

# YAG\_laser

December 1, 2021

```
[2]: # -*- coding: utf-8 -*-
      """
      Created on Sat Nov 27 15:36:20 2021

      @author: ruben
      """

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

      def odes(x,t):
          #adicionar constantes
          R1 = 1
          R2 = 0.95
          l_g = 1 #m
          sigma_0 = 5.92e-23 #m2
          N = 1e25 #part/m3
          R_p = 1.8 #s-1
          nt = (1-R1*R2)/(2*l_g*sigma_0*N)
          eta = 1e-12
          tau21 = 1.2e-3
          tau = 3e-6 #s
          taup = tau/(1-R1*R2)
          Rlinha_p = R_p*taup

          #relacionar cada EDO com um elemento de vetor
          N = x[0]
          I = x[1]

          #definir as EDOS
          dNdt = Rlinha_p*(1-N) - (taup/tau21)*N - (taup/tau21)*2*I*N
          dIdt = -I + (N/nt)*I + eta*(N/nt)

          return [dNdt, dIdt]

      #condições iniciais das EDOS
```

```

x0 = [0,0]

#delcarar um vetor temporal (janela de tempo de integração)
t = np.linspace(0,15,15000)

#função para resolver as EDOS
x = odeint(odes, x0, t)

N = x[:,0]
I = x[:,1]

#plotar os resultados

plt.title('População N_normalizada')
plt.plot(t,N , c = 'r')
plt.ylabel('$N_2/N$')
plt.xlabel('$t/\tau_p$')
plt.show()

plt.title('Intensidade Isat_normalizada')
plt.ylabel('$I/I_{sat}$')
plt.xlabel('$t/\tau_p$')
plt.plot(t,I, c= 'r')

plt.show()

```

