



## 1ª Tarefa Avaliativa

Aluno: Estêvão Mendes Santana

Matrícula: 201413383

Data: 26 de maio de 2023

### 1) REATORES BATELADA

#### 1.1) Problema 13-7b

(a) Para uma operação adiabática e com temperatura inicial de 278 K, plote  $T$ ,  $I'$ ,  $r_g$ ,  $-r_S$ ,  $CC$  e  $CS$  em função do tempo até 300 horas. Discuta as tendências observadas.

(b) Repita (a) e aumente a temperatura inicial com incrementos de 10°C até atingir 330 K e descreva o que você encontrou. Plote a concentração de células após 24 horas de operação em função da temperatura de alimentação.

(c) Qual a área de troca térmica que deveria ser adicionada para maximizar o número total de células ao final de 24 horas de operação? Para uma temperatura inicial de 310 K e uma temperatura constante de fluido refrigerante de 290 K, qual seria a concentração celular após 24 horas?

a) Primeiramente gerou-se os plots da Temperatura, Fator  $I'$  da equação de Aiba et al (Equação 1),  $r_G$  (taxa de crescimento celular),  $-r_S$  (taxa de consumo de substrato),  $C_c$  (concentração de células) e  $C_s$  (concentração de substrato) ao longo do tempo em uma condição adiabática, ou sem um sistema de troca térmica.

Analisando os plots podemos observar que inicialmente as taxas de reação permanecem relativamente baixas e elevando-se lentamente entre 0 e 60 horas. Estas saltam abruptamente entre 60 e 70 horas atingindo um pico de 4 g/dm<sup>3</sup>.h em 68 horas e após isto caindo bruscamente e tendendo a zero até o final o período reacional.

Observado o fator  $I'$  da equação de Aiba et al, que estabelece uma relação inversamente proporcional entre a velocidade específica de crescimento celular e a temperatura. Comparando-a ao plot de temperatura pode-se observar que a mesma acarreta elevação deste fator, atingindo um pico de 1, o valor máximo possível, por volta de 310K (37°C) de temperatura e cando próximo a zero após esta temperatura.

$C_s$  e  $C_c$  seguem a tendência das taxas de reação, variando lentamente no início, sofrendo uma rápida variação entre 65 e 60 horas e estagnando após este período até o final da reação.

$$\mu(T) = \mu(310 \text{ K}) I' = \mu_{1\text{máx}} \left[ \frac{0,0038 \cdot T \exp[21,6 - 6700/T]}{1 + \exp[153 - 48.000/T]} \right] \frac{C_s}{K_M + C_s}$$

### Equação 1 – Equação de Aiba et al

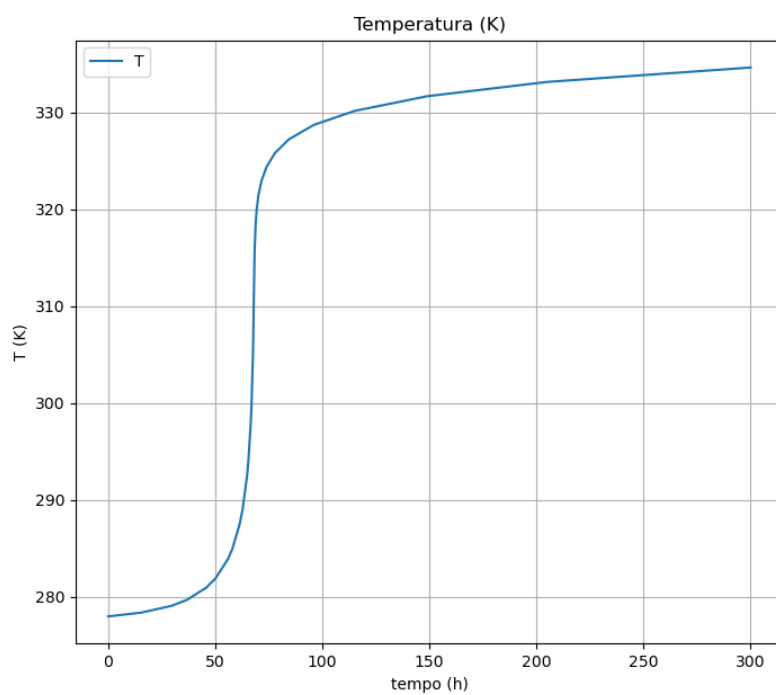


Figura 1 – Evolução da temperatura do reator em 300 horas de reação

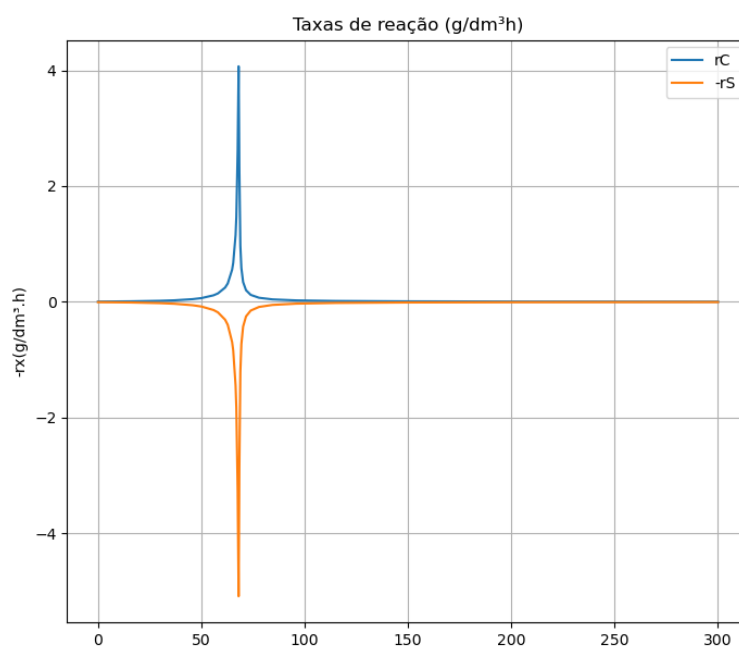


Figura 2 – Taxas de geração de células ( $r_C$ ) e consumo de substrato ( $-r_S$ ), respectivamente.

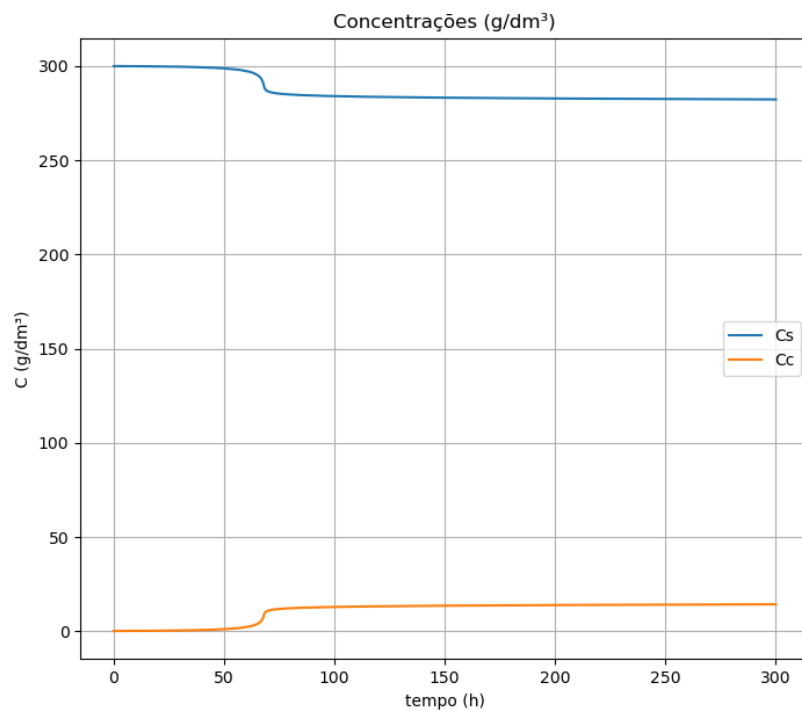


Figura 3 – Concentração de substrato ( $C_s$ ) e de células ( $C_c$ ) por tempo.

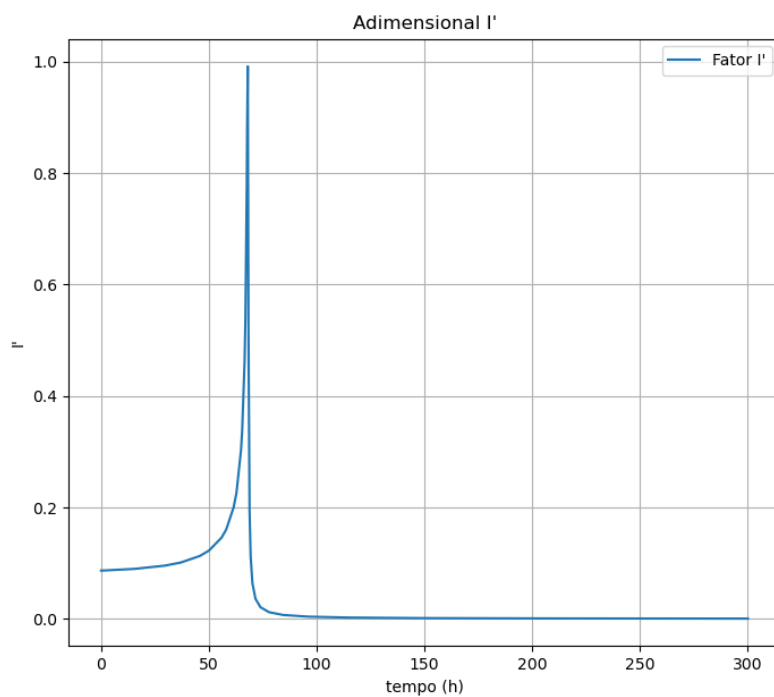


Figura 4 – Fator  $I'$  da equação de Aiba et al.

b) Variando a temperatura inicial até 330K (57°C) e analisando em um intervalo de 24h percebe-se que praticamente não há nenhuma variação nas concentrações de células e substrato, o fator I' e as taxas de reação estão próximas de zero.

