

Modelagem e Resolução Numérica de Pêndulos Simples Amortecidos e Visualização Didática com Manim:

Esboço

Hézio Delfino Lopes de Alcântara 231028343*

IF-UnB

(Data: 20 de janeiro de 2026)

I. FFT E FINDPEAKS

De modo geral, a análise por FFT é utilizada aqui apenas como etapa de calibração para o algoritmo de detecção de picos. A Transformada Rápida de Fourier permite identificar a frequência dominante presente no sinal, i.e., a componente oscilatória de maior energia. A partir dessa frequência f_0 , estima-se o período fundamental $T = 1/f_0$, que é então convertido em uma distância mínima entre picos em número de amostras (distance $\approx 0.6T/\Delta t$).

```
# ---- Estima período via FFT ----
dt = np.median(np.diff(t_win))
yf = np.fft.rfft(signal - np.mean(signal))
xf = np.fft.rfftfreq(len(t_win), dt)

idx = np.argmax(np.abs(yf)[1:]) + 1
f0 = xf[idx]
period_est = 1.0 / f0

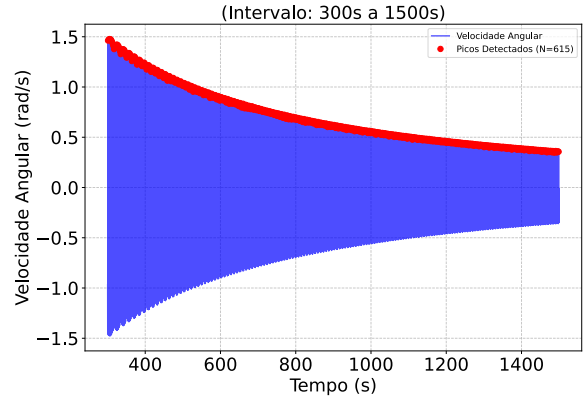
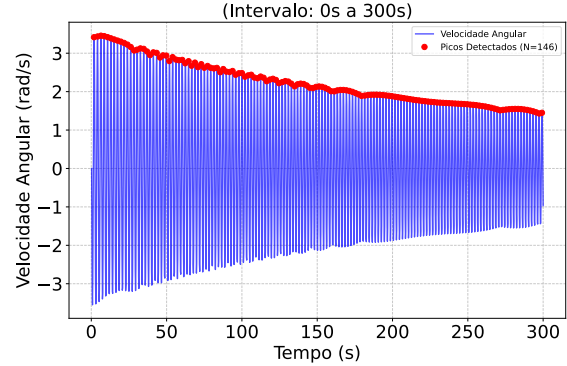
# ---- converte em distancia minima entre picos ----
distance_samples = int(0.6 * period_est / dt)

# ---- Usa a distância no find_peaks ----
peaks, props = find_peaks(signal, distance=distance_samples)
```

Dessa forma, esse valor ajusta automaticamente a escala temporal típica da oscilação, permitindo que o `find_peaks` identifique apenas picos físicos reais e ignore flutuações de ruído. Repare que `distance_samples` é usado no argumento `distance` do `find_peaks`.

II. VELOCIDADE X TEMPO

Com isso, foi possível pegar as velocidades máximas medidas pelo sensor



III. VELOCIDADE MÁXIMA X TEMPO

claramente um início com amortecimento forte (0-300s **figura 1**), onde a energia cai rápido, e uma fase posterior em que o sistema passa a perder energia de forma mais suave, típica de amortecimento viscoso/linear. A velocidade máxima decai o tempo todo, como esperado em um pêndulo amortecido. No regime de (300-1500s **figura 2**), a curva se aproxima lentamente de zero. Sugere pelo menos dois regimes de velocidade.

