# exer_2_pbr_adiabatico

October 8, 2020

Exercício 2 - Desidrogenação de Etanol em Reator de Leito Fixo
Nome: João Eduardo Levandoski RA: 265243

.
.

$C_2H_5OH \rightarrow CH_3CHO + H_2 \qquad \Delta H^o = 82,5 kJmol^{-1}$

$-r_E = \frac{kK_E(P_E - \frac{P_A P_H}{K})}{(1+K_E P_E + K_H P_H)^2}$

Com a taxa de reação acima fornecida e com os dados de como as constantes Ks variam com a temperatura podemos definir a taxa em qualquer temperatura.
A contante de equilíbrio pode ser obtida com a seguinte correlação:

$K = \exp(\frac{-\Delta G^o}{RT})$

As variações de pressões parciais foram obtidas com a seguinte correlação, onde $P_0 = P$:

$P_j = \frac{P_{A0}(\theta_j + \nu_j X)}{1+\epsilon X}\left(\frac{P}{P_0}\right)\frac{T_0}{T}$

A variação da temperatura é obtida com o balanço de energia:

$$T = \frac{X\ (-\Delta H^o) + (Cp_E + \frac{F_I}{F_{E0}})\ Cp_I)T_0 + X\ T_R \Delta Cp}{(Cp_E + \frac{F_I}{F_{E0}}\ Cp_I) + X\ \Delta Cp}$$

$\Delta Cp = Cp_A + Cp_H - Cp_E$

Método utilizado:
Construção de uma função que faz o cálculo de T e $r_e$ para cada conversão fornecida
Inicia com uma conversão 0 e T = T0;
Retorno $\frac{F_{E0}}{-r_E}$, T, $-r_E$;
$Cp_s$ são calculados a cada nova iteração com as equações pertinentes;
Mais passo na conversão até 0,75;
Retorno $\frac{F_{E0}}{-r_E}$, T, $-r_E$;
Construção do gráfico de Levenspiel $\frac{F_{E0}}{-r_E}$ versus $X$;
Cálculo da área abaixo da curva com método numérico de Simpson;

.
.
.

Conclusões:
Massa de Catalisador para pressão de 20 atm e T de 750 K: 115.2475 kg
A massa de catalisador é altamente dependente da temperatura de entrada no reator
A massa de catalisador é mais sensível a variações de temperatura que pressão, como demonstrado nos gráficos abaixo

[1]:
```
import numpy as np
from scipy.integrate import simps
```

```
import matplotlib.pyplot as plt
```

[2]:
```
def function(X, T, T0, P0):
    TR = 298 # K

    FE0 = 945894.82 # mol/h de Ethanol na entrada do reator
    FI = 2417286.76 # mol/h de Inertes (água) na entrada do reator

    yE0 = FE0 / (FE0 + FI)
    PE0 = yE0*P0 # atm

    epsilon = 0.28

    PE = (PE0*(1-X)/(1+epsilon*X))*(T0/T) # atm
    PH = (PE0*X/(1+epsilon*X))*(T0/T) # atm
    PA = (PE0*X/(1+epsilon*X))*(T0/T) # atm

    R = 1.987 # cal/mol*K
    R_u = 8.314 #J/mol*K

    dGA = -133000 # J/mol
    dGE = -167900 # J/mol
    dGH = 0 # J/mol

    dG_padrao = dGA + dGH - dGE # J/mol

    dHrx = 82500 # J/mol*K

    CpA = 8.314*(1.693+0.017978*T - 0.000006158*T**2) # J/mol*K
    CpE = 8.314*(3.518+0.020001*T - 0.000006002*T**2) # J/mol*K
    CpH = 8.314*(3.249+0.000422*T + (0.083*10**5 / (T**2))) # J/mol*K
    CpI = 8.314*(3.47+0.00145*T + (0.121*10**5 / (T**2))) # J/mol*K

    dCp = CpA + CpH - CpE # J/mol*K

    T_num = X * (-dHrx) + (CpE + (FI/FE0)*CpI)*T0 + X*TR*dCp
    T_den =  (CpE + (FI/FE0)*CpI) + X*dCp
    T = T_num/T_den    # K

    K = np.exp(-dG_padrao/(R_u*T))

    KE = np.exp(5560/(R*T) - 5.97) # atm^-1
    KA = np.exp(11070/(R*T) - 9.40) # atm^-1
    KH = np.exp(6850/(R*T) - 7.18) # atm^-1
    k = np.exp(-15300/(R*T) + 15.32) #mol/ g*h

    rE = k*KE*(PE - (PA*PH)/K) / (1 + KE*PE + KA*PA + KH*PH)**2
```

