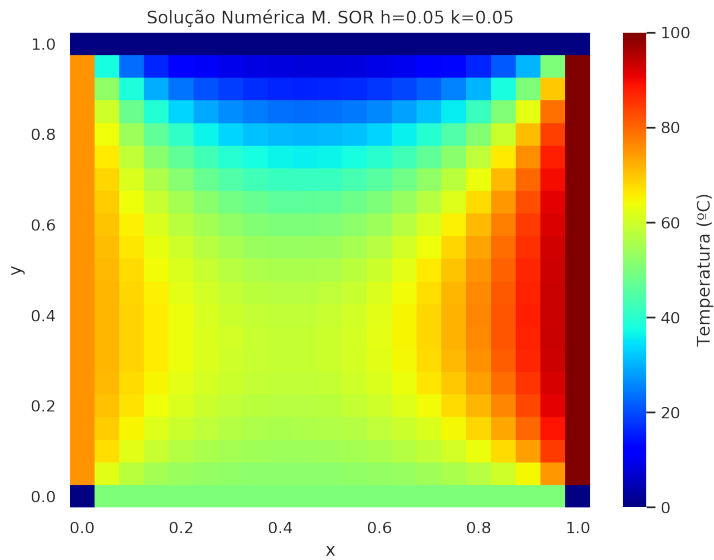


Figura 12: Solução numérica usando Método SOR com $\omega = 1.729453$.

Com o maior refinamento feito na malha, podemos então ver na prática a real diferenciação na convergência dos métodos, enquanto que o método de Gauss-Seidel teve praticamente uma convergência duas vezes mais rápida que o de Jacobi, o método de SOR com escolha ótima de ω , teve ainda menos iterações que o de Gauss-Seidel.

Caso 3: Refinamento $h = k = 0.025$

Nesse caso faremos um refinamento ainda maior.

Método	Iterações	Tempo(s)
Jacobi	+2000	138.830170
Gauss-Seidel	1511	92.368674
SOR	108	1.084296

Tabela 7: Comparação número de iterações e tempo necessário para o cálculo da solução numérica para $h = k = 0.025$ e tolerância 10^{-3} .

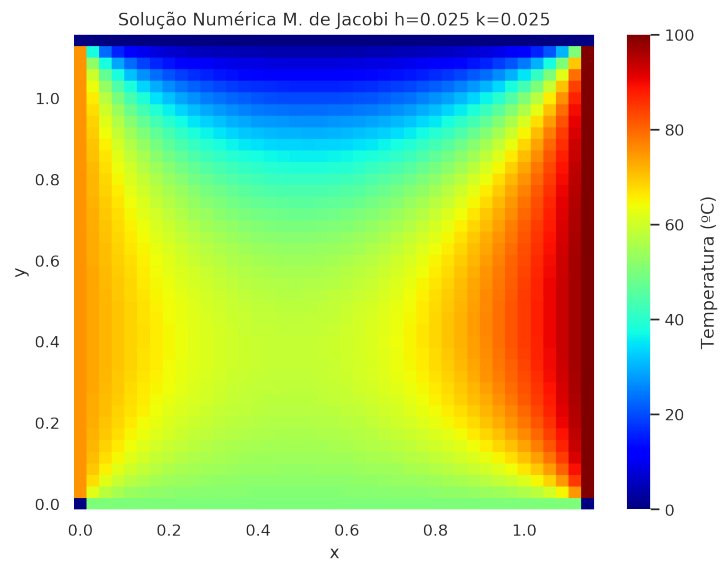


Figura 13: Solução numérica usando Método de Jacobi.

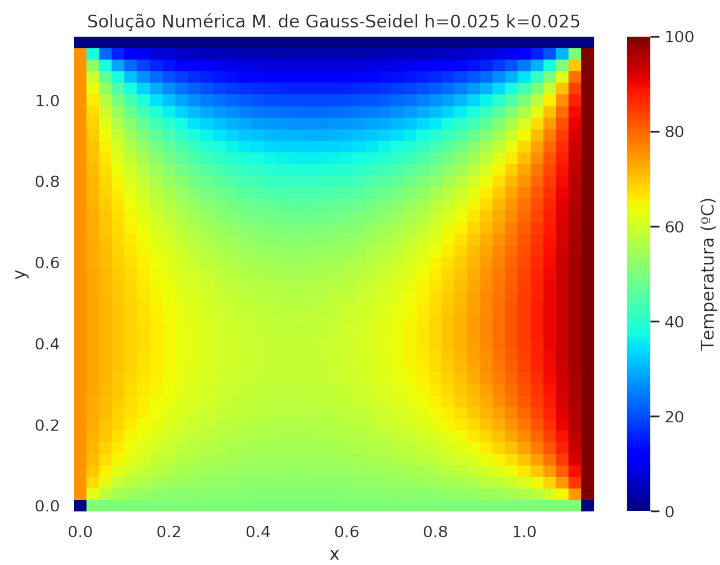


Figura 14: Solução numérica usando Método de Gauss-Seidel.

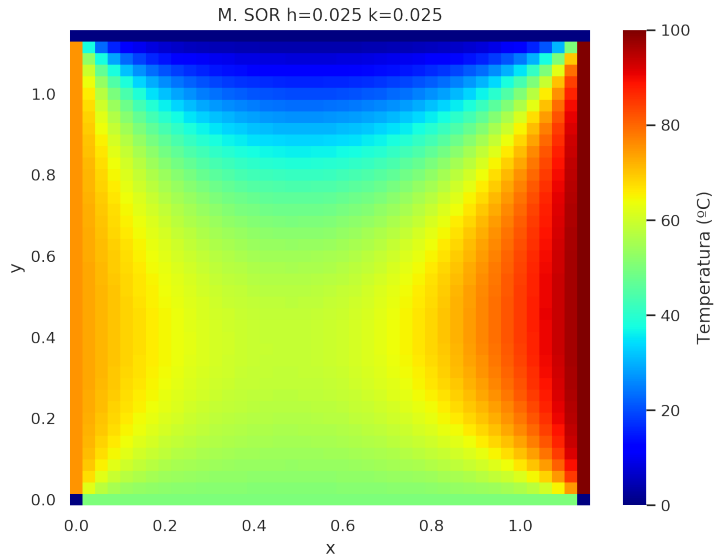


Figura 15: Solução numérica usando Método SOR com $\omega = 1.854497$.

Podemos observar pela tabela que o método de Jacobi ultrapassou o limite de iterações máximo de 2000, e ainda estava longe de alcançar a tolerância de 10^{-3} , ficando com uma diferença entre a iteração 1999 e 20000 de 0.0117.

