



Instituto Federal do Espírito Santo – Campus Serra
Rodovia ES-010 – Km 6,5 – Manguinhos – 29173-087 – Serra – ES

Bacharelado em Sistemas de Informação

Álgebra Linear - Primeira Atividade Prática - 2025/1

Professor: Fidelis Zanetti de Castro

Nome: _____

Valor: 10 pontos

Introdução

Nesta atividade prática, será explorado o processo de escalonamento parcial de sistemas lineares utilizando quatro bibliotecas Python amplamente utilizadas: NumPy, SciPy, SymPy e Numba. O objetivo é analisar como diferentes abordagens computacionais podem ser aplicadas para resolver sistemas lineares com o uso de escalonamento parcial e avaliar seu desempenho em termos de número de operações e tempo gasto.

Serão trabalhados sistemas lineares de diferentes ordens, desde aqueles associados a pequenas matrizes dos coeficientes (de ordem 10×10) até “grandes” matrizes (de ordem 500×500), variando as ordens de 10 em 10. Para garantir reprodutibilidade, todos os alunos devem fixar a semente de aleatoriedade como 42 antes da geração das matrizes. Além disso, será medido o número de operações realizadas e o tempo total gasto para cada biblioteca.

Ao final, deverão ser plotados gráficos comparando o desempenho das bibliotecas, bem como respondidas perguntas que exploram os aspectos teóricos e práticos do uso dessas ferramentas.

Especificação da Atividade

1. Geração dos Sistemas Lineares

Deverão ser gerados aleatoriamente sistemas lineares de diferentes ordens n , onde $n \in \{10, 20, 30, \dots, 500\}$. Cada sistema será representado por uma matriz aumentada $[A|B]$, onde:

- A é a matriz dos coeficientes (de dimensão $n \times n$).
- B é o vetor coluna dos termos independentes (de dimensão $n \times 1$).

Todos os alunos devem fixar a semente de aleatoriedade como 42 antes da geração dos dados e é importante que os números gerados sejam números reais entre 0 e 1, tanto para A , como para B . Isso garantirá que os resultados sejam consistentes entre diferentes execuções e implementações.

2. Escalonamento Parcial

O escalonamento parcial consiste em transformar a matriz aumentada $[A|B]$ de modo que:

- O pivô de cada linha não-nula seja igual a 1.
- Todos os elementos abaixo do pivô sejam iguais a zero.
- Todas as linhas nulas devem ficar situadas abaixo das linhas não nulas.

O processo deve ser implementado nas quatro bibliotecas (NumPy, SciPy, SymPy e Numba) e realizado até obter a forma escalonada parcial.

Exemplo de Escalonamento Parcial Passo a Passo

Considere a matriz aumentada $[A|B]$ de ordem 3:

$$[A|B] = \left[\begin{array}{ccc|c} 2 & -1 & 3 & 5 \\ 4 & 2 & -1 & 3 \\ -1 & 3 & 2 & 7 \end{array} \right].$$

Passo 1: Normalize a primeira linha para que o pivô seja igual a 1:

$$L_1 := L_1 \cdot \frac{1}{2}$$

Resultado:

$$\left[\begin{array}{ccc|c} 1 & -\frac{1}{2} & \frac{3}{2} & \frac{5}{2} \\ 4 & 2 & -1 & 3 \\ -1 & 3 & 2 & 7 \end{array} \right].$$

Passo 2: Zere o elemento abaixo do pivô na segunda linha:

$$L_2 := L_2 + (-4)L_1$$

Resultado:

$$\left[\begin{array}{ccc|c} 1 & -\frac{1}{2} & \frac{3}{2} & \frac{5}{2} \\ 0 & 4 & -7 & -7 \\ -1 & 3 & 2 & 7 \end{array} \right].$$

Passo 3: Zere o elemento abaixo do pivô na terceira linha:

$$L_3 := L_3 + (1)L_1$$

Resultado:

$$\left[\begin{array}{ccc|c} 1 & -\frac{1}{2} & \frac{3}{2} & \frac{5}{2} \\ 0 & 4 & -7 & -7 \\ 0 & \frac{5}{2} & \frac{7}{2} & \frac{19}{2} \end{array} \right].$$