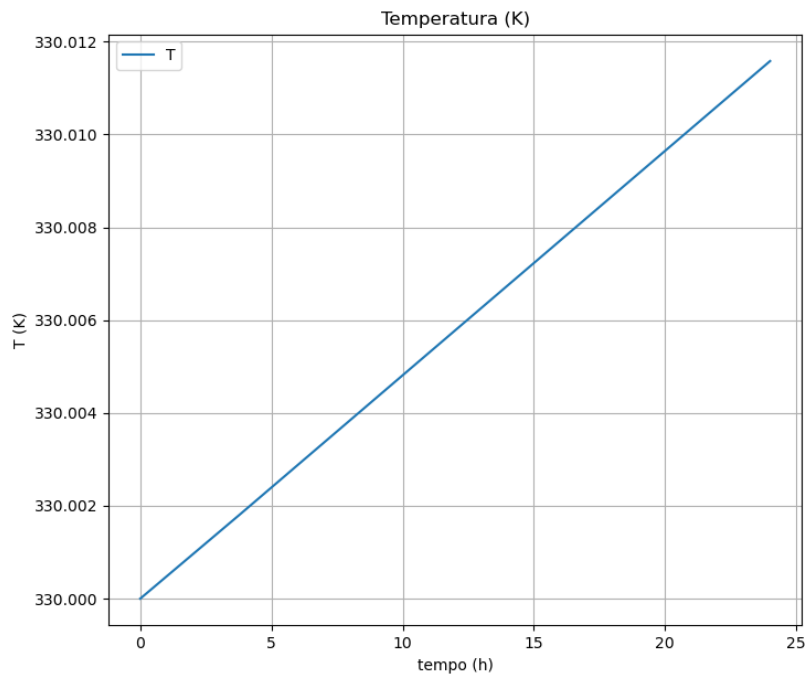


Figura 5 – Temperatura da reação

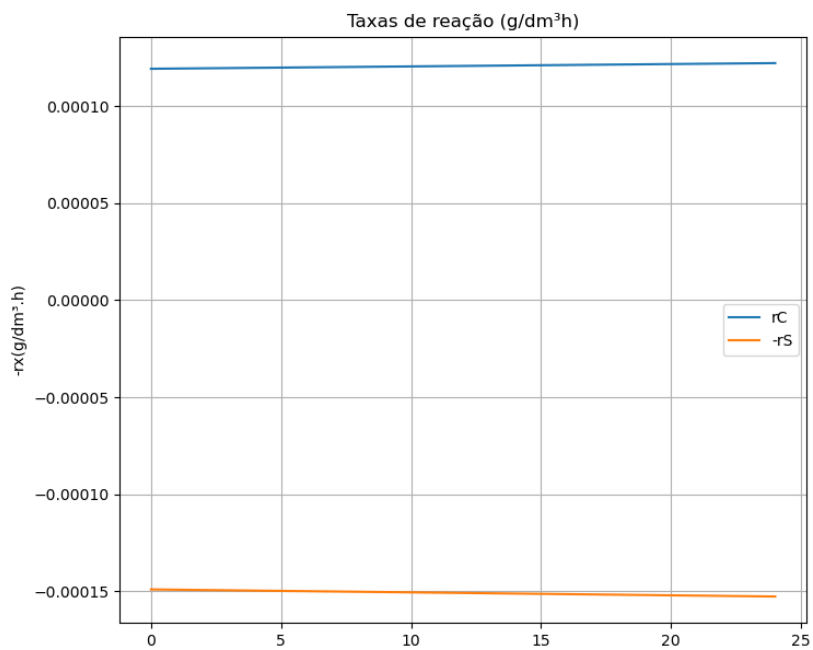


Figura 6 – Taxas de reação

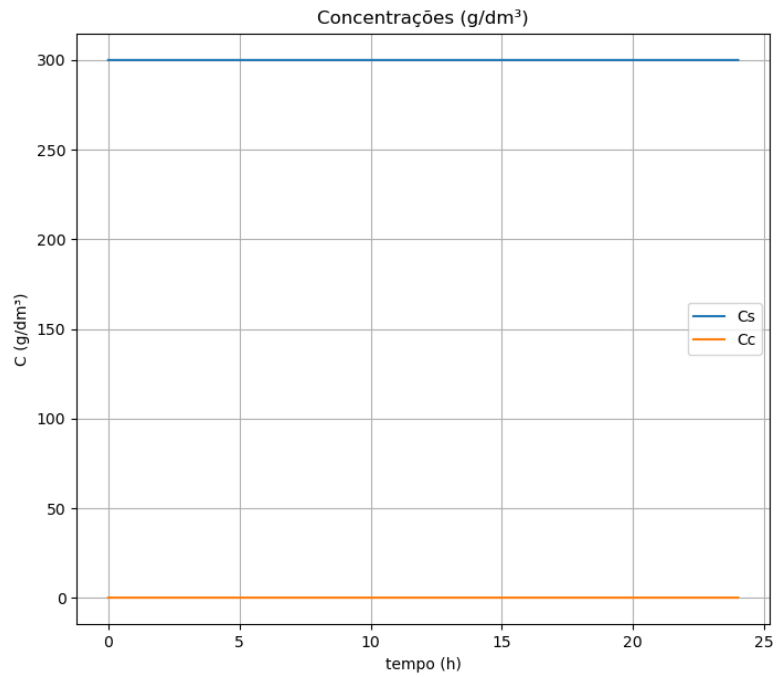


Figura 7 – Concentração das espécies

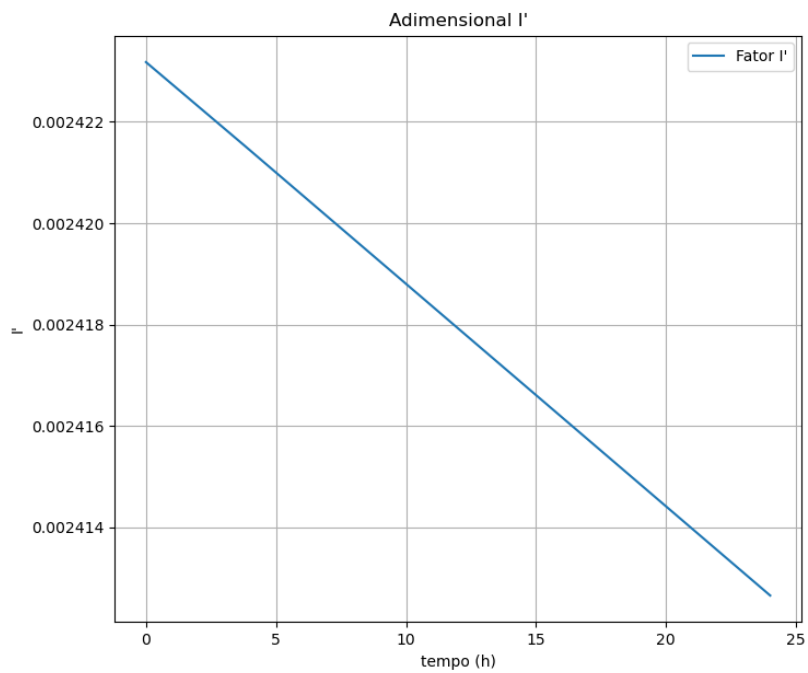


Figura 8 – Fator I'

c) Para uma temperatura inicial de 310K, com temperatura de fluido refrigerante de 290K a melhor área de troca térmica para maximar a geração de células é aproximadamente 0,43 m² e concentração final de células de 21,71 g/dm³. Para obter a área ótima foi utilizado a função ‘minimize’ do módulo ‘optimize’ da biblioteca scipy. Um enxerto do código usado encontra-se a seguir.

Trecho do código para otimização da área.

```
...

def min(A):
    res = solve_ivp(edo,(0,24),(Cs0,Cc0,T0),method='Radau',args=(A))
    return 1/res.y[1][-1]

opt = minimize(min,A)
Aopt = opt.x

print(f'Área ótima: {Aopt} m²')

# Usando a melhor área para obter a concentração máxima
res = solve_ivp(edo,(0,24),(Cs0,Cc0,T0),method='Radau',args=(Aopt))

print(f'Concentração máxima de células em 24h: {res.y[1]} g/dm³')

...
```