Caso 3: Comparação de valores de ω

Neste caso focaremos no método de SOR, como foi dito anteriormente na seção sobre esse método, a escolha do ω ótimo não é uma tarefa fácil, porém podemos usar características especiais da matriz A a nosso favor. Dado que a matriz de iteração G_J é simétrica, podemos utilizar de métodos computacionais eficientes para realizar o cálculo dos autovalores da matriz, encontrando a solução em metade do tempo, dado que podemos usar apenas uma das metades da matriz além de evitar encontrar autovalores complexos, e assim encontrar o raio espectral $\rho(G_J)$ de modo a descobrir o ω ótimo.

Quando analisamos a malha onde $h = k = 0.5$ temos a seguinte matriz de iteração:

$$\begin{bmatrix} & -0.25 & & -0.25 & & & \\ -0.25 & & -0.25 & & -0.25 & & \\ & -0.25 & & & & -0.25 & \\ -0.25 & & & -0.25 & & -0.25 & \\ & -0.25 & -0.25 & & -0.25 & & -0.25 \\ & & -0.25 & -0.25 & & & -0.25 \\ & & & -0.25 & & -0.25 & \\ & & & & -0.25 & -0.25 & -0.25 \\ & & & & & -0.25 & \\ & & & & & & -0.25 \end{bmatrix}$$

Encontramos os seguintes autovalores:

$$\begin{bmatrix} -7.07106781 \times 10^{-01} \\ -3.53553391 \times 10^{-01} \\ -3.53553391 \times 10^{-01} \\ -8.35134609 \times 10^{-17} \\ 8.98370475 \times 10^{-18} \\ 9.66820898 \times 10^{-17} \\ 3.53553391 \times 10^{-01} \\ 3.53553391 \times 10^{-01} \\ 7.07106781 \times 10^{-01} \end{bmatrix}$$

Logo temos que $\rho(G_J) = 7.07106781 \times 10^{-01}$ e assim obtemos que $\omega_{otimo} = 1.1715728752538095$ que nos leva a necessitar de apenas 10 iterações para fazer o convergir.

ω	Iterações
0.2	143
0.4	69
0.6	43
0.8	28
1.0	18
1.2	10
1.4	15
1.6	25
1.8	55
2.0	+2000

Tabela 8: Número de iterações para diferentes valores de ω e com $h = k = 0.5$ e tolerância 10^{-3} .

Para a malha onde $h = k = 0.05$ temos que $\omega_{otimo} = 1.729453817281731$ sendo necessárias apenas 54 iterações para que o método convirja, na tabela a seguir fazemos um comparativo do número de iterações necessárias para que o método convirja.

ω	Iterações
0.2	+2000
0.4	1402
0.6	869
0.8	585
1.0	406
1.2	280
1.4	185
1.6	108
1.8	66
2.0	+2000

Tabela 9: Número de iterações para diferentes valores de ω e com $h = k = 0.05$ e tolerância 10^{-3} .

Pela tabelas podemos observar que de fato o método segue o que foi dito na teoria, para valores maiores ou iguais que 2 ou menores ou iguais a 0 o método não converge,

o que é o que se pode notar pelas duas tabelas, vendo que nesses o método ultrapassa o limite de iterações com erros ainda absurdos, podemos exemplificar o caso onde $\omega = 2$ o erro depois de ultrapassar o limite máximo de iterações ficou em 657.540, nem perto da tolerância desejada de 10^{-3} .

Solução b)

Para construir o sistema devemos trabalhar com os seguintes pontos discretizados:

$$\begin{aligned}
(i=1, j=1) \quad & U_{2,1} + U_{0,1} - 4U_{1,1} + U_{1,0} + U_{1,2} = 0 \\
(i=1, j=2) \quad & U_{2,2} + U_{0,2} - 4U_{1,2} + U_{1,1} + U_{1,3} = 0 \\
(i=1, j=3) \quad & U_{2,3} + U_{0,3} - 4U_{1,3} + U_{1,2} + U_{1,4} = 0 \\
(i=2, j=1) \quad & U_{3,1} + U_{1,1} - 4U_{2,1} + U_{2,0} + U_{2,2} = 0 \\
(i=2, j=2) \quad & U_{3,2} + U_{1,2} - 4U_{2,2} + U_{2,1} + U_{2,3} = 0 \\
(i=2, j=3) \quad & U_{3,3} + U_{1,3} - 4U_{2,3} + U_{2,2} + U_{2,4} = 0 \\
(i=3, j=1) \quad & U_{4,1} + U_{2,1} - 4U_{3,1} + U_{3,0} + U_{3,2} = 0 \\
(i=3, j=2) \quad & U_{4,2} + U_{2,2} - 4U_{3,2} + U_{3,1} + U_{3,3} = 0 \\
(i=3, j=3) \quad & U_{4,3} + U_{2,3} - 4U_{3,3} + U_{3,2} + U_{3,4} = 0 \\
(i=4, j=1) \quad & U_{5,1} + U_{3,1} - 4U_{4,1} + U_{4,0} + U_{4,2} = 0 \\
(i=4, j=2) \quad & U_{5,2} + U_{3,2} - 4U_{4,2} + U_{4,1} + U_{4,3} = 0 \\
(i=4, j=3) \quad & U_{5,3} + U_{3,3} - 4U_{4,3} + U_{4,2} + U_{4,4} = 0
\end{aligned}$$

Os pontos $U_{5,j}$ são pontos fantasmas, ou seja, estão fora do domínio discretizado, utilizamos uma manipulação algébrica utilizando das diferenças finitas centradas, como mostrado na Seção sobre Contorno de Neumann, obtemos:

$$\frac{U_{5,j} - U_{3,j}}{2h} = 0 \quad \Rightarrow \quad U_{5,j} = U_{3,j}$$

Ficamos então com:

$$(i = 4, j = 1) \quad U_{5,1} + U_{3,1} - 4U_{4,1} + U_{4,0} + U_{4,2} = 0$$

$$(i = 4, j = 2) \quad U_{5,2} + U_{3,2} - 4U_{4,2} + U_{4,1} + U_{4,3} = 0$$

$$(i = 4, j = 3) \quad U_{5,3} + U_{3,3} - 4U_{4,3} + U_{4,2} + U_{4,4} = 0$$

$$\Downarrow$$

$$(i = 4, j = 1) \quad U_{3,1} + U_{3,1} - 4U_{4,1} + U_{4,0} + U_{4,2} = 0$$

$$(i = 4, j = 2) \quad U_{3,2} + U_{3,2} - 4U_{4,2} + U_{4,1} + U_{4,3} = 0$$

$$(i = 4, j = 3) \quad U_{3,3} + U_{3,3} - 4U_{4,3} + U_{4,2} + U_{4,4} = 0$$

$$\Downarrow$$

$$(i = 4, j = 1) \quad 2U_{3,1} - 4U_{4,1} + U_{4,0} + U_{4,2} = 0$$

$$(i = 4, j = 2) \quad 2U_{3,2} - 4U_{4,2} + U_{4,1} + U_{4,3} = 0$$

$$(i = 4, j = 3) \quad 2U_{3,3} - 4U_{4,3} + U_{4,2} + U_{4,4} = 0$$

O que gera a seguinte matriz A , para o sistema linear:

$$\underbrace{\begin{bmatrix} -4 & 1 & & & & & & & \\ 1 & -4 & 1 & & & & & & \\ & 1 & -4 & & & & & & \\ & & & & & & & & \\ 1 & & & -4 & 1 & & & & \\ & 1 & & 1 & -4 & 1 & & & \\ & & 1 & & 1 & -4 & & & \\ & & & 1 & & & -4 & 1 & 1 \\ & & & & 1 & & 1 & -4 & 1 & 1 \\ & & & & & 1 & & 1 & -4 & \\ & & & & & & 2 & & & -4 & 1 \\ & & & & & & & 2 & & 1 & -4 & 1 \\ & & & & & & & & 2 & & 1 & -4 \end{bmatrix}}_A \underbrace{\begin{bmatrix} U_{1,1} \\ U_{1,2} \\ U_{1,3} \\ U_{2,1} \\ U_{2,2} \\ U_{2,3} \\ U_{3,1} \\ U_{3,2} \\ U_{3,3} \\ U_{4,1} \\ U_{4,2} \\ U_{4,3} \end{bmatrix}}_U = \underbrace{\begin{bmatrix} -125 \\ -75 \\ -75 \\ -50 \\ 0 \\ 0 \\ -50 \\ 0 \\ 0 \\ -50 \\ 0 \\ 0 \end{bmatrix}}_r$$

Podemos notar que para este caso algumas das entradas da matriz são alteradas, o que leva a perda da simetria da matriz, o que complica ainda mais para encontrar o ω ótimo para o método SOR.

Resolvemos então esse sistema linear utilizando dos métodos iterativos de Jacobi,

Gauss-Seidel e SOR com $\omega = 1.4508871307183546$, inicialmente em uma malha de 5x5 e obtemos os seguintes resultados:

	0	0	0	0
75.0	37.6317	23.6894	18.3384	16.9234
75.0	51.8375	38.7882	32.7415	31.0178
75.0	55.9306	46.8853	42.8230	41.6657
	50.0	50.0	50.0	50.0

Tabela 10: Aproximações obtidas pelo Método de Jacobi

	0	0	0	0
75.0	37.6319	23.6899	18.3391	16.9243
75.0	51.8377	38.7888	32.7424	31.0188
75.0	55.9307	46.8856	42.8234	41.6664
	50.0	50.0	50.0	50.0

Tabela 11: Aproximações obtidas pelo Método de Gauss-Seidel

	0	0	0	0
75.0	37.6320	23.6901	18.3393	16.9245
75.0	51.8379	38.7891	32.7428	31.0192
75.0	55.9309	46.8859	42.8238	41.6667
	50.0	50.0	50.0	50.0

Tabela 12: Aproximações obtidas pelo Método SOR

	0	0	0	0
75.0	37.63	23.69	18.33	16.92
75.0	51.83	38.78	32.74	31.01
75.0	55.93	46.88	42.82	41.66
	50.0	50.0	50.0	50.0

Tabela 13: Resultados obtidos por Fortuna

Observando os resultados obtidos por Fortuna e os experimentos realizados neste trabalho, podemos constatar que os métodos obedecem a teoria mostrada nas seções iniciais, gerando resultados condizentes, com valor de erro inferiores a 10^{-2} quando comparados com Fortuna.

Método	Iterações	Tempo(s)
Jacobi	51	0.008162
Gauss-Seidel	27	0.001577
SOR	16	0.003421

Tabela 14: Comparação número de iterações e tempo necessário para o cálculo da solução numérica para $h = k = 0.25$ e tolerância 10^{-3} .

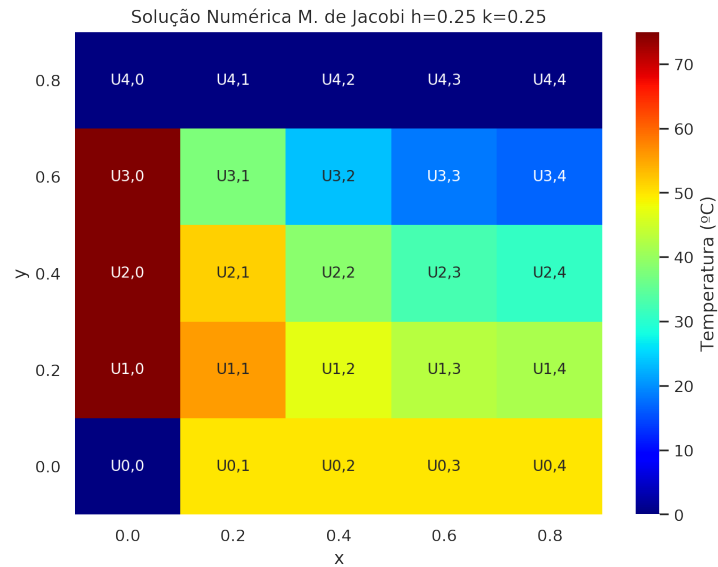


Figura 16: Solução numérica usando Método de Jacobi.

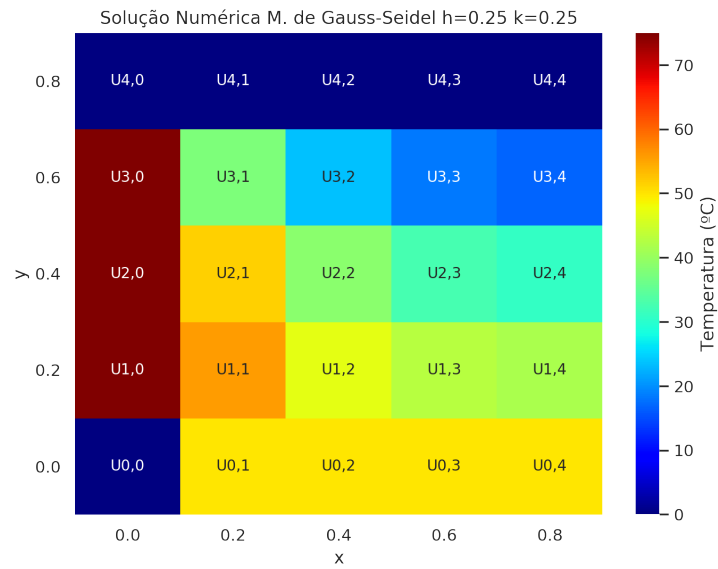


Figura 17: Solução numérica usando Método de Gauss-Seidel.

