ICT - UNIFESP - São José dos Campos

Cálculo Numérico *Segundo Semestre de 2025* Turma IC *Prof. Dr. Thadeu Alves Senne*

Projeto Computacional 2 - Entrega até às 23h59 do dia 03/12/2025

Este projeto consiste na implementação computacional dos Métodos de Gauss-Jacobi e de Gauss Seidel, para encontrar a solução aproximada de um sistema linear

$$Ax = b\ ,$$

em que $A \in \mathbb{R}^{n\times n}$ é uma matriz não-singular (com $a_{ii} \neq 0$ para todo $i = 1, \ldots, n$), $x \in \mathbb{R}^n$ e $b \in \mathbb{R}^n$. Os programas poderão ser desenvolvidos em *Julia* ou em *Python*.

Siga as instruções abaixo para o desenvolvimento deste trabalho:

- Este projeto poderá ser realizado em grupos de até 3 estudantes, e está dividido em dois exercícios (Exercício I e Exercício II), descritos mais adiante. Em cada um destes dois exercícios, será necessário utilizar os programas implementados para os Métodos de Gauss-Jacobi e de Gauss-Seidel.

- Para cada método, o(a) usuário(a) deverá fornecer os seguintes dados de entrada, e o programa deverá retornar os seguintes dados de saída:

  Dados de entrada: matriz A, vetor b, ponto inicial x_0, tolerância ε, número máximo de iterações maxit.

  Dados de saída: solução aproximada x^*, norma-infinito do resíduo $b - Ax^*$, erro relativo (na norma-infinito) em x^*.

- Os dois programas deverão imprimir na tela de comando uma tabela contendo os resultados obtidos em cada iteração. Esta tabela deverá ter a seguinte forma:

  Nesta tabela, denotamos por k o índice da iteração, normres a norma-infinito do resíduo $b - Ax^{(k)}$, e normrel o erro relativo (também na norma-infinito) na iteração k .

- Os programas deverão ser implementados utilizando os *notebooks* da interface *JupyterLab* (https: //jupyter.org/), cujos códigos deverão ser escritos dois arquivos separados com a extensão .jl (no caso de implementação em *Julia*) ou .ipynb (no caso de implementação em *Python*).

- A parte escrita referente à resolução das (que poderá ser manuscrita ou digitada) deverá ser feita em um único arquivo .pdf. É fundamental que as tabelas impressas para cada teste computacional estejam presentes na parte escrita do trabalho. Este arquivo e os programas implementados, cujos códigos deverão estar em arquivos separados (com as extensões .jl ou .ipynb do *notebook* do *JupyterLab*) deverão ser enviados ao professor em uma única pasta compactada .zip via Google Classroom, até às 23h59 do dia 03/12/2025.

- A pasta deverá ser nomeada seguindo os padrões abaixo, em função do número de integrantes do grupo de trabalho:

  Nome_Projeto2_CN_Turma_IC

  ou

  Nome1_Nome2_Projeto2_CN_Turma_IC

  ou

  Nome1_Nome2_Nome3_Projeto2_CN_Turma_IC ,

  Além disso, é necessário escrever os nomes completos dos(as) integrantes do grupo e os respectivos RAs no topo da primeira página do arquivo .pdf.

- ATENÇÃO: Os projetos cujas pastas não estejam nomeadas de acordo com o padrão acima e/ou sem nomes e RAs na primeira página do arquivo .pdf receberão nota zero. Não serão aceitos arquivos em outros formatos que não sejam aqueles mencionados acima.

- Caso o projeto tenha sido feita em dupla ou em trio, será necessário descrever explicitamente o que foi feito por cada membro do grupo.

Abaixo estão algumas observações e orientações para a implementação computacional dos Métodos de Gauss-Jacobi e de Gauss-Seidel.

- Conforme vimos em aula, na formulação dos Métodos de Gauss-Jacobi e de Gauss-Seidel, a matriz de coeficientes $A \in \mathbb{R}^{n\times n}$(com $a_{ii} \neq 0$ para todo $i = 1, \dots, n$) do sistema linear $Ax = b$ é decomposta na soma

  $$A = L + D + U ,$$

  em que L, D e U são, respectivamente, a parte triangular inferior, a parte diagonal e a parte triangular superior da matriz A . Para obter estas matrizes em *Julia* ou em *Python*, utilize os comandos tril, diag e triu (os nomes destes comandos são os mesmos para estas duas linguagens). Consulte o help de cada linguagem para saber como utilizar estes comandos de forma adequada.

