

# Relaxamento

Paulo Ricardo Lisboa de Almeida

2021

## Conteúdo da Aula

- Relaxamento
- Comparação entre métodos

## 1 Relaxamento

Lembrando do método de Gauss-Seidel, onde a função de iteração é:

$$\begin{cases} x_1^{(k+1)} = \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \dots - a_{1n}x_n^{(k)}) \\ x_2^{(k+1)} = \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)}) \\ x_3^{(k+1)} = \frac{1}{a_{33}}(b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} - a_{34}x_4^{(k)} - \dots - a_{3n}x_n^{(k)}) \\ \vdots \\ x_n^{(k+1)} = \frac{1}{a_{nn}}(b_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \dots - a_{n,n-1}x_{n-1}^{(k+1)}) \end{cases}$$

Podemos modificar o método de Gauss-Seidel a fim de melhorar sua convergência através do método de Relaxamento. Para isso, após calcular cada  $x_i$ , um  $x_{i(r)}$ , onde o  $(r)$  indica que é a versão relaxada de  $x_i$ , é computado como

$$x_{i(r)}^{(k+1)} = \lambda x_i^{(k+1)} + (1 - \lambda)x_{i(r)}^{(k)}$$

onde  $\lambda$  é um fator de peso tal que  $0 < \lambda < 2$ . Os valores relaxados são uma ponderação entre o novo valor calculado, e o valor anterior.

- Se  $\lambda = 1$ , o método é o mesmo de Gauss-Seidel.
- Se  $\lambda < 1$ , é realizada uma média ponderada entre o novo valor e o valor antigo.
  - Nesse caso, o método é comumente chamado de *sub-relaxamento*.
  - Pode melhorar a convergência do método, ou fazer um sistema que não estava convergindo convergir.
- Se  $\lambda > 1$ , um peso maior é dado para o valor mais novo.
  - Nesse caso, o método é comumente chamado de *sobrerrelaxamento simultâneo* – *Successive Over-Relaxation* ou *SOR*.

- Assumimos que estamos “indo na direção correta”, e desejamos aumentar a velocidade da convergência.

**Exemplo 1:**

Resolver o seguinte sistema utilizando sobre-relaxamento onde  $\lambda = 1.2$ . Considerar o erro relativo máximo  $\varepsilon = 0.1$  e  $x^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ .

$$\begin{cases} -3x_1 + 12x_2 = 9 \\ 10x_1 - 2x_2 = 8 \end{cases}$$

Função de iteração:

$$\begin{cases} x_1^{(k+1)} = \frac{8 + 2x_2}{10} \\ x_2^{(k+1)} = \frac{9 + 3x_1}{12} \end{cases}$$

Iteração  $k = 1$

Calcular  $x_1^{(1)}$

$$x_1^{(1)} = \frac{8 + 2 * 0}{10} = 0.8$$

Computando a versão relaxada de  $x_1$ :

$$x_{1(r)}^{(1)} = 1.2x_1^{(1)} + (1 - 1.2)x_1^{(0)} = 1.2 * 0.8 + (-0.2) * 0 = \mathbf{0.96}$$

Calcular  $x_2^{(1)}$

$$x_2^{(1)} = \frac{9 + 3 * \mathbf{0.96}}{12} = 0.99$$

Computando a versão relaxada de  $x_2$ :

$$x_{2(r)}^{(1)} = 1.2x_2^{(1)} + (1 - 1.2)x_2^{(0)} = 1.2 * 0.99 + (-0.2) * 0 = 1.188$$

Observe que durante os cálculos, a **versão relaxada mais recente** é utilizada (exemplo destacado em azul).

Temos então:

$$x^{(1)} = \begin{pmatrix} \mathbf{0.96} \\ \mathbf{1.188} \end{pmatrix}$$

Erro relativo:

$$\begin{aligned} |x_1^{(1)} - x_1^{(0)}| &= 0.96 \\ |x_2^{(1)} - x_2^{(0)}| &= 1.188 \end{aligned}$$

$$d^{(1)} = 1.188$$

$$d_r^{(1)} = \frac{1.188}{1.188} = 1 > \varepsilon$$

Iteração  $k = 2$

$$x_1^{(2)} = \frac{8 + 2 * 1.188}{10} = 1.0376$$

$$x_{1(r)}^{(2)} = 1.2x_1^{(2)} + (1 - 1.2)x_{1(r)}^{(1)} = 1.2 * 1.0376 + (-0.2) * 0.96 = 1.05312$$

$$x_2^{(2)} = \frac{9 + 3 * 1.05312}{12} = 1.01328$$

$$x_{2(r)}^{(2)} = 1.2x_2^{(2)} + (1 - 1.2)x_{2(r)}^{(1)} = 1.2 * 1.01328 + (-0.2) * 1.188 = 0.978336$$

Então:

$$x^{(2)} = \begin{pmatrix} 1.05312 \\ 0.978336 \end{pmatrix}$$

Erro relativo:

$$|x_1^{(2)} - x_1^{(1)}| = 0.09312$$

$$|x_2^{(2)} - x_2^{(1)}| = 0.209664$$

$$d^{(1)} = 0.209664$$

$$d_r^{(1)} = \frac{0.209664}{1.05312} = 0.1990884 > \varepsilon$$

Iteração k = 3

$$x_1^{(3)} = \frac{8 + 2 * 0.978336}{10} = 0.9956672$$

$$x_{1(r)}^{(3)} = 1.2x_1^{(3)} + (1 - 1.2)x_{1(r)}^{(2)} = 1.2 * 0.9956672 + (-0.2) * 1.05312 = 0.984177$$

$$x_2^{(3)} = \frac{9 + 3 * 0.984177}{12} = 0.996044$$

Computando a versão relaxada de  $x_2$ :

$$x_{2(r)}^{(3)} = 1.2x_2^{(3)} + (1 - 1.2)x_{2(r)}^{(2)} = 1.2 * 0.996044 + (-0.2) * 0.978336 = 0.999586$$