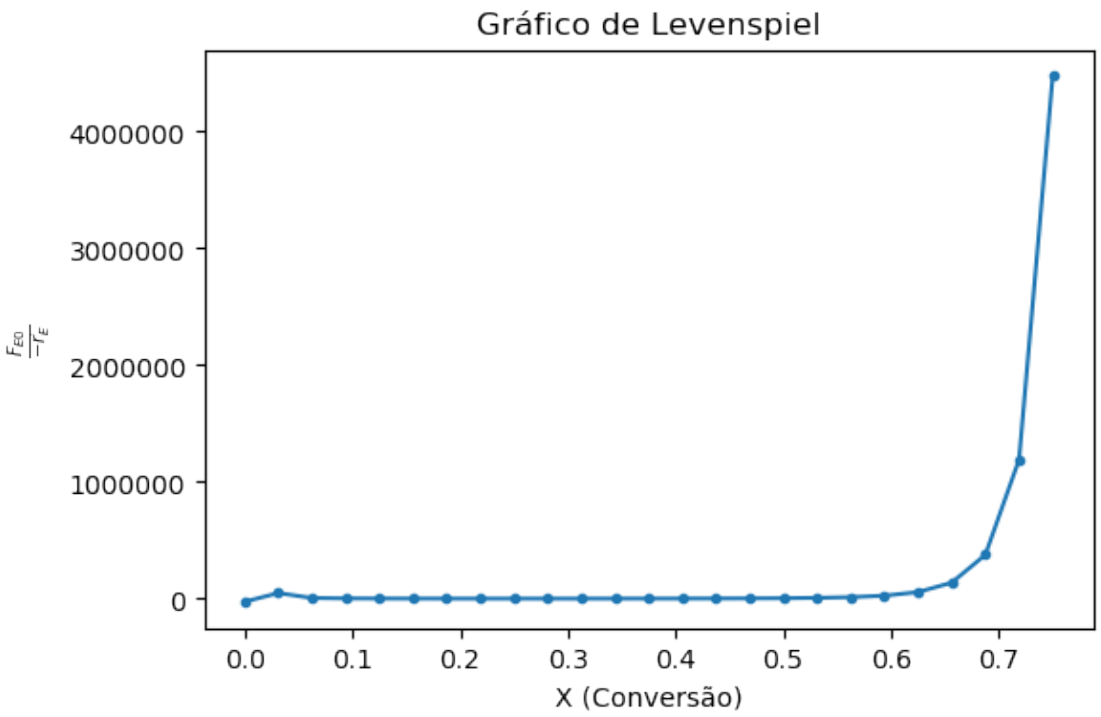
```
#    print("T {}\n X {}\n FE {}\n FI {}\n FA {}\n rE {}".format(T, X, FE, FI,␣
 ↪FA, rE))
#    print("CpA {}\n CpH {}\n CpE {}\n CpI {}\n dCp {}\n".format(CpA, CpH, CpE,␣
 ↪CpI, dCp))
#     print(PE, PA)
    return FE0/-rE, T, -rE
```

[3]:
```
x = np.linspace(0, 0.75, 25)
T = T0 = 750 # K
P0 = 20 # atm

sol = []
T_axis = []
re_axis = []
reac_axis = []

for i in x:
    reac, T, re = function(i, T, T0, P0)
    reac_axis.append(reac)
    T_axis.append(T)
    re_axis.append(re)

massa_cat = simps(reac_axis, x)
```