IV. VELOCIDADE MÁXIMA X AMPLITUDE

para um pendulo ideal sem atrito, quanto maior a amplitude, maior a velocidade no ponto mais baixo. No

* heziodelfino@gmail.com

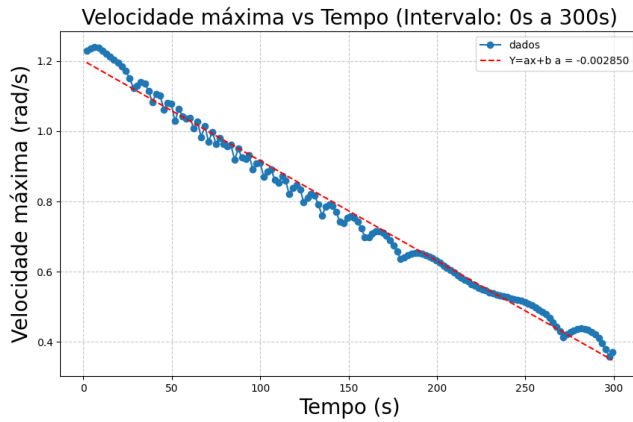


Figura 1: regime de amortecimento 'forte'

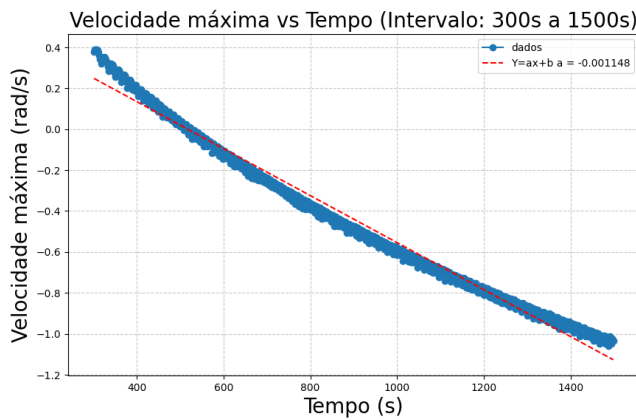
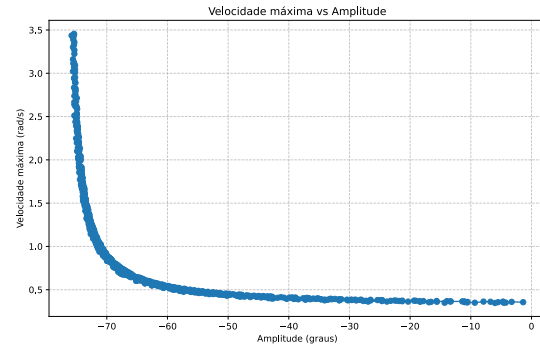
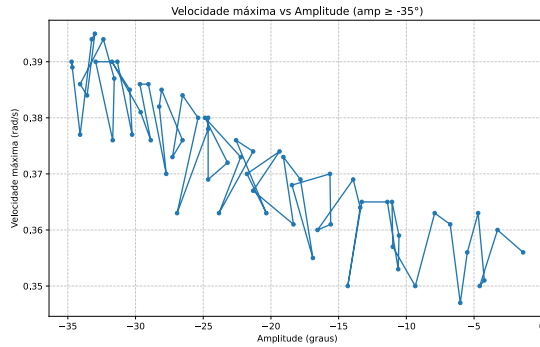
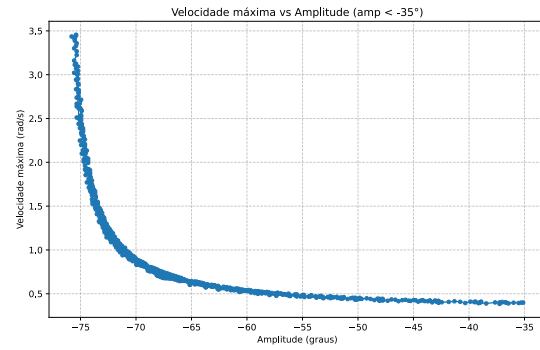


Figura 2: regime de amortecimento 'fraco'



entanto, observando o gráfico e sabendo q tem amortecimento, quanto maior a amplitude menor a velocidade máxima atingida.

