

# Reator em batelada com e sem troca de calor

João Eduardo Levandoski, RA: 265243

14 de Outubro de 2020

Neste caso a reação possui lei de velocidade de primeira ordem, logo o balanço de massa para um reator do tipo batelada sem entrada nem saída fica como da Equação 1

$$\frac{dX}{dt} = k(1 - X) \quad (1)$$

Um reator em batelada com variação de temperatura pode ser modelado com ou sem troca de calor, porém para obter a variação de temperatura no tempo utilizamos a Equação 2

$$\frac{dT}{dt} = \frac{(Q + (-\Delta H_{rx})(-r_o V))}{N_{o0}C_{po}} \quad (2)$$

Para construção do algoritmo foi considerado com troca de calor, e caso não houvesse troca o termo Q foi igualado a 0. O cálculo do sistema de EDOs formados pelas Equações 1 e 2 foi feito com o uso da linguagem Python, o código completo está presente em anexo.

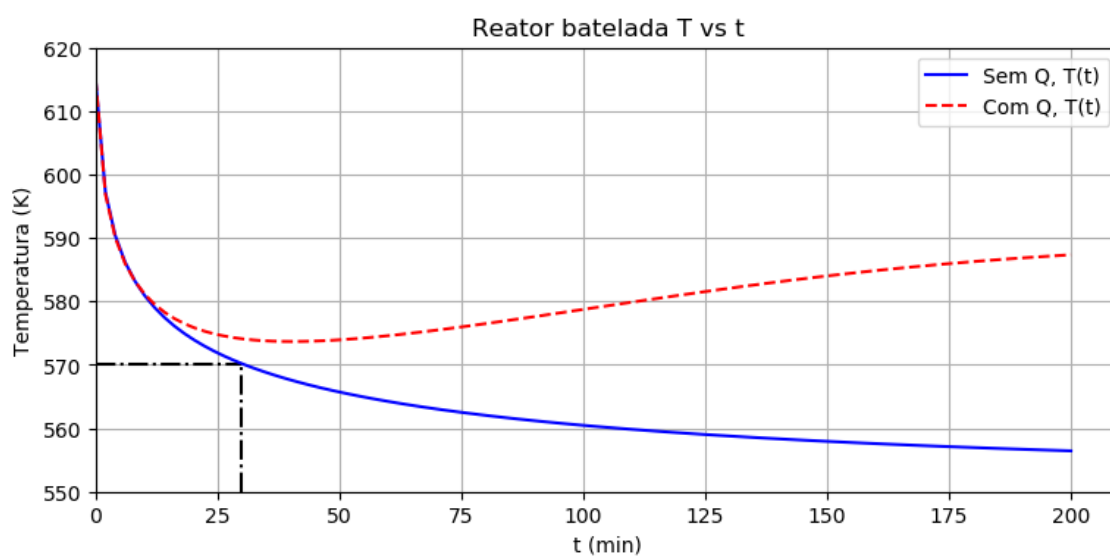


Figura 1 – Temperatura versus t

Nas definições do problema foi fornecida a informação de que a reação só ocorre de modo satisfatório na faixa de temperatura entre 570 K e 615 K, desde modo analisando a Figura 1 pode-se notar que quando não há troca de calor a temperatura fica abaixo de 570 K no tempo  $t = 30$  min (linha -.), e a conversão pode ser analisada na Figura 2, onde  $X = 0,72$ , logo a reação sem troca de calor deverá parar aos 30 min sem atingir conversão completa. Quando a troca de calor é computada é possível notar que menor temperatura durante a reação fica acima de 570 K (573,67 K), sendo assim a reação pode prosseguir até a conversão de  $x = 0,999$  aos 200 min.