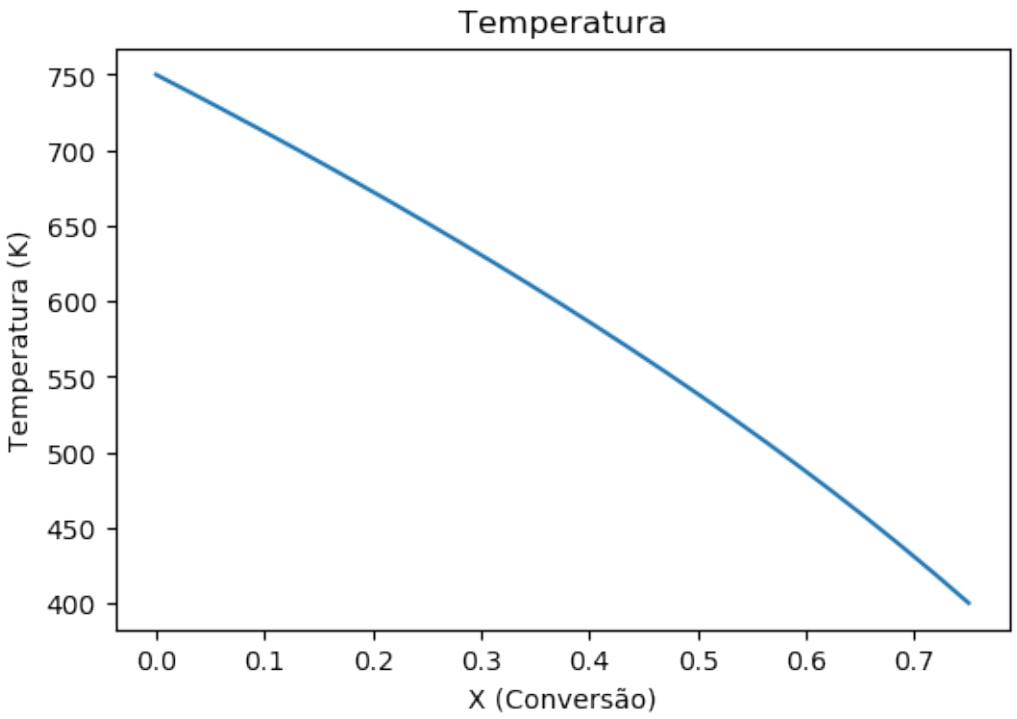
[4]:
```
print(f"Massa de Catalisador para pressão de 20 atm e T de 750 K: {massa_cat/
 ↪1000:5.7} kg")
```

```
Massa de Catalisador para pressão de 20 atm e T de 750 K: 115.2475 kg
```

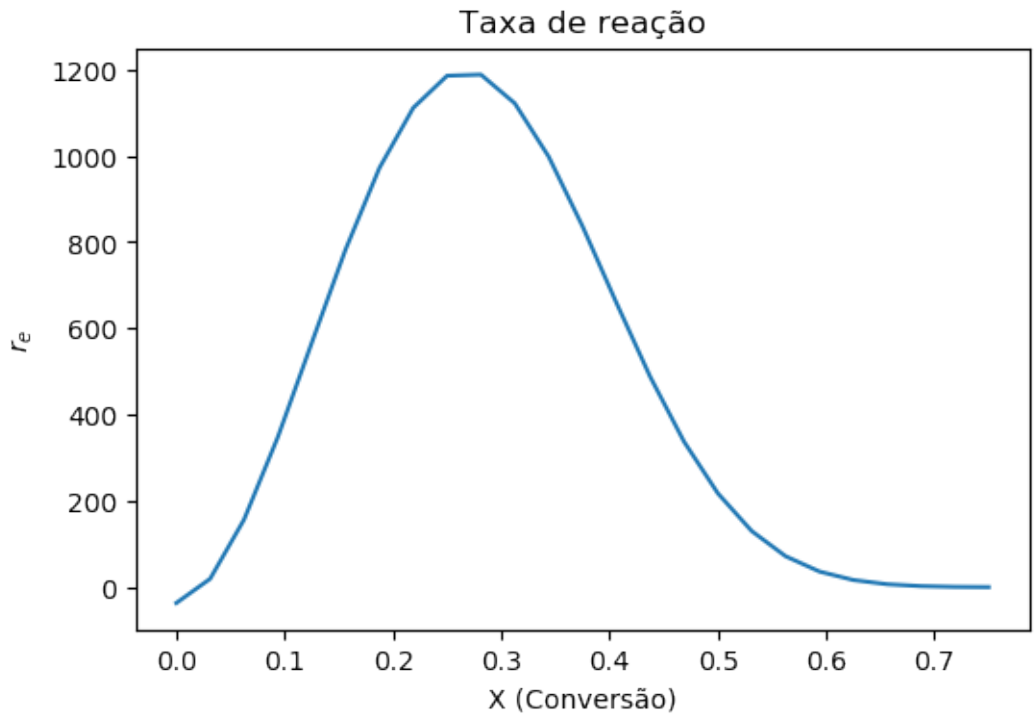
[5]:
```
fig = plt.figure(dpi=100)
plt.title("Gráfico de Levenspiel")
plt.xlabel("X (Conversão)")
plt.ylabel(r"$\frac{F_{E0}}{-r_E}$")
plt.plot(x, reac_axis, marker=".")
plt.show()
```