V. CORREÇÃO DOS GRÁFICOS DE VELOCIDADE MÁXIMA VS AMPLITUDE

Ajustei o sentido de leitura do vetor da coluna de velocidade angular, medido pelo Pasco. O erro era um simples descuido na hora de plotar os dados no gráfico, o que anteriormente resultava numa interpretação física inversa à esperada.

```
# Aq inverte o vetor da velocidade angular
signal = df_plot[COLUNA_VELOCIDADE_ANGULAR].values[::-1]
#o trecho acima conserta o erro cometido anteriormente (18 de out)
```

VI. ESPAÇO DE FASE COM MANIM: ESTUDO INICIAL

A. Entendendo o espaço de fase

Pelo fato de não termos conseguido fazer o ajuste ainda. Achei coerente usar um modelo de pêndulo não-linear com amortecimento viscoso para criar um cenário hipotético que reproduza qualitativamente as fea- tures observadas (grandes amplitudes, decaimento de energia). Gereis três conjuntos com amplitude inicial 1.8 rad e amortecimentos $b = 0.05, 0.15, 0.5$; com isso consigo uma animação no Manim mostrando espaço de fase e comparar com os pontos experimentais. Isso per-

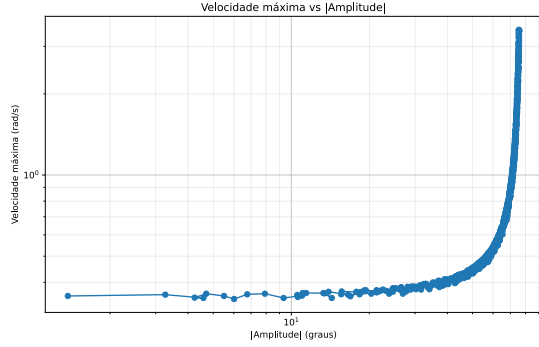


Figura 3: O comportamento não é linear: em pequenas amplitudes, a velocidade máxima varia pouco, mantendo a energia praticamente constante. À medida que a amplitude aumenta, a velocidade máxima cresce muito mais rapidamente do que prevê a aproximação linear. Pode-se concluir que, para grandes amplitudes, o pêndulo opera claramente fora do regime linear.

mite discutirmos: (i) se a forma geral da órbita está correta, (ii) se o ruído experimental é compatível com o modelo, e (iii) parâmetros a ajustar para tentar o fit real.

Este trecho implementa a equação diferencial do pêndulo amortecido e um gerador de dados sintéticos para simulações numéricas.

```
def pendulum_ode(t, y, b, g, L, m):
    theta, omega = y
    return [omega, - (g / L) * np.sin(theta) - (b / (m * L * L)) * omega]
```

o retorno da função é:

$$\dot{\theta} = \omega, \quad (1)$$

$$\dot{\omega} = \frac{-g}{L} \sin \theta - \frac{b}{mL^2} \omega. \quad (2)$$

dps disso defini uma função que gera uma simulação completa do pêndulo com tempo máximo de até 20 segundos:

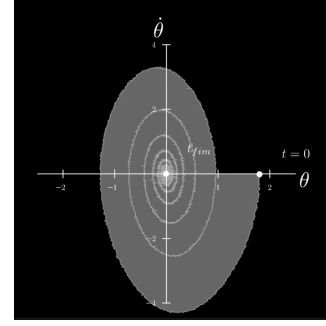
```
def make_dataset(theta0=1.6, omega0=0.0,
b=0.15, g=9.81,
L=1.0, m=1.0,
t_max=20.0, N=4000, noise_sigma=0.0):
```

o processo interno leva uma discretização temporal

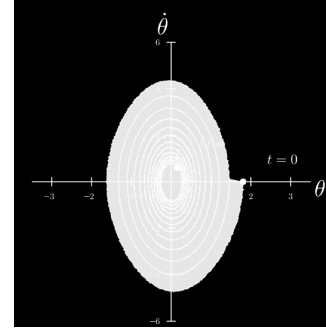
```
t_eval = np.linspace(0, t_max, N)
```

e também a resolução numérica da EDO:

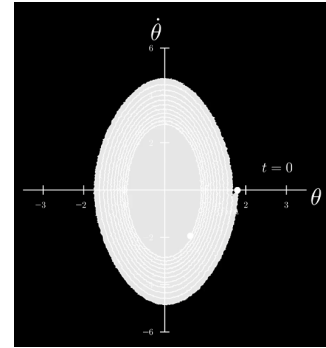
```
sol = solve_ivp(
    pendulum_ode,
    [0, t_max],
    [theta0, omega0],
    args=(b, g, L, m),
    t_eval=t_eval,
    atol=1e-9,
    rtol=1e-8)
```



(a) Amortecimento forte ($b = 0.5$)



(b) Amortecimento médio ($b = 0.15$)



(c) Amortecimento fraco ($b = 0.05$)

Figura 4: Resultados das simulações.

