



Introdução a Métodos Computacionais em Equações Diferenciais Parciais

Lista de exercícios para a prova (parte de conhecimentos gerais)

Lista 1 – D: Seja x^* o ponto fixo, isto é, $x^* = g(x^*)$, e defina

$$e^{(n)} = |x_n - x^*|.$$

Como $x_{n+1} = g(x_n)$ e $x^* = g(x^*)$, pelo Teorema do Valor Médio existe ξ_n entre x_n e x^* tal que

$$|x_{n+1} - x^*| = |g(x_n) - g(x^*)| = |g'(\xi_n)| |x_n - x^*|.$$

Como $|g'(x)| \leq k$ para todo $x \in (a, b)$, obtemos

$$e^{(n+1)} = |x_{n+1} - x^*| \leq k|x_n - x^*| = k e^{(n)}.$$

Aplicando essa desigualdade repetidamente,

$$e^{(n)} \leq k e^{(n-1)} \leq k^2 e^{(n-2)} \leq \dots \leq k^n e^{(0)}.$$

Portanto,

$$e^{(n)} \leq k^n e^{(0)}.$$

Como $0 < k < 1$, temos $k^n \rightarrow 0$. Logo,

$$0 \leq e^{(n)} \leq k^n e^{(0)} \rightarrow 0,$$

e, pelo teorema do confronto,

$$e^{(n)} \rightarrow 0.$$

Assim,

$$x_n \rightarrow x^*,$$

ou seja, o processo iterativo converge para o ponto fixo.

Novamente pelo Teorema do Valor Médio, para $n \geq 1$,

$$|x_{n+1} - x_n| = |g(x_n) - g(x_{n-1})| \leq k|x_n - x_{n-1}|.$$

Iterando,

$$|x_{n+1} - x_n| \leq k|x_n - x_{n-1}| \leq k^2|x_{n-1} - x_{n-2}| \leq \dots \leq k^n|x_1 - x_0|.$$

Logo, podemos estimar

$$|x_{n+1} - x_n| \leq k^n|x_1 - x_0|.$$

Escrevemos

$$x_m - x_n = (x_m - x_{m-1}) + (x_{m-1} - x_{m-2}) + \dots + (x_{n+1} - x_n).$$

Pela desigualdade triangular,

$$|x_m - x_n| \leq \sum_{j=n}^{m-1} |x_{j+1} - x_j|.$$

Usando a estimativa anterior,

$$|x_m - x_n| \leq \sum_{j=n}^{m-1} k^j |x_1 - x_0|.$$

Colocando k^n em evidência,

$$|x_m - x_n| \leq k^n |x_1 - x_0| (1 + k + k^2 + \dots + k^{m-n-1}).$$

Portanto,

$$|x_m - x_n| \leq k^n |x_1 - x_0| \sum_{r=0}^{m-n-1} k^r.$$

Como já vimos, $x_m \rightarrow x^*$. Assim,

$$|x_m - x_n| \rightarrow |x^* - x_n| = e^{(n)}.$$

Tomando o limite $m \rightarrow \infty$ na desigualdade anterior e usando que $0 < k < 1$, obtemos

$$e^{(n)} \leq k^n |x_1 - x_0| \sum_{r=0}^{\infty} k^r.$$

Como a série geométrica converge,

$$\sum_{r=0}^{\infty} k^r = \frac{1}{1-k}.$$

Logo,

$$e^{(n)} \leq \frac{k^n}{1-k} |x_1 - x_0|.$$

Essas desigualdades mostram não só que o método do ponto fixo converge, mas também fornecem estimativas para o erro cometido após n iterações conhecendo apenas x_0 e x_1 .

Lista 1 – H1: Considerando o processo iterativo não-homogêneo

$$x_{n+2} - 5x_{n+1} + 6x_n = 4^n + n^2 + 3.$$

A solução geral é $x_n = x_n^{(h)} + x_n^{(p)}$, onde $x_n^{(h)}$ é a solução da equação homogênea e $x_n^{(p)}$ é uma solução particular.

A equação homogênea associada é

$$x_{n+2} - 5x_{n+1} + 6x_n = 0.$$

Procuramos soluções da forma $x_n = r^n$. Substituindo, obtemos

$$r^{n+2} - 5r^{n+1} + 6r^n = 0,$$

ou seja,

$$r^n(r^2 - 5r + 6) = 0.$$

Como $r \neq 0$, segue

$$r^2 - 5r + 6 = 0.$$

As raízes são

$$r_1 = 2, \quad r_2 = 3.$$

Portanto,

$$x_n^{(h)} = C_1 2^n + C_2 3^n,$$

com $C_1, C_2 \in \mathbb{R}$.

