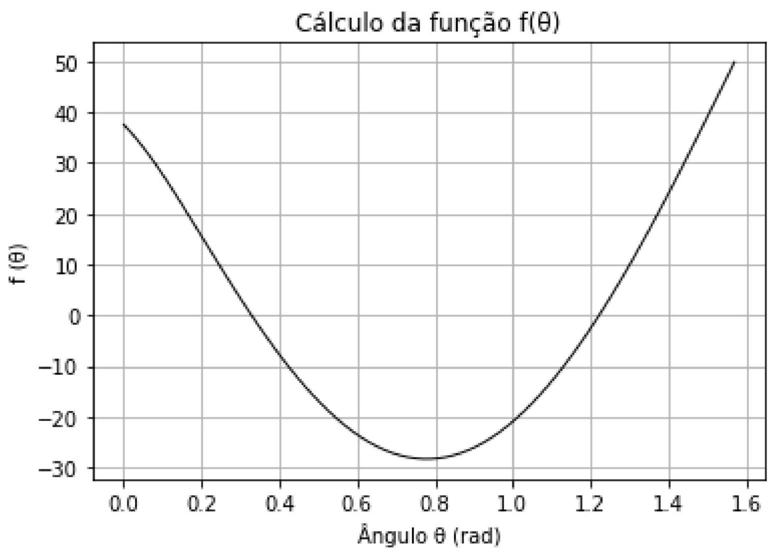
1. Construa um gráfico de f(θ) em função de θ usando a equação (6)

```
import matplotlib.pyplot as plt
import numpy as np

def f(x, yo, vo, g, theta):

  f_theta = (x-(((vo*np.cos(theta))/g)*(((vo)*(np.sin(theta)))+np.sqrt(((vo**2)*(np.sin(th

  return f_theta

angulomaximo = np.pi/2

arrayangulo = []

arrayangulo = np.arange(0, angulomaximo, 0.01)

arrayresultado = []

for i in arrayangulo:
  arrayresultado.append(f(50.0, 1.0, 27.778, 9.98, i))

plt.plot (arrayangulo, arrayresultado, linewidth=1.0, color = 'black')
plt.xlabel('Ângulo θ (rad)')
plt.ylabel('f (θ)')
plt.grid(True)
plt.title('Gráfico da função f(θ)')
plt.show()
```



2. Utilize um dos métodos vistos em aula para determinar os valores de θ que correspondem aos zeros dessa função, ou seja, onde a função cruza o eixo das abscissas

Método da secante :

$$\frac{f(b)-f(a)}{b-a}=\frac{f(a)-0}{a-c}$$

E utilizaremos **a = 0,3** e **b = 1,3**

```
a_1 = 0.3 #valores inferiores
a_2 = 1.3

b_1 = 0.6 #valores superiores
b_2 = 1.6

f_theta = lambda theta: (50-((27.8*np.cos(theta)/9.98)*(27.8*np.sin(theta)+np.sqrt((27.8**

def secante(f_theta, a, b, e = 1e-5):

    erro_g = 1000

    while erro_g > e:
        x = (a-((b-a)/(f_theta(b)-f_theta(a)))*f_theta(a))

        if abs(x-a) < abs(x-b):
            b = x
        else:
            a = x

        erro_g = abs((b-a)/b)

    return x

print('Assim, obtemos pelo método da secante θ1 e θ2, os quais são os ângulos em que a fun

print(f'\n')

theta_1 = secante(f_theta, a_1, b_1)
print(f'θ1 = {theta_1} rad')

print(f'\n')

theta_2 = secante(f_theta, a_2, b_2)
print(f'θ2 = {theta_2} rad')
```

```
Assim, obtemos pelo método da secante θ1 e θ2, os quais são os ângulos em que a funçã


θ1 = 0.3279165440602312 rad


θ2 = 1.2228824518211696 rad
```