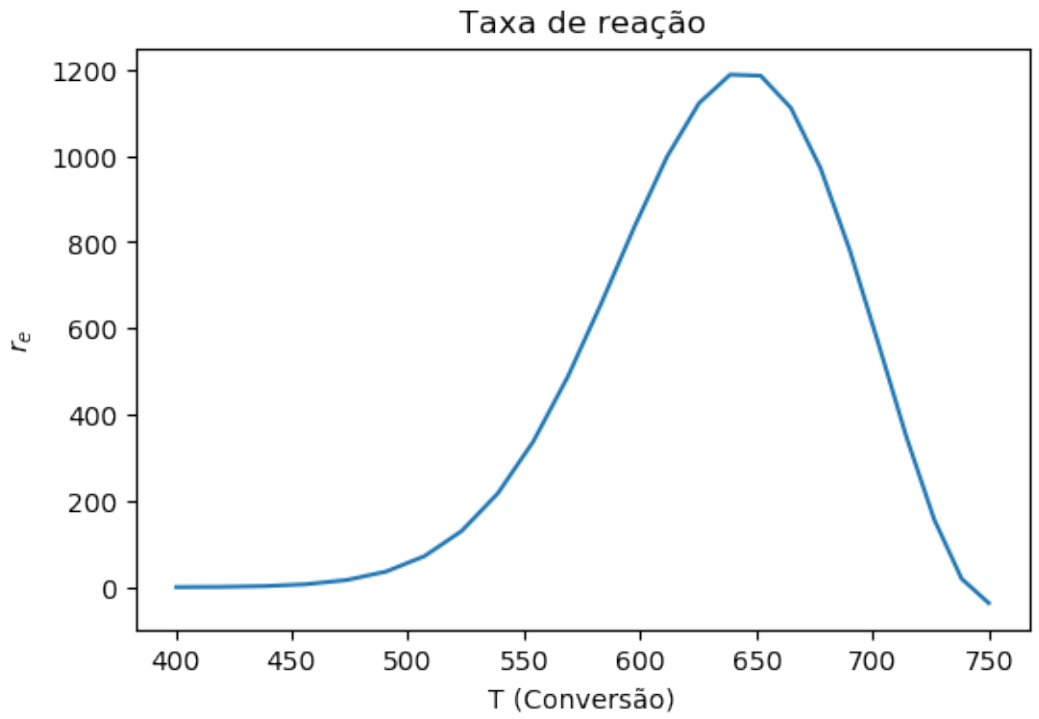

[6]:
```
fig = plt.figure(dpi=100)
plt.title("Temperatura")
plt.xlabel("X (Conversão)")
plt.ylabel("Temperatura (K)")
plt.plot(x, T_axis)
plt.show()
```




[7]:
```
fig = plt.figure(dpi=100)
plt.title("Taxa de reação")
plt.xlabel("X (Conversão)")
plt.ylabel("$r_e$")
plt.plot(x, re_axis)
plt.show()
```

