Bibliotecas

```
import sympy as sp
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import animation
from matplotlib.animation import PillowWriter
from sympy.printing import latex
```

Simbolos do sympy

```
t, g, l, m, k = sp.symbols('t g l m k')

theta = sp.symbols(r'theta', cls=sp.Function)
theta = theta(t)
theta_dot = sp.diff(theta, t)
theta_ddot = sp.diff(theta_dot, t)

u = sp.symbols(r'u', cls=sp.Function)
u = u(t)
u_dot = sp.diff(u, t)
u_ddot = sp.diff(u_dot, t)
```

Equações 'x' e 'y'

```
x = (l+u)*sp.sin(theta)
y = -(l+u)*sp.cos(theta)
```

Equação energia cinética

```
T1 = sp.Rational(1,2)*m*(sp.diff(x, t)**2 + sp.diff(y, t)**2)

T = T1

T
```

$$\frac{m\left(\left(-\left(-l-u(t)\right)\sin\left(\theta(t)\right)\frac{d}{dt}\theta(t)-\cos\left(\theta(t)\right)\frac{d}{dt}u(t)\right)^2+\left(\left(l+u(t)\right)\cos\left(\theta(t)\right)\frac{d}{dt}\theta(t)+\sin\left(\ldots\right.\right.\right.}{2}$$

Equação energia potencial

```
U1 = y*m*g
U2 = sp.Rational(1,2)*k*(u**2)

U = U1 + U2

U
```

$$gm\left(-l-u(t)\right)\cos\left(\theta(t)\right)+\frac{ku^2(t)}{2}$$

Lagrangeano

```
L = T - U

L
```

$$-gm\left(-l-u(t)\right)\cos\left(\theta(t)\right)-\frac{ku^2(t)}{2}+\frac{m\left(\left(-\left(-l-u(t)\right)\sin\left(\theta(t)\right)\frac{d}{dt}\theta(t)-\cos\left(\theta(t)\right)\frac{d}{dt}u(t)\right)^2+\left(\left(l+u(t)\right)\cos\left(\theta(t)\right)\frac{d}{dt}\theta(t)+\sin\left(\ldots\right.\right.\right.}{2}$$

EDO theta(t)

```
eq = sp.diff(L, theta) - sp.diff(sp.diff(L, theta_dot), t)
EDO1 = sp.simplify(eq)

EDO1
```

$$-m\left(gl\sin\left(\theta(t)\right) + gu(t)\sin\left(\theta(t)\right) + l^2\frac{d^2}{dt^2}\theta(t) + 2lu(t)\frac{d^2}{dt^2}\theta(t) + 2l\frac{d}{dt}\theta(t)\frac{d}{dt}u(t) + u^2(t)\frac{d}{d}\right.$$

EDO u(t)

```
eq2 = sp.diff(L, u) - sp.diff(sp.diff(L, u_dot), t)
EDO2 = sp.simplify(eq2)
```

```
EDO2
```

$$gm\cos\left(\theta(t)\right) - ku(t) + lm\left(\frac{d}{dt}\theta(t)\right)^2 + mu(t)\left(\frac{d}{dt}\theta(t)\right)^2 - m\frac{d^2}{dt^2}u(t)$$

Solução das EDO

```
sols = sp.solve([EDO1, EDO2], [theta_ddot, u_ddot], simplify=False, rational=False)
```

```
sols
```

```
{Derivative(theta(t), (t, 2)): -g*sin(theta(t))/(l + u(t)) - 2*Derivative(theta(t), t)*Derivative(u(t), t)/(l + u(t)),
 Derivative(u(t), (t, 2)): g*cos(theta(t)) - k*u(t)/m + l*Derivative(theta(t), t)**2 + u(t)*Derivative(theta(t), t)**2}
```

```
dz1dt_f = sp.lambdify((theta, theta_dot, u, u_dot, g, l, m, k), sols[theta_ddot])
dz2dt_f = sp.lambdify((theta, theta_dot, u, u_dot, g, l, m, k), sols[u_ddot])
dthetadt_f = sp.lambdify(theta_dot, theta_dot)
dudt_f = sp.lambdify(u_dot, u_dot)
```

```
def dSdt(S, t, g, l, m, k):
    theta, z1, u, z2 = S
    return [dthetadt_f(z1),
            dz1dt_f(theta, z1, u, z2, g, l, m, k),
            dudt_f(z2),
            dz2dt_f(theta, z1, u, z2, g, l, m, k)]
```