- A matriz A e o vetor b deverão ser armazenados como matrizes esparsas. Pesquise como fazer isso na linguagem escolhida para desenvolver este trabalho.

- A implementação dos dois métodos deverá ser feita na versão matricial. Em aula, vimos que a sequência $\left\{x^{(k)}\right\}_{k=0}^{+\infty}$ gerada por estes dois métodos é dada por

  $$x^{(k+1)} = P\,x^{(k)} + c ,$$

  em que

  $$P = -D^{-1}(L + U) \text{ e } c = D^{-1}b$$

  para o Método de Gauss-Jacobi, e

  $$P = -(D + L)^{-1}U \text{ e } c = (D + L)^{-1}b$$

para o M´etodo de Gauss-Seidel.

Na pr´atica, n˜ao calculamos inversas de matrizes, pois isto ´e bastante caro sob o ponto de vista computacional. Tendo isto em mente, o ponto $x^{(k+1)}$ ´e obtido por meio da resolu¸c˜ao dos sistemas lineares

$$Dx^{(k+1)} = b - (L + U)x^{(k)}$$

no M´etodo de Gauss-Jacobi, e

$$(D + L)x^{(k+1)} = b - Ux^{(k)}$$

no M´etodo de Gauss-Seidel.

Em *Julia*, um sistema linear $Cx = d$ pode ser resolvido por meio do

comando x = C\d

e, em *Python*, ´e necess´ario utilizar a fun¸c˜ao spsolve do m´odulo sparse.linalg da biblioteca SciPy:

```
x = sparse.linalg.spsolve(C,d)
```

ˆ Para auxiliar o desenvolvimento das rotinas, apresentamos a seguir um “pseudoc´odigo” do M´etodo de Gauss-Jacobi:

<u>Entrada:</u> matriz A, vetor b, ponto inicial x_0, tolerˆancia ε, n´umero m´aximo de itera¸c˜oes maxit. *Inicialize* $k = 0$. *Defina* $E_{rel} = 1$ *e* $E_{res} = 1$.

Defina $D = \text{diag}(A)$, $L = \text{tril}(A,-1)$, $U = \text{triu}(A,1)$ e $M = L + U$.

Enquanto $(E_{rel} > \varepsilon)$ e $(E_{res} > \varepsilon)$ e $(k \leq \text{maxit})$

Resolva $Dx = b - Mx_0$.

Calcule $E_{res} = \| b - Ax \|_\infty$.

Calcule $E_{rel} = \dfrac{\| x - x_0 \|_\infty}{\| x \|_\infty}$.

Atualize $x_0 = x$.

Atualize $k = k + 1$.

<u>Sa´ıda</u>: x, E_{res} e E_{rel} .

Exerc´ıcio I: Seja o sistema linear $Ax = b$, em que $A \in \mathbb{R}^{n\times n}$ e $b \in \mathbb{R}^n$ s˜ao tais que

$$a_{ij} = \begin{cases} \alpha, & \text{se } j = i, \\ -\beta, & \text{se } j = i-1 \text{ ou } j = i+1 \\ 0, & \text{caso contr´ario} \end{cases} \quad \text{e } b_i = \begin{cases} \alpha - \beta, & \text{se } i = 1 \text{ ou } i = n, \\ \alpha - 2\beta, & \text{caso contr´ario.} \end{cases}$$

e α , β s˜ao n´umeros reais positivos. Em cada teste a seguir, tome $n = 20$, o ponto inicial $x^{(0)}$ como sendo o vetor nulo de $\mathbb{R}^n$, a tolerˆancia $\varepsilon = 10^{-8}$, e o n´umero m´aximo de itera¸c˜oes maxit = 2000 .