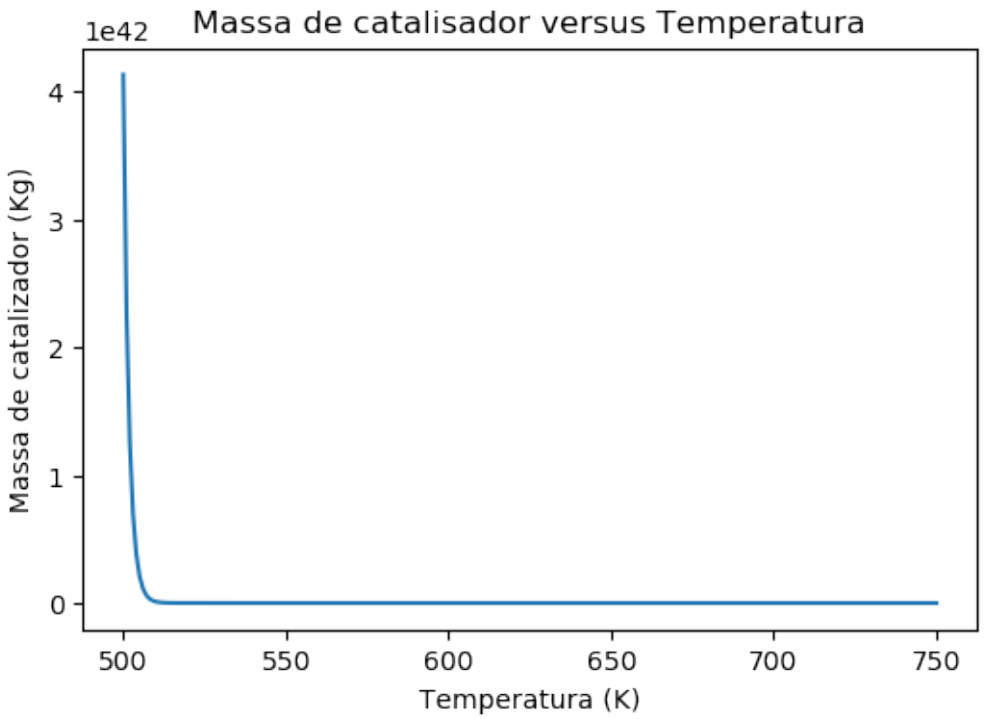



[8]:
```
fig = plt.figure(dpi=100)
plt.title("Taxa de reação")
plt.xlabel("T (Conversão)")
plt.ylabel("$r_e$")
plt.plot(T_axis, re_axis)
plt.show()
```



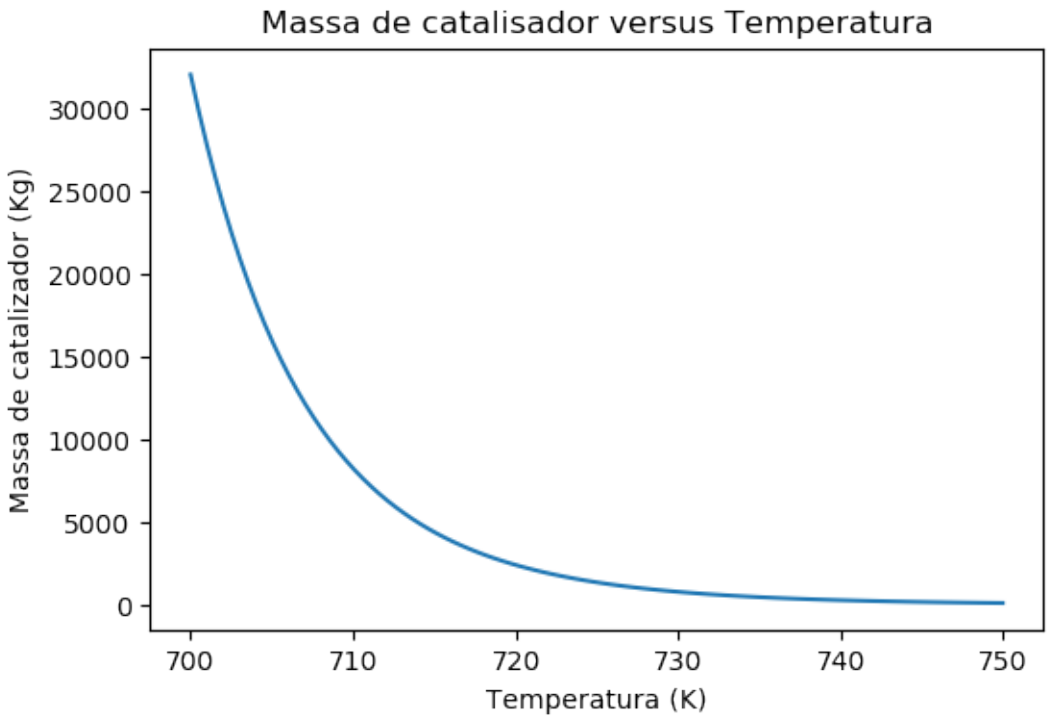

[9]:
```
temps = np.linspace(500, 750, 250)
massas = []
for Ti in temps:
    x = np.linspace(0, 0.75, 25)
    T = T0 = Ti # K
    P0 = 20 # atm
    reac_axis = []

    for i in x:
        reac, T, re = function(i, T, T0, P0)
        reac_axis.append(reac)

    massa_cat = simps(reac_axis, x)/1000
    massas.append(massa_cat)
```

