Resolução da Lista de Exercícios

Francisco Davi Belo Rodrigues

1 de novembro de 2025

1 Questão 1

Enunciado e dados

Consideram-se dois tanques cilíndricos interligados em série. O tanque 1 recebe uma alimentação constante e descarrega no tanque 2, que por sua vez escoa para o ambiente. As vazões de saída de cada tanque dependem do nível interno segundo a relação empírica $Q_i = k_i\sqrt{h_i}$. Os parâmetros fornecidos são resumidos na Tabela 1.

Parâmetro	**Valor**
Vazão de alimentação Q_0	$20\ \mathrm{m^3\,h^{-1}}$
Diâmetro do tanque 1 D_1	4 m
Diâmetro do tanque 2 D_2	3 m
Constante da válvula 1 k_1	$14\ \mathrm{m^{2.5}\,h^{-1}}$
Constante da válvula 2 k_2	$12\ \mathrm{m^{2.5}\,h^{-1}}$
Nível inicial no tanque 1 $h_1(0)$	3 m
Nível inicial no tanque 2 $h_2(0)$	2 m

Tabela 1: Dados operacionais da Questão 1.

Formulação do modelo

O modelo dinâmico é obtido a partir dos balanços de massa (ou volume, dado que a densidade é constante) em cada tanque. Para um volume de controle genérico com densidade constante ρ, tem-se

$$\frac{d(\rho V)}{dt} = \rho Q_{\mathrm{in}} - \rho Q_{\mathrm{out}},$$

o que conduz ao balanço volumétrico

$$\frac{dV}{dt} = Q_{\mathrm{in}} - Q_{\mathrm{out}}.$$

No tanque 1, o volume contido é $V_1 = A_1 h_1$, com A_1 indicado na Eq. (1). O balanço volumétrico resulta em

$$A_1 \frac{dh_1}{dt} = Q_0 - Q_1,$$

em que Q_0 é a vazão de alimentação e Q_1 a vazão de saída do tanque 1. De modo análogo, para o tanque 2 obtém-se $V_2 = A_2 h_2$ e

$$A_2 \frac{dh_2}{dt} = Q_1 - Q_2,$$

com Q_1 proveniente do tanque 1 e Q_2 a descarga para o ambiente.

As vazões de saída seguem a correlação empírica das válvulas e as áreas dos tanques são calculadas pela geometria cilíndrica. Mantendo t em horas, têm-se as equações numeradas finais:

$$A_i = \frac{\pi D_i^2}{4}, \quad i = 1, 2, \tag{1}$$

$$Q_1 = k_1 \sqrt{h_1}, \tag{2}$$

$$Q_2 = k_2 \sqrt{h_2}, \tag{3}$$

$$A_1 \frac{dh_1}{dt} = Q_0 - Q_1, \tag{4}$$

$$A_2 \frac{dh_2}{dt} = Q_1 - Q_2, \tag{5}$$

com condições iniciais $h_1(0) = 3$ m e $h_2(0) = 2$ m. Este conjunto de equações está pronto para utilização em ambientes de simulação como o EMSO, onde os parâmetros podem ser definidos separadamente sem substituição numérica antecipada.

Resolução numérica

O sistema diferencial foi integrado em $0 \leq t \leq 20$ h empregando o método Runge–Kutta de quarta/quinta ordem adaptativo (`solve_ivp` do SciPy) com passo máximo equivalente a 10 s após conversão interna de unidades no script de apoio. A implementação registra também as trajetórias discretizadas (t, h_1, h_2) em arquivo auxiliar para rastreabilidade.

```
import numpy as np
from math import pi
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

Q0 = 20.0 # m^3/h
D1 = 4.0 # m
D2 = 3.0 # m
k1 = 14.0 # m^{2.5}/h
k2 = 12.0 # m^{2.5}/h
h1_0 = 3.0 # m
h2_0 = 2.0 # m

A1 = pi * (D1 ** 2) / 4.0
A2 = pi * (D2 ** 2) / 4.0

T_sim = 20.0

def model(t, y):
    h1, h2 = y
    q1 = k1 * np.sqrt(max(h1, 0.0))
    q2 = k2 * np.sqrt(max(h2, 0.0))
    dh1dt = (Q0 - q1) / A1
    dh2dt = (q1 - q2) / A2
    return [dh1dt, dh2dt]

sol = solve_ivp(
    fun=model,
    t_span=(0.0, T_sim),
    y0=[h1_0, h2_0],
    max_step=0.01,
    dense_output=True
    )

```

```python
t_hours = np.linspace(0.0, T_sim, 100)
h1 = sol.sol(t_hours)[0]
h2 = sol.sol(t_hours)[1]

plt.figure(figsize=(6, 4))
plt.plot(t_hours, h1, label="h1 (m)")
plt.plot(t_hours, h2, label="h2 (m)")
plt.xlabel("Tempo (h)")
plt.ylabel("Nível (m)")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig("figuras/questao1_niveis.png", dpi=300)

with open("figuras/questao1_niveis.dat", "w", encoding="utf-8") as f:
    f.write("tempo_h h1_m h2_m\n")
    for t, hv1, hv2 in zip(t_hours, h1, h2):
        f.write(f"{t:.6f} {hv1:.6f} {hv2:.6f}\n")

print("h1 final:", h1[-1])
print("h2 final:", h2[-1])
```

Listing 1: Script Python utilizado para a integracao numerica da Questao 1.