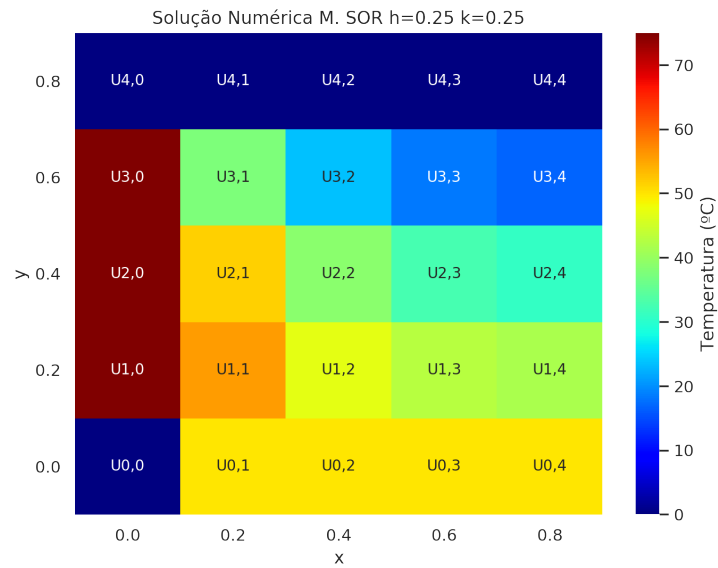


Figura 18: Solução numérica usando Método SOR com $\omega = 1.4508871307183546$.

Caso 2: Refinamento $h = k = 0.05$

Nesse caso faremos um refinamento ainda maior. E para o método de SOR, usaremos $\omega = 1.8$. Obtemos então os seguintes resultados:

Método	Iterações	Tempo(s)
Jacobi	1089	1.565939
Gauss-Seidel	584	0.611001
SOR	56	0.029911

Tabela 15: Comparação número de iterações e tempo necessário para o cálculo da solução numérica para $h = k = 0.05$ e tolerância 10^{-3} .

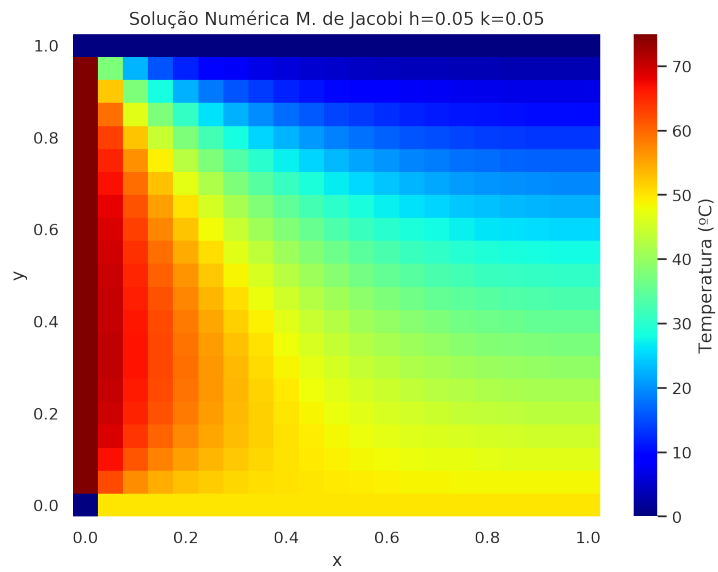


Figura 19: Solução numérica usando Método de Jacobi.

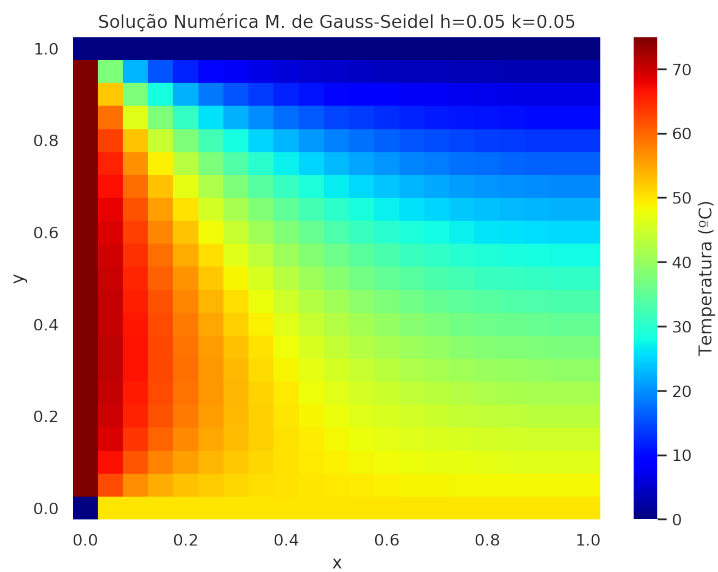


Figura 20: Solução numérica usando Método de Gauss-Seidel.

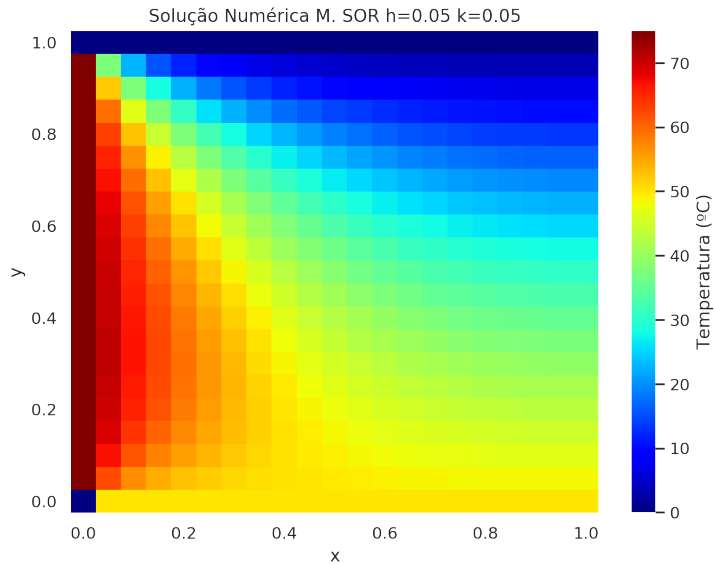


Figura 21: Solução numérica usando Método SOR.

