

# Fundamentos de Modelagem Computacional

## Trabalho do Módulo Técnicas Computacionais

Grazione de Souza

PPGMC/IPRJ/UERJ

2021/2

A iteração de Jacobi é um dos métodos iterativos mais antigos (a sua revisão é importante em termos dos fundamentos dos métodos iterativos).

Considera-se um sistema de equações na forma

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + \dots + a_{1n}x_n = d_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + \dots + a_{2n}x_n = d_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 + \dots + a_{3n}x_n = d_3$$

$$\vdots + \quad \quad \vdots + \quad \quad \vdots + \quad \dots \quad \quad \vdots = \vdots$$

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + a_{n4}x_4 + \dots + a_{nn}x_n = d_n \quad (1)$$

onde  $a_{ij} \neq 0$  para  $i = 1, 2, 3, \dots, n$ .

Reorganizando o sistema,

$$x_1 = \frac{d_1 - a_{12}x_2 - a_{13}x_3 - a_{14}x_4 - \dots - a_{1n}x_n}{a_{11}},$$

$$x_2 = \frac{d_2 - a_{21}x_1 - a_{23}x_3 - a_{24}x_4 - \dots - a_{2n}x_n}{a_{22}},$$

$$x_3 = \frac{d_3 - a_{31}x_1 - a_{32}x_2 - a_{34}x_4 - \dots - a_{3n}x_n}{a_{33}},$$

$$\vdots + \quad \quad \vdots + \quad \quad \vdots + \quad \dots \quad \quad \vdots = \vdots$$

$$x_n = \frac{d_n - a_{n1}x_1 - a_{n2}x_2 - a_{n3}x_3 - \dots - a_{n,n-1}x_{n-1}}{a_{nn}}. \quad (2)$$

Denotando uma iteração nova por  $(k + 1)$  e uma iteração antiga por  $(k)$ ,

$$x_1^{(k+1)} = \frac{d_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - a_{14}x_4^{(k)} - \dots - a_{1n}x_n^{(k)}}{a_{11}},$$

$$x_2^{(k+1)} = \frac{d_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - a_{24}x_4^{(k)} - \dots - a_{2n}x_n^{(k)}}{a_{22}},$$

$$x_3^{(k)} = \frac{d_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)} - a_{34}x_4^{(k)} - \dots - a_{3n}x_n^{(k)}}{a_{33}},$$

$$\vdots + \quad \quad \vdots + \quad \quad \vdots + \quad \dots \quad \quad \vdots = \vdots$$

$$x_n^{(k+1)} = \frac{d_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - a_{n3}x_3^{(k)} - \dots - a_{n,n-1}x_{n-1}^{(k)}}{a_{nn}}. \quad (3)$$

O algoritmo segue

$$x_i^{(k+1)} = \frac{d_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(k)}}{a_{ii}}. \quad (4)$$

para  $i = 1, 2, \dots, n$ .

Ao fim de uma iteração, testa-se a convergência usando, por exemplo,

$$|x_i^{(k+1)} - x_i^{(k)}| \leq \epsilon$$

para  $i = 1, 2, \dots, n$ , onde  $\epsilon$  é uma tolerância numérica. Se o teste de convergência é satisfeito, as iterações são interrompidas e tem a solução para o passo de tempo considerado.

Se o teste não é satisfeito, valores  $x_1, x_2, \dots, x_n$  na iteração antiga, ( $k$ ), são atualizados pelos valores calculados na iteração nova, ( $k+1$ ), e o processo recomeça.

A iteração de Gauss-Seidel é uma modificação da iteração de Jacobi que utiliza a informação mais recente disponível, de forma que

$$x_1^{(k+1)} = \frac{d_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - a_{14}x_4^{(k)} - \dots - a_{1n}x_n^{(k)}}{a_{11}},$$

$$x_2^{(k+1)} = \frac{d_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - a_{24}x_4^{(k)} - \dots - a_{2n}x_n^{(k)}}{a_{22}},$$

$$x_3^{(k)} = \frac{d_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} - a_{34}x_4^{(k)} - \dots - a_{3n}x_n^{(k)}}{a_{33}},$$

$$\vdots + \quad \quad \vdots + \quad \quad \vdots + \quad \dots \quad \quad \vdots = \vdots$$

$$x_n^{(k+1)} = \frac{d_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - a_{n3}x_3^{(k+1)} - \dots - a_{n,n-1}x_{n-1}^{(k+1)}}{a_{nn}}.$$

(5)

Como a iteração de Gauss-Seidel usa informações mais recentes, ele geralmente converge em um número menor de iterações do que a iteração de Jacobi. A iteração de Gauss-Seidel requer menor armazenamento em relação a Jacobi.

A iteração de Gauss-Seidel converge se

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}| \quad (6)$$

O algoritmo segue

$$x_i^{(k+1)} = \frac{d_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)}}{a_{ii}} \quad (7)$$

para  $i = 1, 2, \dots, n$ .

Ao fim de uma iteração, testa-se a convergência usando, por exemplo,

$$|x_i^{(k+1)} - x_i^{(k)}| \leq \epsilon$$

para  $i = 1, 2, \dots, n$ , onde  $\epsilon$  é uma tolerância numérica. O teste de convergência segue o raciocínio discutido para a iteração de Jacobi.



Por outro lado, nos métodos do tipo SOR (*Successive Overrelaxation*), objetiva-se acelerar a taxa de convergência utilizando-se um fator de relaxação,  $\omega$ , como no caso do PSOR (*Point Successive Overrelaxation*),

$$x_i^{(k+1)} = \omega \frac{d_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)}}{a_{ii}} - (\omega - 1) x_i^{(k)} \quad (8)$$

onde  $\omega$  está entre 1 e 2, sendo que existem técnicas para se calcular o  $\omega$  ótimo que mais acelera a convergência, como discutido por Ertekin et al. (2002).