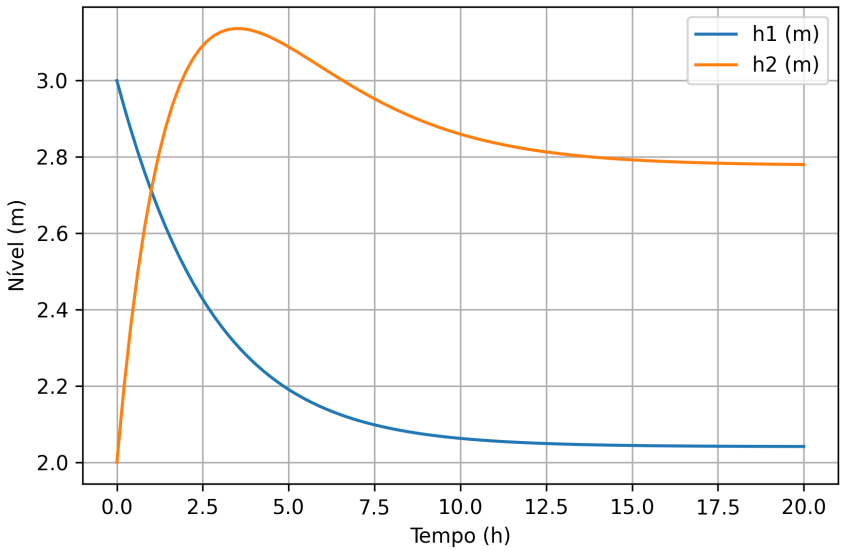

Figura 1: Perfis temporais simulados dos níveis h_1 e h_2 durante 20 horas.

2 Questão 2

Enunciado e dados

Um tanque de mistura perfeitamente agitado recebe duas correntes de entrada e tem uma corrente de saída. As correntes contêm três componentes (A, B e C) em diferentes proporções. A vazão de saída depende do nível de líquido no tanque segundo a relação $F_3 = \rho_3 k\sqrt{h}$, onde ρ_3 é a densidade da corrente de saída. Os parâmetros fornecidos são resumidos na Tabela 2.

Parâmetro	Valor
Vazão da corrente 1 F_1	10 $\mathrm{kg\,min^{-1}}$
Fração mássica de A na corrente 1 x_{A1}	0,6
Fração mássica de B na corrente 1 x_{B1}	0,0
Fração mássica de C na corrente 1 x_{C1}	0,4
Vazão da corrente 2 F_2	8 $\mathrm{kg\,min^{-1}}$
Fração mássica de A na corrente 2 x_{A2}	0,0
Fração mássica de B na corrente 2 x_{B2}	0,7
Fração mássica de C na corrente 2 x_{C2}	0,3
Densidade do componente A ρ_A	1200 $\mathrm{kg\,m^{-3}}$
Densidade do componente B ρ_B	1400 $\mathrm{kg\,m^{-3}}$
Densidade do componente C ρ_C	1000 $\mathrm{kg\,m^{-3}}$
Área da seção transversal A	0,2 $\mathrm{m^2}$
Parâmetro da válvula k	0,02 $\mathrm{m^{2.5}\,min^{-1}}$
Massa inicial de A m_{A0}	20 kg
Massa inicial de B m_{B0}	20 kg
Massa inicial de C m_{C0}	40 kg

Tabela 2: Dados operacionais da Questão 2.

Formulação do modelo

O modelo dinâmico é obtido a partir dos balanços de massa individuais para cada componente. Para um componente genérico i (A, B ou C), o balanço mássico no volume de controle resulta em

$$\frac{dm_i}{dt} = F_1 x_{i1} + F_2 x_{i2} - F_3 x_{i3},$$

onde m_i é a massa do componente i no tanque, x_{i1} e x_{i2} são as frações mássicas nas correntes de entrada, e x_{i3} é a fração mássica na corrente de saída.

Como o tanque é perfeitamente agitado, a composição de saída é igual à composição interna. Assim, a fração mássica de cada componente na corrente de saída é dada por

$$x_{i3} = \frac{m_i}{m_{\text{total}}}, \quad m_{\text{total}} = m_A + m_B + m_C. \tag{6}$$

Para calcular a densidade da mistura na corrente de saída, considera-se o comportamento de uma mistura ideal, na qual os volumes dos componentes puros são aditivos. Essa hipótese é adequada para misturas líquidas sem interações moleculares significativas (mistura ideal). Assim, o volume total da mistura é dado pela soma dos volumes individuais:

$$V_{\text{total}} = V_A + V_B + V_C.$$

Como cada volume parcial pode ser expresso em termos da massa e densidade do componente ($V_i = m_i/\rho_i$), obtém-se

$$V_{\text{total}} = \frac{m_A}{\rho_A} + \frac{m_B}{\rho_B} + \frac{m_C}{\rho_C}.$$