Passo 4: Normalize a segunda linha para que o pivô seja igual a 1:

$$L_2 := L_2 \cdot \frac{1}{4}$$

Resultado:

$$\left[\begin{array}{ccc|c} 1 & -\frac{1}{2} & \frac{3}{2} & \frac{5}{2} \\ 0 & 1 & -\frac{7}{4} & -\frac{7}{4} \\ 0 & \frac{5}{2} & \frac{7}{2} & \frac{19}{2} \end{array} \right].$$

Passo 5: Zere o elemento abaixo do pivô na terceira linha:

$$L_3 := L_3 + \left(-\frac{5}{2}\right) L_2$$

Resultado:

$$\left[\begin{array}{ccc|c} 1 & -\frac{1}{2} & \frac{3}{2} & \frac{5}{2} \\ 0 & 1 & -\frac{7}{4} & -\frac{7}{4} \\ 0 & 0 & \frac{63}{8} & \frac{147}{8} \end{array} \right].$$

Passo 6: Normalize a terceira linha para que o pivô seja igual a 1:

$$L_3 := L_3 \cdot \frac{8}{63}$$

Resultado:

$$\left[\begin{array}{ccc|c} 1 & -\frac{1}{2} & \frac{3}{2} & \frac{5}{2} \\ 0 & 1 & -\frac{7}{4} & -\frac{7}{4} \\ 0 & 0 & 1 & \frac{7}{3} \end{array} \right].$$

Contagem de Operações

No contexto deste exercício, uma operação é definida como qualquer operação aritmética envolvida no processo de escalonamento, incluindo:

- Multiplicações e divisões (e.g., normalizar um pivô dividindo cada elemento da linha por um valor).
- Adições e subtrações (e.g., zerar elementos abaixo de um pivô usando combinações lineares de linhas).

Por exemplo:

- Dividir uma linha por 2 envolve uma operação para cada elemento da linha. Se a linha tiver m elementos, isso conta como m operações.
- Subtrair uma linha multiplicada por um fator de outra linha envolve duas operações para cada elemento da linha: uma multiplicação e uma subtração. Se a linha tiver m elementos, isso conta como $2m$ operações.