Então:

$$x^{(3)} = \begin{pmatrix} 0.984177 \\ 0.999586 \end{pmatrix}$$

Erro relativo:

$$|x_1^{(3)} - x_1^{(2)}| = 0.068943$$

$$|x_2^{(3)} - x_2^{(2)}| = 0.021250$$

$$d^{(1)} = 0.068943$$

$$d_r^{(1)} = \frac{0.068943}{0.999586} = 0.068972 < \varepsilon$$

Logo,  $\bar{x} = x^{(3)}$

**Problema do método do relaxamento:** o valor de  $\lambda$  depende do problema sendo tratado e é comumente encontrado de maneira empírica. Existem teoremas que podem ser usados para se selecionar  $\lambda$ , mas eles servem apenas para sistemas específicos. No geral, o método do relaxamento é utilizado quando um mesmo sistema precisa ser resolvido múltiplas vezes (com diferentes vetores  $b$  por exemplo), onde nesse caso pode valer a pena gastar recursos computacionais para se definir um bom  $\lambda$  empiricamente.

## 2 Comparação entre métodos

Os métodos diretos resolvem os problemas em uma quantidade finita de passos, e sempre convergem para a resposta quando ela existe, mas:

- Podem sofrer com problemas de representação numérica. Exemplo: ao se realizar operações de multiplicação ou divisão com valores muito próximos de zero.
  - Esses problemas podem ser mitigados através de técnicas de pivoteamento. Veja detalhes na literatura da disciplina.
- Muitos problemas podem conter matrizes esparsas (com muitos zeros). Os métodos diretos exigem que mesmo matrizes esparsas sejam representadas em suas totalidades nos nossos computadores.
  - Muitos problemas do mundo real podem conter milhões de equações.
  - Ocupam muita memória.

Já os métodos iterativos, nem sempre convergem para a resposta, mas possuem as seguintes vantagens:

- Pelo fato de serem iterativos, podem sofrer menos com os problemas de representação numérica.
- Preservam a esparsidade das matrizes, sendo interessantes para problemas com matrizes “grandes” e esparsas.

O método de Gauss-Seidel precisa em média da metade da quantidade de passos para se chegar na resposta com a precisão desejada, quando comparado com Gauss-Jacobi (FILHO, 2016).

No entanto, por precisar de  $x_{i-1}$  na iteração atual antes de poder calcular  $x_i$ , o método de Gauss-Seidel não é paralelizável. Dessa forma, utilizar o método de Gauss-Jacobi ou Gauss-Seidel pode depender se temos ou não recursos computacionais para se computar múltiplos  $x_i$  em paralelo.

### CURIOSIDADE

Em simulação de fluidos, os sistemas muitas vezes são gigantescos (exemplo: ocupam 128GiB de memória). A resolução desses sistemas geralmente é feita em paralelo utilizando-se, por exemplo, supercomputadores, com milhares de unidades de processamento.

Nesses casos, técnicas comuns para se resolver os sistemas incluem o método de Gauss-Jacobi, ou variantes do método de Gauss-Seidel que permitem paralelismo, como Red-Black Gauss-Seidel.

Veja um vídeo com o resultado de simulações:

<https://www.youtube.com/watch?v=p67-Qiad5zc>.

### 3 Exercícios

1) Resolva o seguinte sistema linear utilizando Gauss-Seidel e valor inicial  $x^{(0)} = [1 \ 1 \ 1]^T$ . Considere um erro relativo máximo  $\varepsilon < 3 \times 10^{-3}$  ou um máximo de 7 iterações.

a) Utilizando o método original ( $\lambda = 1$ ).

b) Utilizando o sobre-relaxamento com  $\lambda = 1.25$ .

$$\begin{cases} 4x_1 + 3x_2 = 24 \\ 3x_1 + 4x_2 - x_3 = 30 \\ -x_2 + 4x_3 = -24 \end{cases}$$

2) Resolva o seguinte sistema linear utilizando Gauss-Seidel e valor inicial  $x^{(0)} = [0 \ 0 \ 0]^T$ . Considere um erro relativo máximo  $\varepsilon < \times 10^{-4}$  ou um máximo de 10 iterações.

a) Utilizando o método original ( $\lambda = 1$ ).

b) Utilizando o sobre-relaxamento com  $\lambda = 1.06$ .

$$\begin{cases} 9x_1 + 4x_2 = 20 \\ 4x_1 + 9x_2 - x_3 = 12 \\ -x_2 + 9x_3 = 51 \end{cases}$$

### 4 Licença

Esta obra tem a licença [Creative Commons “Atribuição-CompartilhaIgual 4.0 Internacional”](https://creativecommons.org/licenses/by-sa/4.0/).



### Referências

CHAPRA, S. *Métodos Numéricos Aplicados com MATLAB® para Engenheiros e Cientistas* - 3.ed. [S.l.]: AMGH Editora, 2013. ISBN 9788580551778.

FILHO, A. *Fundamentos de Cálculo Numérico*. [S.l.]: Bookman Editora, 2016. ISBN 9788582603857.

PIRES, A. d. A. *Cálculo numérico: prática com algoritmos e planilhas*. [S.l.]: Editora Atlas, 2014. ISBN 9788522498826.

RUGGIERO, M.; LOPES, V. da R. *Cálculo numérico: aspectos teóricos e computacionais*. [S.l.]: Makron Books do Brasil, 1996.