Por outro lado, o volume total relaciona-se com a massa total e a densidade da mistura por $V_{\text{total}} = m_{\text{total}}/\rho_3$. Igualando as duas expressões, tem-se

$$\frac{m_{\text{total}}}{\rho_3} = \frac{m_A}{\rho_A} + \frac{m_B}{\rho_B} + \frac{m_C}{\rho_C}.$$

Dividindo ambos os lados por m_{total} e utilizando a definição de fração mássica $x_i = m_i/m_{\text{total}}$, chega-se a

$$\frac{1}{\rho_3} = \frac{x_A}{\rho_A} + \frac{x_B}{\rho_B} + \frac{x_C}{\rho_C},$$

de onde resulta a densidade da mistura na corrente de saída:

$$\rho_3 = \frac{1}{\frac{x_A}{\rho_A} + \frac{x_B}{\rho_B} + \frac{x_C}{\rho_C}}. \tag{7}$$

O volume total no tanque é obtido pela soma dos volumes de cada componente:

$$V = \frac{m_A}{\rho_A} + \frac{m_B}{\rho_B} + \frac{m_C}{\rho_C}, \tag{8}$$

e o nível de líquido é

$$h = \frac{V}{A}. \tag{9}$$

A vazão de saída é calculada por

$$F_3 = \rho_3 k \sqrt{h}. \tag{10}$$

O sistema de equações diferenciais finais é

$$\frac{dm_A}{dt} = F_1 x_{A1} + F_2 x_{A2} - F_3 x_A, \tag{11}$$

$$\frac{dm_B}{dt} = F_1 x_{B1} + F_2 x_{B2} - F_3 x_B, \tag{12}$$

$$\frac{dm_C}{dt} = F_1 x_{C1} + F_2 x_{C2} - F_3 x_C, \tag{13}$$

com condições iniciais $m_A(0) = 20$ kg, $m_B(0) = 20$ kg e $m_C(0) = 40$ kg.

Resolução numérica

O sistema diferencial foi integrado em $0 \le t \le 60$ min empregando o método Runge–Kutta de quarta/-quinta ordem adaptativo (`solve_ivp` do SciPy) com passo máximo equivalente a 0,01 min. A implementação registra também as trajetórias discretizadas (t, h, x_A, x_B, x_C) em arquivo auxiliar para rastreabilidade.

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

F1 = 10.0
xA1 = 0.6
xB1 = 0.0
xC1 = 0.4
F2 = 8.0
xA2 = 0.0
xB2 = 0.7
xC2 = 0.3
rhoA = 1200.0
rhoB = 1400.0
rhoC = 1000.0
A = 0.2
k = 0.02
mA0 = 20.0
mB0 = 20.0
mC0 = 40.0

T_sim = 60.0

def model(t, y):
    mA, mB, mC = y
    m_total = mA + mB + mC
```

```
    xA = mA / m_total
    xB = mB / m_total
    xC = mC / m_total
    rho3 = 1.0 / (xA / rhoA + xB / rhoB + xC / rhoC)
    V = mA / rhoA + mB / rhoB + mC / rhoC
    h = V / A
    F3 = rho3 * k * np.sqrt(max(h, 0.0))
    dmAdt = F1 * xA1 + F2 * xA2 - F3 * xA
    dmBdt = F1 * xB1 + F2 * xB2 - F3 * xB
    dmCdt = F1 * xC1 + F2 * xC2 - F3 * xC
    return [dmAdt, dmBdt, dmCdt]

sol = solve_ivp(
    fun=model,
    t_span=(0.0, T_sim),
    y0=[mA0, mB0, mC0],
    max_step=0.01,
    dense_output=True
    )

t_min = np.linspace(0.0, T_sim, 200)
mA = sol.sol(t_min)[0]
mB = sol.sol(t_min)[1]
mC = sol.sol(t_min)[2]

m_total = mA + mB + mC
xA = mA / m_total
xB = mB / m_total
xC = mC / m_total
V = mA / rhoA + mB / rhoB + mC / rhoC
h = V / A

fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(6, 6))

ax1.plot(t_min, h, label="h (m)", color='blue')
ax1.set_xlabel("Tempo (min)")
ax1.set_ylabel("Nível (m)")
ax1.legend()
ax1.grid(True)

ax2.plot(t_min, xA, label="xA", color='red')
ax2.plot(t_min, xB, label="xB", color='green')
ax2.plot(t_min, xC, label="xC", color='purple')
ax2.set_xlabel("Tempo (min)")
ax2.set_ylabel("Fração mássica")
ax2.legend()
ax2.grid(True)

plt.tight_layout()
plt.savefig("figuras/questao2_tanque.png", dpi=300)

with open("figuras/questao2_tanque.dat", "w", encoding="utf-8") as f:
    f.write("tempo_min h_m xA xB xC\n")
    for t, hval, xa, xb, xc in zip(t_min, h, xA, xB, xC):
        f.write(f"{t:.6f} {hval:.6f} {xa:.6f} {xb:.6f} {xc:.6f}\n")

print("h final:", h[-1])
print("xA final:", xA[-1])
print("xB final:", xB[-1])
```