B. Phase flow

De modo geral, ao escolhermos um ponto no espaço de fase observamos a órbita correspondente — sua evolução no tempo. Já o *phase flow* reúne um conjunto de condições iniciais distintas e mostra simultaneamente as órbitas geradas por todas elas.

Na simulação, foram colocadas 10 condições iniciais diferentes conforme a lista

```
cis = [
    (1.5, 0), (1.0, 1.2), (2.0, -0.5), (-1.5, 1.0),
    (-2.2, -1.5), (0.5, 2.5), (2.8, 0.8), (-0.8, -2.2),
    (1.8, -2.5), (0.2, 0.1)]
```

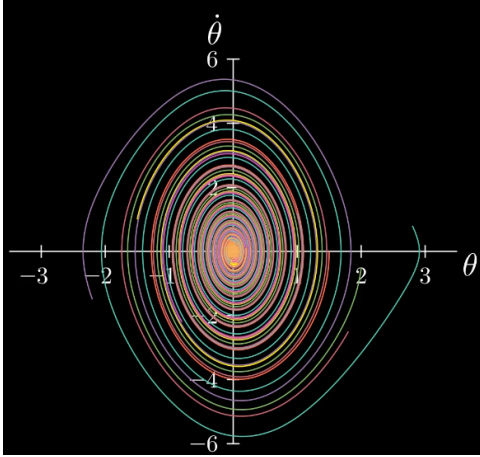


Figura 5: simulação de *phase flow* com conjunto de 10 condições iniciais diferentes.

VII. TESTANDO PARA DIFERENTES REGIMES DE AMORTECIMENTO

A. Regime Super-crítico

Para

$$\ddot{\theta} + 2\gamma\dot{\theta} + \omega_0^2\theta = 0, \quad \gamma > \omega_0, \quad (3)$$

a solução geral é

$$\theta(t) = C_1 e^{-(\gamma+\alpha)t} + C_2 e^{-(\gamma-\alpha)t}, \quad \alpha = \sqrt{\gamma^2 - \omega_0^2}. \quad (4)$$

As condições iniciais são

$$\theta(0) = \theta_0, \quad \dot{\theta}(0) = 0. \quad (5)$$

Resolvendo, obtemos o sistema

$$\begin{cases} C_1 + C_2 = \theta_0, \\ -(\gamma + \alpha)C_1 - (\gamma - \alpha)C_2 = 0. \end{cases} \quad (6)$$

Daí segue que

$$C_1 = \frac{\gamma - \alpha}{2\alpha} \theta_0, \quad C_2 = \frac{\gamma + \alpha}{2\alpha} \theta_0. \quad (7)$$

No código, ficamos com

```
theta0 = 0.1
gamma = 2.0
alpha = np.sqrt(gamma**2 - omega0**2)
C1 = (gamma - alpha)/(2*alpha) * theta0
C2 = (gamma + alpha)/(2*alpha) * theta0
theta_super = (C1 * np.exp(-(gamma + alpha)*t) +
               C2 * np.exp(-(gamma - alpha)*t))
```

B. Regime Crítico

$$\ddot{\theta} + 2\gamma\dot{\theta} + \omega_0^2\theta = 0, \quad \gamma = \omega_0 \quad (8)$$

A solução geral é:

$$\theta(t) = (C_1 + C_2 t) e^{-\gamma t} \quad (9)$$

Aplicando as condições iniciais desejadas, elas implicam em:

$$\theta(0) = \theta_0, \quad \dot{\theta}(0) = 0. \quad (10)$$

e

$$C_1 = \theta_0, \quad C_2 = \gamma\theta_0 \quad (11)$$

No código fica:

```
gamma_crit = omega0
x_crit = A * np.exp(-gamma_crit*t) * (1 + gamma_crit*t)
```