Como o termo não-homogêneo é

$$f(n) = 4^n + n^2 + 3,$$

e 4 não é raiz da equação característica, propomos

$$x_n^{(p)} = A 4^n + B n^2 + C n + D.$$

Substituindo na recorrência, calculamos separadamente.

Para o termo $A 4^n$,

$$A 4^{n+2} - 5A 4^{n+1} + 6A 4^n = A 4^n (16 - 20 + 6) = 2A 4^n.$$

Como queremos que esse termo produza 4^n , temos

$$2A = 1 \implies A = \frac{1}{2}.$$

Para o polinômio $q(n) = Bn^2 + Cn + D$, precisamos de

$$q(n+2) - 5q(n+1) + 6q(n) = n^2 + 3.$$

Calculando,

$$q(n+2) = B(n+2)^2 + C(n+2) + D,$$

$$q(n+1) = B(n+1)^2 + C(n+1) + D.$$

Desenvolvendo e agrupando os coeficientes, obtemos

$$q(n+2) - 5q(n+1) + 6q(n) = 2Bn^2 + (-6B + 2C)n + (-B - 3C + 2D).$$

Igualando a $n^2 + 3$, vem

$$2B = 1,$$

$$-6B + 2C = 0,$$

$$-B - 3C + 2D = 3.$$

Da primeira equação,

$$B = \frac{1}{2}.$$

Da segunda,

$$-6 \cdot \frac{1}{2} + 2C = 0 \implies -3 + 2C = 0 \implies C = \frac{3}{2}.$$

Da terceira,

$$-\frac{1}{2} - 3 \cdot \frac{3}{2} + 2D = 3,$$

$$-\frac{1}{2} - \frac{9}{2} + 2D = 3,$$

$$-5 + 2D = 3,$$

$$2D = 8,$$

$$D = 4.$$

Assim,

$$x_n^{(p)} = \frac{1}{2} 4^n + \frac{1}{2} n^2 + \frac{3}{2} n + 4.$$

A solução geral do processo iterativo é

$$x_n = C_1 2^n + C_2 3^n + \frac{1}{2} 4^n + \frac{1}{2} n^2 + \frac{3}{2} n + 4$$

com $C_1, C_2 \in \mathbb{R}$.

Se forem dadas condições iniciais x_0 e x_1 , os valores de C_1 e C_2 podem ser determinados resolvendo o sistema

$$\begin{cases} C_1 + C_2 + \frac{1}{2} + 4 = x_0, \\ 2C_1 + 3C_2 + 2 + \frac{1}{2} + \frac{3}{2} + 4 = x_1. \end{cases}$$

Lista 2 – B3: Usando Newton-Raphson, com $f(x) = x^6 - 3$ cuja raiz positiva é $\sqrt[6]{3}$ em 6 iterações obtemos

Iteração 1: $x = 1.333333333333$, $f(x) = 2.619e+00$

Iteração 2: $x = 1.229763454861$, $f(x) = 4.588e-01$

Iteração 3: $x = 1.202574359818$, $f(x) = 2.463e-02$

Iteração 4: $x = 1.200942518723$, $f(x) = 8.339e-05$
 Iteração 5: $x = 1.200936955240$, $f(x) = 9.658e-10$
 Iteração 6: $x = 1.200936955176$, $f(x) = 0.000e+00$
 Raiz aproximada: 1.200936955176. Com 5 casas decimais: 1.20094.

Lista 2 – B5: A raiz aproximada é $x \approx 1.072916131$, ou, com 6 casas decimais, $x \approx 1.072916$.

Com os mesmos pontos iniciais usuais:

Newton–Raphson: $x_0 = 1.0$;

Secante: $x_0 = 1.0, x_1 = 1.1$;

Bisseção: intervalo $[1.0, 1.1]$,

obtemos

Newton–Raphson: $x = 1.072916131$, iteracoes = 4

Secante: $x = 1.072916131$, iteracoes = 6

Bissecao: $x = 1.072916131$, iteracoes = 30

Afastando um pouco o chute inicial para

Newton–Raphson: $x_0 = 3.0$;

Secante: $x_0 = 0.2, x_1 = 3.1$;

Bisseção: intervalo $[0.2, 3.1]$,

encontramos

Newton–Raphson: $x = 1.072916131121$, iteracoes = 7

Secante: $x = 1.072916131121$, iteracoes = 9

Bissecao: $x = 1.072916131097$, iteracoes = 35

ou seja, o método mais rápido é o Newton–Raphson. A secante também converge rapidamente, mas fica um pouco atrás. A bisseção é a mais lenta, pois sua convergência é linear.

