



Introdução a Métodos Computacionais em Equações Diferenciais Parciais (1ª. PROVA - 26/10/26)

Questão – A1:

Equação:

$$y'' + 4y' + 3y = 1, \quad y(0) = 1, \quad y'(0) = 0.$$

Definimos

$$u_1(t) = y(t), \quad u_2(t) = y'(t).$$

Então

$$\begin{aligned} u_1' &= u_2, \\ u_2' &= y'' = -4y' - 3y + 1 = -4u_2 - 3u_1 + 1. \end{aligned}$$

Logo, o sistema é

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ -3 & -4 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

com condição inicial

$$\begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Em notação vetorial,

$$u' = Au + b, \quad A = \begin{pmatrix} 0 & 1 \\ -3 & -4 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad u(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Questão – A2:

O método de Euler explícito é

$$u^{(n+1)} = u^{(n)} + h f(t_n, u^{(n)}),$$

onde

$$f(t, u) = Au + b.$$

Assim,

$$u^{(n+1)} = u^{(n)} + h(Au^{(n)} + b) = (I + hA)u^{(n)} + hb.$$

Portanto, o processo iterativo matricial é

$$u^{(n+1)} = M_{\text{exp}} u^{(n)} + hb,$$

com

$$M_{\text{exp}} = I + hA = \begin{pmatrix} 1 & h \\ -3h & 1 - 4h \end{pmatrix}, \quad hb = \begin{pmatrix} 0 \\ h \end{pmatrix}.$$

Questão – A3:

O ponto fixo u^* satisfaz

$$u^* = M_{\text{exp}} u^* + hb \implies (I - M_{\text{exp}})u^* = hb.$$

Como

$$I - M_{\text{exp}} = \begin{pmatrix} 0 & -h \\ 3h & 4h \end{pmatrix},$$

temos

$$\begin{cases} -h u_2^* = 0, \\ 3h u_1^* + 4h u_2^* = h. \end{cases}$$

Da primeira, $u_2^* = 0$ (pois $h > 0$). Da segunda, $3h u_1^* = h$, logo $u_1^* = \frac{1}{3}$. Portanto,

$$u^* = \begin{pmatrix} \frac{1}{3} \\ 0 \end{pmatrix}.$$

Esse é exatamente o ponto de equilíbrio da EDO:

$$3y = 1 \implies y = \frac{1}{3}, \quad y' = 0.$$

Estabilidade

A estabilidade do ponto fixo é determinada pelos autovalores de M_{exp} . Calculamos

$$\det(M_{\text{exp}} - \lambda I) = \begin{vmatrix} 1 - \lambda & h \\ -3h & 1 - 4h - \lambda \end{vmatrix} = (1 - \lambda)(1 - 4h - \lambda) + 3h^2.$$

Expandindo:

$$\lambda^2 - (2 - 4h)\lambda + (1 - 4h + 3h^2) = 0.$$

Os autovalores são

$$\lambda_{1,2} = \frac{(2 - 4h) \pm \sqrt{(2 - 4h)^2 - 4(1 - 4h + 3h^2)}}{2}.$$

Calculando o discriminante:

$$\Delta = (2 - 4h)^2 - 4(1 - 4h + 3h^2) = 4 - 16h + 16h^2 - 4 + 16h - 12h^2 = 4h^2.$$

Portanto,

$$\lambda_{1,2} = \frac{2 - 4h \pm 2h}{2} = 1 - 2h \pm h.$$

Assim,

$$\lambda_1 = 1 - h, \quad \lambda_2 = 1 - 3h.$$

Condição de estabilidade

O ponto fixo é **assintoticamente estável** se

$$|\lambda_1| < 1 \quad \text{e} \quad |\lambda_2| < 1.$$

Analisando:

$$|1 - h| < 1 \iff 0 < h < 2,$$

$$|1 - 3h| < 1 \iff 0 < h < \frac{2}{3}.$$

A condição mais restritiva é

$$0 < h < \frac{2}{3}.$$

Para $h = \frac{2}{3}$, $|\lambda_2| = 1$ (estabilidade marginal). Para $h > \frac{2}{3}$, $|\lambda_2| > 1$ e o método é **instável**: as iteradas divergem, embora a solução exata da EDO convirja para $y = \frac{1}{3}$.

