

Aula 8

- **PDE elípticas**
- **Métodos iterativos**
- **Método de Jacobi**
- **Método de Gauss-Seidel**
- **Método da relaxação (SOR)**
- **Aplicação: equação de Poisson**



Métodos numéricos para sistemas de equações Lineares

Os métodos numéricos para resolver sistemas de equações lineares podem ser:

- **diretos**

- eliminação de Gauss, eliminação de Gauss–Jordan, método de inversão da matriz, fatorização LU.
- Dão-nos soluções exatas apenas afetadas por erros de arredondamento.
- São numericamente mais eficientes para sistemas com poucas equações (poucas centenas de equações).

- **iterativos ou de relaxação**

- Jacobi, Gauss–Seidel, sobre-relaxação sucessiva.
- A solução é obtida assintoticamente por um processo iterativo.
- São numericamente eficientes para um grande número de equações cuja matriz é diagonalmente dominante (muitos dos casos que resultam da discretização de equações diferenciais).



Métodos Iterativos

Considere o sistema de equações lineares

$$Ax = b$$

É possível obter a solução diretamente a partir da matriz inversa de A :

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

Na realidade, sabe que em geral a matriz inversa não é calculada explicitamente quando se usam métodos diretos. Para introduzir os métodos iterativos ou de relaxação, em vez do caso geral, vamos considerar um sistema de apenas 3 equações:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix},$$

esperando que a discussão se torne assim mais simples de seguir.



Métodos Iterativos

O sistema também pode ser escrito como,

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3, \end{cases} \quad (1)$$

ou como,

$$\begin{cases} x_1 = (-a_{12}x_2 - a_{13}x_3 + b_1)/a_{11} , \\ x_2 = (-a_{21}x_1 - a_{23}x_3 + b_2)/a_{22} , \\ x_3 = (-a_{31}x_1 - a_{32}x_2 + b_3)/a_{33} . \end{cases} \quad (2)$$

Esta última forma não ajuda em nada na resolução direta do sistema, mas é a base para o mais simples dos métodos iterativos, o método de Jacobi.



Método de Jacobi

A ideia é atribuir valores iniciais $\mathbf{x}(0)$ às incógnitas e usar as equações 2 para obter novas estimativas $\mathbf{x}(1)$:

$$\begin{cases} x_1^{(1)} = \left(-a_{12}x_2^{(0)} - a_{13}x_3^{(0)} + b_1 \right) / a_{11} , \\ x_2^{(1)} = \left(-a_{21}x_1^{(0)} - a_{23}x_3^{(0)} + b_2 \right) / a_{22} , \\ x_3^{(1)} = \left(-a_{31}x_1^{(0)} - a_{32}x_2^{(0)} + b_3 \right) / a_{33} . \end{cases}$$

O processo é então repetido:

$$\begin{cases} x_1^{(k+1)} = \left(-a_{12}x_2^{(k)} - a_{13}x_3^{(k)} + b_1 \right) / a_{11} , \\ x_2^{(k+1)} = \left(-a_{21}x_1^{(k)} - a_{23}x_3^{(k)} + b_2 \right) / a_{22} , \\ x_3^{(k+1)} = \left(-a_{31}x_1^{(k)} - a_{32}x_2^{(k)} + b_3 \right) / a_{33} , \end{cases}$$

até que, espera-se, os valores obtidos estejam suficientemente próximos das soluções exatas \mathbf{x} .



Método de Jacobi — algoritmo

1. Define-se a matriz A de elementos a_{ij} e o vetor b .

2. Escolhem-se valores iniciais para x_{old}

3. Para $i = 1, 2, \dots$

$$x_{new}(i) = -\frac{1}{a_{ii}} \sum_{j \neq i} a_{ij} x_{old}(j) + \frac{1}{a_{ii}} b(i)$$

4. Calcula-se os vetores $\|x_{new} - x_{old}\|$ ou $\|Ax_{new} - b\|$

5. Quando a média ou o valor máximo de um dos vetores acima é maior que uma dada tolerância, faz-se $x_{old} = x_{new}$ e volta-se ao ponto 3.

6. Quando o valor referido acima é menor que a tolerância, considera-se que o método convergiu. A solução numérica é x_{new} .



Convergência dos métodos iterativos

Escreveu-se o algoritmo do método, mas só se pode aplicar se ele convergir, ou seja, se

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}$$

Para se discutir esta questão, podem escrever-se as equações do método de Jacobi aplicado ao nosso exemplo da seguinte forma:

$$\begin{cases} a_{11}x_1^{(k+1)} = -a_{12}x_2^{(k)} - a_{13}x_3^{(k)} + b_1, \\ a_{22}x_2^{(k+1)} = -a_{21}x_1^{(k)} - a_{23}x_3^{(k)} + b_2, \\ a_{33}x_3^{(k+1)} = -a_{31}x_1^{(k)} - a_{32}x_2^{(k)} + b_3, \end{cases} \quad (3)$$

e decompor a matriz \mathbf{A} numa matriz diagonal \mathbf{D} , numa matriz \mathbf{U} (*upper*) que contém todos os elementos de \mathbf{A} acima da diagonal principal e numa matriz \mathbf{L} (*lower*) que contém todos os elementos de \mathbf{A} abaixo da diagonal principal:

$$\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U}.$$



Convergência dos métodos iterativos

No nosso exemplo,

$$\mathbf{D} = \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ a_{31} & a_{32} & 0 \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} 0 & a_{12} & a_{13} \\ 0 & 0 & a_{23} \\ 0 & 0 & 0 \end{pmatrix}.$$

É fácil de ver que as equações 3 podem ser escritas na forma matricial como,

$$\mathbf{D} \mathbf{x}(k+1) = -(\mathbf{L} + \mathbf{U}) \mathbf{x}(k) + \mathbf{b}$$

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

Para estudar a convergência dos métodos iterativos, define-se a matriz \mathbf{T} e o vetor \mathbf{c} , tais que

$$\mathbf{x}(k+1) = \mathbf{T} \mathbf{x}(k) + \mathbf{c} \quad (4)$$



Convergência dos métodos iterativos

Como se viu, no método de Jacobi,

$$\mathbf{T} = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}) \quad \text{e} \quad \mathbf{c} = \mathbf{D}^{-1} \mathbf{b}.$$

Raio Espectral

O raio espectral da matriz \mathbf{T} é dado pelo maior dos valores absolutos dos seus valores próprios

$$\rho(\mathbf{T}) = |\lambda_i|_{\max} \quad (5)$$

Convergência dos métodos iterativos

Os métodos iterativos estudados nesta aula convergem para todos os valores iniciais $\mathbf{x}^{(0)}$ e para todos os \mathbf{b} se e só se

$$\rho(\mathbf{T}) < 1 \quad (6)$$



Convergência dos métodos iterativos

Taxa de Convergência

A taxa de convergência $r(\mathbf{T})$, definida como o aumento, por iteração, do número de casas decimais corretas da solução numérica é

$$r(\mathbf{T}) = -\log_{10} \rho(\mathbf{T}) \quad (7)$$

Quando o raio espectral está próximo de 1, a taxa de convergência pode ser aproximada por

$$r(\mathbf{T}) \simeq \frac{1 - \rho(\mathbf{T})}{\ln 10} \simeq \frac{1 - \rho(\mathbf{T})}{2.30} \quad (8)$$

12-05-2020



Método de Gauss–Seidel

Um melhoramento do método de Jacobi consiste em calcular cada $x_i^{(k+1)}$ usando a solução atual dos outros elementos caso ela já tenha sido atualizada, isto é, usar $x_{j<i}^{(k+1)}$ e $x_{j>i}^{(k)}$:

$$\begin{cases} x_1^{(k+1)} = \left(-a_{12}x_2^{(k)} - a_{13}x_3^{(k)} + b_1 \right) / a_{11} , \\ x_2^{(k+1)} = \left(-a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} + b_2 \right) / a_{22} , \\ x_3^{(k+1)} = \left(-a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} + b_3 \right) / a_{33} . \end{cases}$$



Método de Gauss-Seidel — algoritmo

1. Define-se a matriz A de elementos a_{ij} e o vetor b .
2. Escolhem-se valores iniciais para x_{old}
3. Faz-se $x_{\text{new}} = x_{\text{old}}$.
4. Para $i = 1, 2, \dots$

$$x_{\text{new}}(i) = -\frac{1}{a_{ii}} \sum_{j \neq i} a_{ij} x_{\text{new}}(j) + \frac{1}{a_{ii}} b(i)$$

5. Calcula-se os vetores $\|x_{\text{new}} - x_{\text{old}}\|$ ou $\|Ax_{\text{new}} - b\|$
6. Quando a média ou o valor máximo de um dos vetores acima é maior que uma dada tolerância, faz-se $x_{\text{old}} = x_{\text{new}}$ e volta-se ao ponto 4.
7. Quando o valor referido acima é menor que a tolerância, considera-se que o método convergiu. A solução numérica é x_{new} .



