Carlos Eduardo Amorim Gusmão

Implementação dos métodos numéricos Gauss-Seidel e SOR baseado em Gauss-Seidel

Belo Horizonte
28 de maio de 2025

Carlos Eduardo Amorim Gusmão

Implementação dos métodos numéricos Gauss-Seidel e SOR baseado em Gauss-Seidel

Universidade Federal de Minas Gerais
Bacharelado em Engenharia Mecânica

Orientador: Prof. Lucas Ribeiro Alves

Belo Horizonte
28 de maio de 2025

Introdução

O objetivo deste trabalho foi implementar e analisar o desempenho de dois métodos iterativos clássicos para a resolução de sistemas lineares: o método de Gauss-Seidel e o método de Relaxação Sucessiva (SOR - Successive Over-Relaxation). Esses métodos foram aplicados a sistemas lineares cujas soluções são conhecidas previamente, de modo a permitir a validação dos resultados obtidos numericamente. A implementação foi realizada na linguagem C++, com comentários explicativos ao longo do código para descrever cada etapa do processo computacional. Utilizou-se um mesmo critério de parada para ambos os métodos, baseado na precisão do erro relativo entre iterações consecutivas, assegurando uma comparação justa entre os algoritmos.

Durante a execução, foram armazenados os valores dos erros a cada iteração, o que possibilitou a análise da taxa de convergência de cada método. Os resultados foram organizados em gráficos que representam a evolução do erro em função do número de iterações, permitindo uma avaliação visual da eficiência de convergência. A comparação entre Gauss-Seidel e SOR evidencia as vantagens e limitações de cada abordagem, especialmente no que se refere à escolha do parâmetro de relaxação no método SOR.

Desenvolvimento

O código foi desenvolvido na linguagem C++ usando a programação modularizada, com uma função para cada etapa:

- Função para Gauss-Seidel
- Função para SOR-Gauss-Seidel
- Função principal main

O método de Gauss-Seidel tem como ideia principal isolar uma das variáveis e resolver normalmente a equação gerada pelo isolamento, usando a fórmula geral:

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

Para que esse método convirja, é necessário que o sistema seja uma matriz diagonal dominante, ou seja, os valores em que i=j, devem ser maiores que o módulo da soma dos outros valores da mesma linha. Para tanto, foi escolhido o seguinte sistema:

$$\begin{cases} 4x + -y = 15 \\ -x + 4y - z = 10 \\ -y + 3z = 10 \end{cases}$$

Em formato de matriz, é:

$$\left[\begin{array}{ccc|c} 4 & -1 & 0 & 15 \\ -1 & 4 & -1 & 10 \\ 0 & -1 & 3 & 10 \end{array}\right]$$

Diferentemente do método de Jacobi, Gauss-Seidel usa os valores imediatos encontrados ,em vez de usar valores pré definidos em toda iteração. Mas também possui chutes inciais. Aqui foram

$$x = 0; y = 0; z = 0$$

SOR baseado em Gauss-Seidel consiste em aprimorar Gauss-Seidel, usando um λ, cujo valor pode, ou não, contribuir para que convirja mais rápido, assim chegando a solução real de forma muito mais rápido se bem ajustado. Aqui foram feitos diversos testes para que fosse escolhido um valor aceitável para λ, chegando ao valor de 1,05. Esse método também tem uma fórmula geral e consiste em:

$$x_i^{\text{novo}} = \lambda x_i^{\text{novo}} + (1 - \lambda) x_i^{\text{velho}}$$

No código o cálculo da variação da solução entre iterações é realizado por meio da variável dx, que representa a norma da diferença entre os vetores solução de duas iterações consecutivas. Essa variação é usada como critério de parada do método: quando dx se torna menor que um valor previamente definido:

$$\varepsilon = 10^{-}6$$

Além disso, a cada iteração, é calculado o erro da solução atual em relação ao sistema original. Esse erro é obtido por:

$$||A\vec{x} - \vec{b}||$$

Feito isso, a cada iteração realizada, são impressas as soluções, o dx e o erro.

Análise dos dados gerados

A partir dos erros, foi gerada a tabela, usada posteriormente para gerar o gráfico.