[10]:
```
fig = plt.figure(dpi=100)
plt.title("Massa de catalisador versus Temperatura")
plt.ylabel("Massa de catalizador (Kg)")
plt.xlabel("Temperatura (K)")
plt.plot(temps, massas)
plt.show()
```



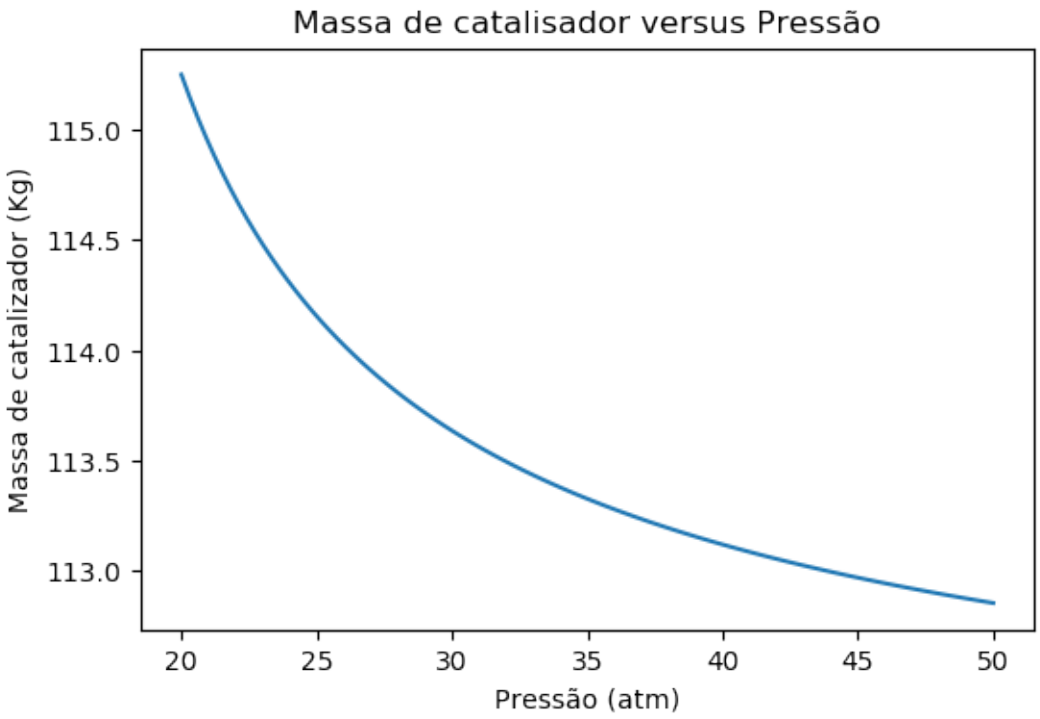

[11]:
```
temps = np.linspace(700, 750, 100)
massas = []
for Ti in temps:
    x = np.linspace(0, 0.75, 25)
    T = T0 = Ti # K
    P0 = 20 # atm
    reac_axis = []

    for i in x:
        reac, T, re = function(i, T, T0, P0)
        reac_axis.append(reac)

    massa_cat = simps(reac_axis, x)/1000
    massas.append(massa_cat)
```

[12]:
```
fig = plt.figure(dpi=100)
plt.title("Massa de catalisador versus Temperatura")
plt.ylabel("Massa de catalizador (Kg)")
plt.xlabel("Temperatura (K)")
plt.plot(temps, massas)
plt.show()
```




[13]:
```
pressoes = np.linspace(20, 50, 100)
massas = []
for P0 in pressoes:
    x = np.linspace(0, 0.75, 25)
    T = T0 = 750 # K
    reac_axis = []

    for i in x:
        reac, T, re = function(i, T, T0, P0)
        reac_axis.append(reac)

    massa_cat = simps(reac_axis, x)/1000
    massas.append(massa_cat)
```

[14]:
```
fig = plt.figure(dpi=100)
plt.title("Massa de catalisador versus Pressão")
plt.ylabel("Massa de catalizador (Kg)")
plt.xlabel("Pressão (atm)")
plt.plot(pressoes, massas)
plt.show()
```


Massa de catalisador versus Pressão
Massa de catalizador (Kg)
115.0
114.5
114.0
113.5
113.0
20
25
30
35
40
45
50
Pressão (atm)

[ ]: