

Métodos iterativos ou de relaxação

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

$$A\mathbf{x} = \mathbf{b}.$$

Um método iterativo converte o sistema num outro equivalente, da forma

$$\mathbf{x} = T\mathbf{x} + \mathbf{c},$$

usa uma primeira estimativa $\mathbf{x}^{(0)}$ e calcula uma sequência de soluções aproximadas, dadas por

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c},$$

parando quando soluções consecutivas já são tão próximas quanto se queira ou quando os resíduos são tão pequenos quanto se queira.

Método de Jacobi - exemplo

Considere o sistema

$$\begin{array}{rclclcl}
 10x_1 & -x_2 & +2x_3 & & = & 6 \\
 -x_1 & +11x_2 & -x_3 & +3x_4 & = & 25 \\
 2x_1 & -x_2 & +10x_3 & -x_4 & = & -11 \\
 & 3x_2 & -x_3 & +8x_4 & = & 15
 \end{array}$$

é equivalente a

$$\begin{array}{rclclcl}
 x_1 & = & & \frac{1}{10}x_2 & -\frac{1}{5}x_3 & +\frac{3}{5} \\
 x_2 & = & \frac{1}{11}x_1 & & +\frac{1}{11}x_3 & -\frac{3}{11}x_4 +\frac{25}{11} \\
 x_3 & = & -\frac{1}{5}x_1 & +\frac{1}{10}x_2 & & +\frac{1}{10}x_4 -\frac{11}{10} \\
 x_4 & = & & -\frac{3}{8}x_2 & +\frac{1}{8}x_3 & +\frac{15}{8}
 \end{array}$$

Método de Jacobi - exemplo

A matriz T e o vetor c são

$$T = \begin{bmatrix} 0 & \frac{1}{10} & -\frac{1}{5} & 0 \\ \frac{1}{11} & 0 & \frac{1}{11} & -\frac{3}{11} \\ -\frac{1}{5} & \frac{1}{10} & 0 & \frac{1}{10} \\ 0 & -\frac{3}{8} & \frac{1}{8} & 0 \end{bmatrix}, \quad c = \begin{bmatrix} \frac{3}{5} \\ \frac{25}{11} \\ -\frac{11}{10} \\ \frac{15}{8} \end{bmatrix}$$

Pode-se iniciar o método iterativo com a estimativa $\mathbf{x}^{(0)} = (0 \ 0 \ 0 \ 0)^T$ e obter as seguintes usando

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$$

Método de Jacobi - matrizes gerais

$$A = D - L - U$$

- D é uma matriz diagonal cuja diagonal é a diagonal principal de A .
- $-L$ é uma matriz que só tem elementos não nulos abaixo da diagonal principal (lower) e iguais aos da matriz A .
- $-U$ é uma matriz que só tem elementos não nulos acima da diagonal principal (upper) e iguais aos da matriz A .

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

Matrizes do método de Jacobi

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

Método de Jacobi - algoritmo

- 1 Define-se a matriz A cujos elementos são a_{ij} e o vetor \mathbf{b} .
- 2 Escolhem-se valores iniciais para $x_{\text{old}}(i)$

3

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

- 4 Calcula-se

$$\text{dif} = \|\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}}\| \quad \text{ou} \quad \text{dif} = \|\mathbf{A}\mathbf{x}_{\text{new}} - \mathbf{b}\|$$

- 5 Se $\text{dif} > \text{tolerancia}$, faz-se $x_{\text{old}}(i) = x_{\text{new}}(i)$ e volta-se ao ponto 3.
- 6 Se $\text{dif} < \text{tolerancia}$, convergiu. A solução numérica é $x_{\text{new}}(i)$.

Método de Gauss–Seidel

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

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

Método de Gauss–Seidel

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

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

Método de Gauss–Seidel - algoritmo

- 1 Define-se a matriz A cujos elementos são a_{ij} e o vetor \mathbf{b} .
- 2 Escolhem-se valores iniciais para $x_{\text{old}}(i)$
- 3 Para todos os pontos, faz-se $x_{\text{new}}(i) = x_{\text{old}}(i)$

4

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

- 5 Calcula-se

$$\text{dif} = \|\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}}\| \quad \text{ou} \quad \text{dif} = \|\mathbf{A}\mathbf{x}_{\text{new}} - \mathbf{b}\|$$

- 6 Se $\text{dif} > \text{tolerancia}$, faz-se $x_{\text{old}}(i) = x_{\text{new}}(i)$ e volta-se ao ponto 3.
- 7 Se $\text{dif} < \text{tolerancia}$, convergiu. A solução numérica é $x_{\text{new}}(i)$.

Convergência dos métodos iterativos

O método deve convergir, ou seja

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

Seja \mathbf{e}_k o erro na iteração k

$$\mathbf{e}_k = \mathbf{x} - \mathbf{x}^{(k)}$$

Subtraindo as equações $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ e $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$, obtemos

$$\mathbf{e}_k = T\mathbf{e}_{k-1}$$

que resulta em

$$\mathbf{e}_k = T^k \mathbf{e}_0$$

Logo para que o erro tenda para zero quando k tende para infinito é necessário que $T^k \mathbf{e}_0$ tenda para zero.

Convergência dos métodos iterativos

Existe um resultado que nos garante que $\lim_{k \rightarrow \infty} A^k \mathbf{x} = 0$.

Seja $\rho(A)$ o raio espectral da matriz A , definido como

$$\rho(A) = |\lambda_i|_{\max},$$

onde λ_i são os valores próprios da matriz A .

Se $\rho(A) < 1$ então $\lim_{k \rightarrow \infty} A^k \mathbf{x} = 0$.

Os métodos iterativos estudados acima são convergentes se

$$\rho(T) < 1$$

- Os métodos de Jacobi e de Gauss-Seidel são convergentes ou não dependendo do caso. No entanto, existem teoremas que permitem saber a convergência de um ou outro pelas características da matriz A .
- É desejável escolher um método cujo $\rho(T) < 1$ seja mínimo.

Métodos de sub- ou sobre-relaxação

Métodos de sub ou sobre-relaxação são métodos que permitem desacelerar ou acelerar a convergência do método de Gauss-Seidel.

- desacelerar quando o Gauss-Seidel não converge;
- acelerar quando o Gauss-Seidel já converge.

Seja \mathbf{d} a diferença entre duas soluções aproximadas consecutivas

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

Se pretendemos que a variação entre as duas soluções seja diferente poderemos escrever genericamente

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

- $\alpha < 1$ - desaceleramos relativamente ao método de Gauss-Seidel;
- $\alpha > 1$ - aceleramos relativamente ao método de Gauss-Seidel - denominado **método de sobre-relaxação sucessiva**;

Note que $\alpha = 1$ corresponde ao método de Gauss-Seidel.

Métodos de sub- ou sobre-relaxação

A ideia é usar o Gauss–Seidel

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

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

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

Para as componentes fica

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

Métodos de sub- ou sobre-relaxação - algoritmo

- 1 Define-se a matriz A cujos elementos são a_{ij} e o vetor \mathbf{b} .
- 2 Escolhem-se valores iniciais para $x_{\text{old}}(i)$
- 3 Para todos os pontos, faz-se $x_{\text{new}}(i) = x_{\text{old}}(i)$

4

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

- 5 Calcula-se

$$\text{dif} = \|\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}}\| \quad \text{ou} \quad \text{dif} = \|\mathbf{A}\mathbf{x}_{\text{new}} - \mathbf{b}\|$$

- 6 Se $\text{dif} > \text{tolerancia}$, faz-se $x_{\text{old}}(i) = x_{\text{new}}(i)$ e volta-se ao ponto 3.
- 7 Se $\text{dif} > \text{tolerancia}$, convergiu. A solução numérica é $x_{\text{new}}(i)$.

Métodos de sub- ou sobre-relaxação

Para avaliar a convergência e taxa de convergência temos que determinar o raio espectral da matriz T .

Neste caso

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

ou seja

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

ou

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

Matriz T do método de relaxação

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

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:

- 1 **Condições fronteira de Dirichlet** – quando ϕ é conhecido na fronteira.
- 2 **Condições fronteira de Neumann** – quando conhecemos a derivada normal de ϕ na fronteira.
- 3 **Condições fronteira mistas** - quando temos uma mistura das condições fronteira dos dois tipos anteriores.

Discretização da equação de Poisson

Consideremos 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 o x , espaçados por Δx , e M_y pontos segundo o 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, obtemos:

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

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

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

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

Método de Jacobi para a equação de Poisson

A discretização por diferenças finitas transforma a equação de Poisson num sistema de equações algébricas lineares

$$A\mathbf{V} = \mathbf{b}$$

que pode ser resolvido por métodos diretos ou iterativos. Em FC vamos resolvê-la por métodos iterativos.

Método de Jacobi

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

com as devidas alterações nos pontos fronteira.

Método de Jacobi para a equação de Poisson

No método de Jacobi aplicado à equação de Poisson:

- a matriz $T = D^{-1}(L + U) = \frac{1}{4}(L + U)$;
- o vetor $c = -\frac{1}{4}h^2 f(i, j)$.

Pode-se calcular os valores próprios da matriz T :

$$\lambda_{mn} = \frac{1}{2} \left[\cos \frac{m\pi}{M_x} + \cos \frac{n\pi}{M_y} \right], \quad m = 1, 2, \dots, M_x - 2, \quad n = 1, 2, \dots, M_y - 2$$

que cujos valores absolutos são sempre menores que 1. Os λ 's máximos são para $m = n = 1$. Para M_x e M_y grandes o λ máximo é aproximado por

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

que é muito próximo de 1. Logo a convergência é muito lenta.