Método de Gauss-Seidel

As equações do método de Gauss–Seidel no exemplo considerado podem ser escritas como

$$\begin{cases} a_{11}x_1^{(k+1)} = -a_{12}x_2^{(k)} - a_{13}x_3^{(k)} + b_1, \\ a_{21}x_1^{(k+1)} + a_{22}x_2^{(k+1)} = -a_{23}x_3^{(k)} + b_2, \\ a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)} + a_{33}x_3^{(k+1)} = +b_3, \end{cases}$$

ou seja,

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

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

Comparando com $\mathbf{x}^{(k+1)} = T\mathbf{x}^{(k)} + \mathbf{c}$, obtêm-se as matrizes para o método de Gauss–Seidel:

$$T = -(L + D)^{-1}U \quad \text{e} \quad \mathbf{c} = (L + D)^{-1} \mathbf{b}.$$



Métodos de sub- ou sobre-relaxação

A ideia é partir do método de Gauss–Seidel,

$$\tilde{x}_i^{(k+1)} = -\frac{1}{a_{ii}} \left(\sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} + \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right) + \frac{b_i}{a_{ii}}$$

que promove uma variação na solução de $\mathbf{d} = \tilde{\mathbf{x}}^{(k+1)} - \mathbf{x}^{(k)}$. No entanto, atualiza-se a solução como $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha \mathbf{d}$ ou seja

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha(\tilde{\mathbf{x}}^{(k+1)} - \mathbf{x}^{(k)}) = (1 - \alpha) \mathbf{x}^{(k)} + \alpha \tilde{\mathbf{x}}^{(k+1)}$$

Para as componentes fica

$$x_i^{(k+1)} = (1 - \alpha)x_i^{(k)} + \frac{\alpha}{a_{ii}} \left[-\sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} + b_i \right]$$



Método de sub- ou sobre-relaxação — algoritmo

1. Define-se a matriz A de elementos a_{ij} e o vetor b .

2. Escolhem-se valores iniciais para x_{old}

3. Faz-se $x_{new} = x_{old}$.

4. Para $i = 1, 2, \dots$

$$x_{new}(i) = (1 - \alpha)x_{old}(i) + \frac{\alpha}{a_{ii}} \left[-\sum_{j \neq i} a_{ij} x_{new}(j) + b_i \right]$$

5. Calcula-se os vetores $\|x_{new} - x_{old}\|$ ou $\|Ax_{new} - b\|$

6. Quando a média ou o valor máximo de um dos vetores acima é maior que uma dada tolerância, faz-se $x_{old} = x_{new}$ e volta-se ao ponto 4.

7. Quando o valor referido acima é menor que a tolerância, considera-se que o método convergiu.
A solução numérica é x_{new} .



Métodos de sub- ou sobre-relaxação

Para avaliar a convergência e taxa de convergência tem que se determinar o raio espectral da matriz T . Neste caso,

$$a_{ii}x_i^{(k)} + \alpha \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} = (1 - \alpha)a_{ii}x_i^{(k)} - \alpha \sum_{j=i+1}^n a_{ij}x_j^{(k)} + \alpha b_i$$

ou seja

$$(\mathbf{D} - \alpha\mathbf{L}) \mathbf{x}^{(k)} = [(1 - \alpha)\mathbf{D} - \alpha\mathbf{U}] \mathbf{x}^{(k)} + \alpha \mathbf{b}$$

Ou

$$\mathbf{x}^{(k)} = (\mathbf{D} - \alpha\mathbf{L})^{-1} [(1 - \alpha)\mathbf{D} - \alpha\mathbf{U}] \mathbf{x}^{(k)} + \alpha (\mathbf{D} - \alpha\mathbf{L})^{-1} \mathbf{b}$$

Comparando com $\mathbf{x}^{(k+1)} = \mathbf{T}\mathbf{x}^{(k)} + \mathbf{c}$, obtêm-se as matrizes para o método da relaxação:

$$\mathbf{T} = (\mathbf{D} - \alpha\mathbf{L})^{-1} [(1 - \alpha)\mathbf{D} - \alpha\mathbf{U}] \quad \text{e} \quad \mathbf{c} = \alpha (\mathbf{D} - \alpha\mathbf{L})^{-1} \mathbf{b}.$$



PDEs elípticas

PDEs elípticas descrevem situações físicas estacionárias ou de equilíbrio.

Exemplos:

- Equação de Laplace:

$$\nabla^2 \phi = 0$$

- Equação de Poisson

$$\nabla^2 \phi = f$$

- Equação de Helmholtz:

$$\nabla^2 \phi + k^2 \phi = 0$$



Tipo de condições fronteira

As PDEs elípticas ocorrem como problemas de valor fronteira, sendo estas condições fronteira de 3 tipos:

Condições fronteira de Dirichlet – quando ϕ é conhecido na fronteira.

Condições fronteira de Neumann – quando se conhece a derivada normal de ϕ na fronteira.

Condições fronteira mistas - quando se tem uma mistura das condições fronteira dos dois tipos anteriores.



Discretização da equação de Poisson

Considere-se a equação de Poisson para o potencial elétrico num domínio bidimensional, $V \equiv V(x, y)$. Escrevendo o Laplaciano por extenso:

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

O problema é discretizado numa grelha numérica de M_x pontos segundo x , espaçados por Δx , e M_y pontos segundo y , espaçados por Δy .



Discretização da equação de Poisson

O potencial em (x_i, y_j) é aproximado por $V(i, j)$. Usando diferenças finitas centradas para aproximar as segundas derivadas, obtém-se:

$$\frac{\partial^2 V(i, j)}{\partial x^2} \approx \frac{V(i + 1, j) - 2V(i, j) + V(i - 1, j)}{(\Delta x)^2}$$

$$\frac{\partial^2 V(i, j)}{\partial y^2} \approx \frac{V(i, j + 1) - 2V(i, j) + V(i, j - 1)}{(\Delta y)^2}$$

Se $\Delta x = \Delta y = h$, a equação para $V(i, j)$ nos pontos interiores fica:

$$-4V(i, j) + V(i + 1, j) + V(i - 1, j) + V(i, j + 1) + V(i, j - 1) = h^2 f(i, j) \quad (9)$$



Discretização da equação de Poisson — Métodos iterativos

O método de Jacobi, por exemplo, escreve-se simplesmente como,

$$V^{(k+1)}(i,j) = \frac{1}{4} [V^{(k)}(i+1,j) + V^{(k)}(i-1,j) + V^{(k)}(i,j+1) + V^{(k)}(i,j-1) - h^2 f(i,j)]$$

com as devidas alterações nos pontos fronteira.

Os algoritmos dos slides 6, 12 e 15 podem ser facilmente adaptados para a resolução da equação de Poisson.



Método de Jacobi para a equação de Poisson — algoritmo

1. Escolhem-se valores iniciais para V_{old}
2. Para todos os pontos que não pertencem à fronteira faz-se

$$V_{new}(i,j) = \frac{1}{4} [V_{old}(i+1,j) + V_{old}(i-1,j) + V_{old}(i,j+1) + V_{old}(i,j-1) - h^2 f(i,j)]$$

3. Calcula-se a matriz $\|V_{new} - V_{old}\|$
4. Quando a média ou o valor máximo da matriz acima é maior que uma dada tolerância, faz-se $V_{old} = V_{new}$ e volta-se ao ponto 2.
5. Quando o valor referido acima é menor que a tolerância, considera-se que o método convergiu. A solução numérica é V_{new} .



Método de Gauss-Seidel para a equação de Poisson — algoritmo

1. Escolhem-se valores iniciais para V_{old}
2. Faz-se $V_{new} = V_{old}$.
3. Para todos os pontos que não pertencem à fronteira faz-se

$$V_{new}(i,j) = \frac{1}{4} [V_{new}(i+1,j) + V_{new}(i-1,j) + V_{new}(i,j+1) + V_{new}(i,j-1) - h^2 f(i,j)]$$

4. Calcula-se a matriz $\|V_{new} - V_{old}\|$
5. Quando a média ou o valor máximo da matriz acima é maior que uma dada tolerância, faz-se $V_{old} = V_{new}$ e volta-se ao ponto 3.
6. Quando o valor referido acima é menor que a tolerância, considera-se que o método convergiu. A solução numérica é V_{new} .



Método da sobre-relaxação sucessiva para a equação de Poisson — algoritmo

1. Escolhem-se valores iniciais para V_{old}
2. Faz-se $V_{new} = V_{old}$.
3. Para todos os pontos que não pertencem à fronteira faz-se

$$V_{new}(i, j) = (1 - \alpha)V_{old}(i, j) + \frac{\alpha}{4} [V_{new}(i + 1, j) + V_{new}(i - 1, j) + V_{new}(i, j + 1) + V_{new}(i, j - 1) - h^2 f(i, j)]$$
4. Calcula-se a matriz $\|V_{new} - V_{old}\|$
5. Quando a média ou o valor máximo da matriz acima é maior que uma dada tolerância, faz-se $V_{old} = V_{new}$ e volta-se ao ponto 3.
6. Quando o valor referido acima é menor que a tolerância, considera-se que o método convergiu. A solução numérica é V_{new} .



Convergência do método de Jacobi para a equação de Poisson

A matriz A do exemplo, com a sua estrutura de blocos, pode ser facilmente generalizada para outros valores de M_x e de M_y (diferentes ou não de M_x). As matrizes T de cada método podem ser escritas e os seus valores próprios e raios espectrais determinados.

Os valores próprios da matriz T no caso do método de Jacobi aplicado à equação de Poisson a duas dimensões, num domínio retangular são:

$$\lambda_{pq} = \frac{1}{2} \left[\cos \frac{p\pi}{M_x - 1} + \cos \frac{q\pi}{M_y - 1} \right]$$

Com $p = 1, 2, \dots, M_x-2$, e $q = 1, 2, \dots, M_y-2$

Os valores absolutos destes valores próprios são sempre menores que 1.



Convergência do método de Jacobi para a equação de Poisson

Os λ máximos ocorrem para $p = q = 1$. Para M_x e M_y grandes, vem que

$$\rho = |\lambda|_{max} \simeq 1 - \frac{1}{4} \left(\frac{\pi^2}{M_x^2} + \frac{\pi^2}{M_y^2} \right)$$

que é muito próximo de 1. Para $M = M_x = M_y$,

$$\rho \simeq 1 - \frac{1}{2} \frac{\pi^2}{M^2}$$

e a taxa de convergência é aproximadamente proporcional a M^{-2} .

O número de iterações necessárias para um mesmo nível de convergência é então proporcional a M^2 . Como o número de pontos da grelha também aumenta com M^2 , o tempo de simulação aumenta com M^4 .



Convergência do método de Gauss–Seidel aplicado à equação de Poisson

Os valores próprios são agora

$$\lambda_{pq} = \frac{1}{2} \left[\cos \frac{p\pi}{M_x - 1} + \cos \frac{q\pi}{M_y - 1} \right]^2$$

Com $p = 1, 2, \dots, M_x - 2$, e $q = 1, 2, \dots, M_y - 2$

Usando os mesmos argumentos aplicados ao método de Jacobi, vem

$$\rho = |\lambda|_{\max} \simeq 1 - \frac{1}{2} \left(\frac{\pi^2}{M_x^2} + \frac{\pi^2}{M_y^2} \right)$$

Para $M = M_x = M_y$,

$$\rho \simeq 1 - \frac{\pi^2}{M^2}$$



Convergência do método de Gauss–Seidel aplicado à equação de Poisson

A taxa de convergência é agora aproximadamente o dobro da do método de Jacobi:

$$r \simeq \frac{1 - \rho}{\ln 10}$$

$$r \simeq \frac{\pi^2}{M^2 \ln 10}$$

O número de iterações necessárias para um mesmo nível de convergência continua a ser proporcional a M^2 . O tempo de simulação continua a aumentar com M^4 , no entanto o método de Gauss–Seidel é de convergência mais rápida, requerendo metade das iterações necessárias para o método de Jacobi para convergir dentro da mesma tolerância.

12-05-2020



Método de sobre-relaxação sucessiva aplicado à equação de Poisson

Como o método de Gauss-Seidel aplicado à equação de Poisson é convergente, é comum usar um método de sobre-relaxação sucessiva.

O valor de α que minimiza o número de passos necessários para atingir a convergência, α_{opt} , depende do tipo de problema, do número de dimensões do problema e do tipo de condições fronteira.

Para a equação de Poisson num domínio retangular, pode-se mostrar que

$$\alpha_{opt} \simeq 2 - \frac{2\pi}{M}$$

O raio espectral para α_{opt} é dado por

$$\rho_{opt} \simeq 1 - \frac{2\pi}{M}$$

Quando se usa α_{opt} , o número de iterações necessárias para um mesmo nível de convergência passa a ser proporcional a M . O tempo de simulação aumenta com M^3 .