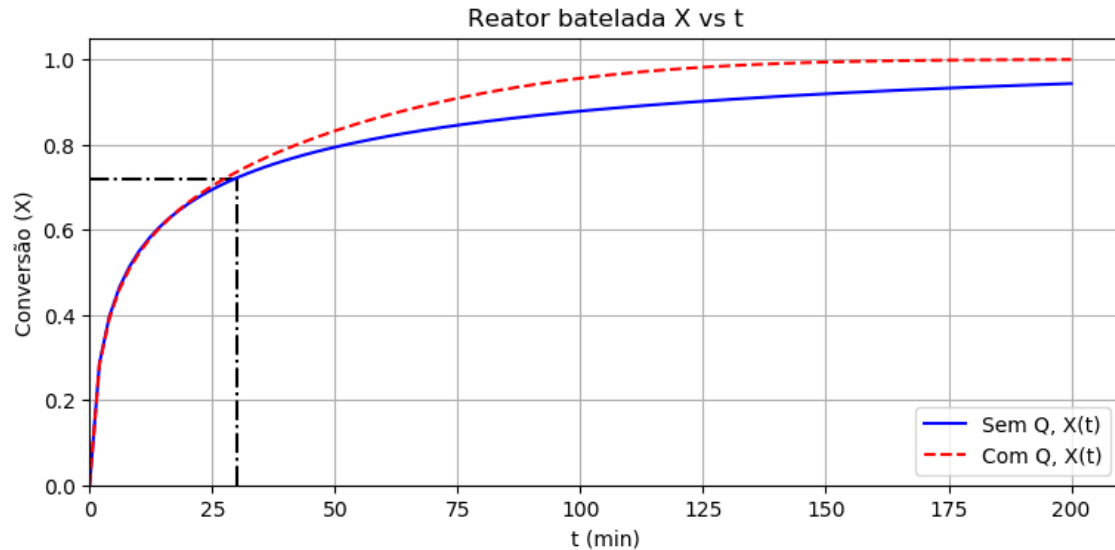


Figura 2 – Conversão versus t

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Tue Oct 13 17:53:07 2020
5
6  @author: levandoski
7  """
8  import numpy as np
9  import matplotlib.pyplot as plt
10 from scipy.integrate import odeint
11
12 def bat(y, t, troca):
13     X, T = y
14     TR = 298 # K
15
16     A = 1.94e15 # min-1
17     Ea = 186000 # J mol-1
18     R = 8.314 # J mol-1 K-1
19
20     k = A*np.exp(-Ea/(R*T)) # min-1
21
22     deltaHpad = 60000 # J mol-1
23
24     Cpo = 966.15 # J mol-1 K-1
25     Cpa = 123 # J mol-1 K-1
26     Cps = 843.15 # J mol-1 K-1
27

```

```

28
29 # # dados trocador <----- #####
30 if troca:
31     U = 230*60 #W m^-2 K^ 1 * 60 s min^-1
32     mc = 10 #kg min^-1
33     Cpc = 20000 # J kg^-1 K^-1
34     Ta1 = 590 # K
35     d = 1.2 # m
36     h = d*((5**(1/2)) - 2)/2 # m
37     Acin = 2*np.pi*d**2
38     Acal = 3.1416 * (h**2 + (d**2)/4) #m2
39     Area = Acin + Acal
40
41     Q = mc*Cpc*(Ta1 - T)*(1 - np.exp(-U*Area/(mc*Cpc)))
42 else:
43     Q = 0
44
45 rho_o = 900 # kg m^-3
46
47 V = 2.5 # m^3
48
49 deltaCps = Cps + Cpa - Cpo
50
51
52 deltaH = deltaHpad + deltaCps*(T - TR)
53
54 mo = V * rho_o # kg
55
56 Mo = 0.3846 # kg mol^-1
57 Na0 = mo/Mo
58
59
60 ra = -k*Na0*(1 - X)/V
61
62 dTdt = (Q + (-deltaH)*(-ra*V))/(Na0 * Cpo)
63
64 dXdt = k * (1 - X)
65
66 dydt = [dXdt, dTdt]
67 return dydt
68
69 #y = X, T
70 y0 = [0, 615]
71 t = np.linspace(0, 200, 100)
72
73 troca = True
74 sol = odeint(bat, y0, t, args=(troca, ))
75 X_q, T_q = sol[:, 0], sol[:, 1]
76
77 troca = False
78 sol = odeint(bat, y0, t, args=(troca, ))
79 X, T = sol[:, 0], sol[:, 1]
80
81
82
83 plt.figure(dpi=100, figsize=(9, 4))
84 plt.plot(t, X, 'b', label='Sem Q, X(t)')
85 plt.plot(t, X_q, 'r', label='Com Q, X(t)', linestyle='--')
86 plt.legend(loc='lower right')
87 plt.xlabel('t (min)')

```

```

88     plt.ylabel('Convers o (X)')
89     plt.title('Reator batelada X vs t')
90     plt.vlines(30, 0, 0.72, linestyle="-.")
91     plt.hlines(0.72, 0, 30, linestyle="-.")
92     plt.ylim(0, 1.05)
93     plt.xlim(0, 210)
94     plt.grid()
95     plt.show()
96
97
98     plt.figure(dpi=100, figsize=(9, 4))
99     plt.plot(t, T, 'b', label='Sem Q, T(t)')
100    plt.plot(t, T_q, 'r', label='Com Q, T(t)', linestyle='--')
101    plt.legend(loc='best')
102    plt.xlabel('t (min)')
103    plt.ylabel('Temperatura (K)')
104    plt.title('Reator batelada T vs t')
105    plt.vlines(30, 550, 570, linestyle="-.")
106    plt.hlines(570, 0, 30, linestyle="-.")
107    plt.ylim(550, 620)
108    plt.xlim(0, 210)
109    plt.grid()
110    plt.show()
111

```

Listing 1 – Código para calculo de sistemas de EDOs