Método de Jacobi – algoritmo

No caso da **equação de Poisson**, o algoritmo é da forma:

- 1 Escolhem-se valores iniciais para $V_{\text{old}}(i, j)$
- 2 Para todos os pontos (que não pertencem às fronteiras), faz-se

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

- 3 Calcula-se

$$\text{dif} = \|\mathbf{V}_{\text{new}} - \mathbf{V}_{\text{old}}\| \quad \text{ou} \quad \text{dif} = \|\mathbf{A}\mathbf{V}_{\text{new}} - \mathbf{b}\|$$

- 4 Se $\text{dif} > \text{tolerancia}$, faz-se, para todos os pontos, $V_{\text{old}} = V_{\text{new}}$ e volta-se ao ponto 2.
- 5 Se $\text{dif} < \text{tolerancia}$, convergiu. A solução numérica é $V_{\text{new}}(i, j)$.

Método de Jacobi para a equação de Poisson

Outros aspetos a ter em conta em relação ao método de Jacobi:

- Tem que se ter muito cuidado para não alterar os valores do potencial nas fronteiras.
- Analisando o caso $M_x = M_y = \dots = M$, o número de iterações necessárias para um mesmo nível de convergência, a d dimensões, varia com M^d . Como o número de pontos da grelha também aumenta com M^d , o tempo de simulação aumenta com M^{2d} .

O número de iterações pode ser reduzido por um fator de dois, usando o método de Gauss–Seidel.

Método de Gauss–Seidel aplicado à equação de Poisson

Método de Gauss–Seidel

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

Como analisado acima $T = (-4I - L)^{-1}U$, onde I é a matriz identidade, cujos valores próprios são

$$\lambda_{mn} = \frac{1}{4} \left[\cos \frac{m\pi}{M_x} + \cos \frac{n\pi}{M_y} \right]^2$$

$$m = 1, 2, \dots, M_x - 2, n = 1, 2, \dots, M_y - 2.$$

Método de Gauss–Seidel

O λ máximo pode ser aproximado, neste caso, por

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

Logo o método de Gauss-Seidel é de convergência mais rápida e requer metade das iterações do método de Jacobi para convergir dentro da mesma tolerância.

Método de Gauss–Seidel — Algoritmo

O algoritmo para a **equação de Poisson**:

- ➊ Escolhem-se valores iniciais para $V_{\text{old}}(i, j)$
- ➋ Para todos os pontos, faz-se $V_{\text{new}}(i, j) = V_{\text{old}}(i, j)$
- ➌ Para todos os pontos (que não pertencem às fronteiras):

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

- ➍ Calcula-se

$$\text{dif} = \|\mathbf{V}_{\text{new}} - \mathbf{V}_{\text{old}}\| \quad \text{ou} \quad \text{dif} = \|\mathbf{A}\mathbf{V}_{\text{new}} - \mathbf{b}\|$$

- ➎ Se $\text{dif} > \text{tolerancia}$, faz-se, para todos os pontos, $V_{\text{old}} = V_{\text{new}}$ e volta-se ao ponto 3.
- ➏ Se $\text{dif} < \text{tolerancia}$, convergiu. A solução numérica é $V_{\text{new}}(i, j)$.

Método de sobre-relaxação sucessiva

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 depende do número de dimensões do problema e do tipo de condições fronteira.

Com a melhor escolha de α , consegue-se reduzir o número de relaxações necessárias por um fator de M em relação ao método de Jacobi.

Método de sobre-relaxação sucessiva — Algoritmo

- 1 Escolhem-se valores iniciais para $V_{\text{old}}(i, j)$
- 2 Para todos os pontos, faz-se $V_{\text{new}}(i, j) = V_{\text{old}}(i, j)$
- 3 Para todos os pontos (que não pertencem às fronteiras):

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

O novos valores são calculados fazendo médias com uma mistura de valores velhos e novos.

- 4 Calcula-se

$$\text{dif} = \|\mathbf{V}_{\text{new}} - \mathbf{V}_{\text{old}}\| \quad \text{ou} \quad \text{dif} = \|\mathbf{A}\mathbf{V}_{\text{new}} - \mathbf{b}\|$$

- 5 Se $\text{dif} > \text{tolerancia}$, faz-se, para todos os pontos, $V_{\text{old}} = V_{\text{new}}$ e volta-se ao ponto 3.
- 6 Se $\text{dif} < \text{tolerancia}$, convergiu. A solução numérica é $V_{\text{new}}(i, j)$.

Escolha de α no método de sobre-relaxação

Pode provar-se que o valor próprio máximo, em valor absoluto, da matriz T é mínimo para

$$\alpha_{\text{opt.}} = \frac{2}{1 + \sqrt{1 - (\lambda_{\text{max}}^{\text{Jacobi}})^2}}$$

Usando o valor encontrado atrás

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

e $M_x = M_y = M$, obtemos

$$\alpha_{\text{opt}} \simeq \frac{2}{1 + \pi/M}$$