Questão – A4:

O método de Euler implícito é

$$u^{(n+1)} = u^{(n)} + h f(t_{n+1}, u^{(n+1)}).$$

Com $f(t, u) = Au + b$:

$$u^{(n+1)} = u^{(n)} + h (Au^{(n+1)} + b).$$

Isolando $u^{(n+1)}$:

$$(I - hA)u^{(n+1)} = u^{(n)} + hb,$$

$$u^{(n+1)} = M_{\text{imp}} u^{(n)} + (I - hA)^{-1}hb,$$

com

$$M_{\text{imp}} = (I - hA)^{-1}.$$

Calculando:

$$I - hA = \begin{pmatrix} 1 & -h \\ 3h & 1 + 4h \end{pmatrix}, \quad \det(I - hA) = (1)(1 + 4h) - (-h)(3h) = 1 + 4h + 3h^2.$$

A inversa é

$$(I - hA)^{-1} = \frac{1}{1 + 4h + 3h^2} \begin{pmatrix} 1 + 4h & h \\ -3h & 1 \end{pmatrix}.$$

Logo,

$$M_{\text{imp}} = \frac{1}{1 + 4h + 3h^2} \begin{pmatrix} 1 + 4h & h \\ -3h & 1 \end{pmatrix}.$$

E o termo constante:

$$(I - hA)^{-1}hb = \frac{h}{1 + 4h + 3h^2} \begin{pmatrix} h \\ 1 \end{pmatrix} = \frac{1}{1 + 4h + 3h^2} \begin{pmatrix} h^2 \\ h \end{pmatrix}.$$

Assim, o processo iterativo é

$$u^{(n+1)} = \frac{1}{1 + 4h + 3h^2} \begin{pmatrix} 1 + 4h & h \\ -3h & 1 \end{pmatrix} u^{(n)} + \frac{1}{1 + 4h + 3h^2} \begin{pmatrix} h^2 \\ h \end{pmatrix}.$$

Ponto fixo (Euler implícito)

O ponto fixo satisfaz

$$u^* = M_{\text{imp}} u^* + (I - hA)^{-1}hb \iff (I - M_{\text{imp}})u^* = (I - hA)^{-1}hb.$$

Mas, como $M_{\text{imp}} = (I - hA)^{-1}$, temos

$$(I - M_{\text{imp}}) = (I - hA)^{-1}((I - hA) - I) = (I - hA)^{-1}(-hA).$$

Logo,

$$(I - hA)^{-1}(-hA)u^* = (I - hA)^{-1}hb \implies -hA u^* = hb.$$

Como $h > 0$,

$$-Au^* = b \implies Au^* = -b.$$

Ou seja,

$$\begin{pmatrix} 0 & 1 \\ -3 & -4 \end{pmatrix} \begin{pmatrix} u_1^* \\ u_2^* \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}.$$

Da primeira equação, $u_2^* = 0$. Da segunda, $-3u_1^* = -1$, logo $u_1^* = \frac{1}{3}$. Portanto,

$$u^* = \begin{pmatrix} \frac{1}{3} \\ 0 \end{pmatrix},$$

o mesmo ponto fixo do método explícito (e do equilíbrio da EDO).

Estabilidade (Euler implícito)

Os autovalores de M_{imp} são os inversos dos autovalores de $I - hA$. Calculamos

$$\det(I - hA - \mu I) = \begin{vmatrix} 1 - \mu & -h \\ 3h & 1 + 4h - \mu \end{vmatrix} = (1 - \mu)(1 + 4h - \mu) + 3h^2.$$

Expandindo:

$$\mu^2 - (2 + 4h)\mu + (1 + 4h + 3h^2) = 0.$$

Os autovalores de $I - hA$ são

$$\mu_{1,2} = \frac{(2 + 4h) \pm \sqrt{(2 + 4h)^2 - 4(1 + 4h + 3h^2)}}{2}.$$

O discriminante:

$$\Delta = (2 + 4h)^2 - 4(1 + 4h + 3h^2) = 4 + 16h + 16h^2 - 4 - 16h - 12h^2 = 4h^2.$$

Logo,

$$\mu_{1,2} = \frac{2 + 4h \pm 2h}{2} = 1 + 2h \pm h.$$

Assim,

$$\mu_1 = 1 + h, \quad \mu_2 = 1 + 3h.$$

Os autovalores de $M_{\text{imp}} = (I - hA)^{-1}$ são

$$\lambda_1 = \frac{1}{1 + h}, \quad \lambda_2 = \frac{1}{1 + 3h}.$$

Para todo $h > 0$,

$$0 < \lambda_1 < 1, \quad 0 < \lambda_2 < 1.$$

Portanto, o ponto fixo é assintoticamente estável para todo $h > 0$ e o método de Euler implícito é **incondicionalmente estável** para este problema.