C. Regime Subcrítico

Por fim, para pequenas amplitudes:

$$\ddot{\theta} + 2\gamma\dot{\theta} + \omega_0^2\theta = 0 \quad \text{com} \quad 0 < \gamma < \omega_0 \quad (12)$$

No subcrítico

$$\gamma^2 - \omega_0^2 < 0 \quad (13)$$

Defina

$$\omega_d = \sqrt{\omega_0^2 - \gamma^2} \quad (14)$$

E então

$$r_{\pm} = -\gamma \pm i\omega_d \quad (15)$$

Cuja solução real é:

$$\theta(t) = e^{-\gamma t} (C_1 \cos(\omega_d t) + C_2 \sin(\omega_d t)) \quad (16)$$

Que possui forma equivalente a

$$\theta(t) = A e^{-\gamma t} \cos(\omega_d t + \phi) \quad (17)$$

No código fica:

```
gamma_sub = 0.05
omega_d = np.sqrt(omega0**2 - gamma_sub**2)
x_sub = A * np.exp(-gamma_sub*t) * np.cos(omega_d * t)
```

D. Simulação com dados sintéticos e visualização do Espaço de Fase no *Manim*

Com o gráfico em mãos, é possível também extrair e organizar os dados correspondentes a cada regime de amortecimento em arquivos `.npz` distintos. Esses arquivos serão utilizados posteriormente em um código independente, desenvolvido em Python com a biblioteca *open-source* **Manim**, com o objetivo de simular e visualizar o espaço de fase do sistema.

A seguir, apresenta-se o código responsável pelo carregamento de um *dataset* e pelo salvamento separado dos dados referentes aos regimes subcrítico (ou subamortecido), crítico e supercrítico.

```

# ===== SALVANDO DATASETS =====

# Subcrítico
np.savez(
    "pendulo_subcritico.npz",
    t=t,
    x_clean=x_sub,
    x_noisy=x_noise0,
    A=A,
    omega0=omega0,
    gamma=gamma_sub
)

# Crítico
np.savez(
    "pendulo_critico.npz",
    t=t,
    x_clean=x_crit,
    x_noisy=x_noise2,
    A=A,
    omega0=omega0,
    gamma=gamma_crit
)

# Supercrítico
np.savez(
    "pendulo_supercritico.npz",
    t=t,
    x_clean=x_super,
    x_noisy=x_noise1,
    theta0=theta0,
    omega0=omega0,
    gamma=gamma_super
)

# ===== CARREGANDO DATASETS =====
data = np.load("pendulo_subcritico.npz")

t = data["t"]
x = data["x_noisy"]
gamma = data["gamma"]

```

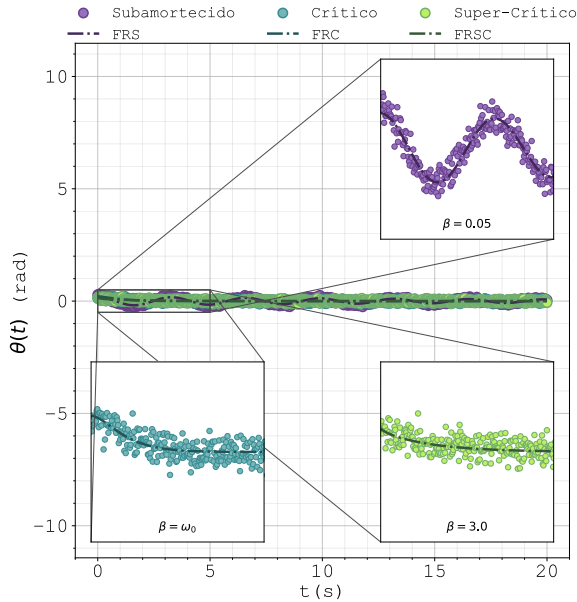
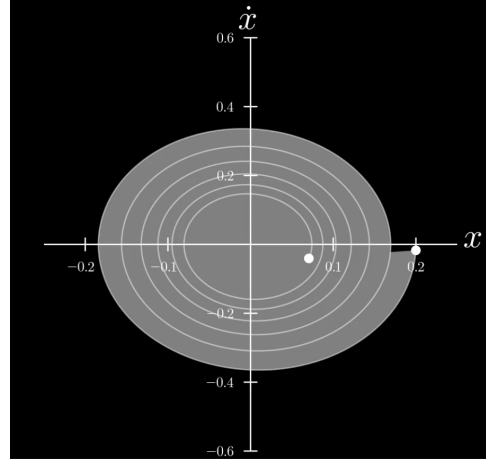