```
86 print("xC final:", xC[-1])
```

Listing 2: Script Python utilizado para a integracao numerica da Questao 2.

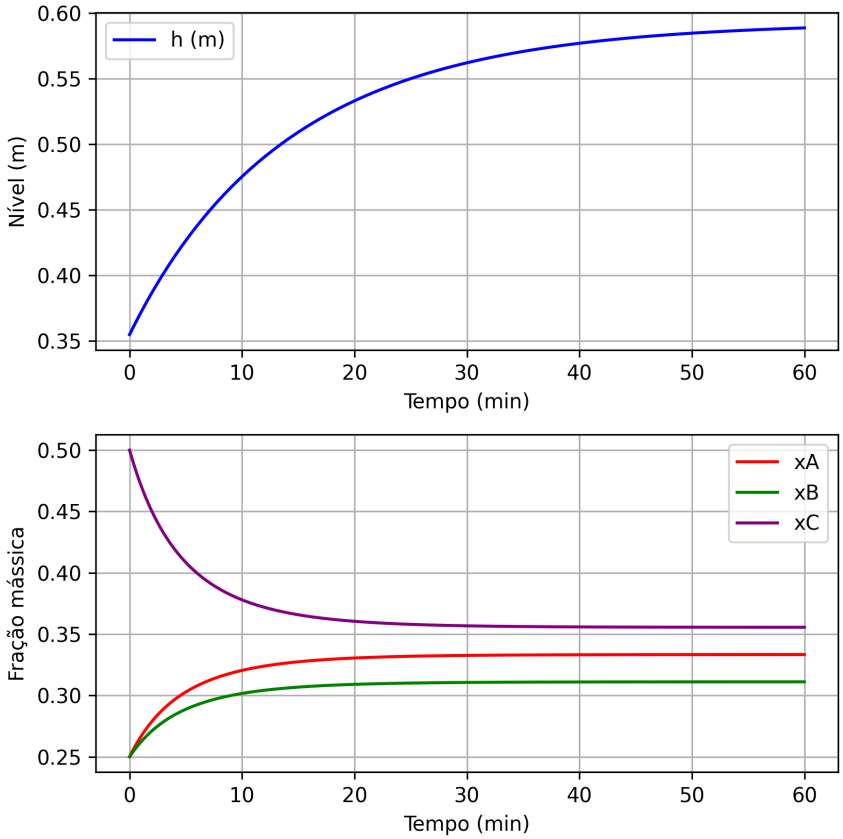

Figura 2: Perfis temporais simulados do nível h e das frações mássicas x_A, x_B e x_C durante 60 minutos.

3 Questão 3

Enunciado e dados

Um reator do tipo BSTR (batelada) é utilizado para executar a seguinte reação reversível:

$$\mathrm{A} + \mathrm{B} \rightleftharpoons \mathrm{C} + \mathrm{D}.$$

As taxas [mol·L^{-1}·min^{-1}] das reações direta e inversa podem ser estimadas segundo as seguintes equações:

$$r_d = k_d C_A^{1,1} C_B^{1,4}, \tag{14}$$

$$r_i = k_i C_C^{1,31} C_D^{1,2}, \tag{15}$$

em que C_A, C_B, C_C e C_D são as concentrações [mol·L^{-1}], T é a temperatura [K], e as constantes cinéticas k_d e k_i são dadas por

$$k_d = 50\,000 \exp\left(-\frac{4\,000}{T}\right), \tag{16}$$

$$k_i = 30\,000 \exp\left(-\frac{5\,000}{T}\right). \tag{17}$$

Inicialmente são colocados no reator somente os reagentes A e B, com concentrações de 0,5 $\mathrm{mol\,L^{-1}}$ e 0,8 $\mathrm{mol\,L^{-1}}$, respectivamente. O reator apresenta um sistema de ajuste de temperatura que inicialmente está fixado em 400 K. Essa temperatura é mantida constante durante os primeiros 5 minutos de batelada. Passado esse tempo, inicia-se uma rampa de diminuição da temperatura de forma que ao final da batelada a temperatura alcance o valor de 350 K.

Formulação do modelo

O modelo dinâmico é obtido a partir dos balanços de massa para cada componente no reator em batelada. Para um reator em batelada, não há termos de entrada ou saída de massa, logo o balanço de massa para cada componente é dado por

$$\frac{dC_i}{dt} = r_i,$$

onde C_i é a concentração do componente i e r_i é a taxa líquida de geração/consumo do respectivo componente.

Para a reação reversível $\mathrm{A} + \mathrm{B} \rightleftharpoons \mathrm{C} + \mathrm{D}$, as velocidades das reações direta e inversa são dadas em (14) e (15), e a taxa líquida de reação é

$$r_{\mathrm{net}} = r_d - r_i, \tag{18}$$

onde r_d é a velocidade da reação direta e r_i a da reação inversa.

Os balanços de massa para cada componente seguem a estequiometria da reação:

$$\frac{dC_A}{dt} = -r_{\mathrm{net}}, \tag{19}$$

$$\frac{dC_B}{dt} = -r_{\mathrm{net}}, \tag{20}$$

$$\frac{dC_C}{dt} = \ r_{\mathrm{net}}, \tag{21}$$

$$\frac{dC_D}{dt} = \ r_{\mathrm{net}}, \tag{22}$$

com condições iniciais

$$C_A(0) = 0{,}5\ \mathrm{mol\,L^{-1}}, \quad C_B(0) = 0{,}8\ \mathrm{mol\,L^{-1}}, \quad C_C(0) = 0\ \mathrm{mol\,L^{-1}}, \quad C_D(0) = 0\ \mathrm{mol\,L^{-1}}.$$

A temperatura do reator segue o perfil de operação especificado:

$$T(t) = \begin{cases} 400\ \mathrm{K}, & \text{se } t \le 5\ \mathrm{min}, \\ 400 - \dfrac{50}{10}(t-5)\ \mathrm{K}, & \text{se } t > 5\ \mathrm{min}, \end{cases} \tag{23}$$

ou seja, permanece em 400 K nos primeiros 5 min e decresce linearmente até 350 K ao final da batelada (15 min).

A conversão do reagente A é definida como

$$X_A = \frac{C_{A0} - C_A}{C_{A0}}, \tag{24}$$

onde $C_{A0} = 0{,}5\ \mathrm{mol\,L^{-1}}$ é a concentração inicial de A.

Resolução numérica

O sistema de equações ordinais foi integrado em $0 \le t \le 15$ min empregando o método Runge–Kutta de quarta/quinta ordem adaptativo (`solve_ivp` do SciPy) com passo máximo equivalente a 0,01 min. A implementação registra também as trajetórias discretizadas $(t, T, C_A, C_B, C_C, C_D, X_A)$ em arquivo auxiliar para rastreabilidade.

```python
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

CA0 = 0.5
CB0 = 0.8
CC0 = 0.0
CD0 = 0.0

T_sim = 15.0

def model(t, y):
    CA, CB, CC, CD = y
    if t <= 5.0:
        T = 400.0
    else:
        T = 400.0 - (400.0 - 350.0) * (t - 5.0) / (T_sim - 5.0)
    # constantes de velocidade direta e inversa atualizadas
    kd = 50000.0 * np.exp(-4000.0 / T)
    ki = 30000.0 * np.exp(-5000.0 / T)
    # taxas com os expoentes especificados
    rd = kd * (CA ** 1.1) * (CB ** 1.4)
    ri = ki * (CC ** 1.31) * (CD ** 1.2)
    r_net = rd - ri
    dCAdt = -r_net
    dCBdt = -r_net
    dCCdt = r_net
    dCDdt = r_net
    return [dCAdt, dCBdt, dCCdt, dCDdt]

sol = solve_ivp(
    fun=model,
    t_span=(0.0, T_sim),
    y0=[CA0, CB0, CC0, CD0],
    max_step=0.01,
    dense_output=True
    )

t_min = np.linspace(0.0, T_sim, 300)
CA = sol.sol(t_min)[0]
CB = sol.sol(t_min)[1]
CC = sol.sol(t_min)[2]
CD = sol.sol(t_min)[3]

T = np.where(t_min <= 5.0, 400.0, 400.0 - (400.0 - 350.0) * (t_min - 5.0) / (T_sim - 5.0))
XA = (CA0 - CA) / CA0

fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(6, 8))

ax1.plot(t_min, T, label="T (K)", color='red')
ax1.set_xlabel("Tempo (min)")
ax1.set_ylabel("Temperatura (K)")
ax1.legend()
ax1.grid(True)

ax2.plot(t_min, CA, label="CA", color='blue')
ax2.plot(t_min, CB, label="CB", color='green')
ax2.plot(t_min, CC, label="CC", color='orange')
```

```
ax2.plot(t_min, CD, label="CD", color='purple')
ax2.set_xlabel("Tempo (min)")
ax2.set_ylabel("Concentração (mol/L)")
ax2.legend()
ax2.grid(True)

ax3.plot(t_min, XA, label="XA", color='black')
ax3.set_xlabel("Tempo (min)")
ax3.set_ylabel("Conversão de A")
ax3.legend()
ax3.grid(True)

plt.tight_layout()
plt.savefig("figuras/questao3_reator.png", dpi=300)

with open("figuras/questao3_reator.dat", "w", encoding="utf-8") as f:
    f.write("tempo_min T_K CA CB CC CD XA\n")
    for t, temp, ca, cb, cc, cd, xa in zip(t_min, T, CA, CB, CC, CD, XA):
        f.write(f"{t:.6f} {temp:.6f} {ca:.6f} {cb:.6f} {cc:.6f} {cd:.6f} {xa:.6f}\n")

print("T final:", T[-1])
print("CA final:", CA[-1])
print("CB final:", CB[-1])
print("CC final:", CC[-1])
print("CD final:", CD[-1])
print("XA final:", XA[-1])
```

Listing 3: Script Python utilizado para a integraÃ§Ã£o numÃ©rica da QuestÃ£o 3.

4 Questão 4

Enunciado e dados

Considera-se um sistema com reator tubular de dispersão axial e reciclo. O modelo adimensional do reator é dado pela equação de dispersão–advecção–reação:

$$\frac{\partial C(t,z)}{\partial t} + \frac{\partial C(t,z)}{\partial z} = \frac{1}{Pe}\,\frac{\partial^2 C(t,z)}{\partial z^2} - Da\,C(t,z), \tag{25}$$

em que $C(t,z)$ é a concentração adimensional, $z \in [0,1]$ é a coordenada axial adimensional, Pe é o número de Péclet e Da é o número de Damköhler.

As condições de contorno são

$$C(t,0) - \frac{1}{Pe}\,\frac{\partial C(t,z)}{\partial z}\bigg|_{z=0} = C_{\text{alim}}, \tag{26}$$

$$\frac{\partial C(t,z)}{\partial z}\bigg|_{z=1} = 0, \tag{27}$$

onde C_{alim} é a concentração de alimentação do reator, obtida a partir do balanço no ponto de mistura:

$$C_{\text{alim}} = \frac{Q_F C_F + R Q_F C(t,1)}{Q_F(1+R)}, \tag{28}$$

sendo R a razão de reciclo ($R = Q_{\text{rec}}/Q_F$), Q_F a vazão de entrada e C_F a concentração na corrente de alimentação fresca.

Os parâmetros fornecidos são resumidos na Tabela 3.

Parâmetro	Valor
Razão de reciclo R	5
Número de Péclet Pe	15
Número de Damköhler Da	1
Concentração da corrente fresca C_F	1
Vazão de entrada Q_F	10
Condição inicial	$C(0,z) = 0$ (reator vazio)

Tabela 3: Dados operacionais da Questão 4.

Discretização espacial

A equação diferencial parcial (25) foi discretizada pelo método de diferenças finitas na coordenada espacial z, empregando $N = 60$ pontos uniformemente distribuídos no intervalo $[0, 1]$. O espaçamento é $\Delta z = 1/(N-1)$. As derivadas espaciais são aproximadas por diferenças centradas:

$$\left.\frac{\partial C}{\partial z}\right|_{z_i} \approx \frac{C_i - C_{i-1}}{\Delta z}, \tag{29}$$

$$\left.\frac{\partial^2 C}{\partial z^2}\right|_{z_i} \approx \frac{C_{i+1} - 2C_i + C_{i-1}}{(\Delta z)^2}, \tag{30}$$

onde $C_i \equiv C(t, z_i)$.

Condição de contorno em $z = 0$: A condição (26) pode ser reescrita como

$$C_0 - \frac{1}{Pe}\,\frac{C_1 - C_0}{\Delta z} = C_{\text{alim}},$$

de onde se obtém

$$C_0 = \frac{Pe\,\Delta z\, C_{\text{alim}} + C_1}{Pe\,\Delta z + 1}. \tag{31}$$

Entretanto, para fins de implementação numérica, pode-se também expressar a derivada em função de C_{alim} diretamente. Utilizando a aproximação por diferenças avançadas na posição $z = 0$, obtém-se:

$$\frac{dC_0}{dt} = -\frac{C_0 - C_{\text{left}}}{\Delta z} + \frac{1}{Pe}\,\frac{C_1 - 2C_0 + C_{\text{left}}}{(\Delta z)^2} - Da\, C_0, \tag{32}$$

onde C_{left} é obtido de (26) utilizando diferenças finitas:

$$C_{\text{left}} = C_{\text{alim}} - \frac{1}{Pe}\,\frac{C_1 - C_{\text{alim}}}{\Delta z}. \tag{33}$$

Pontos internos ($i = 1, 2, \ldots, N-2$): A equação discretizada no interior do domínio é

$$\frac{dC_i}{dt} = -\frac{C_i - C_{i-1}}{\Delta z} + \frac{1}{Pe}\,\frac{C_{i+1} - 2C_i + C_{i-1}}{(\Delta z)^2} - Da\, C_i. \tag{34}$$

Condição de contorno em $z = 1$: A condição de Neumann homogênea (27) implica que $\partial C/\partial z = 0$ na saída. Utilizando diferenças centradas, tem-se

$$\frac{C_N - C_{N-1}}{\Delta z} = 0 \quad \Rightarrow \quad C_N = C_{N-1}.$$

A equação de evolução para o último ponto é então

$$\frac{dC_N}{dt} = -\frac{C_N - C_{N-1}}{\Delta z} + \frac{1}{Pe}\,\frac{C_{N-1} - 2C_N + C_{N-1}}{(\Delta z)^2} - Da\, C_N. \tag{35}$$

O sistema resultante consiste em N equações diferenciais ordinárias acopladas, sujeitas à condição inicial $C_i(0) = 0$ para todo i.

Resolução numérica

O sistema de EDOs foi integrado no intervalo $0 \leq t \leq 5$ (unidades de tempo adimensionais) empregando o método Runge–Kutta de quarta/quinta ordem adaptativo (`solve_ivp` do SciPy) com passo máximo de 0,01. A cada instante de tempo, a concentração de alimentação C_{alim} é recalculada segundo (28), utilizando a concentração na posição de saída do reator $C(t, 1)$. A implementação registra também as trajetórias discretizadas $(t, C_{\text{alim}}, C_{z=0,25}, C_{z=0,50}, C_{z=0,75}, C_{z=1})$ em arquivo auxiliar para rastreabilidade.

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

R = 5.0
Pe = 15.0
Da = 1.0
CF = 1.0
QF = 10.0

N = 60
dz = 1.0 / (N - 1)

T_sim = 5.0

def model(t, C_vec):
    C = C_vec.copy()

    C_saida = C[-1]
    C_alim = (QF * CF + R * QF * C_saida) / (QF * (1 + R))

    dCdt = np.zeros(N)

    i = 0
    C_left = C_alim - (1 / Pe) * (C[1] - C_alim) / dz
    dCdt[i] = -(C[i] - C_left) / dz + (1 / Pe) * (C[i+1] - 2*C[i] + C_left) / (dz**2) - Da * C[i]

    for i in range(1, N-1):
        dCdt[i] = -(C[i] - C[i-1]) / dz + (1 / Pe) * (C[i+1] - 2*C[i] + C[i-1]) / (dz**2) - Da * C[i
            ]

    i = N - 1
    C_right = C[i-1]
    dCdt[i] = -(C[i] - C[i-1]) / dz + (1 / Pe) * (C_right - 2*C[i] + C[i-1]) / (dz**2) - Da * C[i]

    return dCdt

C0 = np.zeros(N)

sol = solve_ivp(
    fun=model,
    t_span=(0.0, T_sim),
    y0=C0,
    max_step=0.01,
    dense_output=True
)

t_vals = np.linspace(0.0, T_sim, 300)
C_all = sol.sol(t_vals)

z_vals = np.linspace(0.0, 1.0, N)
```

```
51
52 idx_025 = int(0.25 / dz)
53 idx_050 = int(0.50 / dz)
54 idx_075 = int(0.75 / dz)
55 idx_100 = N - 1
56
57 C_025 = C_all[idx_025, :]
58 C_050 = C_all[idx_050, :]
59 C_075 = C_all[idx_075, :]
60 C_100 = C_all[idx_100, :]
61
62 C_alim_vals = np.zeros_like(t_vals)
63 for j, t in enumerate(t_vals):
64     C_saida = C_all[-1, j]
65     C_alim_vals[j] = (QF * CF + R * QF * C_saida) / (QF * (1 + R))
66
67 fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(8, 8))
68
69 ax1.plot(t_vals, C_alim_vals, label="C_alim", color='red', linewidth=2)
70 ax1.set_xlabel("Tempo (adimensional)")
71 ax1.set_ylabel("Concentração C_alim (adimensional)")
72 ax1.legend()
73 ax1.grid(True)
74
75 ax2.plot(t_vals, C_025, label="z = 0.25", color='blue')
76 ax2.plot(t_vals, C_050, label="z = 0.50", color='green')
77 ax2.plot(t_vals, C_075, label="z = 0.75", color='orange')
78 ax2.plot(t_vals, C_100, label="z = 1.00", color='purple')
79 ax2.set_xlabel("Tempo (adimensional)")
80 ax2.set_ylabel("Concentração (adimensional)")
81 ax2.legend()
82 ax2.grid(True)
83
84 plt.tight_layout()
85 plt.savefig("figuras/questao4_reator_dispersao.png", dpi=300)
86
87 with open("figuras/questao4_reator_dispersao.dat", "w", encoding="utf-8") as f:
88     f.write("tempo C_alim C_z025 C_z050 C_z075 C_z100\n")
89     for t, ca, c1, c2, c3, c4 in zip(t_vals, C_alim_vals, C_025, C_050, C_075, C_100):
90         f.write(f"{t:.6f} {ca:.6f} {c1:.6f} {c2:.6f} {c3:.6f} {c4:.6f}\n")
91
92 print("C_alim final:", C_alim_vals[-1])
93 print("C (z=0.25) final:", C_025[-1])
94 print("C (z=0.50) final:", C_050[-1])
95 print("C (z=0.75) final:", C_075[-1])
96 print("C (z=1.00) final:", C_100[-1])
```

Listing 4: Script Python utilizado para a integraÃ§Ã£o numÃ©rica da QuestÃ£o 4.

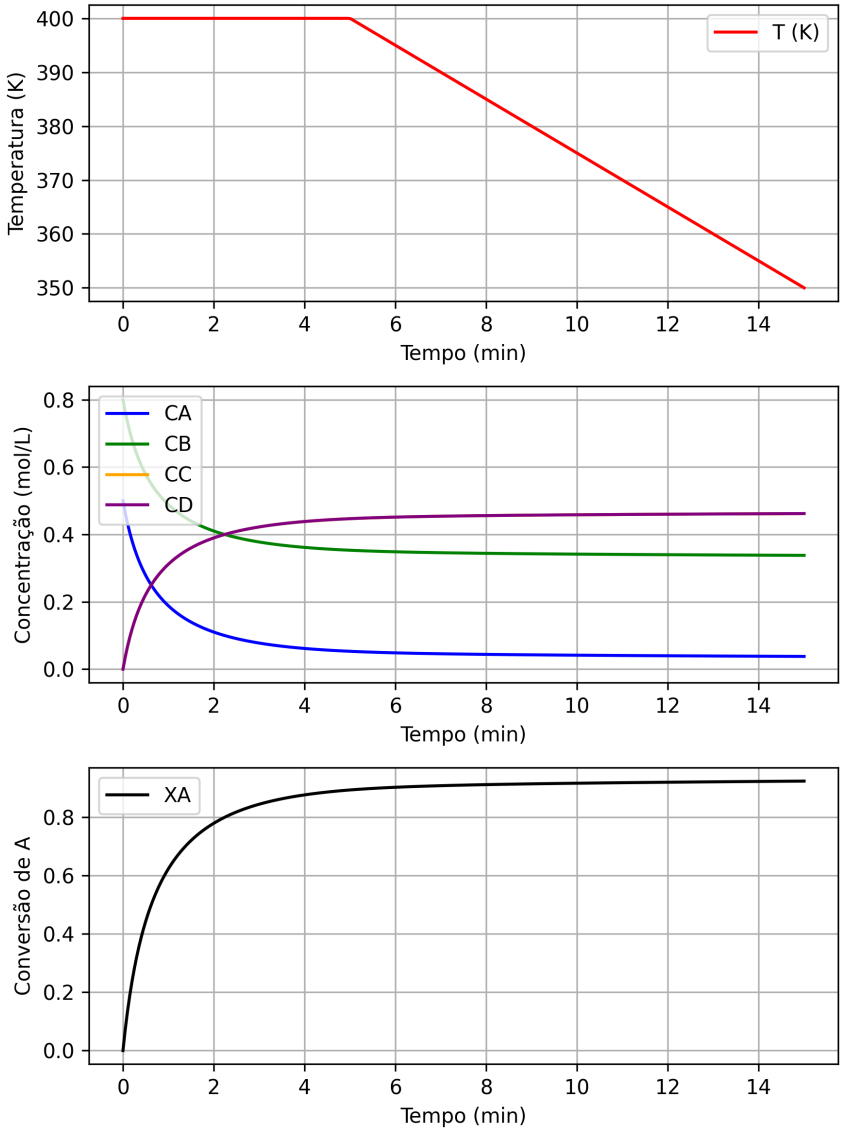

Figura 3: Perfis temporais simulados da temperatura, concentrações e conversão do reagente A durante 15 minutos de batelada.

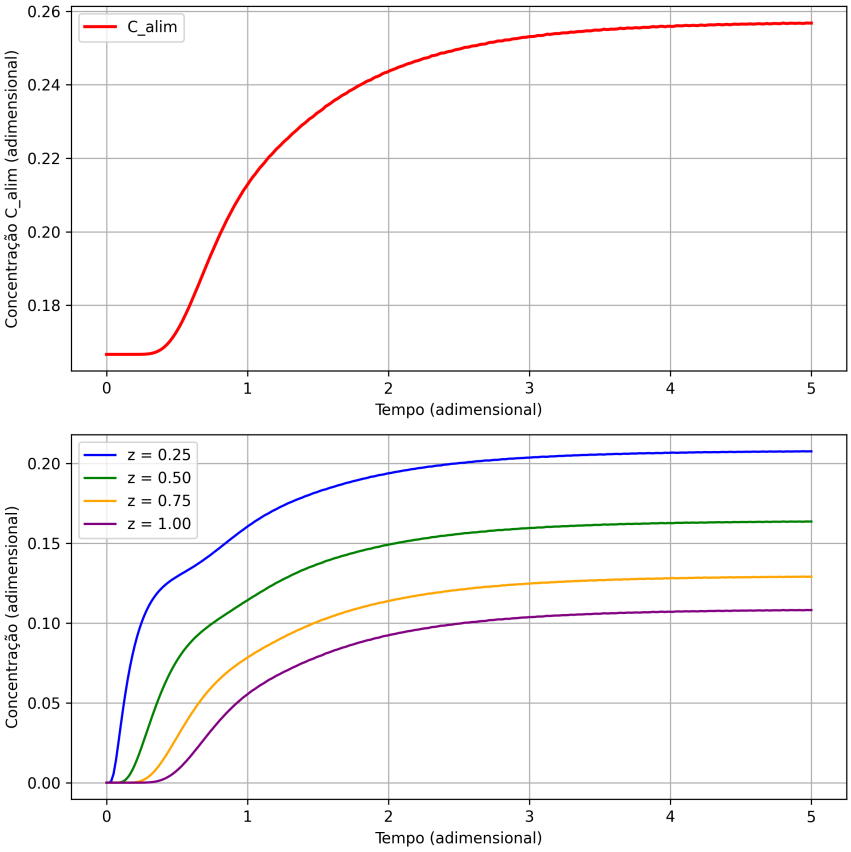

Figura 4: Perfis temporais simulados da concentração de alimentação C_{alim} e das concentrações em diferentes posições axiais do reator ($z = 0{,}25$, $z = 0{,}50$, $z = 0{,}75$ e $z = 1{,}00$) durante 5 unidades de tempo adimensionais.