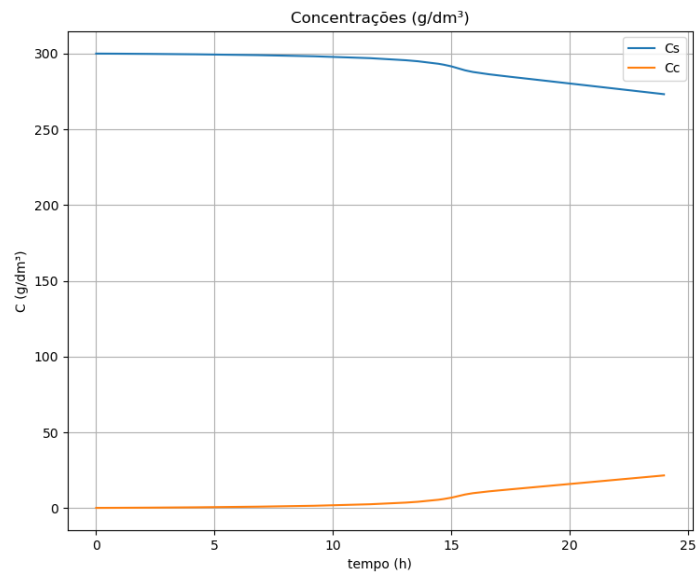


Figura 9 – Concentração com área otimizada

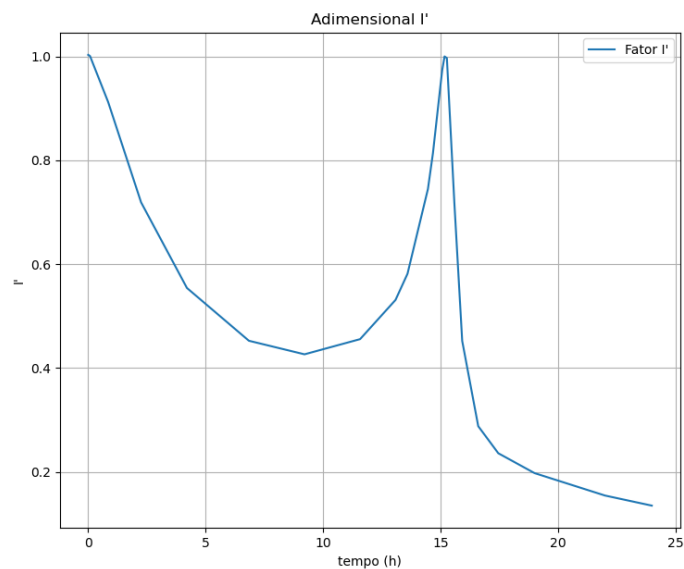


Figura 10– Fator I' otimizado

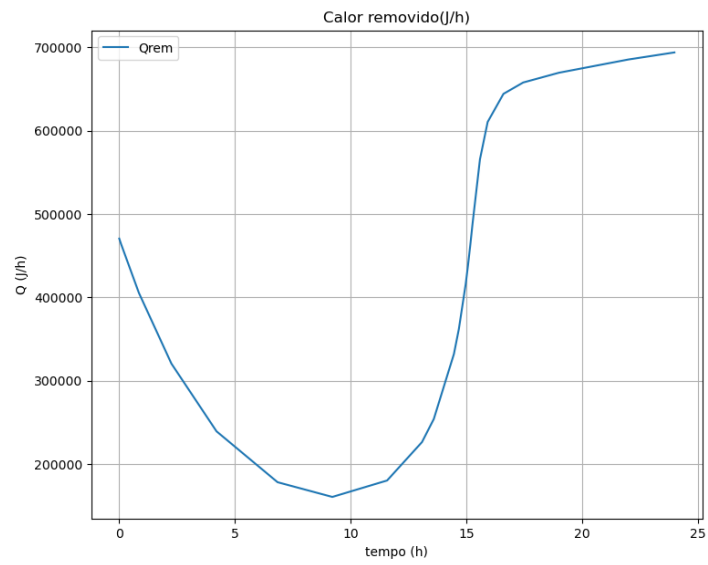


Figura 11 – Calor removido

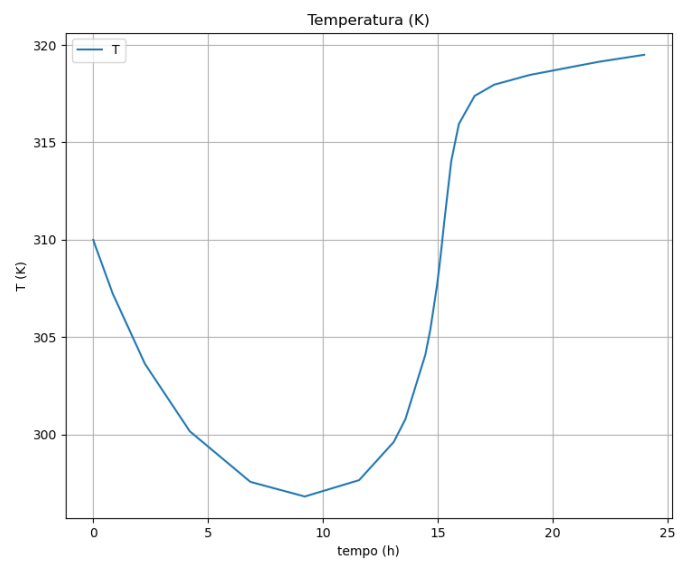


Figura 12 – Temperatura no reator otimizada

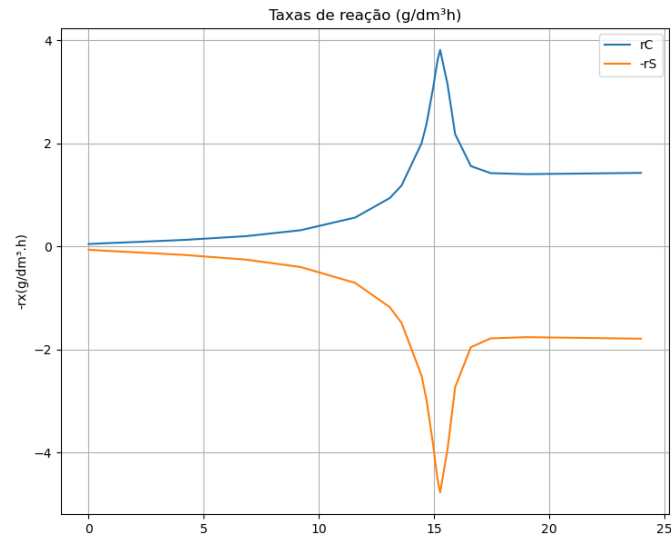


Figura 13 -Taxa de reação otimizada

Código-fonte do problema 1 em Python 3.11

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

# = Funções auxiliares = #
# = Velocidade de reação = #
def u_mi(T,Cs):
    c = Cs/(Km+Cs)
    return uMax * I_l(T) * c

def I_l(T):
    a = 0.0038*T*np.exp(21.6 - 6700/T)
    b = 1 + np.exp(153 - 48000/T)
    return a/b

def Qrem(T):
    return U*A*(T-Ta)
    #return mc*Cpc*(T-Ta)*(1-np.exp(1-(U*A)/(mc*Cpc)))

def rC(T,Cs):
    return u_mi(T,Cs) * Cc

def rS(rc):
    return -1/Ycs * rc

def edo(t,F):
    Cs = F[0]
    Cc = F[1]
    T = F[2]
    dCcDt = u_mi(T,Cs) * Cc
    #dCcDt = uMax * Cs/(Km+Cs) * Cc
    dCsDt = rS(dCcDt)
    dTdt = ((dCcDt*V)*(-dHrx) - Qrem(T))/(V*Cps*rho)
    return [dCsDt,dCcDt,dTdt]

# = Dados do problema = #

V = 25 # dm³
Ycs = 0.8 # g cel/g subst
Km = 5.0 # g/dm³
uMax = 0.5 # h⁻¹
Cps = 5 # J/g/K
```



```

mc = 100 # kg/h
rho = 1000 # g/dm³
dHrx = -2e4 # J/g cel
Cpc = 4.2 # J/g.K
U = 5e4 # J/(h.K.m²)

A = 0.31370*1.5 # m²
Ta = 290 # K

# Condições iniciais do problema
Cc0 = 0.1 # g/dm³
Cs0 = 300 # g/dm³
#T0 = 273+35 # K
T0 = 310 #K

res = solve_ivp(edo,(0,24),(Cs0,Cc0,T0),method='Radau')

t = res.t
Cs = res.y[0]
Cc = res.y[1]
T = res.y[2]

# Plot
fig1, ax1 = plt.subplots()
ax1.plot(res.t, Cs,label='Cs')
ax1.plot(res.t, Cc,label='Cc')
ax1.set_title('Concentrações (g/dm³)')
ax1.set_ylabel('C (g/dm³)')
ax1.set_xlabel('tempo (h)')
ax1.legend()
ax1.grid()

fig2,ax2 = plt.subplots()
ax2.plot(res.t, T, label='T')
ax2.set_title('Temperatura (K)')
ax2.set_ylabel('T (K)')
ax2.set_xlabel('tempo (h)')
ax2.legend()
ax2.grid()

fig3, ax3 = plt.subplots()
ax3.plot(t, rC(T,Cs),label='rC')
ax3.plot(t, rS(rC(T,Cs)),label='-rS')
ax3.set_title('Taxas de reação (g/dm³.h)')
ax3.set_ylabel('-rx(g/dm³.h)')
ax3.grid()
ax3.legend()

fig4, ax4 = plt.subplots()
ax4.plot(t, I_l(T),label="Fator I'")
ax4.set_xlabel('tempo (h)')
ax4.set_ylabel("I'")
ax4.set_title("Adimensional I'")
ax4.grid()
ax4.legend()

fig5, ax5 = plt.subplots()
ax5.plot(res.t, Qrem(T),label='Qrem')
ax5.set_title('Calor removido(J/h)')
ax5.set_ylabel('Q (J/h)')
ax5.set_xlabel('tempo (h)')
ax5.legend()
ax5.grid()

plt.show()

```

## Código-fonte do problema 1.1 em Python 3.11 com a rotina de otimização

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

# = Funções auxiliares = #
# = Velocidade de reação = #
def u_mi(T,Cs):
    c = Cs/(Km+Cs)
    return uMax * I_l(T) * c

def I_l(T):
    a = 0.0038*T*np.exp(21.6 - 6700/T)
    b = 1 + np.exp(153 - 48000/T)
    return a/b

def Qrem(T,A):
    return U*A*(T-Ta)
    #return mc*Cpc*(T-Ta)*(1-np.exp(1-(U*A)/(mc*Cpc)))

def rC(T,Cs):
    return u_mi(T,Cs) * Cc

def rS(rc):
    return -1/Ycs * rc

def edo(t,F,A):
    Cs = F[0]
    Cc = F[1]
    T = F[2]
    dCcdt = u_mi(T,Cs) * Cc
    #dCcdt = uMax * Cs/(Km+Cs) * Cc
    dCsdt = rS(dCcdt)
    dTdt = ((dCcdt*V)*(-dHrx) - Qrem(T,A))/(V*Cps*rho)
    return [dCsdt,dCcdt,dTdt]

# = Dados do problema = #

V = 25 # dm³
Ycs = 0.8 # g cel/g subst
Km = 5.0 # g/dm³
uMax = 0.5 # h⁻¹
Cps = 5 # J/g/K
mc = 100 # kg/h
rho = 1000 # g/dm³
dHrx = -2e4 # J/g cel
Cpc = 4.2 # J/g.K
U = 5e4 # J/(h.K.m²)

A = 0.31370 # m²
Ta = 290 # K

# Condições iniciais do problema
Cc0 = 0.1 # g/dm³
Cs0 = 300 # g/dm³
#T0 = 273+35 # K
T0 = 310 #K

def min(A):
    res = solve_ivp(edo,(0,24),(Cs0,Cc0,T0),method='Radau',args=(A))
    return 1/res.y[1][-1]

opt = minimize(min,A)
Aopt = opt.x

print(f'Área ótima: {Aopt}')
```

# Usando a melhor área encontrada

```
res = solve_ivp(edo,(0,24),(Cs0,Cc0,T0),method='Radau',args=(Aopt))

print(f'Concentração máxima de células em 24h: {res.y[1]} g/dm³')
```

```

t = res.t
Cs = res.y[0]
Cc = res.y[1]
T = res.y[2]

# Plot
fig1, ax1 = plt.subplots()
ax1.plot(res.t, Cs, label='Cs')
ax1.plot(res.t, Cc, label='Cc')
ax1.set_title('Concentrações (g/dm³)')
ax1.set_ylabel('C (g/dm³)')
ax1.set_xlabel('tempo (h)')
ax1.legend()
ax1.grid()

fig2, ax2 = plt.subplots()
ax2.plot(res.t, T, label='T')
ax2.set_title('Temperatura (K)')
ax2.set_ylabel('T (K)')
ax2.set_xlabel('tempo (h)')
ax2.legend()
ax2.grid()

fig3, ax3 = plt.subplots()
ax3.plot(t, rC(T, Cs), label='rC')
ax3.plot(t, rS(rC(T, Cs)), label='-rS')
ax3.set_title('Taxas de reação (g/dm³h)')
ax3.set_ylabel('-rx(g/dm³.h)')
ax3.grid()
ax3.legend()

fig4, ax4 = plt.subplots()
ax4.plot(t, I_l(T), label="Fator I'")
ax4.set_xlabel('tempo (h)')
ax4.set_ylabel("I'")
ax4.set_title("Adimensional I'")
ax4.grid()
ax4.legend()

fig5, ax5 = plt.subplots()
ax5.plot(res.t, Qrem(T, Aopt), label='Qrem')
ax5.set_title('Calor removido(J/h)')
ax5.set_ylabel('Q (J/h)')
ax5.set_xlabel('tempo (h)')
ax5.legend()
ax5.grid()

plt.show()

```

### 1.2) Problema 13.9

A reação do Problema P11-3A será conduzida em um reator batelada de 10 dm<sup>3</sup>. Plote e analise a temperatura e a concentração de A, B e C em função do tempo para os seguintes casos:

(a) Operação adiabática.

(b) Valores de UA de 10.000, 40.000 e 100.000 J/(min · K).

