import numpy as np

import matplotlib.pyplot as plt

# Parâmetros do sistema (Oscilador Amortecido)

m = 1.0 # Massa (kg)

k = 10.0 # Constante da mola (N/m)

b = 0.5 # Coeficiente de amortecimento (N.s/m)

x0 = 5.0 # Deslocamento inicial (m)

v0 = 0.0 # Velocidade inicial (m/s)

# Função para calcular a solução analítica

def oscilador\_amortecido(t, m, k, b, x0, v0):

gamma = b / (2 \* m)

omega\_0 = np.sqrt(k / m)

omega = np.sqrt(omega\_0\*\*2 - gamma\*\*2)

A = np.sqrt(x0\*\*2 + (v0 + gamma \* x0)\*\*2 / omega\*\*2)

phi = np.arctan((v0 + gamma \* x0) / (omega \* x0))

x\_t = A \* np.exp(-gamma \* t) \* np.cos(omega \* t - phi)

return x\_t

# Tempo de simulação

t = np.linspace(0, 20, 1000)

# Cálculo da solução

x\_analitico = oscilador\_amortecido(t, m, k, b, x0, v0)

# Definição das fases (Constância, Inconstância, Finitude)

gamma = b / (2 \* m)

tau = 1 / gamma # Tempo de relaxamento (63% da redução)

t\_fases = {

"Constância": t <= 0.5 \* tau,

"Inconstância": (t > 0.5 \* tau) & (t <= 3 \* tau),

"Finitude": t > 3 \* tau

}

# Criar o gráfico

plt.figure(figsize=(10, 5))

plt.plot(t, x\_analitico, label="Oscilador Amortecido", color="blue", linestyle="-")

# Marcar as fases

colors = {"Constância": "green", "Inconstância": "orange", "Finitude": "red"}

for fase, mask in t\_fases.items():

plt.fill\_between(t, x\_analitico, where=mask, color=colors[fase], alpha=0.1, label=fase)

# Linhas de transição

plt.axvline(0.5 \* tau, color='gray', linestyle=':', linewidth=1)

plt.axvline(3 \* tau, color='gray', linestyle=':', linewidth=1)

plt.text(0.5 \* tau + 0.2, 4, r'$t = \tau/2$', rotation=90, fontsize=8)

plt.text(3 \* tau + 0.2, 4, r'$t = 3\tau$', rotation=90, fontsize=8)

# Personalização do gráfico

plt.title("Damped Oscillator: Constancy, Inconstancy, and End")

plt.xlabel("Time (s)")

plt.ylabel("Displacement (m)")

plt.axhline(0, color='black', linewidth=0.5)

plt.legend(loc="upper right")

plt.grid(True, linestyle="--", alpha=0.6)

# Exibir o gráfico

plt.show()