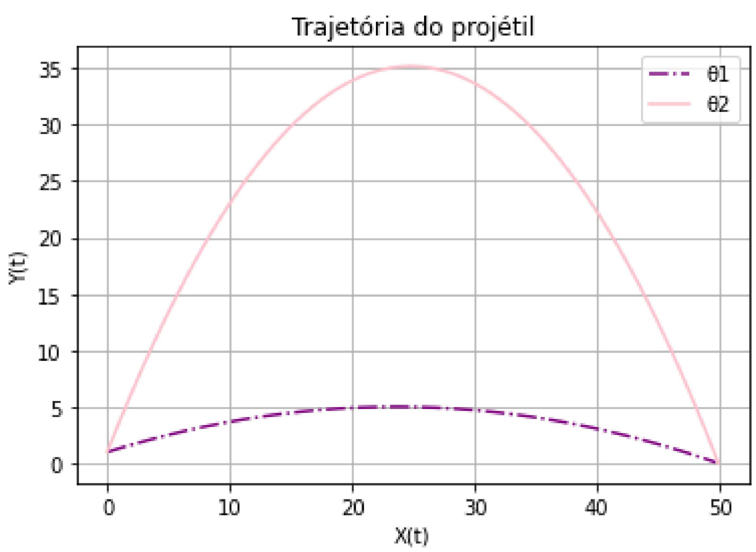
3. Para cada um dos ângulos θ, grafique a correspondente trajetória do projétil, y(t) × x(t). Você pode usar a equação (4) para determinar o tempo no qual o projétil atinge o alvo. Coloque as duas trajetórias em uma mesma figura, indicando na legenda do gráfico qual o valor de θ cada trajetória corresponde.

equação (4) :

$$t = \frac{1}{g}\left[vosen\left(\theta\right) + \sqrt{vo^2 sen^2\left(\theta\right) + 2gyo}\right]$$

```
def tempo_alvo(t):
  tempo_t = ( (1/9.98)* (((27.778)* (np.sin(t))) + (np.sqrt((((27.778)**2)*((np.sin(t))**2
  return tempo_t


tempo_t_1 = tempo_alvo(theta_1)
tempo_t_2 = tempo_alvo(theta_2)


def horizontal(ta, tt):
  x = (27.778)*(np.cos(ta)*(tt))
  return x

def vertical(ta, tt):
  y = (1 + ((27.778)*(tt)*(np.sin(ta))) - ((1/2)*(9.98)*(tt)**2))
  return y


arrayx_1 = np.arange(0, tempo_t_1, 0.01)
arrayvazio1x = []
for time in arrayx_1:
    arrayvazio1x.append(horizontal(theta_1, time))

arrayy1 = np.arange(0, tempo_t_1, 0.01)
arrayvazio1y = []
for time in arrayy1:
  arrayvazio1y.append(vertical(theta_1, time))

arrayx_2 = np.arange(0, tempo_t_2, 0.01)
arrayvazio_2x = []
for time in arrayx_2:
    arrayvazio_2x.append(horizontal(theta_2, time))


arrayy2 = np.arange(0, tempo_t_2, 0.01)
arrayvazio_2y = []
for time in arrayy2:
  arrayvazio_2y.append(vertical(theta_2, time))

plt.plot(arrayvazio1x, arrayvazio1y,'-.b', linewidth=1.5, label ='θ1', color = 'purple')
plt.plot(arrayvazio_2x, arrayvazio_2y, '-r',linewidth=1.5,label = 'θ2' , color = 'pink')
plt.ylabel('Y(t)')
```

```
plt.xlabel('X(t)')
plt.grid(True)
plt.title('Trajetória do projétil')
plt.legend()
plt.show()
```



link do trabalho: https://colab.research.google.com/drive/1KRe95Urws1YDNTgWDO4_2ZbDI-ICcHNH?usp=sharing

✓ 0s conclusão: 16:16