Lista 2 – D1: Seja $f(x) = 0$ e suponha que temos um intervalo $[a_i, b_i]$ tal que

$$f(a_i)f(b_i) < 0.$$

A reta secante pelos pontos $(a_i, f(a_i))$ e $(b_i, f(b_i))$ é

$$y - f(a_i) = \frac{f(b_i) - f(a_i)}{b_i - a_i}(x - a_i).$$

Fazendo $y = 0$, obtemos a nova aproximação

$$x^{(i+1)} = \frac{a_i f(b_i) - b_i f(a_i)}{f(b_i) - f(a_i)}$$

ou, equivalentemente,

$$x^{(i+1)} = a_i - \frac{f(a_i)(b_i - a_i)}{f(b_i) - f(a_i)}.$$

Depois escolhe-se o novo intervalo mantendo a mudança de sinal:

$$\begin{cases} \text{se } f(a_i)f(x^{(i+1)}) < 0, & a_{i+1} = a_i, \quad b_{i+1} = x^{(i+1)}, \\ \text{caso contrário,} & a_{i+1} = x^{(i+1)}, \quad b_{i+1} = b_i. \end{cases}$$

Assim, a raiz permanece sempre contida no intervalo $[a_{i+1}, b_{i+1}]$.

Em cada iteração, toma-se a corda que une os pontos extremos do intervalo que contém a raiz. A nova aproximação é a interseção dessa corda com o eixo x . A fronteira cujo valor de f tem o mesmo sinal que $f(x^{(i+1)})$ é substituída, de modo que a raiz continue “presa” entre os dois extremos.

Lista 2 – D2: Considere

$$f(x) = \sin(x) - \frac{x}{2}.$$

A solução não nula positiva está no intervalo $[1, 2]$, pois

$$f(1) = \sin(1) - 0.5 > 0, \quad f(2) = \sin(2) - 1 < 0.$$

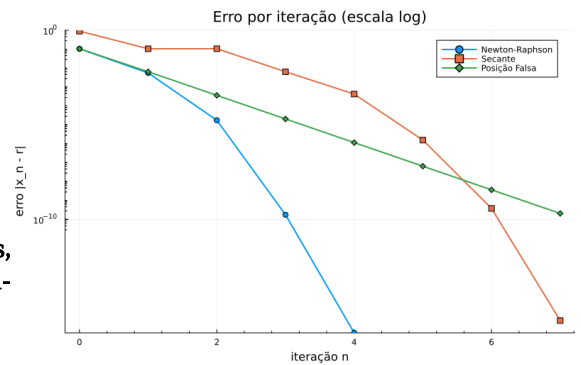
A raiz positiva é

$$x^* \approx 1.895494267033981.$$

Lista 2 – D3: Usando tolerância 10^{-8} :

Método	Iterações	Aproximação
Newton–Raphson, $x_0 = 2$	4	1.8954942670
Secante, $x_0 = 1, x_1 = 2$	6	1.8954942670
Posição Falsa, $[1, 2]$	8	1.8954942670

Portanto, para este problema e estes chutes iniciais, Newton–Raphson é o mais rápido; depois a Secante; por último a Posição Falsa.



A Posição Falsa garante o confinamento da raiz, mas em geral tem convergência linear.

Lista 3 – D1: Um programa em Julia ficaria algo como

```

1  using Printf, LinearAlgebra
2
3  function lu_pivoteada(A::AbstractMatrix)
4      n = size(A, 1)
5      size(A, 2) == n || error("A matriz deve ser quadrada")
6      U = Float64.(copy(A))
7      L = Matrix{Float64}(I, n, n)
8      p = collect(1:n)
9
10     for k in 1:n-1
11         piv = k
12         maxabs = abs(U[k, k])
13         for i in k+1:n
14             if abs(U[i, k]) > maxabs
15                 maxabs = abs(U[i, k]); piv = i
16             end
17         end
18         maxabs == 0 && error("Matriz singular")
19
20         if piv != k
21             U[k, :], U[piv, :] = copy(U[piv, :]), copy(U[k, :])
22             p[k], p[piv] = p[piv], p[k]
23             if k > 1
24                 L[k, 1:k-1], L[piv, 1:k-1] = copy(L[piv, 1:k-1]), copy(L[k, 1:k-1])
25             end
26         end
27
28         for i in k+1:n
29             L[i, k] = U[i, k] / U[k, k]
30             U[i, k:n] -= L[i, k] .* U[k, k:n]
31         end
32     end
33
34     any(abs.(diag(U)) .< eps()) && error("Matriz singular")
35     return L, U, p
36 end
37
38 function substituicao_direta(L::AbstractMatrix, b::AbstractVector)