1. Mostre algebricamente que, para qualquer escolha de α e β positivos, a solu¸c˜ao do sistema linear

em quest˜ao ´e

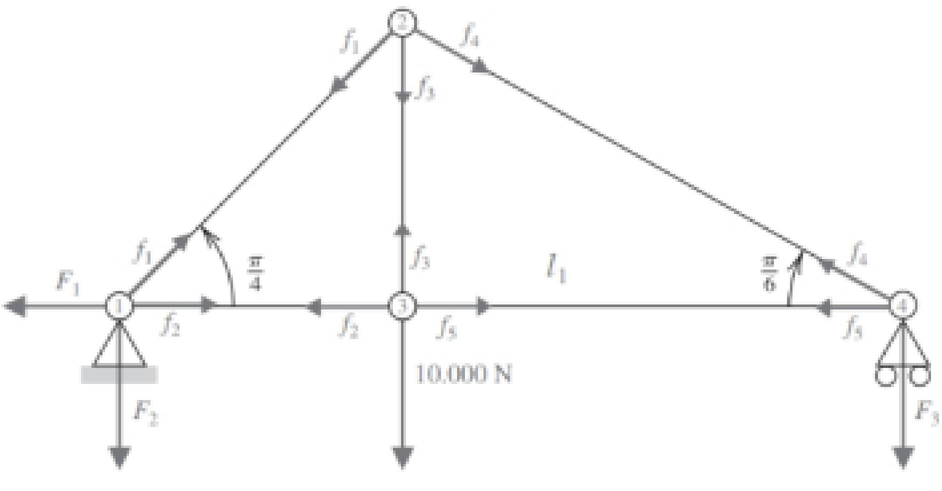
$x^* =$ 11.. . 1 ou seja, $x^*_i = 1$ para todo $i = 1, 2, . . . , n$.

2. Elabore uma rotina que calcula a matriz A e outra rotina que calcula o vetor b, fornecendo como dados de entrada os valores de α, β e n .

3. Aplique os M´etodos de Gauss-Jacobi e de Gauss-Seidel, no caso em que $\alpha = 4$ e $\beta = 1$. Para cada um dos dois testes, fa¸ca dois gr´aficos: um deles exibindo a varia¸c˜ao do valor de E_{res} e outro mostrando a varia¸c˜ao de E_{rel} , ambos em fun¸c˜ao do ´ındice das itera¸c˜oes. Estes dois gr´aficos dever˜ao estar em escala logar´ıtmica no eixo y .

4. Considere $\alpha = 1$ e $\beta = 1$. Aplique novamente os M´etodos de Gauss-Jacobi e de Gauss-Seidel. Descreva e justifique o comportamento destes dois m´etodos para estes valores de α e β .

Exerc´ıcio II: Treli¸cas s˜ao estruturas leves capazes de suportar cargas pesadas. A figura a seguir mostra uma treli¸ca mantida fixa na extremidade inferior esquerda (1), e livre para se mover horizontalmente na extremidade inferior direita (4), e com juntas de pinos em (1), (2), (3) e (4). Uma carga de 10000 N ´e aplicada na junta (3), e as for¸cas resultantes nas juntas s˜ao dadas por f_1, f_2, f_3, f_4 e f_5. Quando positivas, essas for¸cas indicam tens˜ao nos elementos da treli¸ca e, quando negativas, compress˜ao. O membro de apoio fixo na junta (1) possui uma componente de for¸ca horizontal F_1 e uma componente de for¸ca vertical F_2, enquanto que o membro do suporte m´ovel, dado pela junta (4), tem apenas uma componente de for¸ca horizontal F_3.



Se a treli¸ca estiver em equil´ıbrio est´atico, a soma das for¸cas em cada junta deve ser igual ao vetor nulo e, assim, a soma das componentes horizontais e verticais em cada junta deve ser igual a zero. Isto produz o sistema linear mostrado na tabela abaixo.

Junta	Componente horizontal	Componente vertical

(1)	$-F_1 + \frac{\sqrt{2}}{2} f_1 + f_2 = 0$	$\frac{\sqrt{2}}{2} f_1 - F_2 = 0$
(2)	$-\frac{\sqrt{2}}{2} f_1 + \frac{\sqrt{3}}{2} f_4 = 0$	$-\frac{\sqrt{2}}{2} f_1 - f_3 - \frac{1}{2} f_4 = 0$
(3)	$-f_2 + f_5 = 0$	$f_3 - 10000 = 0$
(4)	$-\frac{\sqrt{3}}{2} f_4 - f_5 = 0$	$\frac{1}{2} f_4 - F_3 = 0$

Resolva este sistema linear utilizando os Métodos de Gauss-Jacobi e de Gauss-Seidel, usando o vetor nulo de $\mathbb{R}^8$ como ponto inicial $x^{(0)}$. Escolha uma tolerância $\varepsilon > 0$ e um número máximo de iterações. Se necessário, faça um reordenamento das equações para garantir a convergência da sequência gerada por cada método.