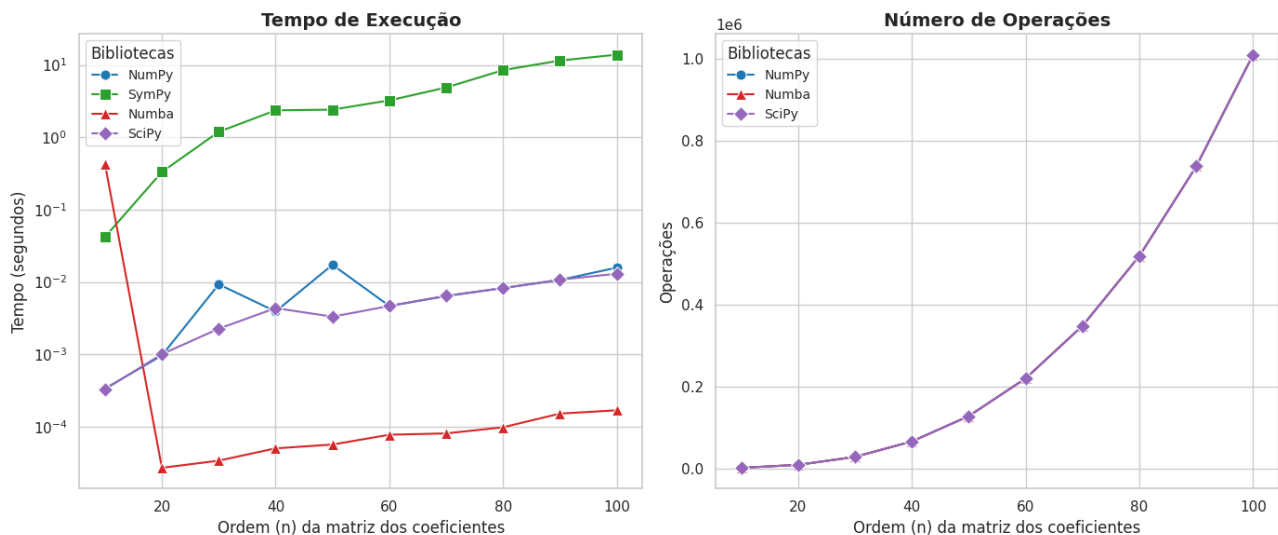
Gráficos Comparativos

Os gráficos de desempenho devem ser plotados da seguinte forma:

- Um único gráfico contendo as curvas de tempo gasto pelas quatro bibliotecas em função de n . Use escala logarítmica no eixo das ordenadas.
- Um único gráfico contendo as curvas do número de operações realizadas pelas quatro bibliotecas em função de n .

Essa organização permitirá uma comparação visual direta entre as bibliotecas.

Seguem exemplos de gráficos que gerei com n variando de 10 a 100, de 10 em 10 unidades:



Entrega dos Códigos

Os códigos desenvolvidos para esta atividade devem ser entregues em um notebook do *Google Colaboratory*, seguindo as diretrizes abaixo:

- **Implementação dos Códigos:** Inclua a implementação completa dos algoritmos de escalonamento parcial utilizando as bibliotecas NumPy , SciPy , SymPy e Numba. Certifique-se de que o código esteja funcional e bem organizado.
- **Comentários Explicativos:** Adicione comentários detalhados ao longo do código para explicar a lógica utilizada, as etapas do escalonamento e o papel de cada biblioteca. Os comentários devem facilitar a compreensão do processo por terceiros.
- **Gráficos Comparativos:** Apresente gráficos comparativos que mostrem o desempenho das bibliotecas em termos de tempo de execução e número de operações realizadas. Todos os gráficos devem estar claramente identificados e incluir legendas explicativas.
- **Respostas às Perguntas:** Responda às perguntas propostas no ambiente de *Markdown* do notebook. As respostas devem ser claras, concisas e bem fundamentadas, demonstrando sua compreensão dos conceitos abordados.
- **Organização Geral:** O notebook deve estar bem estruturado, com seções (células de *Markdown*) que organizem o conteúdo de forma lógica e fácil de seguir. Utilize títulos, subtítulos e espaçamentos adequados para garantir uma boa experiência de leitura.

Perguntas

1. O que é semente de aleatoriedade? Explique a importância de, neste exercício, fixar uma mesma semente de aleatoriedade para a turma toda.
2. Compare as quatro bibliotecas utilizadas (NumPy, SciPy, SymPy, Numba) em termos de desempenho, precisão e facilidade de implementação.
3. Justifique por que o número de operações no escalonamento parcial é da ordem de $O(n^3)$ e discuta como isso se aplica a matrizes grandes.

4. Explique por que o tempo gasto pode variar entre as bibliotecas, mesmo que o número de operações seja semelhante.
5. Explique por que a contagem de operações aritméticas não é aplicável ao SymPy, considerando sua natureza simbólica e seu foco em precisão exata.
6. Por que foi usada uma escala logarítmica no eixo das ordenadas no gráfico do tempo de execução?
7. Em quais cenários você escolheria Numba em vez de NumPy ou SciPy? Justifique sua resposta.
8. Discuta o problema da aritmética de ponto flutuante e como ele pode afetar os resultados do escalonamento parcial.
9. Proponha um método para verificar se os resultados de escalonamento parcial foram os mesmos (a menos de uma tolerância ϵ) para todas as bibliotecas.
10. Quais são as limitações de cada biblioteca no contexto desta atividade? Como essas limitações podem ser mitigadas?
11. Explique como o uso de compilação JIT (Just-In-Time) no Numba pode acelerar algoritmos de Álgebra Linear personalizados.
12. Com base nos resultados obtidos, quais são as implicações práticas de escolher uma biblioteca específica para resolver sistemas lineares em projetos reais?