Questão – B1:

Seja $f_k = f(x_k)$, $x_k = hk$, e os operadores de diferenças

$$\Delta_+ f_k = f_{k+1} - f_k, \quad \Delta_- f_k = f_k - f_{k-1}.$$

Lembremos também que, atuando sobre exponenciais $e^{ik\theta}$, os operadores correspondem a

$$\Delta_+ \leftrightarrow e^{i\theta} - 1, \quad \Delta_- \leftrightarrow 1 - e^{-i\theta},$$

e que o operador derivada corresponde a $hD \leftrightarrow i\theta$. Equivalentemente, em termos do operador de deslocamento ε , $\varepsilon f_k = f_{k+1}$, temos

$$\Delta_+ = \varepsilon - 1, \quad \Delta_- = 1 - \varepsilon^{-1}.$$

Queremos $f'(x_i)$ usando os pontos

$$x_{i-2}, x_{i-1}, x_i, x_{i+1}.$$

Por conveniência, transladamos a origem para x_i : sejam

$$t = \frac{x - x_i}{h}, \quad f(x_i + ht) = P(t),$$

com nós $t = -2, -1, 0, 1$. Queremos $f'(x_i) = \frac{1}{h}P'(0)$.

O polinômio interpolador de Lagrange é

$$P(t) = \sum_{j=-2}^1 f_{i+j} \ell_j(t),$$

com

$$\begin{aligned}\ell_{-2}(t) &= \frac{(t+1)t(t-1)}{(-1)(-2)(-3)} = -\frac{t(t+1)(t-1)}{6}, \\ \ell_{-1}(t) &= \frac{(t+2)t(t-1)}{(1)(-1)(-2)} = \frac{t(t+2)(t-1)}{2}, \\ \ell_0(t) &= \frac{(t+2)(t+1)(t-1)}{(2)(1)(-1)} = -\frac{(t+2)(t+1)(t-1)}{2}, \\ \ell_1(t) &= \frac{(t+2)(t+1)t}{(3)(2)(1)} = \frac{t(t+1)(t+2)}{6}.\end{aligned}$$

Derivando em $t = 0$:

$$\begin{aligned}\ell'_{-2}(0) &= -\frac{1}{6} \cdot \frac{d}{dt}[t(t^2-1)]\Big|_0 = -\frac{1}{6}(3t^2-1)\Big|_0 = \frac{1}{6}, \\ \ell'_{-1}(0) &= \frac{1}{2} \cdot \frac{d}{dt}[t(t^2+t-2)]\Big|_0 = \frac{1}{2}(3t^2+2t-2)\Big|_0 = -1, \\ \ell'_0(0) &= -\frac{1}{2} \cdot \frac{d}{dt}[(t+2)(t^2-1)]\Big|_0 = -\frac{1}{2}(3t^2+4t-1)\Big|_0 = \frac{1}{2}, \\ \ell'_1(0) &= \frac{1}{6} \cdot \frac{d}{dt}[t(t^2+3t+2)]\Big|_0 = \frac{1}{6}(3t^2+6t+2)\Big|_0 = \frac{1}{3}.\end{aligned}$$

Portanto,

$$P'(0) = \frac{1}{6}f_{i-2} - f_{i-1} + \frac{1}{2}f_i + \frac{1}{3}f_{i+1},$$

e, como $f'(x_i) = \frac{1}{h}P'(0)$,

$$f'(x_i) \approx \frac{1}{h} \left(\frac{1}{6}f_{i-2} - f_{i-1} + \frac{1}{2}f_i + \frac{1}{3}f_{i+1} \right).$$

Multiplicando por 6:

$$f'(x_i) \approx \frac{1}{6h} (f_{i-2} - 6f_{i-1} + 3f_i + 2f_{i+1}).$$

Questão – B2:

Temos, em termos do deslocamento ε :

$$\Delta_+ = \varepsilon - 1, \quad \Delta_- = 1 - \varepsilon^{-1}.$$

Da segunda, $\varepsilon^{-1} = 1 - \Delta_-$, logo $\varepsilon = (1 - \Delta_-)^{-1}$. Então

$$\Delta_+ = \varepsilon - 1 = \frac{1}{1 - \Delta_-} - 1 = \frac{\Delta_-}{1 - \Delta_-}.$$

Portanto,

$$\Delta_+ = \frac{\Delta_-}{1 - \Delta_-} = \Delta_- + \Delta_-^2 + \Delta_-^3 + \cdots = \sum_{j=1}^{\infty} \Delta_-^j.$$

Essa é a representação formal de Δ_+ em potências de Δ_- .

Questão – B3:

Queremos que $\mathcal{D}$ represente hD , isto é, que atue sobre exponenciais como $i\theta$. Com $\Delta_+ \leftrightarrow e^{i\theta} - 1$ e $\Delta_- \leftrightarrow 1 - e^{-i\theta}$, temos

$$\mathcal{D} \leftrightarrow \alpha(e^{i\theta} - 1) + \sum_{j \geq 1} \beta_j (1 - e^{-i\theta})^j.$$

Queremos que isso seja igual a $i\theta$ para todo θ . Usando o item B2,

$$\Delta_+ = \sum_{j \geq 1} \Delta_-^j,$$

temos

$$\mathcal{D} = \alpha \sum_{j \geq 1} \Delta_-^j + \sum_{j \geq 1} \beta_j \Delta_-^j = \sum_{j \geq 1} (\alpha + \beta_j) \Delta_-^j.$$