```

```

39     n = length(b)
40     y = similar(b, Float64)
41     for i in 1:n
42         s = b[i]
43         for j in 1:i-1
44             s -= L[i, j] * y[j]
45         end
46         y[i] = s / L[i, i]
47     end
48     return y
49 end
50
51 function substituicao_retroativa(U::AbstractMatrix, b::AbstractVector)
52     n = length(b)
53     x = similar(b, Float64)
54     for i in n:-1:1
55         s = b[i]
56         for j in i+1:n
57             s -= U[i, j] * x[j]
58         end
59         abs(U[i, i]) < eps() && error("U singular na diagonal")
60         x[i] = s / U[i, i]
61     end
62     return x
63 end
64
65 function inversa_via_LU(A::AbstractMatrix)
66     n = size(A, 1)
67     L, U, p = lu_pivoteada(A)
68     X = zeros(Float64, n, n)
69     e = zeros(Float64, n)
70
71     for i in 1:n
72         fill!(e, 0.0)
73         e[i] = 1.0
74         b = e[p]
75         y = substituicao_direta(L, b)
76         x = substituicao_retroativa(U, y)
77         X[:, i] = x
78     end
79     return X
80 end
81
82 # ----- Teste -----
83
84 n = 150
85
86 C = diagm(
87     0 => fill( 2.0, n),      # diagonal principal
88     1 => fill(-1.0, n-1),    # primeira super-diagonal
89     2 => fill(-1.0, n-2),    # segunda super-diagonal
90    -1 => fill(-1.0, n-1),    # primeira sub-diagonal
91    -2 => fill(-1.0, n-2),    # segunda sub-diagonal
92 )
93
94 Cinv = inversa_via_LU(C)    # inversa de C que queremos calcular
95 Id = Matrix{Float64}(I, n, n) # identidade
96
97 println("Matriz C:")
98 display(C); println()
99
100 println("Inversa calculada por LU:")
101 display(Cinv); println()
102
103 println("Verificacao: ||C * Cinv - I||_inf = ",
104     norm(C * Cinv - Id, Inf))
105 println("Comparacao com inv(C): ||Cinv - inv(C)||_inf = ",
106     norm(Cinv - inv(C), Inf))

```

onde $\text{inv}(C)$ é a função inversa nativa do pacote `LinearAlgebra` usado para comparação apenas. Como as matrizes são muito grandes para serem exibidas, deixamos apenas o resultado das últimas linhas:

Verificação: $\|C * \text{Cinv} - I\|_{\text{inf}} = 5.551115123125783\text{e-}16$

Comparação com $\text{inv}(C)$: $\|\text{Cinv} - \text{inv}(C)\|_{\text{inf}} = 1.1657341758564144\text{e-}15$
que mostram que a norma está bem próxima de zero.

Lista 3 – F1: A matriz dos coeficientes é

$$A(\alpha) = \begin{pmatrix} 2 & 1 & 3 \\ 4 & 6 & 8 \\ 6 & \alpha & 10 \end{pmatrix}.$$

Calculando o determinante pela regra de Sarrus:

$$\det A(\alpha) = 2(60 - 8\alpha) - 1(40 - 48) + 3(4\alpha - 36).$$

Simplificando,

$$\det A(\alpha) = 120 - 16\alpha + 8 + 12\alpha - 108 = 20 - 4\alpha.$$

Portanto,

$$\det A(\alpha) = 0 \iff 20 - 4\alpha = 0 \iff \alpha = 5.$$

Assim:

- Se $\alpha \neq 5$, então $\det A(\alpha) \neq 0$, e o sistema tem **solução única**.
- Se $\alpha = 5$, o sistema pode ser inconsistente ou ter infinitas soluções.