Caso 3: Refinamento $h = k = 0.0025$

Para este caso, já que fizemos o uso de uma malha tão refinada, os métodos de Jacobi e Gauss-Seidel ultrapassaram o limite máximo de iterações, sendo que o Jacobi termina com um erro na tolerância de 0.046823, apenas o método SOR com parâmetro $\omega_{otimo} = 1.8509422023650053$ conseguiu aproximar em 186 iterações. Podemos notar ainda que para valores de ω interiores a 1.0 o método de SOR também ultrapassa o limite de iterações. Podemos destacar que era o esperado já que $\omega = 1.0$ o SOR equivale ao método de Gauss-Seidel.

Método	Iterações	Tempo(s)
Jacobi	+2000	147.898482
Gauss-Seidel	+2000	125.118288
SOR	186	1.331988

Tabela 16: Comparação número de iterações e tempo necessário para o cálculo da solução numérica para $h = k = 0.0025$ e tolerância 10^{-3} .

Na tabela a seguir podemos observar os erros finais entre a iteração 1999 e 2000 para os métodos de Jacobi e de Gauss-Seidel que não atingiram a tolerância desejada de 10^{-3} :

Método	Erro Tolerância
Jacobi	0.046823
Gauss-Seidel	0.001892
SOR	0.000960

Tabela 17: Comparação número erro de tolerância final .

Pela tabela podemos notar que especificamente para Gauss-Seidel, apenas algumas iterações a mais seriam o suficiente para que atingisse a tolerância desejada.

Nas figuras 22, 23 e 24, não podemos notar muitas diferenças, dado que mesmo tendo ultrapassado o limite de iterações a diferença entre os resultados de Jacobi, Gauss-Seidel e SOR são pequenos.

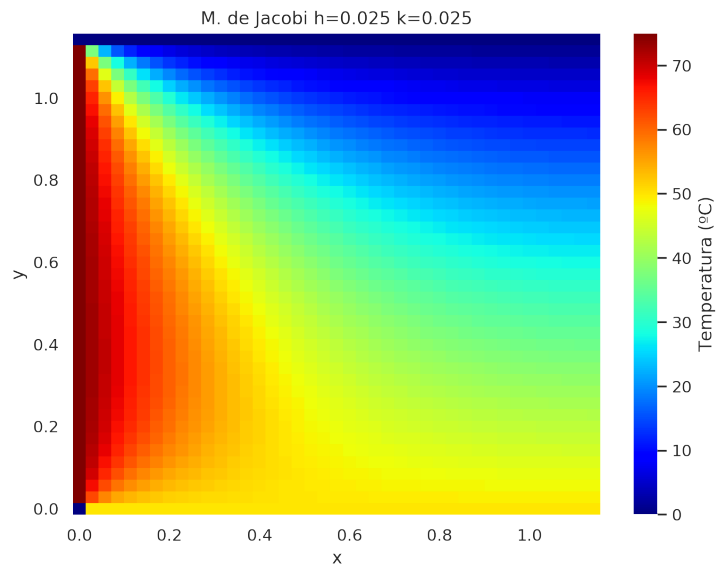


Figura 22: Solução numérica usando Método de Jacobi.

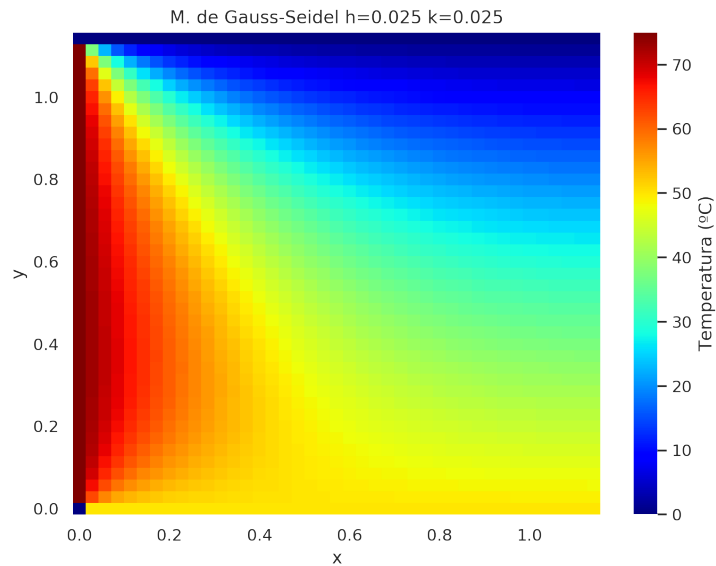


Figura 23: Solução numérica usando Método de Gauss-Seidel.

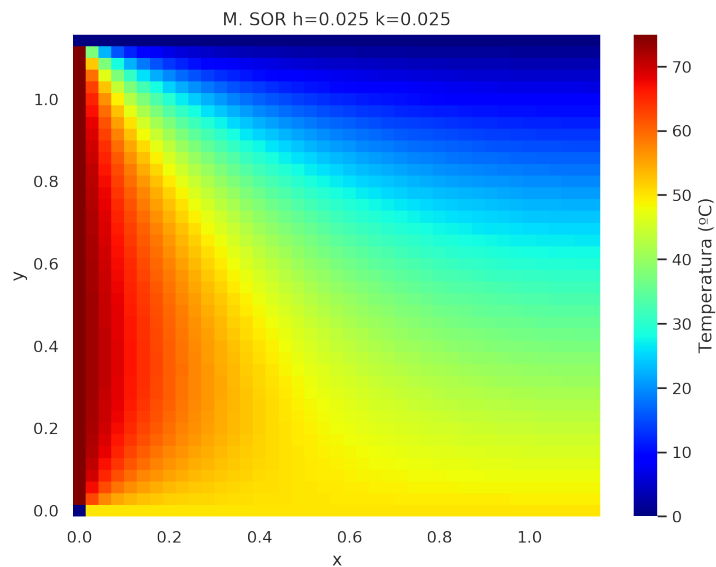


Figura 24: Solução numérica usando Método SOR.

Vale destacar que no resultado da Figura 22, nas regiões de vermelho mais escuro, ou seja, próximos de 70°C e em azul escuro, próximos a 0°C , quando comparamos as mesmas regiões com a Figura 24, podemos observar que a variação de cor/temperatura é muito mais suave na segunda figura, o que devido ao fato que a aproximação pelo método SOR, neste caso, está muito mais próxima do comportamento do fenômeno real.

4.2 Problema 2

Considere o problema elíptico:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2 \quad (33)$$

Considerando um domínio quadrado, ou seja, $D = \{(x, y) \in \mathbb{R}^2 : -1 \leq x \leq 1, -1 \leq y \leq 1\}$ resolva o problema elíptico bidimensional (33), com condições de contorno de Dirichlet, ou seja, $u = 0$ em todos os lados do quadrado.