Por outro lado, o operador hD em termos de Δ_- é obtido de $\Delta_- = 1 - e^{-hD}$, logo $e^{-hD} = 1 - \Delta_-$, e

$$-hD = \ln(1 - \Delta_-) \implies hD = -\ln(1 - \Delta_-) = \sum_{j \geq 1} \frac{\Delta_-^j}{j}.$$

Portanto,

$$hD = \sum_{j \geq 1} \frac{1}{j} \Delta_-^j.$$

Igualando

$$\sum_{j \geq 1} (\alpha + \beta_j) \Delta_-^j = \sum_{j \geq 1} \frac{1}{j} \Delta_-^j,$$

obtemos, para cada $j \geq 1$,

$$\alpha + \beta_j = \frac{1}{j} \implies \boxed{\beta_j = \frac{1}{j} - \alpha, \quad j = 1, 2, 3, \dots}$$

Questão – B4:

Agora queremos

$$\mathcal{D} = \alpha \Delta_+ + \sum_{j=1}^N \beta_j \Delta_-^j + \mathcal{O}(h^{N+1}).$$

Como $\Delta_+ = \varepsilon - 1 = \mathcal{O}(h)$ e $\Delta_-^j = \mathcal{O}(h^j)$, o termo $\alpha \Delta_+$ contribui em todas as ordens. Usando $\Delta_+ = \sum_{j \geq 1} \Delta_-^j$,

$$\mathcal{D} = \alpha \sum_{j \geq 1} \Delta_-^j + \sum_{j=1}^N \beta_j \Delta_-^j = \sum_{j=1}^N (\alpha + \beta_j) \Delta_-^j + \alpha \sum_{j > N} \Delta_-^j.$$

O termo $\alpha \sum_{j > N} \Delta_-^j = \mathcal{O}(h^{N+1})$ somente se $\alpha = 0$; caso contrário, o erro é $\mathcal{O}(h^{N+1})$ com coeficiente α , o que ainda é admissível, mas para que a expansão coincida com hD até ordem N , precisamos que os coeficientes de Δ_-^j , $j = 1, \dots, N$, sejam $1/j$. Isso dá

$$\alpha + \beta_j = \frac{1}{j}, \quad j = 1, \dots, N.$$

Mas α é um único parâmetro e há N equações; para que o sistema seja consistente para todo N , a única escolha é

$$\boxed{\alpha = 0}.$$

De fato, se $\alpha \neq 0$, o termo $\alpha \Delta_+$ contém contribuições de todas as potências de Δ_- , e não é possível ajustar $\beta_1, \dots, \beta_N$ para cancelar todas as ordens $j \leq N$ simultaneamente, a menos que $\alpha = 0$. Com $\alpha = 0$,

$$\mathcal{D} = \sum_{j=1}^N \beta_j \Delta_-^j, \quad \beta_j = \frac{1}{j},$$

e o erro é $\mathcal{O}(h^{N+1})$ (o primeiro termo omitido é $\Delta_-^{N+1}/(N+1) = \mathcal{O}(h^{N+1})$).

Questão – B5: Com $N = 2$ e $\alpha = 0$,

$$\mathcal{D} = \beta_1 \Delta_- + \beta_2 \Delta_-^2, \quad \beta_1 = 1, \quad \beta_2 = \frac{1}{2}.$$

Assim,

$$hD \approx \Delta_- + \frac{1}{2} \Delta_-^2.$$

Em termos de valores da função:

$$\Delta_- f_i = f_i - f_{i-1},$$

$$\Delta_-^2 f_i = \Delta_-(f_i - f_{i-1}) = f_i - 2f_{i-1} + f_{i-2}.$$

Logo,

$$hf'(x_i) \approx (f_i - f_{i-1}) + \frac{1}{2}(f_i - 2f_{i-1} + f_{i-2}),$$

$$hf'(x_i) \approx \frac{3}{2}f_i - 2f_{i-1} + \frac{1}{2}f_{i-2}.$$

Portanto,

$$f'(x_i) \approx \frac{1}{h} \left(\frac{3}{2}f_i - 2f_{i-1} + \frac{1}{2}f_{i-2} \right) = \frac{3f_i - 4f_{i-1} + f_{i-2}}{2h}.$$

Essa é a conhecida fórmula regressiva de 3 pontos, com erro $\mathcal{O}(h^2)$. Mas o enunciado pede $\mathcal{O}(h^3)$. Para obter $\mathcal{O}(h^3)$ com $N = 2$, é necessário usar $\alpha \neq 0$ e ajustar β_1, β_2 de modo que os coeficientes de Δ_-^1 e Δ_-^2 coincidam com 1 e 1/2, e o termo $\alpha\Delta_+$ seja usado para cancelar o erro de ordem h^3 .

Vamos refazer com $\alpha \neq 0$. Escrevemos

$$\mathcal{D} = \alpha\Delta_+ + \beta_1\Delta_- + \beta_2\Delta_-^2.$$

Usando $\Delta_+ = \Delta_- + \Delta_-^2 + \Delta_-^3 + \dots$,

$$\mathcal{D} = (\alpha + \beta_1)\Delta_- + (\alpha + \beta_2)\Delta_-^2 + \alpha\Delta_-^3 + \dots$$

Para coincidir com $hD = \Delta_- + \frac{1}{2}\Delta_-^2 + \frac{1}{3}\Delta_-^3 + \dots$ até ordem 2 (erro $\mathcal{O}(h^3)$), precisamos

$$\alpha + \beta_1 = 1, \quad \alpha + \beta_2 = \frac{1}{2}.$$

O termo de ordem 3 é $\alpha\Delta_-^3$, que é $\mathcal{O}(h^3)$; para que o erro seja $\mathcal{O}(h^3)$, basta que o coeficiente de Δ_-^3 seja *qualquer* (não precisa coincidir com 1/3), pois já é da ordem do erro permitido. Então há liberdade em α . Escolhemos α de modo a obter uma fórmula com pontos à esquerda e um à direita. Por exemplo, tome $\alpha = \frac{1}{3}$ (para que o coeficiente de Δ_-^3 coincida com 1/3, melhorando a constante):

$$\beta_1 = 1 - \frac{1}{3} = \frac{2}{3}, \quad \beta_2 = \frac{1}{2} - \frac{1}{3} = \frac{1}{6}.$$

Então

$$\mathcal{D} = \frac{1}{3}\Delta_+ + \frac{2}{3}\Delta_- + \frac{1}{6}\Delta_-^2.$$

Em valores:

$$\Delta_+ f_i = f_{i+1} - f_i,$$

$$\Delta_- f_i = f_i - f_{i-1},$$

$$\Delta_-^2 f_i = f_i - 2f_{i-1} + f_{i-2}.$$

Logo,

$$hf'(x_i) \approx \frac{1}{3}(f_{i+1} - f_i) + \frac{2}{3}(f_i - f_{i-1}) + \frac{1}{6}(f_i - 2f_{i-1} + f_{i-2}).$$

Agrupando:

$$hf'(x_i) \approx \frac{1}{3}f_{i+1} + \left(-\frac{1}{3} + \frac{2}{3} + \frac{1}{6}\right)f_i + \left(-\frac{2}{3} - \frac{1}{3}\right)f_{i-1} + \frac{1}{6}f_{i-2}.$$

Calculando os coeficientes:

$$-\frac{1}{3} + \frac{2}{3} + \frac{1}{6} = \frac{-2 + 4 + 1}{6} = \frac{3}{6} = \frac{1}{2},$$

$$-\frac{2}{3} - \frac{1}{3} = -1.$$

Portanto,

$$f'(x_i) \approx \frac{1}{h} \left(\frac{1}{6}f_{i-2} - f_{i-1} + \frac{1}{2}f_i + \frac{1}{3}f_{i+1} \right) = \frac{f_{i-2} - 6f_{i-1} + 3f_i + 2f_{i+1}}{6h}.$$

Parte Computacional

Questão – C1:

Como já mencionado na Questão A, a matriz de iteração é

$$M_{\text{exp}} = I + hA = \begin{pmatrix} 1 & h \\ -3h & 1 - 4h \end{pmatrix}$$

cujos autovalores são $\lambda_1 = 1 - h$, $\lambda_2 = 1 - 3h$.

A condição de estabilidade é $|1 - h| < 1$ e $|1 - 3h| < 1 \iff 0 < h < 2/3 \approx 0.6667$.

Com a função abaixo

```
function euler_sistema(h, T)
    N = round(Int, T/h)
    u1 = zeros(N+1); u2 = zeros(N+1)
    u1[1] = 1.0; u2[1] = 0.0
    for n in 1:N
        u1[n+1] = u1[n] + h*u2[n]
        u2[n+1] = u2[n] + h*(1 - 3u1[n] - 4u2[n])
    end
    return 0:h:T, u1, u2
end
```