Verifiquemos o caso $\alpha = 5$. A matriz aumentada fica

$$\left[\begin{array}{ccc|c} 2 & 1 & 3 & 1 \\ 4 & 6 & 8 & 5 \\ 6 & 5 & 10 & 5 \end{array} \right].$$

Fazendo $L_2 \leftarrow L_2 - 2L_1$ e $L_3 \leftarrow L_3 - 3L_1$:

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

Agora $L_3 \leftarrow L_3 - \frac{1}{2}L_2$:

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

Como $0 = \frac{1}{2}$ é impossível, o sistema é **inconsistente** para $\alpha = 5$. Portanto, $\alpha = 5$ não produz solução alguma.

Lista 3 – F2:

Na eliminação gaussiana sem pivoteamento, exige-se que todos os pivôs sejam não nulos na ordem natural.

Passo 1. Pivô $a_{11} = 2 \neq 0$ — OK para todo α .

Após eliminar a primeira coluna:

$$L_2 \leftarrow L_2 - 2L_1, \quad L_3 \leftarrow L_3 - 3L_1,$$

obtem-se

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

Passo 2. O segundo pivô é $a_{22} = 4 \neq 0$ — OK para todo α .
Após eliminar a segunda coluna,

$$L_3 \leftarrow L_3 - \frac{\alpha - 3}{4} L_2,$$

o elemento $(3, 3)$ fica

$$a_{33}^{(2)} = 1 - \frac{\alpha - 3}{4} \cdot 2 = 1 - \frac{\alpha - 3}{2} = \frac{5 - \alpha}{2}.$$

Passo 3. Exige-se $a_{33}^{(2)} \neq 0$, isto é,

$$\frac{5 - \alpha}{2} \neq 0 \iff \alpha \neq 5.$$

Portanto, a eliminação gaussiana **sem pivoteamento** funciona para todo $\alpha \neq 5$. No caso $\alpha = 5$, o terceiro pivô se anula e seria necessário pivoteamento (mas o sistema já é inconsistente, como visto).

Lista 3 – F3:

Como podemos pegar qual valor diferente de 5, escolhemos $\alpha = 1$. Então

$$A = \begin{pmatrix} 2 & 1 & 3 \\ 4 & 6 & 8 \\ 6 & 1 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 5 \\ 5 \end{pmatrix}.$$

Como visto acima, os multiplicadores são

$$m_{21} = \frac{4}{2} = 2, \quad m_{31} = \frac{6}{2} = 3.$$

Após o primeiro passo,

$$U^{(1)} = \begin{pmatrix} 2 & 1 & 3 \\ 0 & 4 & 2 \\ 0 & -2 & 1 \end{pmatrix}.$$

O segundo multiplicador é

$$m_{32} = \frac{-2}{4} = -\frac{1}{2}.$$

Após o segundo passo,

$$U = \begin{pmatrix} 2 & 1 & 3 \\ 0 & 4 & 2 \\ 0 & 0 & 2 \end{pmatrix}.$$

Assim,

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & -\frac{1}{2} & 1 \end{pmatrix}$$

Resolvendo $Ly = b$:

$$\begin{cases} y_1 = 1, \\ 2y_1 + y_2 = 5 \Rightarrow y_2 = 3, \\ 3y_1 - \frac{1}{2}y_2 + y_3 = 5 \Rightarrow 3 - \frac{3}{2} + y_3 = 5 \Rightarrow y_3 = \frac{7}{2}. \end{cases}$$

Então

$$y = \begin{pmatrix} 1 \\ 3 \\ \frac{7}{2} \end{pmatrix}.$$

Resolvendo $Ux = y$:

$$\begin{cases} 2x_3 = \frac{7}{2} \Rightarrow x_3 = \frac{7}{4}, \\ 4x_2 + 2x_3 = 3 \Rightarrow 4x_2 + \frac{7}{2} = 3 \Rightarrow x_2 = -\frac{1}{8}, \\ 2x_1 + x_2 + 3x_3 = 1 \Rightarrow 2x_1 - \frac{1}{8} + \frac{21}{4} = 1 \Rightarrow 2x_1 = 1 + \frac{1}{8} - \frac{21}{4} = -\frac{33}{8} \Rightarrow x_1 = -\frac{33}{16}. \end{cases}$$

Portanto,

$$x = -\frac{33}{16}, \quad y = -\frac{1}{8}, \quad z = \frac{7}{4}.$$

Lista 3 – K1:

O processo é

$$x^{(n+1)} = A x^{(n)}, \quad x^{(0)} = a.$$

Por indução:

$$x^{(1)} = Aa, \quad x^{(2)} = A(Aa) = A^2a, \quad \dots, \quad x^{(n)} = A^n a.$$

