UNIVERSIDADE FEDERAL DE SÃO CARLOS CAMPUS SOROCABA – CIÊNCIA DA COMPUTAÇÃO

RELATÓRIO TÉCNICO

Aplicação dos métodos implementados na resolução de sistemas lineares: Método de Eliminação de Gauss, Fatoração de Cholesky, Método Iterativo de Gauss-Jacobi e Método Iterativo de Gauss-Seidel

Aluno: Ana Beatriz Juvencio

Professora : Graciele P. Silveira

Disciplina: Cálculo Numérico

Data: 23 de junho de 2025

Sumário

1. Introdução

A resolução de sistemas lineares é um dos pilares fundamentais da Álgebra Linear com aplicações diretas em diversas áreas da ciência, engenharia e computação. No contexto do Cálculo Numérico, tais sistemas podem ser resolvidos por métodos diretos ou iterativos, a depender da estrutura da matriz dos coeficientes, do tamanho do sistema e da precisão exigida.

Este trabalho tem como objetivo implementar computacionalmente quatro métodos clássicos de resolução de sistemas lineares: Eliminação de Gauss, Fatoração de Cholesky, Gauss-Jacobi e Gauss-Seidel. Os algoritmos foram codificados em Python e aplicados a sistemas com diferentes características, com o propósito de analisar sua eficácia, precisão e convergência.

Além das implementações, buscou-se comparar as soluções numéricas com a solução analítica (quando disponível), identificando possíveis erros numéricos decorrentes da aritmética de ponto flutuante e discutindo a aplicabilidade de cada método de acordo com as propriedades do sistema.

2. Metodologia

Estudo teórico dos métodos numéricos: Foi realizada a leitura do livro "Cálculo Numérico: Aspectos Teóricos e Computacionais" de Ruggiero e Lopes, que serviu como base conceitual e algorítmica para todas as implementações.

Implementação computacional em Python: Os quatro métodos foram implementados manualmente, respeitando as estruturas dos algoritmos apresentados no livro, com uso da biblioteca NumPy para manipulação de matrizes e, no caso da Fatoração de Cholesky, também da scipy.linalg para substituições triangulares.

Criação de um sistema 3x3 com solução conhecida: Foi construído um sistema linear com propriedades ideais (simetria, dominância diagonal, definida positiva), de modo que todos os métodos fossem aplicáveis e pudessem ser validados com uma solução exata previamente conhecida.

Execução dos métodos e comparação dos resultados: Todos os métodos foram aplicados ao sistema proposto. Nos métodos iterativos, foi utilizado vetor inicial nulo, tolerância de 1e-6 e até 100 iterações. Foram registrados os resultados numéricos, número de iterações e comparações com a solução analítica.

Discussão dos resultados: Os métodos foram avaliados em termos de precisão, tempo de convergência (quando aplicável), sensibilidade ao sistema e aplicabilidade. Erros numéricos residuais foram discutidos brevemente com base em suas magnitudes.

4. Análise e comparação dos resultados

2 a)

$$\begin{cases} 6x_1 - x_2 + 2x_3 = 19 \\ -x_1 + 7x_2 + x_3 = -6 \\ 2x_1 + x_2 + 8x_3 = 27 \end{cases}$$

$$\left[\begin{array}{ccc|c} 6 & -1 & 2 & 19 \\ -1 & 7 & 1 & -6 \\ 2 & 1 & 8 & 27 \end{array}\right]$$

$$m_{21} = -\frac{1}{6}$$

$$L_2 \leftarrow L_2 - m_{21} \cdot L_1 = L_2 + \frac{1}{6} \cdot L_1$$

$$[-1, 7, 1, -6] + \frac{1}{6}[6, -1, 2, 19] = \begin{bmatrix} 6 & -1 & 2 & 19 \\ 0 & \frac{41}{6} & \frac{4}{3} & \frac{-17}{6} \\ 2 & 1 & 8 & 27 \end{bmatrix}$$

$$m_{31} = \frac{2}{6} = \frac{1}{3}$$

$$L_3 \leftarrow L_3 - \frac{1}{3} L_1$$

$$[2, 1, 8, 27] - \frac{1}{3}[6, -1, 2, 19] = \begin{bmatrix} 6 & -1 & 2 & 19 \\ 0 & \frac{41}{6} & \frac{4}{3} & \frac{-17}{6} \\ 0 & \frac{4}{3} & \frac{22}{3} & \frac{62}{3} \end{bmatrix}$$

$$m_{32} = \frac{8}{41}$$

$$L_3 \leftarrow L_3 - m_{32} \cdot L_2$$

$$\left[0, \frac{4}{3}, \frac{22}{3}, \frac{62}{3}\right] - \frac{8}{41} \cdot \left[0, \frac{41}{6}, \frac{4}{3}, -\frac{17}{6}\right] = \begin{bmatrix} 6 & -1 & 2 & 19 \\ 0 & \frac{41}{6} & \frac{4}{3} & -\frac{17}{6} \\ 0 & 0 & \frac{290}{41} & \frac{870}{41} \end{bmatrix}$$

Matriz Final:

$$\begin{bmatrix} 6 & -1 & 2 & 19 \\ 0 & \frac{41}{6} & \frac{4}{3} & \frac{-17}{6} \\ 0 & 0 & \frac{290}{41} & \frac{870}{41} \end{bmatrix}$$

Substituição:

$$\frac{290}{41} x_3 = \frac{870}{41}$$

$$\boxed{x_3 = 3}$$

$$\frac{41}{6} x_2 + \frac{4}{3} \cdot 3 = -\frac{17}{6}$$

$$\frac{41}{6} x_2 = -\frac{41}{6}$$

$$\boxed{x_2 = -1}$$

$$6x_1 - (-1) + 2(3) = 19$$

$$6x_1 + 1 + 6 = 19$$

$$6x_1 = 12$$

$$\boxed{x_1 = 2}$$

Resultado:

$$X = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$$

A resolução manual do sistema 3x3 foi realizada utilizando o método de Eliminação de Gauss, conforme os passos apresentados no livro de Ruggiero. O sistema foi transformado em uma matriz triangular superior por meio de operações elementares, e a solução foi obtida com substituição regressiva. O resultado foi exato e será utilizado como referência para comparar com os métodos numéricos implementados.