Obtenha soluções aproximadas para u com $h = k = 0.2; 0.1$ e 0.05 . Resolva o sistema linear proveniente da discretização por algum dos métodos aprendidos em sala. Compare a solução analítica com a solução numérica.

$$u(x, y) = 1 - y^2 - \frac{32}{\pi^3} \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^3} \operatorname{sech}\left(\frac{(2n+1)\pi}{2}\right) \cosh\left(\frac{(2n+1)\pi x}{2}\right) \cos\left(\frac{(2n+1)\pi y}{2}\right).$$

Figura 25: Solução analítica para o problema.

Para solucionar numericamente este problema, faremos uso novamente das diferenças finitas, de forma análoga ao realizado no Problema 1a. Porém agora trabalhamos com uma equação no modelo de Poisson, obtendo a equação:

$$U_{i-1,j} + U_{i+1,j} - 4U_{i,j} + U_{i,j-1} + U_{i,j+1} = -2h^2$$

Tendo a solução conhecida na fronteira, devemos obter apenas o valor aproximado para os pontos no interior. Devemos então resolver o sistema do tipo $AU = r$. Com um chute inicial de zero para os valores de U e sendo r um vetor do tipo $r = [-0.08, -0.08, \dots, -0.08]^T$. Utilizaremos do Método Iterativo de Gauss-Seidel para resolver o sistema linear com diferentes refinamentos de malha.

Caso 1: $h = k = 0.2$

Para este caso tomaremos uma malha mais grossa, obteremos então uma matriz A de 81×81 com o seguinte padrão:

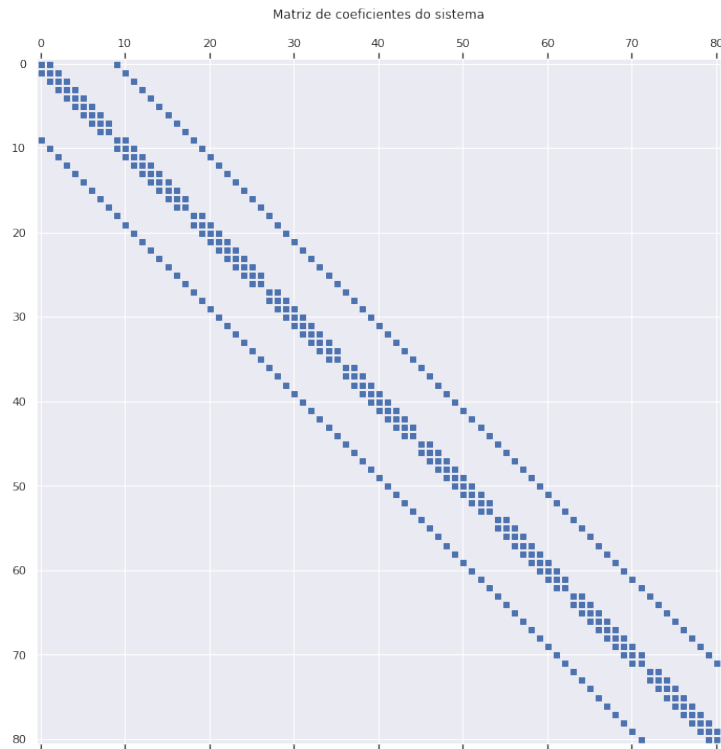


Figura 26: Matriz de coeficientes do sistema

Usando do Método de Gauss-Seidel, que tem uma boa velocidade de convergência e a facilidade de não precisar calcular um parâmetro adicional, como no caso do método SOR, iremos solucionar numericamente o problema, discretizando em uma malha com $h = k = 0.2$:

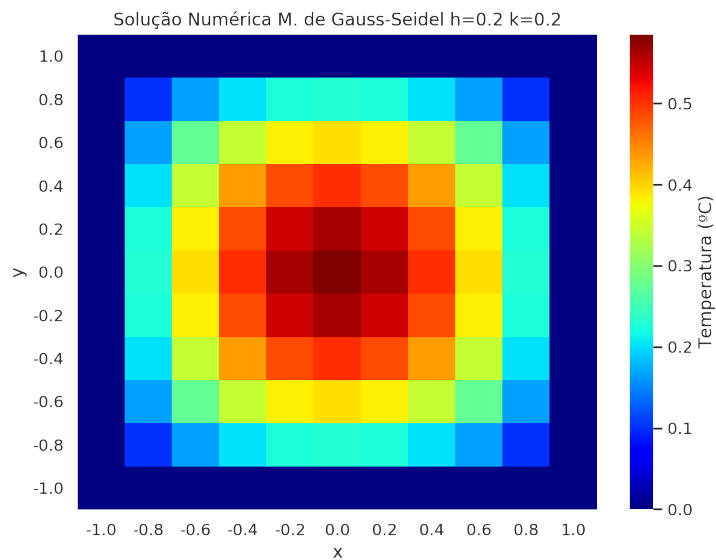


Figura 27: Solução numérica usando Método de Gauss-Seidel com $h = k = 0.2$.

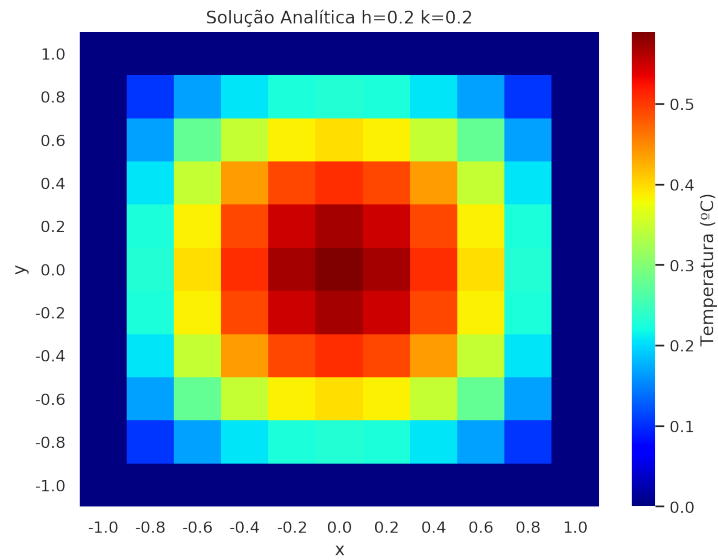


Figura 28: Solução analítica para $h = k = 0.2$.