Portanto,

$$x^{(n)} = A^n a.$$

Como $\{v_1, \dots, v_N\}$ é base de autovetores de A , escrevemos

$$a = \sum_{k=1}^N c_k v_k.$$

Aplicando A^n e usando $A^n v_k = \lambda_k^n v_k$:

$$x^{(n)} = A^n a = A^n \sum_{k=1}^N c_k v_k = \sum_{k=1}^N c_k A^n v_k = \sum_{k=1}^N c_k \lambda_k^n v_k.$$

Logo,

$$x^{(n)} = \lambda_1^n c_1 v_1 + \lambda_2^n c_2 v_2 + \dots + \lambda_N^n c_N v_N = \sum_{k=1}^N \lambda_k^n c_k v_k.$$

Lista 3 – K2:

Suponha que λ_i seja dominante, isto é,

$$|\lambda_i| > |\lambda_k|, \quad k \neq i.$$

Fatorando λ_i^n :

$$x^{(n)} = \lambda_i^n \left[c_i v_i + \sum_{k \neq i} c_k \left(\frac{\lambda_k}{\lambda_i} \right)^n v_k \right].$$

Como $|\lambda_k/\lambda_i| < 1$, todos os termos com $k \neq i$ tendem a zero quando $n \rightarrow \infty$. Assim, para $n \gg 1$,

$$x^{(n)} \approx \lambda_i^n c_i v_i.$$

Em componentes,

$$x_j^{(n)} \approx \lambda_i^n c_i v_{ij},$$

onde v_{ij} é a j -ésima componente de v_i .

Então, para qualquer componente j com $c_i v_{ij} \neq 0$,

$$\frac{x_j^{(n+1)}}{x_j^{(n)}} = \frac{\lambda_i^{n+1} c_i v_{ij}}{\lambda_i^n c_i v_{ij}} = \lambda_i.$$

Portanto,

$$\lambda_i = \lim_{n \rightarrow \infty} \frac{x_j^{(n+1)}}{x_j^{(n)}}.$$

Lista 3 – K3:

Em todos os casos abaixo usamos o método da potência com normalização pela norma ∞ e critério de parada

$$|\lambda^{(n+1)} - \lambda^{(n)}| < 10^{-10}.$$

(a)

$$A = \begin{pmatrix} -1 & 2 & 2 \\ -1 & -4 & -2 \\ -3 & 9 & 7 \end{pmatrix}.$$

Os autovalores são $\lambda = 2, 1, -1$. O dominante é

$$\lambda_1 = 2.$$

O método da potência converge para $\lambda_1 = 2$.

(b)

$$A = \begin{pmatrix} -2 & 0 & 1 \\ -3 & -2 & 0 \\ -2 & 5 & 4 \end{pmatrix}.$$

Os autovalores são $\lambda = 3, -2, -1$ (verificação numérica: traço = 0, determinante = 6). O dominante é

$$\lambda_1 = 3.$$

(c)

$$A = \begin{pmatrix} 1 & -2 & -3 \\ 1 & -5 & 4 \\ 1 & -5 & 4 \end{pmatrix}.$$

Aqui as linhas 2 e 3 são idênticas, então $\det A = 0$ e 0 é autovalor. Calculando o polinômio característico,

$$\det(A - \lambda I) = -\lambda(\lambda^2 + 4\lambda + 1).$$

Como $L_2 - L_3 = 0$, $\text{posto}(A) \leq 2$, logo 0 é autovalor. O traço é

$$\text{tr } A = 1 - 5 + 4 = 0,$$

e a soma dos autovalores é 0. O determinante de A é 0 (linhas LD). Os outros dois autovalores vêm de

$$\lambda^2 + 4\lambda + 1 = 0 \implies \lambda = -2 \pm \sqrt{3}.$$

Portanto,

$$\lambda \approx 0, \quad \lambda \approx -2 + \sqrt{3} \approx -0.2679, \quad \lambda \approx -2 - \sqrt{3} \approx -3.7321.$$

O dominante é

$$\lambda_1 = -2 - \sqrt{3} \approx -3.7321.$$

Lista 3 – K4:

Resolvendo $(A - \lambda_1 I)v = 0$. Uma vez conhecido λ_1 , o autovetor pode ser obtido pelo espaço nulo de $A - \lambda_1 I$ (via eliminação gaussiana ou SVD). Esse método é mais estável e é o recomendado quando se quer v_1 com alta precisão, pois o método da potência fornece v_1 com precisão limitada pela razão $|\lambda_2/\lambda_1|$.