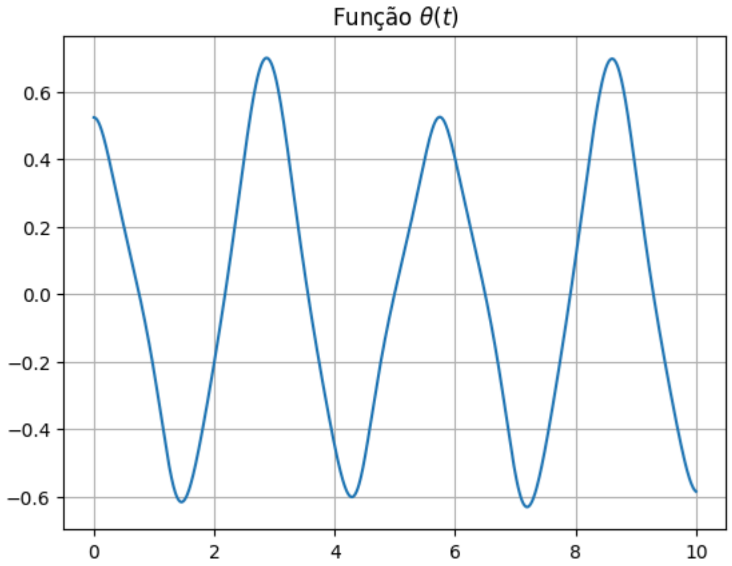
```
t = np.linspace(0, 10, 1000)
g = 9.81
l = 1
m = 1
k = 24
deg = 30
theta0 = deg*np.pi/180
dtheta0 = 0
u0 = 0
du0 = 0

sol = odeint(dSdt, y0 =[theta0, dtheta0, u0, du0], t=t, args=(g, l, m, k))
```

```
the = sol.T[0]
upos = sol.T[2]
thedot = sol.T[1]
uposdot = sol.T[3]
```

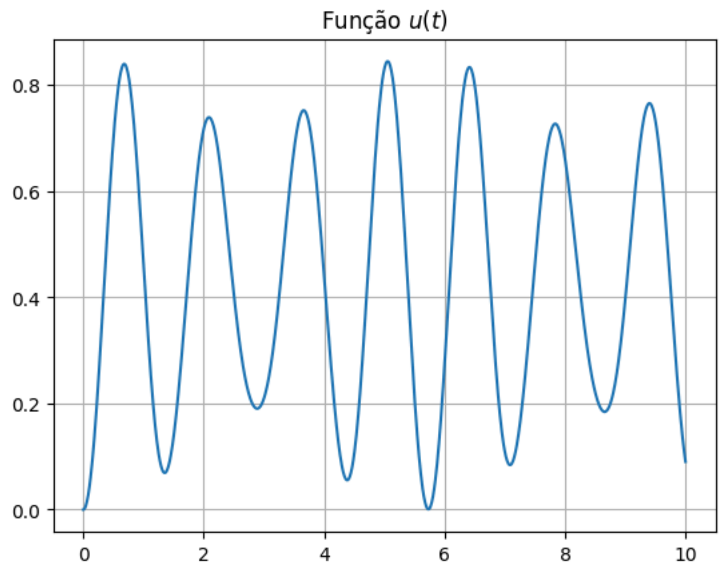
```
def pos(t, the1, u1, l):
    x1 = (l+u1)*np.sin(the1)
    y1 = -(l+u1)*np.cos(the1)
    return [
        x1, y1
    ]
```

```
x11, y11 = pos(t, the, upos, l)
```

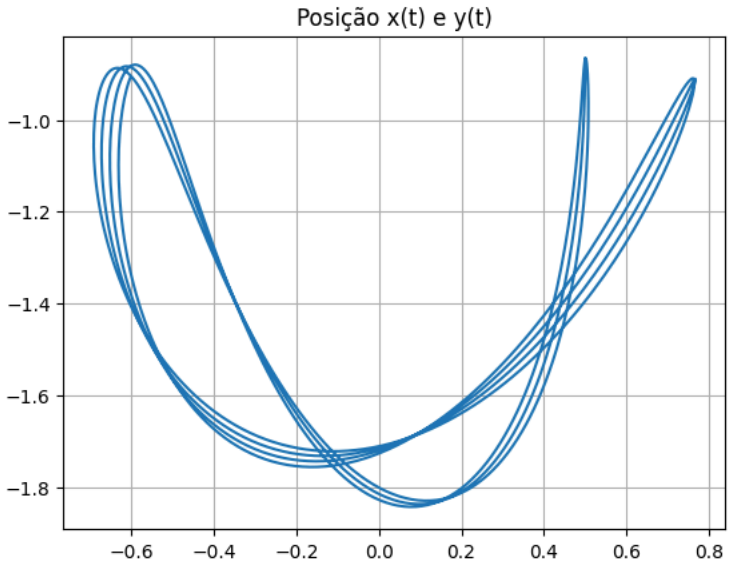
```
plt.title(f'Função ${latex(theta)}$')
plt.plot(t, the)
plt.grid()
plt.show()
```