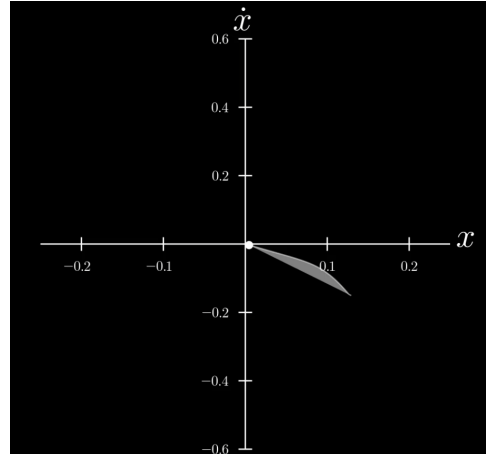
Figura 6: Gráfico mostrando os diferentes regimes de amortecimento com dados sintéticos gerados com ruído normal de 5% pseudoaleatório com $seed = 0$ e começando com amplitude inicial $A_0 = 0.2\text{rad}$.

Nota: Tanto no regime de amortecimento crítico quanto supercrítico, a ausência de oscilações faz com

que a trajetória no espaço de fase não apresente uma estrutura geométrica bem definida. Além disso, a reconstrução da velocidade a partir da derivada discreta de $x(t)$ no regime crítico torna-se numericamente instável para tempos longos, o que dificulta a obtenção de uma trajetória contínua e confiável no espaço de fase.



(a) Espaço de fase para regime de amortecimento Subcrítico ($\beta = 0.05$)



(b) Espaço de fase para regime de amortecimento Supercrítico ($\beta = 3.0$)

VIII. PENSANDO NO MOVIMENTO EM TERMOS DE ENERGIA

Sabemos que a solução geral do oscilador amortecido no regime subcrítico é dada por

$$x(t) = Ae^{-\gamma t} \cos(\omega_d t + \phi), \quad \omega_d = \sqrt{\omega_0^2 - \gamma^2}. \quad (18)$$

Derivando em relação ao tempo, obtemos a velocidade

$$\dot{x}(t) = Ae^{-\gamma t} [-\gamma \cos(\omega_d t + \phi) - \omega_d \sin(\omega_d t + \phi)]. \quad (19)$$

A. Energia cinética e potencial

A energia cinética é dada por

$$T(t) = \frac{1}{2}m\dot{x}^2(t), \quad (20)$$

ou explicitamente,

$$T(t) = \frac{1}{2}mA^2e^{-2\gamma t} [\gamma \cos(\omega_d t + \phi) + \omega_d \sin(\omega_d t + \phi)]^2. \quad (21)$$

Expandindo o quadrado, obtemos

$$T(t) = \frac{1}{2}mA^2e^{-2\gamma t} \left[\gamma^2 \cos^2(\omega_d t + \phi) + \omega_d^2 \sin^2(\omega_d t + \phi) + 2\gamma\omega_d \sin(\omega_d t + \phi) \cos(\omega_d t + \phi) \right]. \quad (22)$$

A energia potencial é

$$U(t) = \frac{1}{2}kx^2(t) = \frac{1}{2}m\omega_0^2 A^2 e^{-2\gamma t} \cos^2(\omega_d t + \phi). \quad (23)$$

B. Energia mecânica do sistema

A energia mecânica total do sistema é dada por

$$E(t) = T(t) + U(t). \quad (24)$$

Como todos os termos de $T(t)$ e $U(t)$ possuem o fator comum $e^{-2\gamma t}$, conclui-se que

$$\boxed{E(t) \propto e^{-2\gamma t}}. \quad (25)$$

Ou seja, enquanto a amplitude do movimento decai como $e^{-\gamma t}$, a energia mecânica do oscilador decai como $e^{-2\gamma t}$.