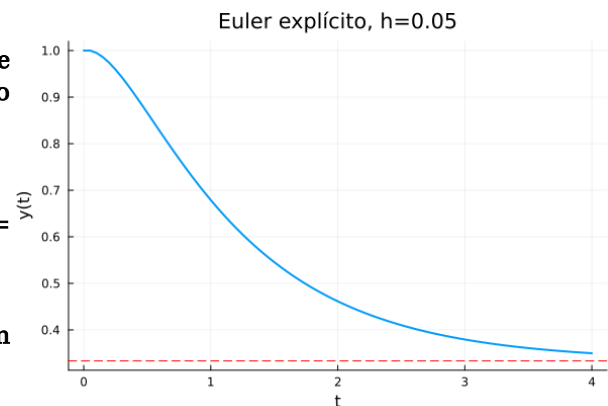
Para $N = 80$, observamos que a solução parte de $y(0) = 1$, decresce suavemente e converge para o equilíbrio

$$y_{\infty} = \frac{1}{3},$$

sem oscilações visíveis. Isso é esperado: para $h = 0.05$,

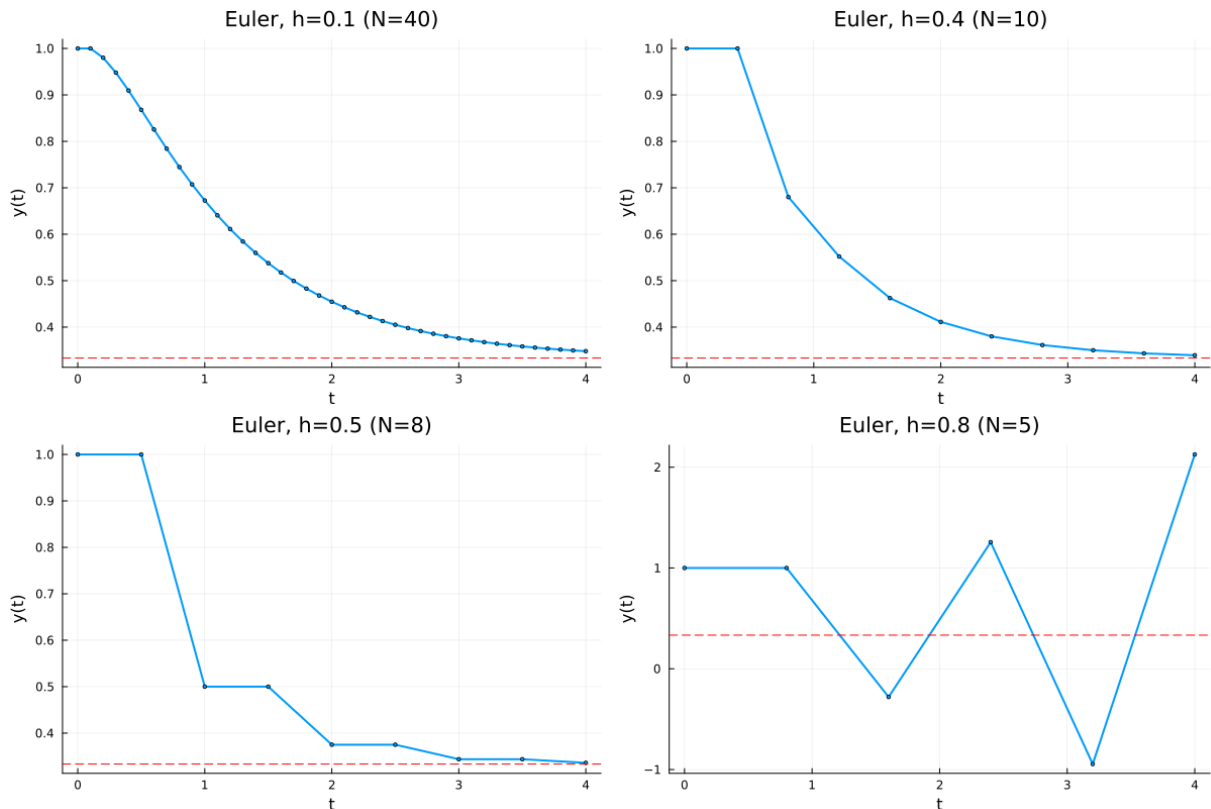
$$\lambda_1 = 0.95, \quad \lambda_2 = 0.85,$$

ambos em $(0, 1)$, de modo que os dois modos decaem monotonamente.



Questão – C2:

Com diferentes números de subintervalos



Subintervalos	h	$\lambda_1 = 1 - h$	$\lambda_2 = 1 - 3h$	Comportamento
80	0.05	0.95	0.85	Convergência monotônica suave
40	0.10	0.90	0.70	Convergência suave, muito próxima da exata
10	0.40	0.60	-0.20	Convergência com oscilação amortecida
8	0.50	0.50	-0.50	Oscilação visível, ainda amortecida
5	0.80	0.20	-1.40	Instável: oscilações crescem

O que se observa é a manifestação numérica direta do critério de estabilidade $0 < h < \frac{2}{3}$ do Euler explícito para este problema: passos pequenos reproduzem a solução suave; passos moderados ($\frac{2}{3} > h \geq \frac{1}{3}$) geram oscilações amortecidas por causa do autovalor negativo; passos grandes ($h > \frac{2}{3}$) produzem divergência.

