

LISTA 2 – Oscilador harmônico, oscilações amortecidas e forçadas, ondas e som

ENTREGA INDIVIDUAL PARA OS TRABALHOS EM GRUPO

PRAZO: 05/05 – 23:59

EXPERIMENTAÇÃO INDIVIDUAL:

1. EXPERIMENTO VIRTUAL - OSCILADOR

Acesse o site:

https://phet.colorado.edu/sims/html/masses-and-springs/latest/masses-and-springs_all.html

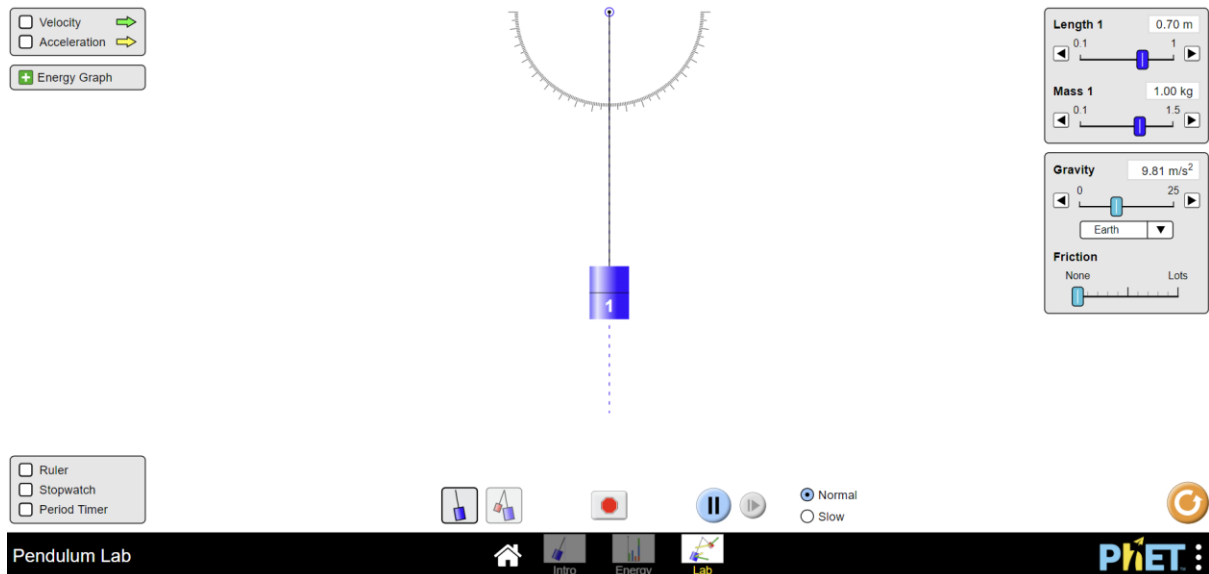
The screenshot shows the PhET 'Masses and Springs' simulation interface. On the left is an 'Energy Graph' with axes for KE, PE_{spring}, PE_{gravity}, E_{total}, and E_{total}. The central part shows a spring-mass system with a mass of 100 g and a spring constant of Small. The right panel contains various controls: checkboxes for Displacement, Natural Length, Mass Equilibrium, Movable Line, and Period Trace; a Gravity slider set to 9.8 m/s²; a Damping slider set to None; and checkboxes for Velocity and Acceleration. A bottom bar includes navigation icons for Intro, Vectors, Energy, Lab, and Office, along with the PhET logo.

- Escolha a **massa** como sendo os 3 primeiros números de seu CPF. Considere uma constante de mola intermediária
- Caso 1: Sem amortecimento (damping) – Determine a gravidade do Planeta X (Em Gravity selecione “Planet X”)
- Caso 2: Mantenha os parâmetros anteriores (massa, constante de mola e gravidade) e ligue o amortecimento i) (intermediário) e ii) intenso (Lots). Descreva os efeitos em termos das energias e oscilações para essas situações.