(c) Use UA = 40.000 J/(min · K) e diferentes temperaturas iniciais para o reator.

a) Para a operação adiabática, obtiveram-se as seguintes curvas para um intervalo de integração de 120 segundos (Figuras 9 e 10)

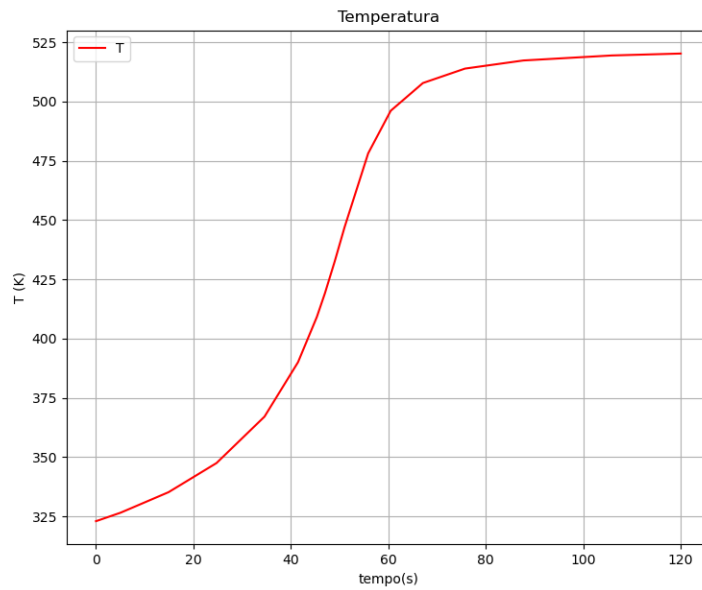


Figura 14 — Temperatura em um sistema adiabático

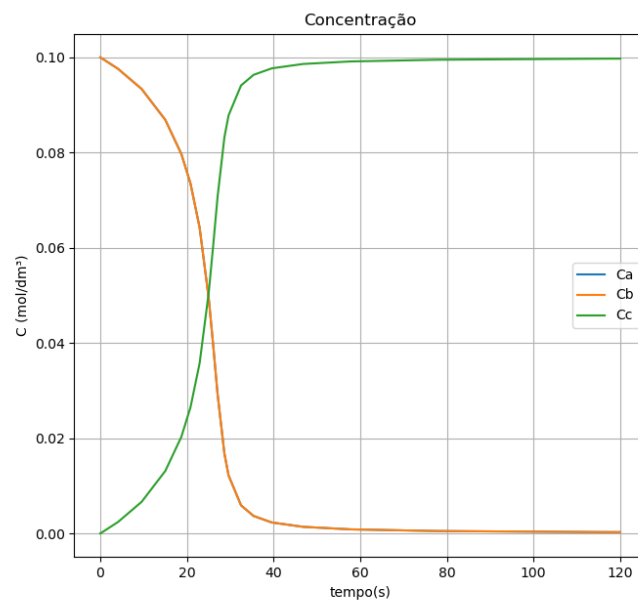


Figura 15 - Concentração no sistema adiabático

b) Para trocadores de calor com coeficiente acoplado de transferência de calor 10.000, 40.000 e 100.000 J/(min · K) (39,83 cal/s.K, 129.33 cal/s.K e 398.33 cal/s.K)

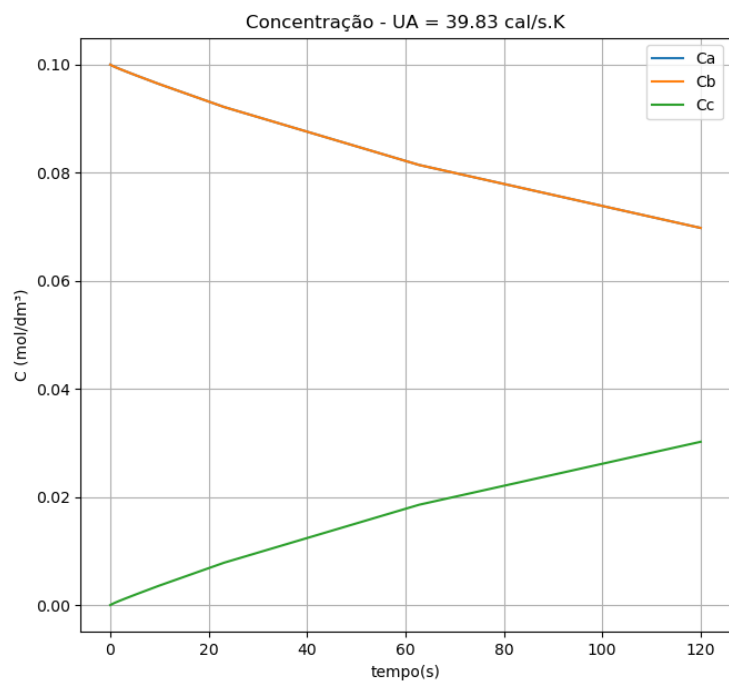


Figura 16 — Concentração com UA = 39,83 cal/s.K

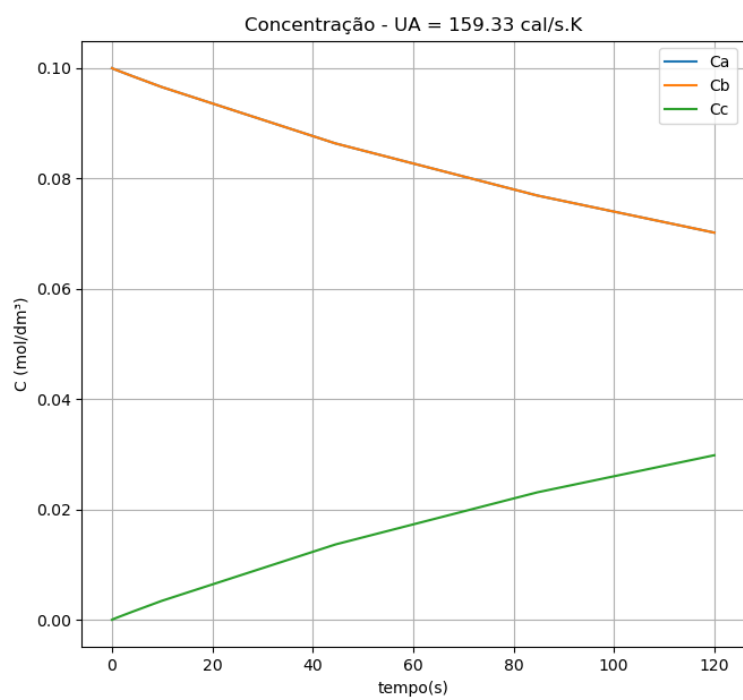


Figura 17 – Concentração com UA = 159,33 cal/s.K

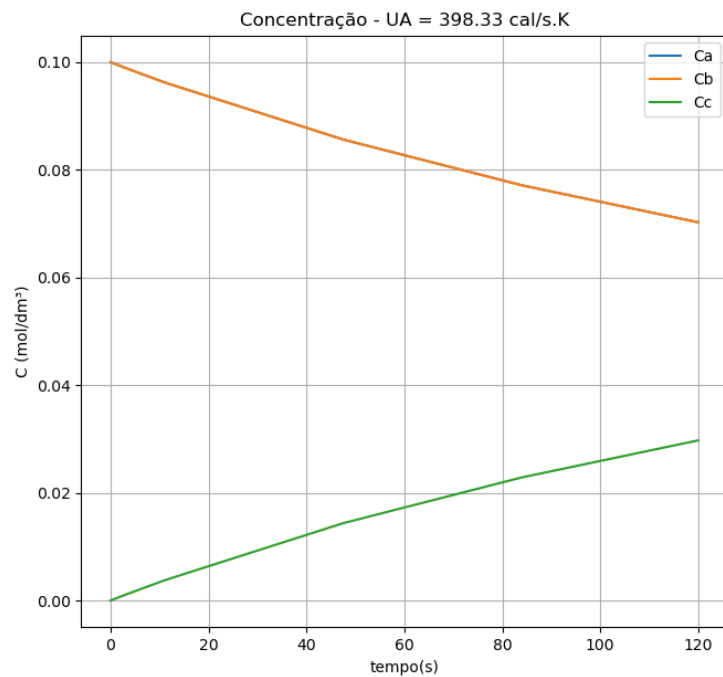


Figura 18 – Concentração com UA = 398,33 cal/s.K

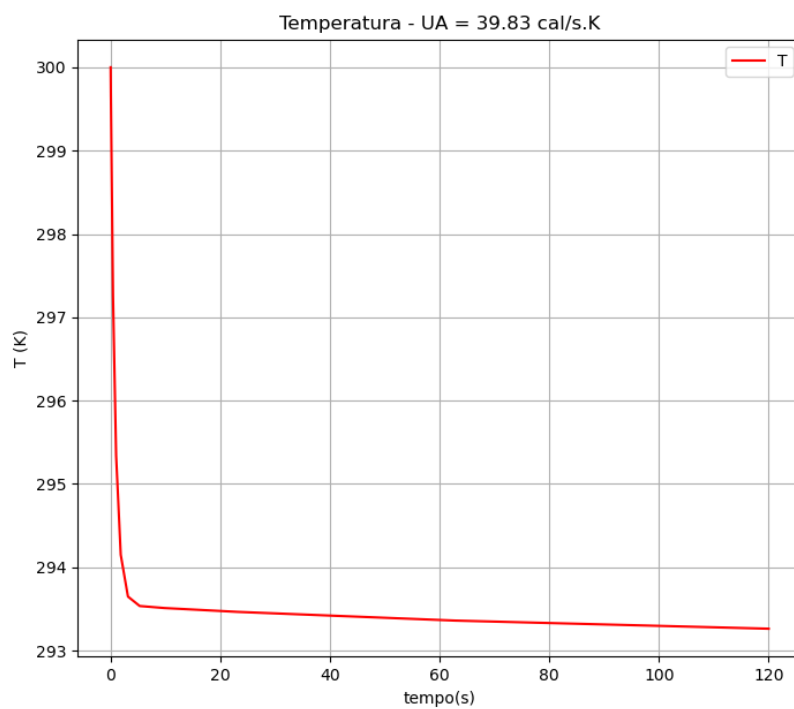


Figura 19 – Temperatura com UA = 39,83 cal/s.K

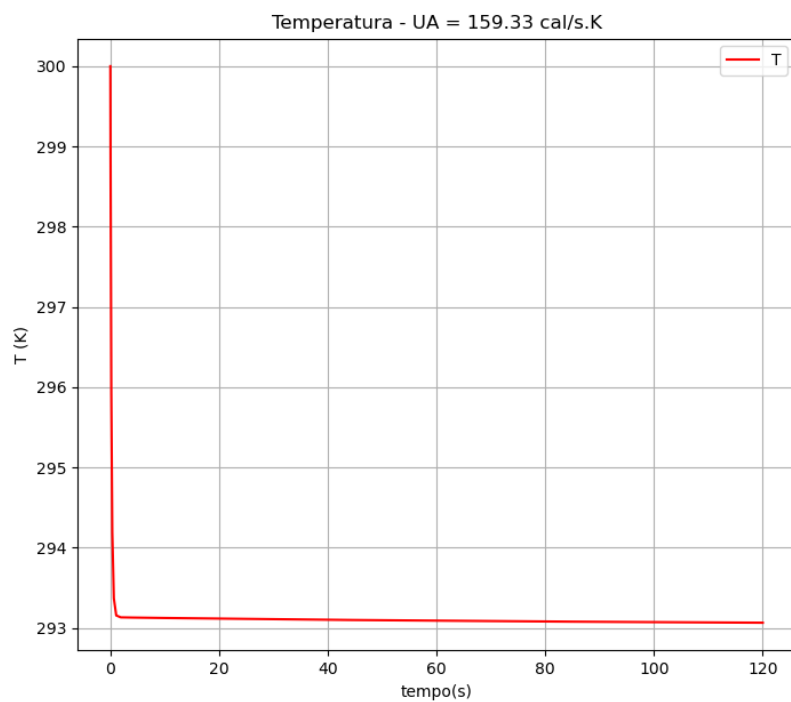


Figura 20 – Temperatura a UA = 159,33 cal/s.K

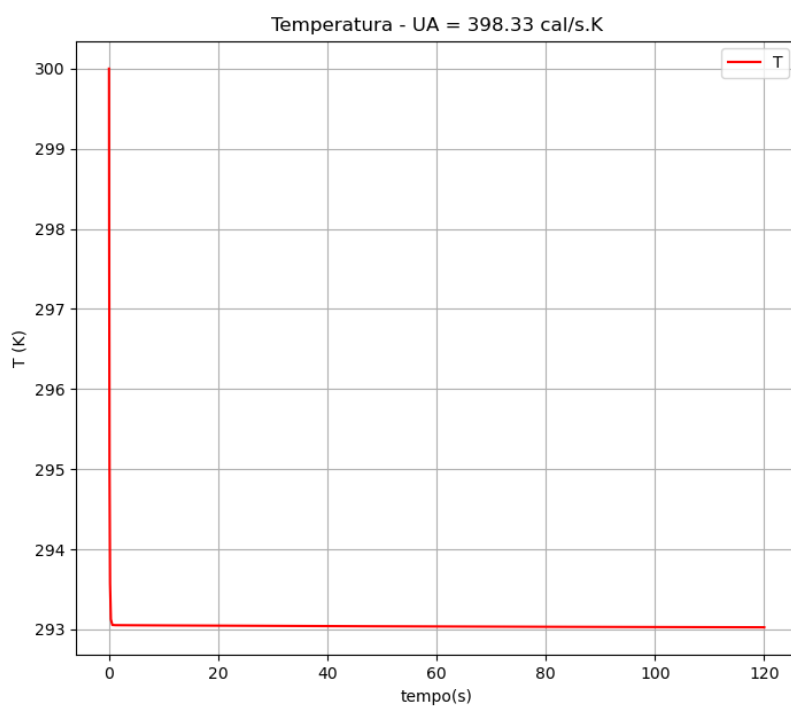


Figura 21 – Temperatura a UA = 398,33 cal/s.K

c) Com  $UA = 40000 \text{ J/min.K}$  para as temperaturas iniciais de  $27^\circ\text{C}$ ,  $35^\circ\text{C}$  e  $45^\circ\text{C}$ .