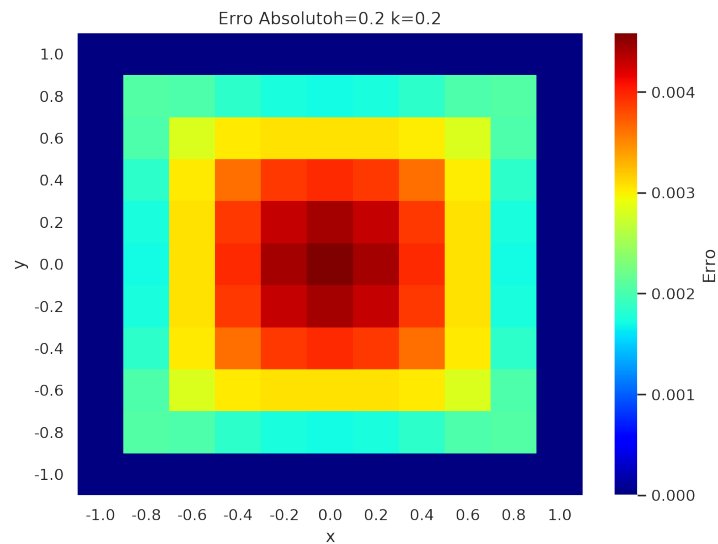


Figura 29: Erro entre a solução numérica do M. de Gauss-Seidel e a solução exata para $h = k = 0.2$.

Caso 2: $h = k = 0.1$

Usando novamente do Método de Gauss-Seidel, iremos solucionar numericamente o problema, discretizando em uma malha de 20x20.

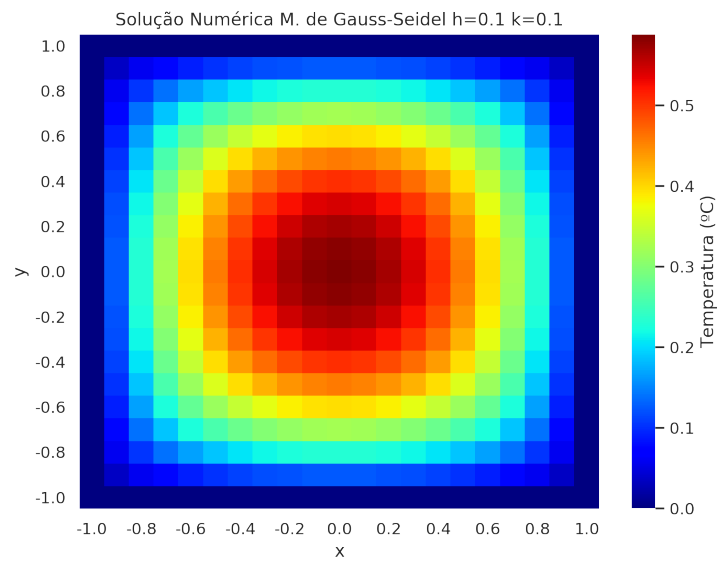


Figura 30: Solução numérica usando Método de Gauss-Seidel com $h = k = 0.1$.

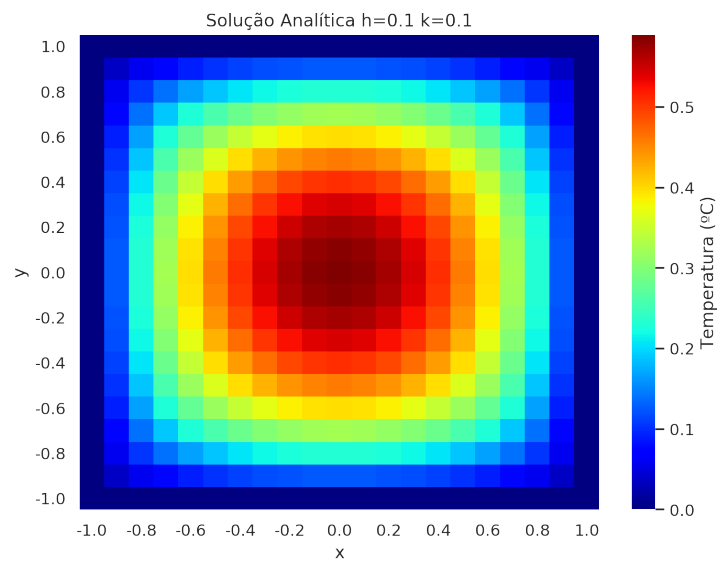


Figura 31: Solução analítica para $h = k = 0.1$.

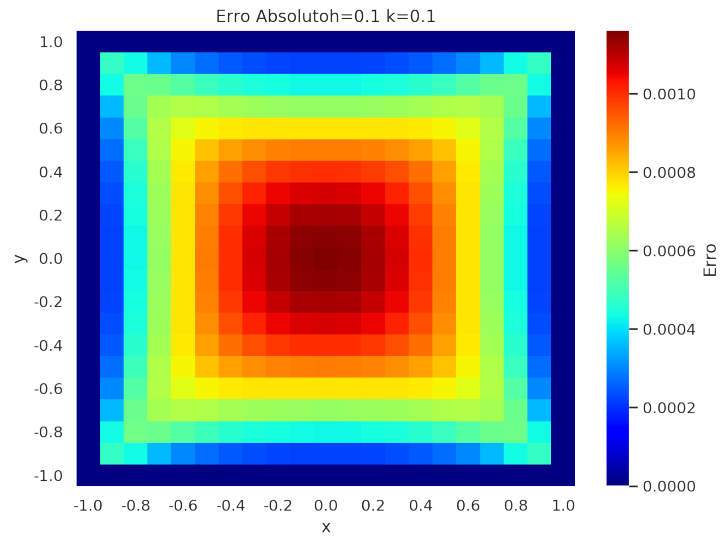


Figura 32: Erro entre a solução numérica do M. de Gauss-Seidel e a solução exata para $h = k = 0.1$.

Obtemos a aproximação com uma tolerância de 10^{-3} em apenas 0.498147 segundos e com 486 iterações.

Caso 3: $h = k = 0.05$

Discretizando em uma malha de 40x40 obtemos a seguinte aproximação:

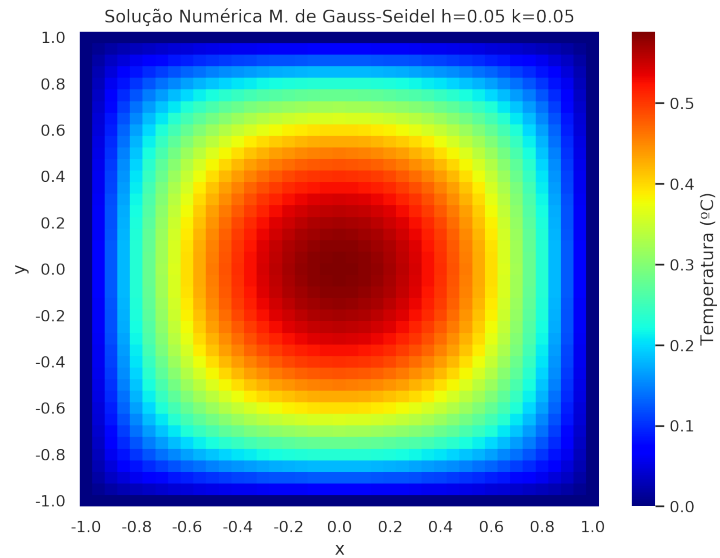


Figura 33: Solução numérica usando Método de Gauss-Seidel com $h = k = 0.05$.

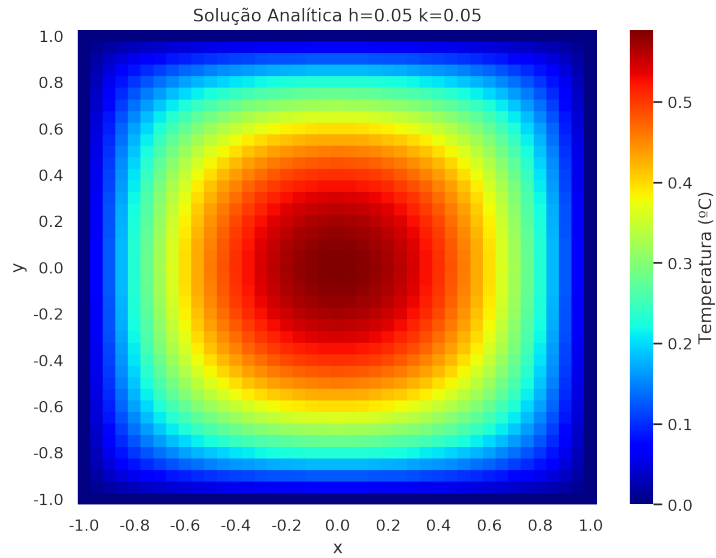


Figura 34: Solução analítica para $h = k = 0.05$.

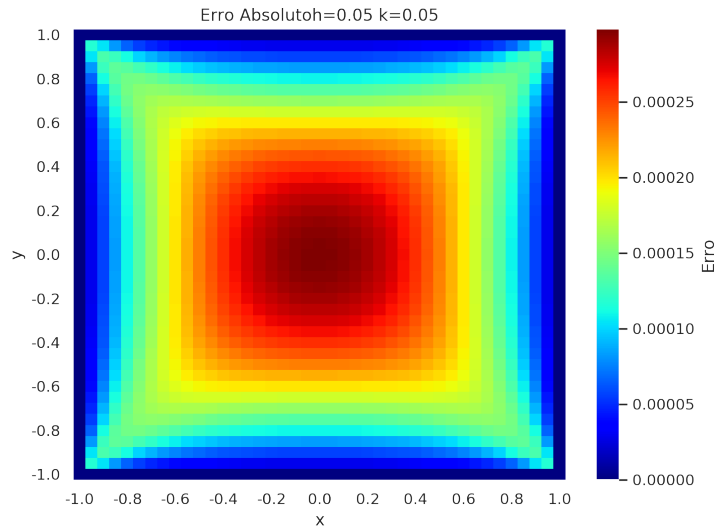


Figura 35: Erro entre a solução numérica do M. de Gauss-Seidel e a solução exata para $h = k = 0.05$.

Obtemos a aproximação com uma tolerância de 10^{-3} em apenas 92.093533 segundos e com 1832 iterações.

Podemos notar observando as Figuras 29, 32, e 35 que os erros obtidos são todos da ordem de 10^{-3} , o que demonstra que o método é adequado. Podemos destacar ainda que perto da região central, em pontos discretizados próximos a o ponto $(0,0)$ encontramos os maiores valores de erros, isso se deve principalmente ao acúmulo dos pequenos erros durante o processo iterativo, dado que os pontos centrais são os mais distantes dos valores analiticamente conhecidos, ou seja, os valores da borda, o mais esperado era que realmente

nestes pontos os valores dos erros fossem superiores, enquanto que para valores mais próximos das bordas/fronteiras, o erro fosse bem menor.

5 Conclusão

Com esse trabalho foi possível exercitar na prática as teorias aprendidas em aula, com o foco nas equações elípticas, utilizando do clássico método das Diferenças Finitas, aliado a os Métodos iterativos de Jacobi, Gauss-Seidel e SOR, foi possível resolver alguns problemas bidimensionais com diferentes condições de contorno e estudando suas especificidades, as condições de contorno são um dos fatores determinantes ao se trabalhar com equações diferenciais numericamente, cada tipo de contorno, seja ele do tipo Dirichlet, Neumann ou misto, deve ser analisado individualmente, e essa mudança pode levar a dificuldades na implementação.

O método SOR é excelente para acelerar a convergência da solução de sistemas de equações lineares, como foi visto no Problema 1, onde em todos os casos vistos, esse método foi o que convergiu mais rapidamente. Porém, a determinação de ω_{otimo} é difícil, pois os teoremas existentes são aplicáveis a um grupo específico de sistemas de equações. Sendo assim, a aplicação do método ainda é limitada. Então em quesitos práticos, muitas vezes é mais natural usar o Método de Gauss-Seidel, podemos elucidar que em casos onde a matriz A é simétrica encontrar ω_{otimo} é mais fácil então faria bastante sentido aplicar o método SOR neste caso, porém para matrizes não simétricas como foi o caso discutido no Problema 1b, o método de Gauss-Seidel se mostra mais adequado.

O método de Gauss-Seidel como esperado pelo que foi discutido pela teoria, tendo uma convergência duas vezes mais rápida que o de Jacobi, foi o que depois do SOR, teve a melhor performance. Outro aspecto que vale ser destacado é que para estes problemas, as matrizes A são esparsas, o que abre muitas possibilidades para explorar essa característica levando a implementações computacionais muito mais eficientes.

Bibliografia

CUMINATO, José Alberto; MENEGUETTE, Messias. Discretização de equações diferenciais parciais: técnicas de diferenças finitas. Sociedade Brasileira de Matemática, 2013.

AGUIRRE, L. A. Introdução à Identificação de Sistemas, Técnicas Lineares e Não lineares Aplicadas a Sistemas Reais. Belo Horizonte, Brasil, EDUFMG. 2004.

BURDEN, R. L. FAIRES, J. D. Análise Numérica. São Paulo: Thomson Pioneira, 2003.

Boyce et al,Elementary Differential Equations and Boundary Value ,WILEY, 2017, ISBN: 978-1-119-16975-8

U.M. Ascher, C. Greif,A First Course in Numerical Methods, SIAM,2011