2. EXPERIMENTO VIRTUAL – PÊNDBULO

Acesso o site:

https://phet.colorado.edu/sims/html/pendulum-lab/latest/pendulum-lab_all.html



Escolha a massa em 1 Kg e altere o comprimento 1 (length 1) como sendo o ultimo digito de seu CPF dividido por 10. Se o ultimo digito for 0, considere o comprimento igual a 0.60.

Caso 1: Sem atrito (Friction) – Determine a gravidade do Planeta X (Em Gravity selecione “Planet X”

Caso 2: Mantenha os parâmetros e introduza o atrito. Determine o tempo necessário para o pêndulo parar de oscilar para atritos intermediários e intenso (Lots). Comente sobre os efeitos nas energias.

3. MODELAGEM NUMÉRICA:

Objetivos:

- i) Aplicar o método de Euler para resolver numericamente a equação diferencial ordinária de um pêndulo.
- ii) Simular e visualizar o comportamento de diferentes tipos de pêndulos:
 - Pêndulo simples
 - Pêndulo amortecido
 - Pêndulo forçado
 - Pêndulo amortecido e forçado
- iii) Analisar os diferentes regimes de amortecimento (crítico, sub-crítico, super-crítico).

Parte 1: Teoria

1. **Pêndulo Simples:**

- Equação: $\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin(\theta) = 0$

2. **Pêndulo Amortecido:**

- Equação: $\frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} + \frac{g}{l} \sin(\theta) = 0$
- Discutir os casos:
 - Amortecimento Crítico
 - Amortecimento Sub-crítico
 - Amortecimento Super-crítico

3. **Pêndulo Forçado:**

- Equação: $\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin(\theta) = A \cos(\omega t)$

4. **Pêndulo Amortecido e Forçado:**

- Equação: $\frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} + \frac{g}{l} \sin(\theta) = A \cos(\omega t)$

Parte 2: Método de Euler

- Explicar o método de Euler para resolver EDOs numericamente.
- Transformar as EDOs de segunda ordem em sistemas de primeira ordem.

Parte 3: Implementação em Python

- Criar um script em Python que implemente o método de Euler para cada tipo de pêndulo.
- Plotar os resultados para diferentes parâmetros (comprimento do pêndulo, coeficiente de amortecimento, amplitude e frequência da força externa).

Parte 4: Análise e Conclusão

- Comparar os comportamentos observados nas simulações com a teoria.
- Discutir a eficácia do método de Euler e possíveis melhorias.

Código de Exemplo em Python:

Aqui está um esqueleto básico que os alunos podem usar como ponto de partida para o código:

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

def pendulum(theta0, omega0, t, dt, g, l, b=0, A=0, omega_f=0):
 theta, omega = theta0, omega0
 history = [(theta, omega)]
 for _ in np.arange(0, t, dt):
 dtheta_dt = omega
 domega_dt = - (g / l) * np.sin(theta) - b * omega + A * np.cos(omega_f * t)
 theta += dtheta_dt * dt
 omega += domega_dt * dt
 history.append((theta, omega))
 return history

Parâmetros
g = 9.81 # aceleração da gravidade
l = 1.0 # comprimento do pêndulo
dt = 0.01 # intervalo de tempo
t = 10 # tempo total de simulação

Simulações
history_simple = pendulum(np.pi/4, 0, t, dt, g, l)
history_damped = pendulum(np.pi/4, 0, t, dt, g, l, b=0.1)
history_forced = pendulum(np.pi/4, 0, t, dt, g, l, A=0.1, omega_f=2)

Plotar resultados
plt.figure(figsize=(12, 8))
for i, history in enumerate([history_simple, history_damped, history_forced]):
 thetas,

 omegas = zip(*history)
 plt.subplot(3, 1, i+1)
 plt.plot(thetas)
 plt.title(["Pêndulo Simples", "Pêndulo Amortecido", "Pêndulo Forçado"][i])
plt.tight_layout()
plt.show()
```
```