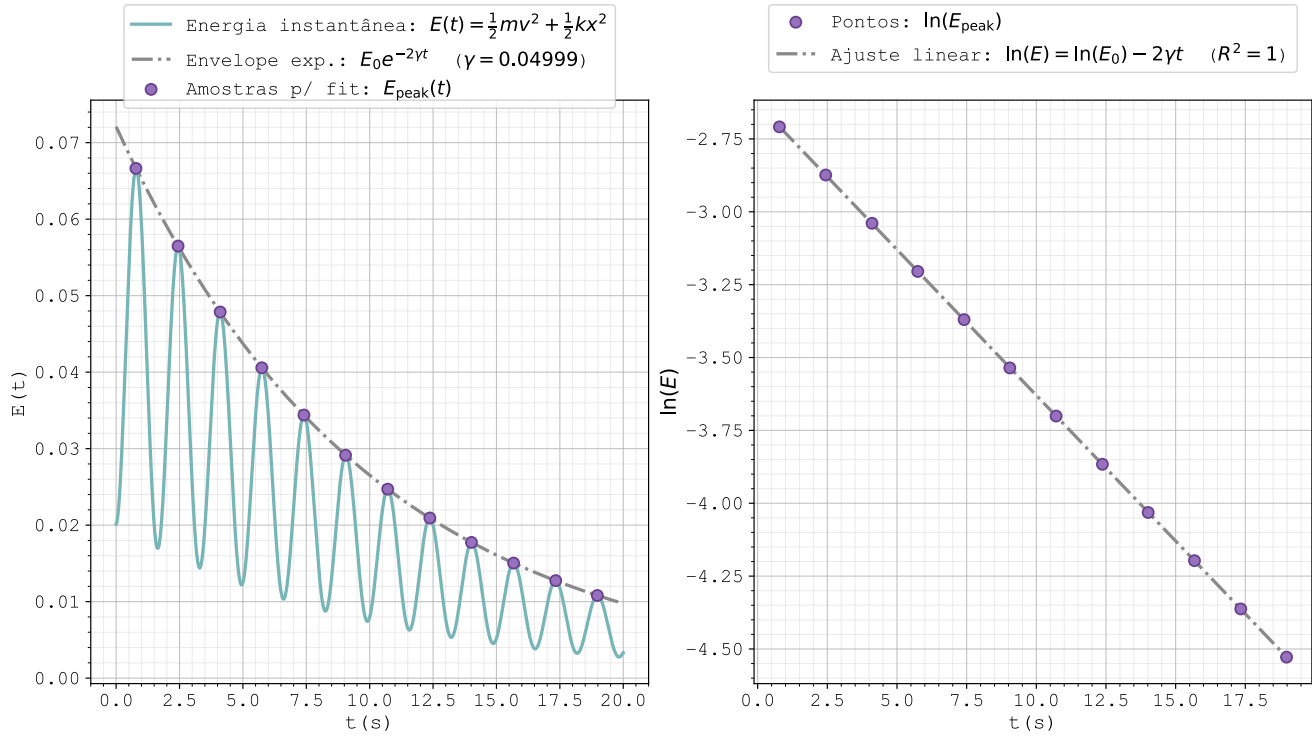


Figura 8: Evolução temporal da energia mecânica do pêndulo amortecido no regime subcrítico. (a) Energia instantânea $E(t) = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}kx^2$, calculada a partir de $x_{\text{clean}}(t)$ e $\dot{x} = dx/dt$, juntamente com o envelope exponencial ajustado $E_0 e^{-2\gamma t}$. Os pontos representam os máximos locais $E_{\text{peak}}(t)$ utilizados no ajuste. (b) Linearização de $\ln(E_{\text{peak}})$ em função do tempo e regressão linear $\ln(E) = \ln(E_0) - 2\gamma t$, a partir da qual se estima γ (com R^2 indicado no gráfico). Percebe-se que o valor de γ extraído da análise da linearização de $\ln(E_{\text{peak}})$ condiz com o input dado inicialmente no dataset dos dados sintéticos para o regime subcrítico, o que mostra coerência com o modelo analisado. Os dados de posição e velocidade foram extraídos do dataset salvo em `pendulosubcritico.npz`