Questão – C3: Aqui foi empregado o seguinte script com comparação do erro em $t = 2.0$ e geração do gráfico de ordem em escala log com ajuste por mínimos quadrados (outros pacotes como CurveFit.jl ou LsqFit.jl podem ser usados para simplificar como o fit do GNUPlot mas não foram necessários).

```

1 function estudo_ordem(tf=2.0)
2     hs = Float64[]
3     erros = Float64[]
4     razoes = Float64[]
5     erro_ant = NaN
6
7     @printf("y_exato(%.1f) = %.12f\n\n", tf, yex(tf))
8     println("      h          y_num(tf)      |erro|          razão")
9
10    for h in (0.1, 0.05, 0.025, 0.0125, 0.00625)
11        _, y, _ = euler_sistema(h, tf)
12        y_num = y[end]
13        err = abs(y_num - yex(tf))
14        razao = isnan(erro_ant) ? NaN : erro_ant/err
15
16        @printf("%.5f   %.10f   %.3e   %6.3f\n", h, y_num, err, razao)
17
18        push!(hs, h); push!(erros, err); push!(razoes, razao)

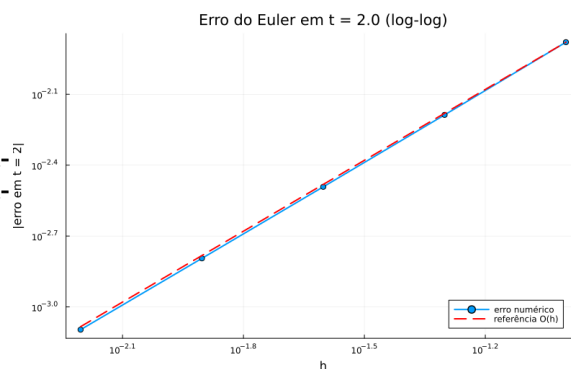
```

```

19     erro_ant = err
20 end
21
22 return hs, erros, razoes
23 end
24
25 hs, erros, razoes = estudo_ordem(2.0)
26
27 # ----- Gráfico do erro em escala log-log -----
28 # Ajuste linear (mínimos quadrados) de log(erro) vs log(h) para estimar a ordem
29 x = log.(hs)
30 y = log.(erros)
31 A = hcat(ones(length(x)), x)
32 coef = A \ y
33 ordem = coef[2]
34
35 p = plot(hs, erros,
36 xscale = :log10, yscale = :log10,
37 marker = :circle, lw = 2,
38 xlabel = "h", ylabel = "|erro em t = 2|",
39 title = "Erro do Euler em t = 2.0 (log-log)",
40 label = "erro numérico",
41 legend = :bottomright,
42 size = (800, 500))

```

onde a reta de referência $\mathcal{O}(h) : C * h$ com C escolhido para passar no primeiro ponto e a Ordem estimada (ajuste log-log) encontrada foi 1.010.



Questão – C3:

Como já vimos, a fórmula obtida nos itens 1 e 5 da Questão B (com $\alpha = \frac{1}{3}$) é a mesma:

$$f'(x_i) \approx \frac{f_{i-2} - 6f_{i-1} + 3f_i + 2f_{i+1}}{6h},$$

com erro teórico $\mathcal{O}(h^3)$. Para comparação, incluímos também a variante do item 5 com $\alpha = 0$ (regressiva de 3 pontos),

$$f'(x_i) \approx \frac{3f_i - 4f_{i-1} + f_{i-2}}{2h},$$

que tem erro $\mathcal{O}(h^2)$. Tomando $f(x) = \sqrt{x}$ em $x_i = 3$, o valor exato é

$$f'(3) = \frac{1}{2\sqrt{3}} \approx 0.2886751345948129.$$

Dessa vez foi incluído `CurveFit.jl` e `CommonSolve.jl` para calcular os coeficientes:

```

1 using CurveFit
2 using CommonSolve: solve
3 using Printf, Plots
4
5 f(x)      = sqrt(x)
6 dfex(x)   = 1/(2*sqrt(x))
7
8 # Formula do item 1 / item 5 com alpha = 1/3 (4 pontos, O(h^3))

```

```

9      df4(f, xi, h) = (f(xi-2h) - 6f(xi-h) + 3f(xi) + 2f(xi+h)) / (6h)
10
11      # Formula do item 5 com alpha = 0 (regressiva 3 pontos, O(h^2))
12      df3reg(f, xi, h) = (3f(xi) - 4f(xi-h) + f(xi-2h)) / (2h)
13
14      xi = 3.0
15      exato = dfex(xi)
16      @printf("f'(3) exato = %.16f\n\n", exato)
17
18      hs = [0.1, 0.05, 0.025, 0.0125, 0.00625, 0.003125]
19
20      erros4 = Float64[]
21      erros3 = Float64[]
22
23      println("      h          df4(h)          |erro4|          df3reg(h)          |erro3|")
24      for h in hs
25          a4 = df4(f, xi, h)
26          a3 = df3reg(f, xi, h)
27          e4 = abs(a4 - exato)
28          e3 = abs(a3 - exato)
29          push!(erros4, e4); push!(erros3, e3)
30          @printf("%.6e   %.15f   %.3e   %.15f   %.3e\n", h, a4, e4, a3, e3)
31      end
32
33      # Ajuste de lei de potencia erro ~ b*h^a via CurveFit.jl
34      sol4 = solve(CurveFitProblem(hs, erros4), PowerCurveFitAlgorithm())
35      sol3 = solve(CurveFitProblem(hs, erros3), PowerCurveFitAlgorithm())
36      b4, a4 = sol4.u
37      b3, a3 = sol3.u
38
39      @printf("\nAjuste item 1/5 (alpha=1/3, 4 pontos): erro ~ %.3e * h^%.4f\n", b4, a4)
40      @printf("Ajuste item 5 (alpha=0, 3 pontos):      erro ~ %.3e * h^%.4f\n", b3, a3)
41
42      # Grafico log-log com retas de referencia
43      p = plot(hs, erros4,
44              xscale = :log10, yscale = :log10,
45              marker = :circle, lw = 2,
46              xlabel = "h", ylabel = "|erro em f'(3)|",
47              title = "Ordem das aproximacoes para f'(3), f(x)=sqrt(x)",
48              label = @sprintf("4 pontos (alpha=1/3), ajuste p=%.2f", a4),
49              legend = :bottomright,
50              size = (800, 500))
51
52      plot!(p, hs, erros3,
53            marker = :square, lw = 2,
54            label = @sprintf("3 pontos regressiva (alpha=0), ajuste p=%.2f", a3))
55
56      href = collect(range(minimum(hs), maximum(hs), length=200))
57      plot!(p, href, erros4[1] .* (href ./ hs[1]).^3,
58            ls = :dash, color = :blue, lw = 1, label = "referencia O(h^3)")
59      plot!(p, href, erros3[1] .* (href ./ hs[1]).^2,
60            ls = :dash, color = :red, lw = 1, label = "referencia O(h^2)")

```

f'(3) exato = 0.2886751345948129

h	df4(h)	erro4	df3reg(h)	erro3
1.000000e-01	0.28867338...	1.754e-06	0.28848997...	1.852e-04
5.000000e-02	0.28867491...	2.208e-07	0.28862933...	4.581e-05
2.500000e-02	0.28867510...	2.771e-08	0.28866371...	1.142e-05
1.250000e-02	0.28867513...	3.466e-09	0.28867227...	2.854e-06
6.250000e-03	0.28867513...	4.333e-10	0.28867442...	7.132e-07
3.125000e-03	0.28867513...	5.417e-11	0.28867496...	1.782e-07

Ajuste item 1/5 (alpha=1/3, 4 pontos): erro ~ 1.75e+00 * h^2.9999

Ajuste item 5 (alpha=0, 3 pontos): erro ~ 1.85e+00 * h^1.9998

- **Item 1 e item 5 (com $\alpha = \frac{1}{3}$):** são a mesma fórmula e o ajuste `CurveFit.jl` confirma

$$\text{erro} \approx 1.75 h^{3.00},$$

isto é, a ordem teórica $\mathcal{O}(h^3)$ é recuperada numericamente. Quando h é reduzido pela metade, o erro cai por um fator $\approx 8 = 2^3$.

- **Item 5 (com $\alpha = 0$):** a variante regressiva de 3 pontos, embora também tenha sido derivada de $N = 2$, tem apenas

$$\text{erro} \approx 1.85 h^{2.00},$$

ou seja, ordem $\mathcal{O}(h^2)$. Reduzindo h pela metade, o erro cai por um fator $\approx 4 = 2^2$.

- **Comparação:** ambas usam $N = 2$ no truncamento do operador $\mathcal{D}$, mas a escolha de α muda a ordem. Com $\alpha = 0$ só há pontos à esquerda (fórmula regressiva); com $\alpha = \frac{1}{3}$, o ponto à direita “paga” a ordem extra e a fórmula de 4 pontos recupera $\mathcal{O}(h^3)$. Esse é exatamente o papel do parâmetro α discutido no item 4: a aproximação $\mathcal{O}(h^3)$ exige $\alpha \neq 0$.

O gráfico log-log evidencia as duas inclinações (3 e 2), que ficam alinhadas com as retas de referência $\mathcal{O}(h^3)$ e $\mathcal{O}(h^2)$. Isso confirma numericamente tanto o resultado analítico do item 1 quanto o do item 5 com $\alpha = \frac{1}{3}$, e mostra que a variante com $\alpha = 0$ perde uma ordem.