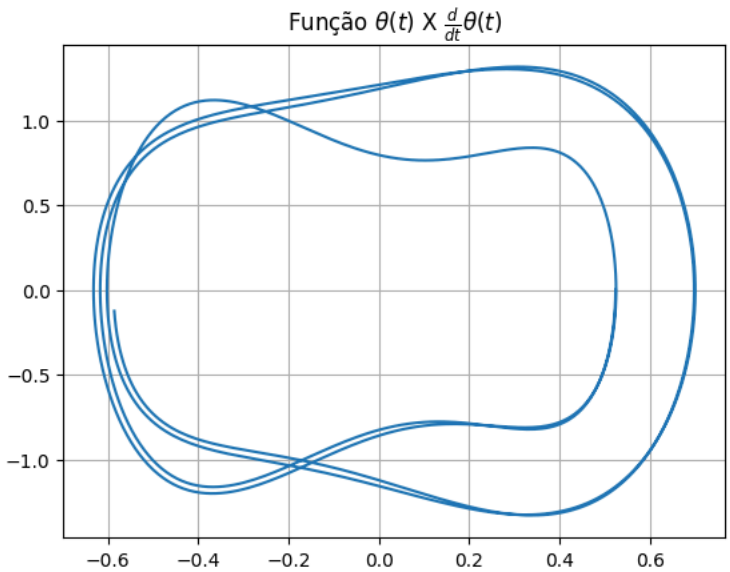

```
plt.title(f'Função ${latex(u)}$')
plt.plot(t, upos)
plt.grid()
plt.show()
```




```
plt.title(f'Posição x(t) e y(t)')
plt.plot(x11, y11)
plt.grid()
plt.show()
```

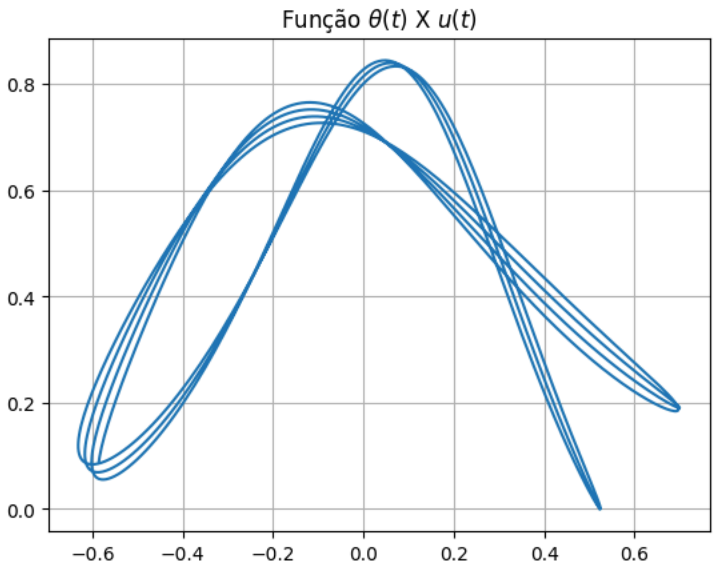



```
plt.title(f'Função ${latex(theta)}$ X ${latex(theta_dot)}$')
plt.plot(the, thedot)
plt.grid()
plt.show()
```




```
plt.title(f'Função ${latex(u)}$ X ${latex(u_dot)}$')
plt.plot(upos, uposdot)
plt.grid()
plt.show()
```

```
plt.title(f'Função ${latex(theta)}$ X ${latex(u)}$')
plt.plot(the, upos)
plt.grid()
plt.show()
```




```
def animate(i):
    ln.set_data([0, x11[i]], [0, y11[i]])
    cur.set_data(x11[:i+1], y11[:i+1])

fig, ax = plt.subplots(1, 1, figsize=(5, 5))
ax.set_xlim(-1.5, 1.5)
ax.set_ylim(-2.5, 0.5)
ax.grid()
ln, = ax.plot([], [], 'bo--', lw=2, markersize=8)
cur, = ax.plot(x11[0], y11[0], 'black', lw=1)

ani = animation.FuncAnimation(fig, animate, frames=1000, interval=50)
ani.save('pendulo.gif', writer='pillow', fps=25)
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-21-6f503a301dc3> in <cell line: 13>()
     11
     12 ani = animation.FuncAnimation(fig, animate, frames=1000, interval=50)
---> 13 ani.save('pendulo.gif', writer='pillow', fps=25)

                         9 frames
/usr/local/lib/python3.10/dist-packages/PIL/ImageChops.py in subtract_modulo(image1, image2)
    235     image1.load()
    236     image2.load()
--> 237     return image1._new(image1.im.chop_subtract_modulo(image2.im))
    238
    239

KeyboardInterrupt:
```