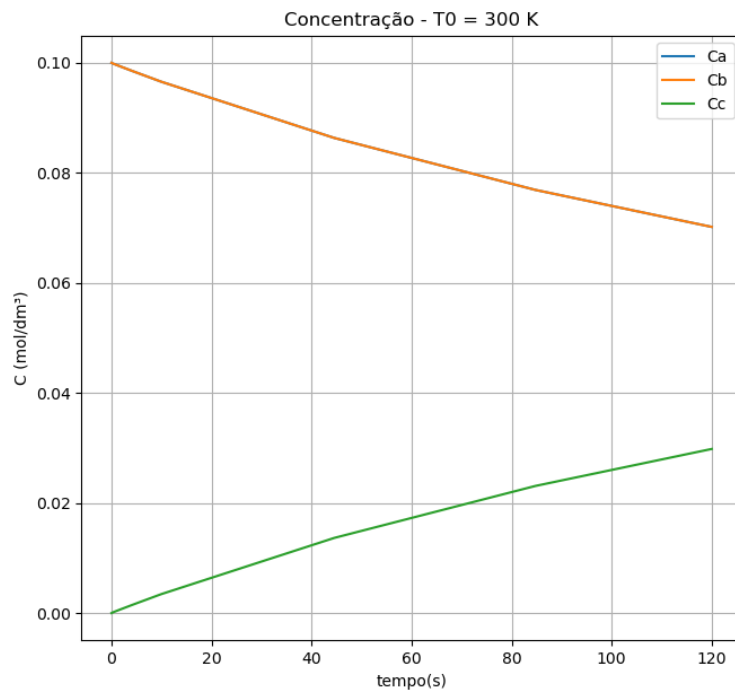


Figura 22

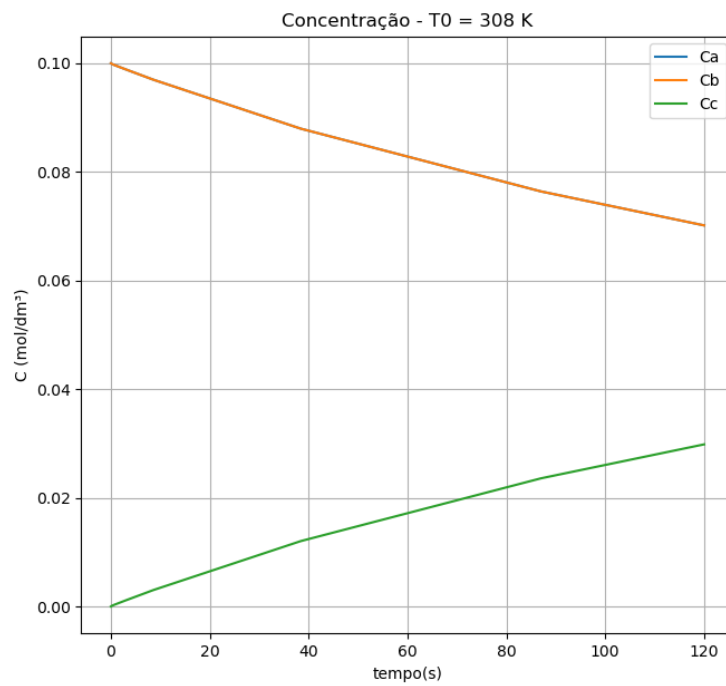


Figura 23



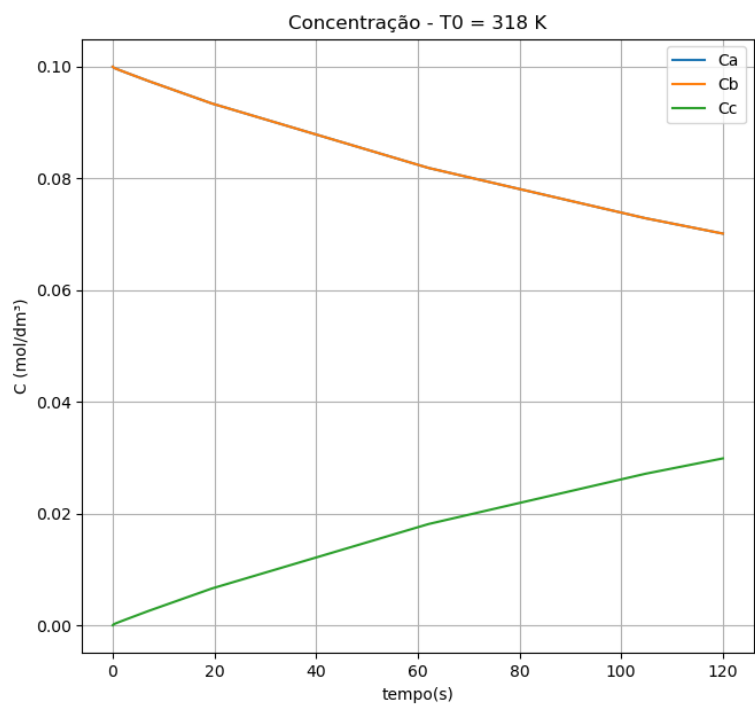


Figura 24

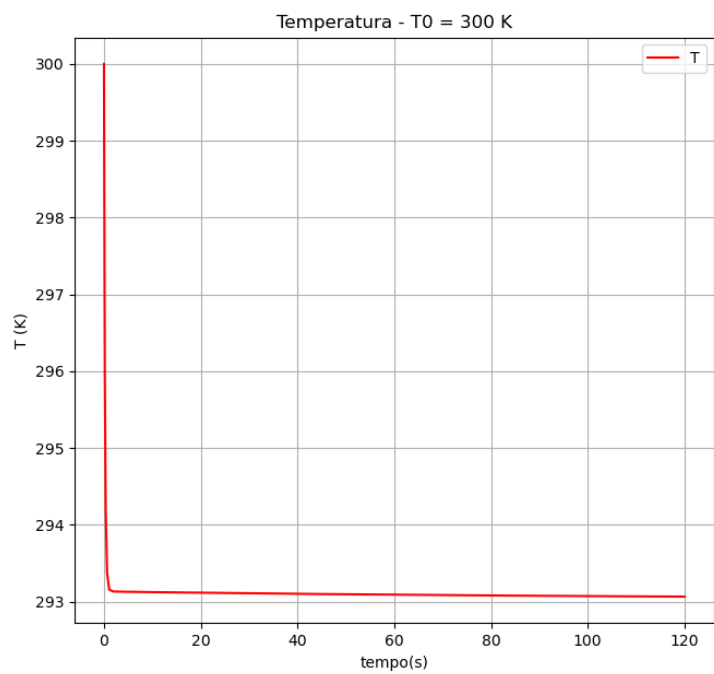


Figura 25

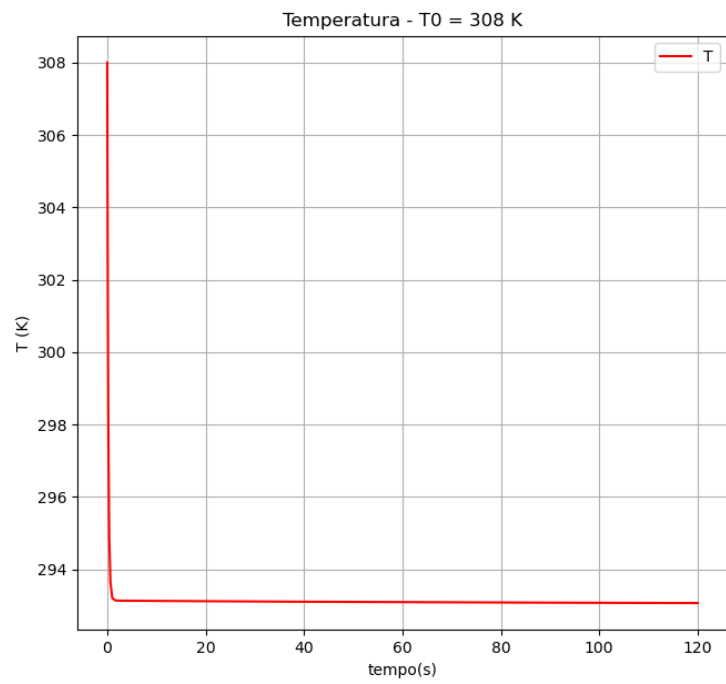


Figura 26

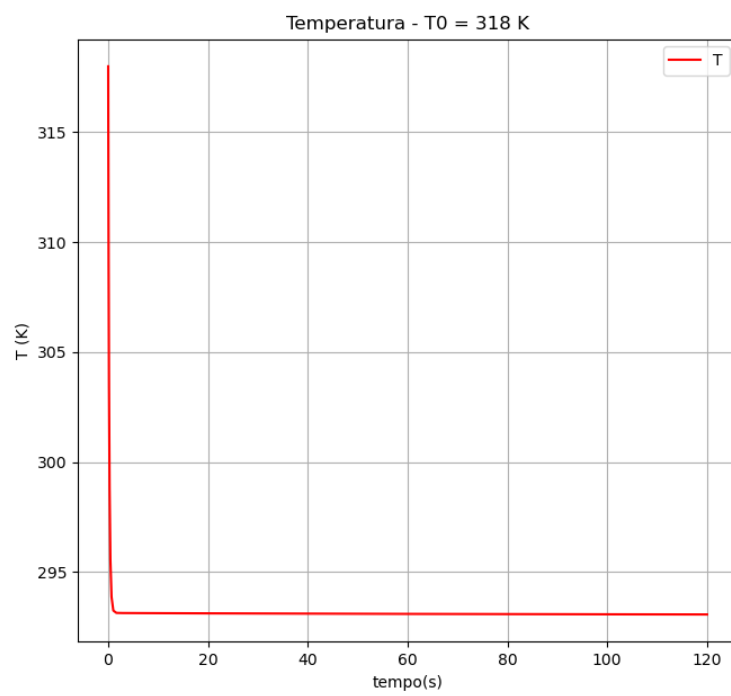


Figura 27

### Código-fonte do problema 3 em Python 3.11

```

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

# Funções auxiliares
def rx(T,X):
    return -(ki(k1,T)*Ca0*(1-X)*Ca0*(theta_b - X))

def ki(k,T):
    return k*np.exp(E/R * (1/Tr - 1/T))

def dHrx(T):
    return deltaHrx + dCp*(T - Tr)

def CiCpi(X):
    Ca = Ca0*(1-X)
    Cb = Ca0*(theta_b - X)
    Cc = Ca0*X
    return Ca*Cpa + Cb*Cpb + Cc*Cpc

def Qgen(T,X):
    return (-rx(T,X)*V)*(-dHrx(T))

def Qrem(T,UA):
    return UA * (T - Ta)

def edo(t,F,UA):
    X = F[0]
    T = F[1]
    dXdt = -rx(T,X)/Ca0
    dTdt = (Qgen(T,X)-Qrem(T,UA))/(V*CiCpi(X))
    return [dXdt, dTdt]

# Dados do problema

a = 1
b = 1
c = 1

Ca0 = 0.1 # mol/dm³
Cb0 = 0.1 # mol/dm³

theta_b = Cb0/Ca0

V = 10 # dm³

R = 1.987 # cal/mol.K

T0 = 27+273 # K
Tr = 273 # K
Ta = 20+273 # K

Ha = -20000 # cal/mol
Hb = -15000 # cal/mol
Hc = -41000 # cal/mol

Cpa = 15 # cal/mol.K
Cpb = Cpa # cal/mol.K
Cpc = 30 # cal/mol.K

# Fator de conversão J → cal
f = 0.2390

UA1 = 10000/60 * f # cal/s.K
UA2 = 40000/60 * f # cal/s.K
UA3 = 100000/60 * f # cal/s.K
Ux = [UA1, UA2, UA3]

E = 10000 # cal/mol

```

```

k1 = 0.01 # dm³/mol.s at 300K

# Dados adicionais

dCp = (c/a)*Cpc - (b/a)*Cpb - Cpa # cal/mol.K

deltaHrx = Hc - Hb - Ha # cal/mol

Tx = [27+273, 35+273, 45+273]

for Ti in Tx:
    TT = tuple([Ti])
    UU = tuple([UA2])
    res = solve_ivp(edo, (0,120), (0,Ti),method="Radau", args=UU)
    fig1, ax1 = plt.subplots()
    ax1.plot(res.t, Ca0*(1-res.y[0]),label='Ca')
    ax1.plot(res.t, Ca0*(theta_b-res.y[0]),label='Cb')
    ax1.plot(res.t, Ca0*res.y[0],label='Cc')
    ax1.set_xlabel('tempo(s)')
    ax1.set_ylabel('C (mol/dm³)')
    ax1.set_title(f'Concentração - T = {Ti:2.2f} K')
    ax1.legend()
    ax1.grid()

    fig2, ax2 = plt.subplots()
    ax2.plot(res.t, res.y[1],label='T',color='r')
    ax2.set_xlabel('tempo(s)')
    ax2.set_ylabel('T (K)')
    ax2.set_title(f'Temperatura - T = {Ti:2.2f} K')
    ax2.legend()
    ax2.grid()

plt.show()

```

## 2) REATORES PFR

### 2.1) Problema 11-3a

*Reação adiabática em reator de escoamento contínuo. Uma alimentação equimolar de A e B entra a 27°C e vazão volumétrica de 2 dm³/s com  $Ca_0 = 0.1 \text{ kmol/m}^3$*

*a) Grafique e analise a conversão e a temperatura como função do volume para um PFR até  $X=0.85$*

*b) Máxima temperatura de entrada até o ponto de ebulição 550K*

a) A temperatura máxima atingida no final da conversão tende 550K, abaixo da temperatura de ebulição dos componentes segundo o enunciado 'b'. A conversão de 0,85 é atingida por volta de 60  $\text{dm}^3$ . Sabendo disto, e dependendo da conversão desejada, um reator menor poderia ser usada neste contexto.

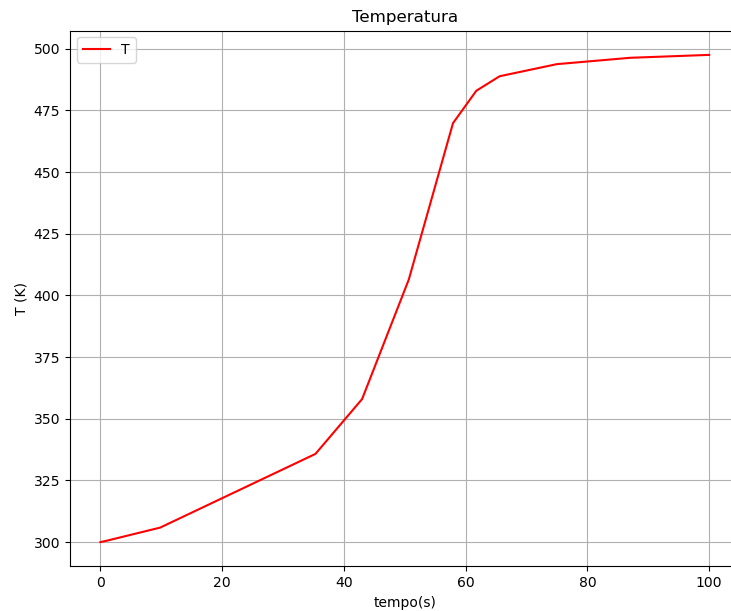


Figura 28 – Temperatura no PFR

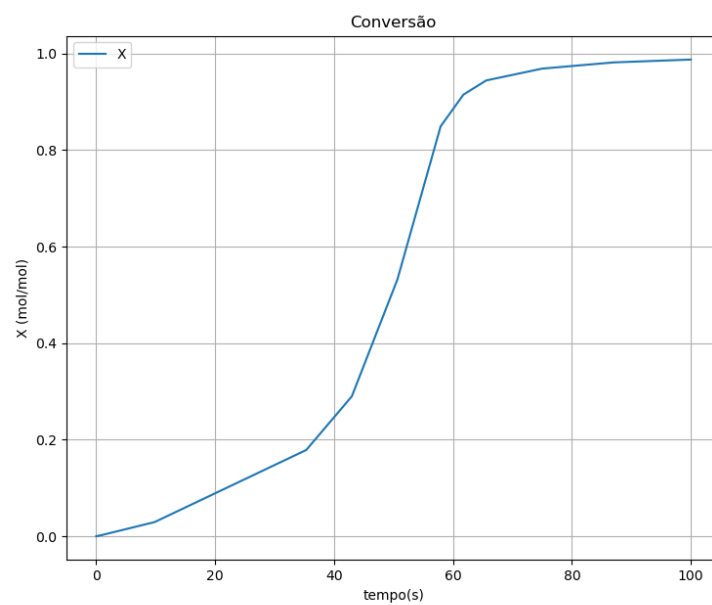


Figura 29 – Conversão

b) Temperatura máxima na alimentação em aproximadamente 350 K (77°C)

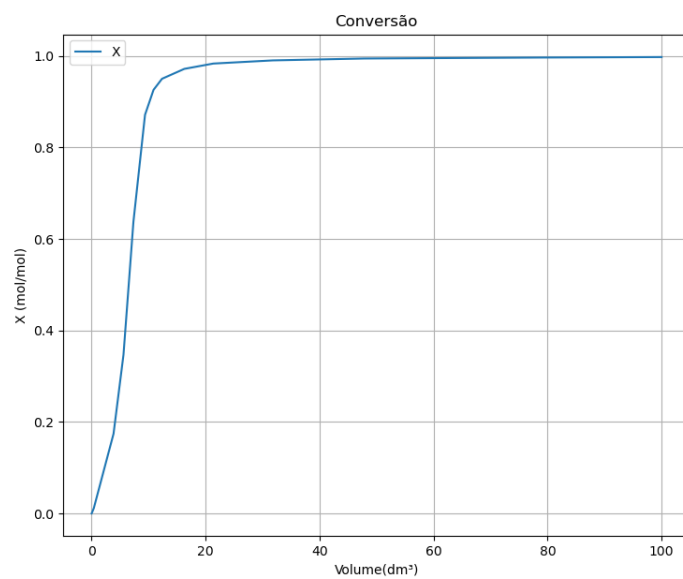


Figura 30 – Conversão com temperatura de alimentação de 350 K

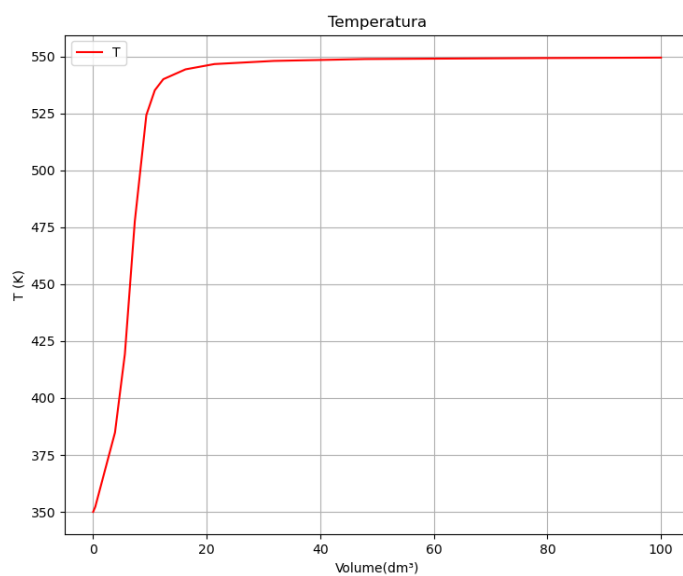


Figura 31 – Temperatura máxima no reator com alimentação a 350K

## Código-fonte do problema 2.1 em Python 3.11

```

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

# == FUNÇÕES AUXILIARES == #
def dHrx(T):
    return deltaHrx + dCp * (T - Tr)

def ki(k,T):
    return k * np.exp(E/R * (1/Tr - 1/T))

def rx(X,T):
    Ca = Ca0 * (1-X)
    Cb = Ca0 * (theta_b - X)
    return -(ki(k,T) * Ca * Cb)

def Qrem(T):
    return UA * (T - Ta)

def pfr(t,F):
    X = F[0]
    T = F[1]

    dXdV = -rx(X,T)/Fa0
    dTdV = (-rx(X,T)*(-dHrx(T)) - Qrem(T))/(Fa0 * (theta_cpi + X*dCp))

    return [dXdV, dTdV]

# == DADOS DO PROBLEMA == #

a = 1
b = 1
c = 1

Ca0 = 0.1 #mol/dm³
Cb0 = Ca0 #mol/dm³
v0 = 2 #dm³/s
Fa0 = Ca0*v0
Fb0 = Fa0
theta_b = Cb0/Ca0

Ha = -20000 #cal/mol
Hb = -15000 #cal/mol
Hc = -41000 #cal/mol
deltaHrx = (c/a)*Hc - (b/a)*Hb - Ha

Cpa = 15 #cal/mol.K
Cpb = Cpa
Cpc = 30 #cal/mol.K
dCp = Cpc - Cpb - Cpa
theta_cpi = Cpa + theta_b*Cpb

k = 0.01 # dm³/mol.S
E = 10000 # cal/mol
R = 1.987 #cal/mol.K

UA = 1000 #cal/dm³.s.K

T0 = 273+77
Tr = 273
Ta = 273+30

res = solve_ivp(pfr, (0,100),(0,T0),method="Radau")

fig1, ax1 = plt.subplots()
fig2, ax2 = plt.subplots()
fig3, ax3 = plt.subplots()

ax1.plot(res.t, res.y[0],label='X')
```

```

ax1.set_xlabel('Volume(dm³)')
ax1.set_ylabel('X (mol/mol)')
ax1.set_title('Conversão')
ax1.grid()
ax1.legend()

ax2.plot(res.t, res.y[1],label='T',color='r')
ax2.set_xlabel('Volume(dm³)')
ax2.set_ylabel('T (K)')
ax2.set_title('Temperatura')
ax2.grid()
ax2.legend()

ax3.plot(res.t, Qrem(res.y[1]),label='Q',color='r')
ax3.set_xlabel('Volume(dm³)')
ax3.set_ylabel('T (K)')
ax3.set_title('Calor (cal/dm³.s)')
ax3.grid()
ax3.legend()

plt.show()

```

2.2) Problema adaptado do Exemplo 1 da aula nº 06 do dia 12 de maio de 2023.

Faça o plot da conversão, temperatura da mistura reacional e o fluido de refrigeração em função do volume de um processo de síntese do propilenoglicol a partir do óxido de etileno em um reator do tipo PFR em um regime adiabático e com trocador de calor concorrente. Qual o volume necessário para haver uma conversão de 0,9? Qual a temperatura máxima atingida pelas correntes de reagente e fluido refrigerante?

Considere os seguintes dados:

Temperatura inicial da água de resfriamento ( $T_a$ ): 408,75 R

Temperatura de alimentação ( $T_0$ ): 75 °F

Temperatura máxima permitida: 125°F ou 584 R

a) Temperatura e conversão para um processo adiabático com volume máximo de 1000 ft<sup>3</sup> (aproximadamente 28 m<sup>3</sup>)

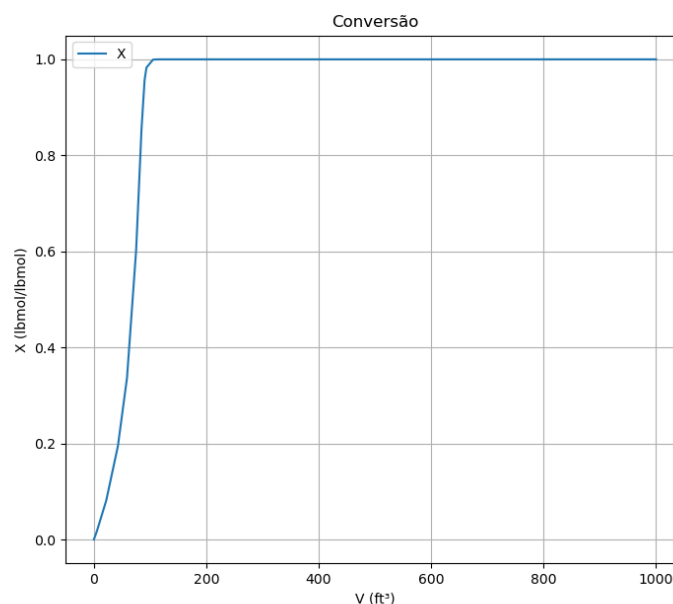




Figura 32 – Conversão do propilenoglicol em processo adiabático

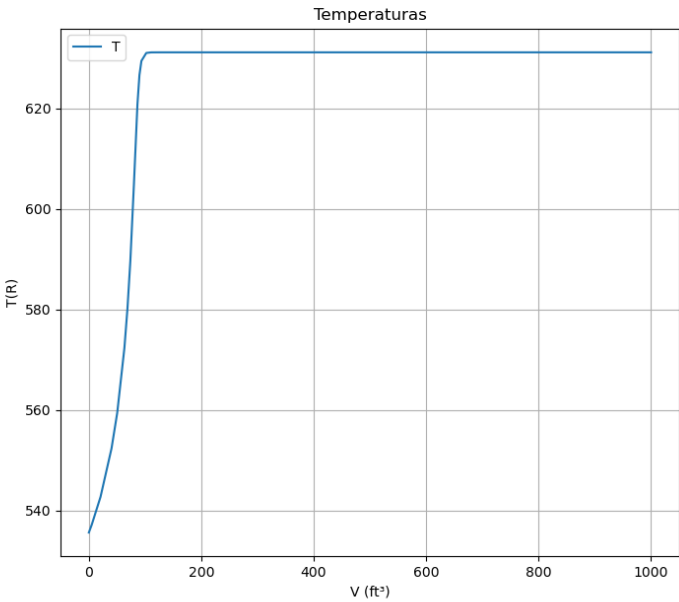


Figura 33 – Temperatura do meio reacional

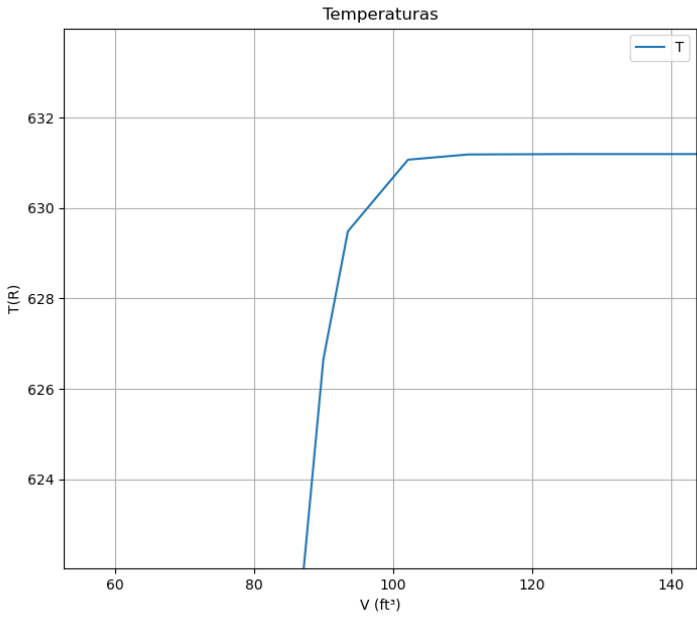


Figura 34 – Temperatura do meio reacional (ampliada)

b) Conversão e temperaturas com troca térmica em regime concorrente

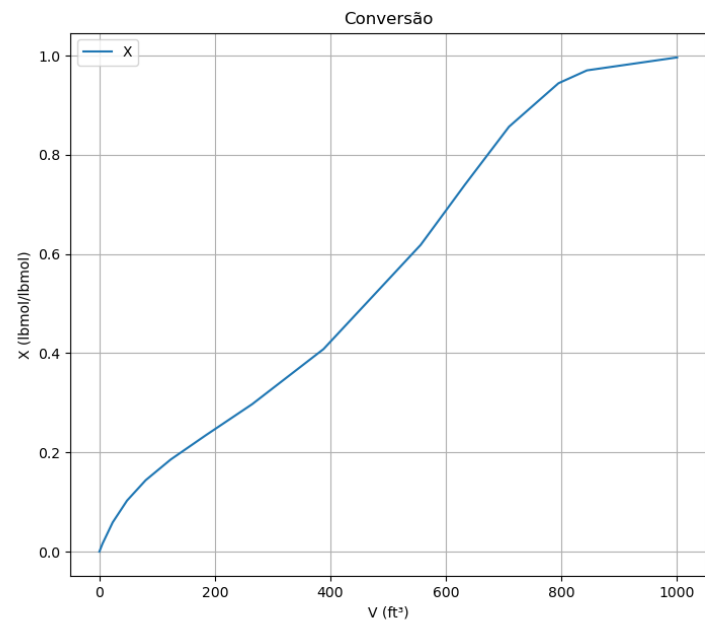


Figura 35 – Conversão em regime concorrente

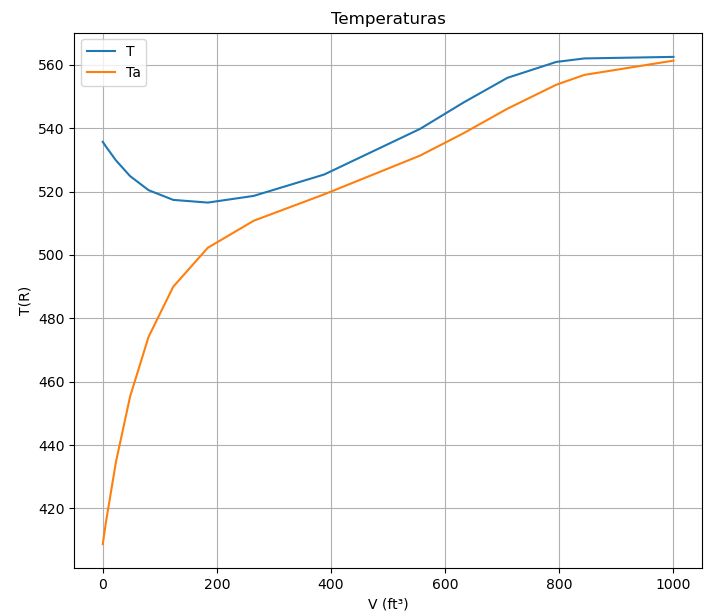


Figura 36 – Temperaturas em regime concorrente

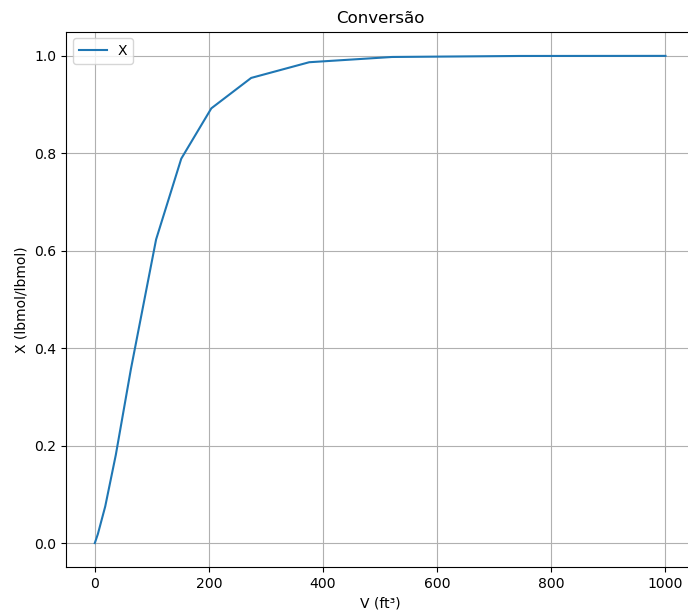


Figura 37 – Conversão em reator com 1/10 de alimentação

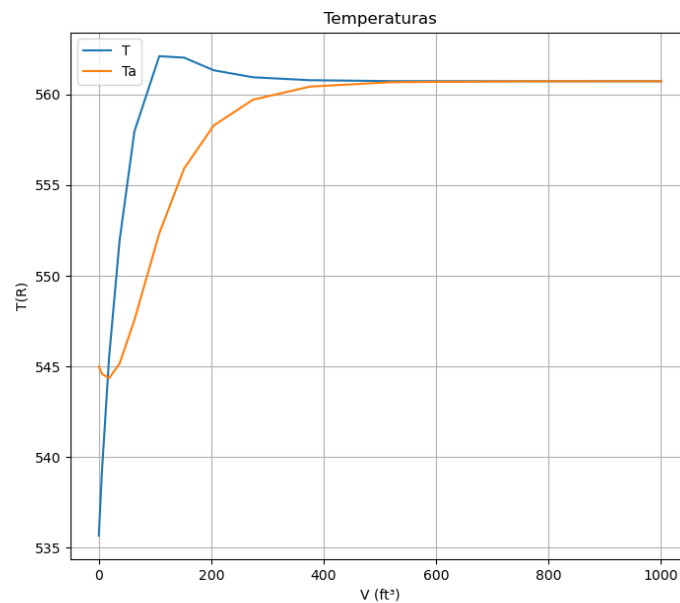


Figura 38 – Temperatura em reator com 1/10 de alimentação.

No sistema adiabático, o sistema atinge a conversão máxima em aproximadamente 100 ft³, entretanto a temperatura ultrapassada o limite para a evaporação do propilenoglicol de 584 R, girando na faixa de 631 R. Em um sistema tubular isto poderia ser perigoso dado o possível aumento de pressão e consequente rompimento dos reatores tubulares. Reatores batelada e CSTR podem ser equipados com sistemas de alívio de pressão com mais facilidade, o que tornaria o processo mais seguro em tais reatores. Com a introdução de um sistema de resfriamento podemos controlar esta temperatura, mas em compensação o volume do reator aumenta drasticamente, dado a

cinética da reação, até atingir uma conversão aceitável. Dividir em reatores menores é uma possibilidade, entretanto pelas simulações (Figuras 31 e 32), não se observam melhorias dados que precisar-se-ia de 10 reatores de aproximadamente 400 ft<sup>3</sup> (11 m<sup>3</sup>).

#### Código-fonte do problema 2.2 em Python 3.11

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

# == FUNÇÕES AUXILIARES == #
def dHrx(T):
    return deltaHrx + dCp * (T - Tr)

def ki(k,T):
    return k * np.exp(E/R * (1/Tr - 1/T))

def rx(X,T):
    Ca = Ca0 * (1-X)
    Cb = Ca0 * (theta_b - X)
    return -(ki(k,T) * Ca * Cb)

def Qrem(T):
    return UA * (T - Ta)

def pfr(t,F):
    X = F[0]
    T = F[1]

    dXdV = -rx(X,T)/Fa0
    dTdV = (-rx(X,T)*(-dHrx(T)) - Qrem(T))/(Fa0 * (theta_cpi + X*dCp))

    return [dXdV, dTdV]

# == DADOS DO PROBLEMA == #

a = 1
b = 1
c = 1

Ca0 = 0.1 #mol/dm³
Cb0 = Ca0 #mol/dm³
v0 = 2 #dm³/s
Fa0 = Ca0*v0
Fb0 = Fa0
theta_b = Cb0/Ca0

Ha = -20000 #cal/mol
Hb = -15000 #cal/mol
Hc = -41000 #cal/mol
deltaHrx = (c/a)*Hc - (b/a)*Hb - Ha

Cpa = 15 #cal/mol.K
Cpb = Cpa
Cpc = 30 #cal/mol.K
dCp = Cpc - Cpb - Cpa
theta_cpi = Cpa + theta_b*Cpb

k = 0.01 # dm³/mol.S
E = 10000 # cal/mol
R = 1.987 #cal/mol.K

UA = 1000 #cal/dm³.s.K

T0 = 273+77
Tr = 273
Ta = 273+30

res = solve_ivp(pfr, (0,100),(0,T0),method="Radau")
```

```

fig1, ax1 = plt.subplots()
fig2, ax2 = plt.subplots()
fig3, ax3 = plt.subplots()

ax1.plot(res.t, res.y[0],label='X')
ax1.set_xlabel('Volume(dm³)')
ax1.set_ylabel('X (mol/mol)')
ax1.set_title('Conversão')
ax1.grid()
ax1.legend()

ax2.plot(res.t, res.y[1],label='T',color='r')
ax2.set_xlabel('Volume(dm³)')
ax2.set_ylabel('T (K)')
ax2.set_title('Temperatura')
ax2.grid()
ax2.legend()

ax3.plot(res.t, Qrem(res.y[1]),label='Q',color='r')
ax3.set_xlabel('Volume(dm³)')
ax3.set_ylabel('T (K)')
ax3.set_title('Calor (cal/dm³.s)')
ax3.grid()
ax3.legend()

plt.show()

```

### 3) REATORES CSTR

#### 3.1) Problema 12-7

Usando os dados do problema P11.3a, calcule a conversão quando a reação é conduzida adiabaticamente em um CSTR de 500 dm<sup>3</sup> quando a reação é conduzida adiabaticamente em um CSTR de 500 dm<sup>3</sup>. Compare com um sistema em série com dois reatores de 250 dm<sup>3</sup>.

A resolução do sistema nos fornece uma conversão de 0,72 e uma temperatura final de 340K (67°C) em um reator CSTR de 500 dm<sup>3</sup>.

Já para dois reatores em série de 250 dm<sup>3</sup>, obteve-se na saída do primeiro reator 0,66 a 344K, e no segundo reator 0,72 a 385K.

Código-fonte do problema 3.1 em Python 3.11. Sistema simples em um reator

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import fsolve

# = FUNÇÕES AUXILIARES = #

def rx(X,T):
    return -(ki(k,T) * Ca0 * (1-X) * Ca0 * (theta_b - X))

def ki(k,T):
    return k * np.exp(E/R * (1/Tr - 1/T))

def dHrx(T):
    return deltaHrx + dCp * (T - Tr)

def theta_cpi(X):
    Ca = Ca0*(1-X)
    Cb = Ca0*(theta_b-X)

```

```

Cc = Ca0*X
return Cpa + Cpb*Cb/Ca + Cpc*Cc/Ca

# == FUNÇÃO OBJETIVO == #

def func(x):
    X = x[0]
    T = x[1]
    F = V*(-rx(X,T)) - Fa0*X
    G = -theta_cpi(X)*(T-T0) - dHrx(T)*X

    return [F,G]

# == DADOS == #

a = 1
b = 1
c = 1

V = 500 # dm³

Tr = 273

v0 = 2 # dm³/s
Ca0 = 0.1 # mol/dm³
Fa0 = v0 * Ca0
Fb0 = Fa0

theta_b = Fb0/Fa0

Ha = -20000 # cal/mol
Hb = -15000 # cal/mol
Hc = -41000 # cal/mol
deltaHrx = (c/a)*Hc - (b/a)*Hb - Ha

Cpa = 15 # cal/mol.K
Cpb = Cpa
Cpc = 30 # cal/mol.K
dCp = (c/a) * Cpc - (b/a)*Cpb - Cpa

k = 0.01 # dm³/mol.s
E = 10000 # cal/mol
R = 1.987 # cal/mol.K

#UA = 20 # cal/m².s.K
#mc = 50 # g/s
#Ta0 = 450 # K
#Cpc = 1 # cal/g.K

T0 = 273+27
x0 = [0.9, 350]

res = fsolve(func, x0)

print(res)

```

Código-fonte do problema 3.1 em Python 3.11. Reatores em série.

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import fsolve

# == FUNÇÕES AUXILIARES == #

def rx(X,T):
    return -(ki(k,T) * Ca0 * (1-X) * Ca0 * (theta_b - X))

def ki(k,T):
    return k * np.exp(E/R * (1/Tr - 1/T))

def dHrx(T):
    return deltaHrx + dCp * (T - Tr)

```

```

def theta_cpi(X):
    Ca = Ca0*(1-X)
    Cb = Ca0*(theta_b-X)
    Cc = Ca0*X
    return Cpa + Cpb*Cb/Ca + Cpc*Cc/Ca

# == FUNÇÃO OBJETIVO == #

def func(x):
    X = x[0]
    T = x[1]
    F = V*(-rx(X,T)) - Fa0*X
    G = -theta_cpi(X)*(T-T0) - dHrx(T)*X

    return [F,G]

# == DADOS == #

a = 1
b = 1
c = 1

V = 250 # dm³

Tr = 273

v0 = 2 # dm³/s
Ca0 = 0.1 # mol/dm³
Fa0 = v0 * Ca0
Fb0 = Fa0

theta_b = Fb0/Fa0

Ha = -20000 # cal/mol
Hb = -15000 # cal/mol
Hc = -41000 # cal/mol
deltaHrx = (c/a)*Hc - (b/a)*Hb - Ha

Cpa = 15 # cal/mol.K
Cpb = Cpa
Cpc = 30 # cal/mol.K
dCp = (c/a) * Cpc - (b/a)*Cpb - Cpa

k = 0.01 # dm³/mol.s
E = 10000 # cal/mol
R = 1.987 # cal/mol.K

#UA = 20 # cal/m³.s.K
#mc = 50 # g/s
#Ta0 = 450 # K
#Cpc = 1 # cal/g.K

T0 = 273+27
x0 = [0.9, 350]
res = fsolve(func, x0)
print(res)

# Recalcular dados
Ca0 = Ca0*(1-res[0])
Fa0 = Ca0*v0
Fb0 = Fa0
theta_b = Fb0/Fa0
T0 = res[1]

x0 = res
res = fsolve(func,x0)
print(res)

```

### 3.2) Problema 12-7a com trocador de calor

Introduzindo uma refrigeração com um fluido de troca térmica em 298K, tem-se no final uma conversão de 0,59 e temperatura final de 319,36K.

Código-fonte do problema 3.2 em Python 3.11.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import fsolve

# == FUNÇÕES AUXILIARES == #

def rx(X,T):
    return -(ki(k,T) * Ca0 * (1-X) * Ca0 * (theta_b - X))

def ki(k,T):
    return k * np.exp(E/R * (1/Tr - 1/T))

def dHrx(T):
    return deltaHrx + dCp * (T - Tr)

def theta_cpi(X):
    Ca = Ca0*(1-X)
    Cb = Ca0*(theta_b-X)
    Cc = Ca0*X
    return Cpa + Cpb*Cb/Ca + Cpc*Cc/Ca

# == FUNÇÃO OBJETIVO == #

def func(x):
    X = x[0]
    T = x[1]
    F = V*(-rx(X,T)) - Fa0*X
    G = (UA/Fa0)*(Ta-T) - theta_cpi(X)*(T-T0) - dHrx(T)*X

    return [F,G]

# == DADOS == #

a = 1
b = 1
c = 1

V = 500 # dm³

Tr = 273
Ta = 298

v0 = 2 # dm³/s
Ca0 = 0.1 # mol/dm³
Fa0 = v0 * Ca0
Fb0 = Fa0

theta_b = Fb0/Fa0

Ha = -20000 # cal/mol
Hb = -15000 # cal/mol
Hc = -41000 # cal/mol
deltaHrx = (c/a)*Hc - (b/a)*Hb - Ha

Cpa = 15 # cal/mol.K
Cpb = Cpa
Cpc = 30 # cal/mol.K
```



```
dCp = (c/a) * Cpc - (b/a)*Cpb - Cpa
```

```
k = 0.01 # dm³/mol.s
```

```
E = 10000 # cal/mol
```

```
R = 1.987 # cal/mol.K
```

```
UA = 20 # cal/m³.s.K
```

```
Ta0 = 450 # K
```

```
T0 = 273+27
```

```
x0 = [0.9, 350]
```

```
res = fsolve(func, x0)
```

```
print(res)
```