Iteração	Erro (Gauss-Seidel)	Erro-SOR
1	5,64618	5,05988
2	1,40683	1,12791
3	0,205163	0,0120131
4	0,0299195	0,000644079
5	0,00436327	0,000130519
6	0,00063631	2,73e-06
7	9,28e-05	1,58e-07
8	1,35e-05	
9	1,97e-06	
10	2,88e-07	

Foi obtido o gráfico:

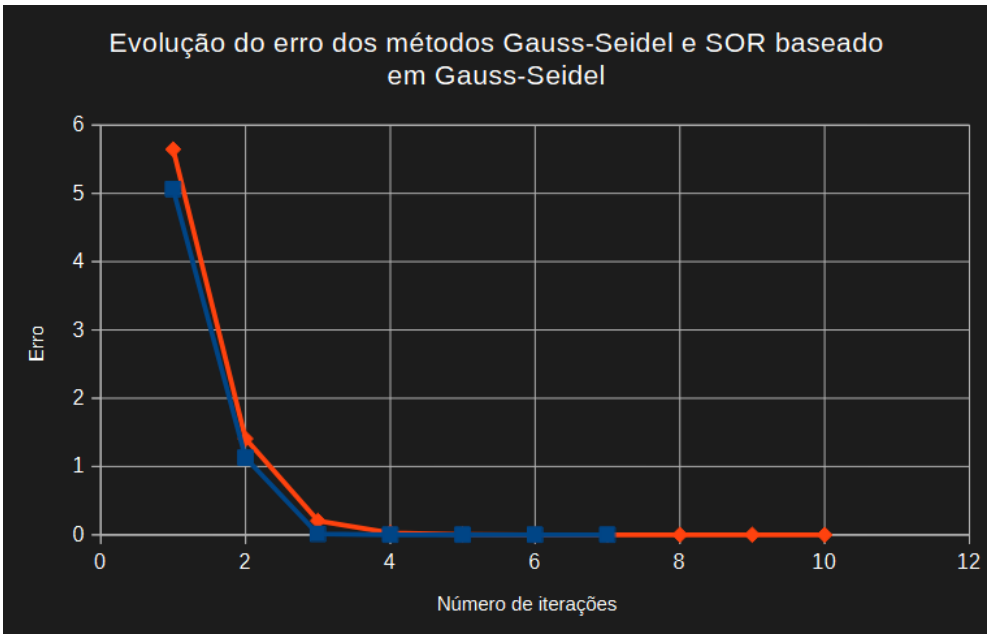

Figura 1 – Gráfico da evolução do erro

O gráfico apresenta a evolução do erro para os métodos Gauss-Seidel (representado pela linha vermelha com marcadores de losango) e SOR baseado em Gauss-Seidel (representado pela linha azul com marcadores de quadrado). O eixo horizontal mostra o "Número de iterações"e o eixo vertical representa o "Erro". Ambos os métodos começam com um erro inicial elevado. Isso é normal, pois as iterações iniciais são geralmente as que produzem as maiores reduções de erro. Nas primeiras 2 a 3 iterações, ambos os métodos mostram uma queda abrupta e significativa no erro. O erro diminui rapidamente após a primeira iteração e a partir da quarta iteração o método de SOR já possui um erro muito próximo de zero, enquando o Gauss-Seidel, está um pouco mais acima. Ao final, percebe-se que para o método de SOR foram necessárias menos iterações em comparação com o outro método.

Conclusão

O gráfico evidencia claramente a superioridade do método SOR sobre o método de Gauss-Seidel tradicional em termos de velocidade de convergência para o problema. Embora ambos os métodos sejam capazes de reduzir o erro e convergir para uma solução de alta precisão, o SOR atinge esse estado de convergência em um número muito menor de iterações. Essa diferença no desempenho entre os métodos ressalta a eficácia do fator de relaxamento (λ) ao método SOR na otimização do processo iterativo de Gauss-Seidel. Ao "super-relaxar"ou "sub-relaxar"as estimativas das variáveis a cada passo, o SOR consegue corrigir o erro de forma mais eficiente, evitando oscilações ou uma convergência muito lenta.

Portanto, a conclusão é que, para problemas com características semelhantes ao exemplificado, o método SOR representa uma escolha mais eficiente computacionalmente, pois permite alcançar a solução desejada com menor esforço, resultando em menor tempo de processamento e recursos computacionais. Essa é uma informação importante para a seleção de algoritmos em aplicações de engenharia e ciência que envolvem a solução numérica de grandes sistemas lineares.

Referências

CHAPRA, STEVEN C. **Métodos numéricos para engenharia**. Tradução: Helena Castro São Paulo, 2008. p.250-254, 27 mai. 2025.