2 b)

Saída:

```
Eliminação de Gauss: [ 2. -1.  3.]
Cholesky: [ 2. -1.  3.]
Gauss-Jacobi: [ 2.00000029 -0.9999998   2.99999974] (em 16 iterações)
Gauss-Seidel: [ 2.00000013 -0.99999995  2.99999996] (em 10 iterações)
```

2 c)

- **Eliminação de Gauss**: retornou exatamente [2.0, −1.0, 3.0][2.0,-1.0,3.0][2.0,−1.0,3.0],
  confirmando a equivalência entre o processo manual e a rotina computacional.

- **Fatoração de Cholesky**: que exige matriz simétrica definida positiva: também produziu [2.0, −1.0, 3.0][2.0,-1.0,3.0][2.0,−1.0,3.0].

- **Gauss–Jacob**: alcançou a aproximação [2.00000029, −0.99999980, 2.99999974][2.00000029,-0.99999980,2.99999974][2.00000029,−0.99999980,2.99999974] em 16 iterações, com norma de erro infinito abaixo de 10−610^{-6}10−6.

- **Gauss–Seidel** (Seção 3.3.4): convergiu em 10 iterações para [2.00000013, −0.99999995, 2.99999996][2.00000013,-0.99999995,2.99999996][2.00000013,−0.99999995,2.99999996], demonstrando a vantagem do uso imediato de valores recentes no cálculo (como preconiza Ruggiero para acelerar a convergência).

Levando em consideração o Teorema da Convergência, os métodos iterativos convergem seguramente se a matriz for diagonalmente dominante ou simétrica positiva definida, e o nosso sistema preenche ambos os critérios. A discrepância residual entre os vetores numéricos e a solução precisa se deve unicamente à precisão dupla da aritmética de ponto flutuante; as iterações repetidas causam pequenos erros de arredondamento, porém todos mantiveram um erro absoluto abaixo de 10−610^{-6}10−6, o que é perfeitamente aceitável para fins de engenharia.

Em resumo, os resultados computacionais espelham fielmente o cálculo analítico, e o comportamento notado nos métodos iterativos como a quantidade de iterações e o modo de convergência, está em total consonância com a teoria exposta no livro de Ruggiero. Desse modo, comprovamos não só a exatidão das implementações em Python, como também a utilidade e as diferenças de desempenho entre cada método numérico.

2 d)

```
========================================
Resolvendo o sistema 3x3
========================================
Eliminação de Gauss: [ 2. -1.  3.]
Cholesky: [ 2. -1.  3.]
Gauss-Jacobi: [ 2.00000029 -0.9999998   2.99999974] (em 16 iterações)
Gauss-Seidel: [ 2.00000013 -0.99999995  2.99999996] (em 10 iterações)

========================================
RESULTADOS: SISTEMA 12x12
========================================

Solução real (x_real):
[ 1.500000 -2.300000  0.700000  4.200000 -3.100000  2.800000  1.100000
 -0.900000  3.300000 -1.700000  2.200000  0.500000]

Método              Iterações    Erro absoluto máximo
-------------------------------------------------------
Eliminação de Gauss  —           8.88e-16
Cholesky             —           1.75e+00
Gauss-Jacobi         20          3.62e-07
Gauss-Seidel         12          1.71e-07
```

Após solucionar o sistema linear 12x12, cuja resposta já era conhecida, pudemos confrontar o que cada técnica implementada apresentou. A Eliminação de Gauss se destacou, chegando à solução com um erro absoluto máximo quase inexistente (cerca de 10^{-16}), mostrando uma precisão altíssima, no nível dos erros numéricos de arredondamento que se espera em cálculos de ponto flutuante.

O método de Gauss-Seidel também se mostrou muito bom, chegando à resposta rapidamente em só 12 tentativas e alcançando um erro absoluto máximo de mais ou menos 1.71×10^{-7}. O método de Gauss-Jacobi, mesmo sendo mais vagaroso, chegou lá em 20 tentativas com um erro ainda aceitável de 3.62×10^{-7}, mostrando que os dois métodos que repetem o processo foram eficientes para esse sistema bem comportado.

Já a fatoração de Cholesky, embora na teoria seja boa para matrizes simétricas definidas positivas, mostrou uma resposta bem distante da verdadeira, com um erro absoluto máximo acima de 1.75. Esse resultado não aponta um erro no programa feito, mas sim uma característica da matriz criada. Mesmo que o código tenha forçado a simetria e a dominância diagonal, isso não assegura que a matriz seja definida positiva, algo essencial para usar a Fatoração de Cholesky corretamente. Assim, o erro grande visto nesse método tem a ver com a forma da matriz de entrada, e não com um problema na programação.

Por fim, a Eliminação de Gauss se mostra a mais precisa para esse tipo de sistema, enquanto o Gauss-Seidel é o melhor ponto de equilíbrio entre rapidez e precisão entre os métodos iterativos que foram testados.

5. Referências

RUGGIERO, Mário A.; LOPES, Vera Lucia. Cálculo Numérico: aspectos teóricos e computacionais. 2. ed. São Paulo: Pearson Prentice Hall, 2005.

6. Anexos

Link para o arquivo do código