

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

Hézio Delfino Lopes de Alcântara 231028343*
 IF-UnB
 (Data: 9 de dezembro de 2025)

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 $\approx 0.6 T/\Delta t$).

```
# ----- 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

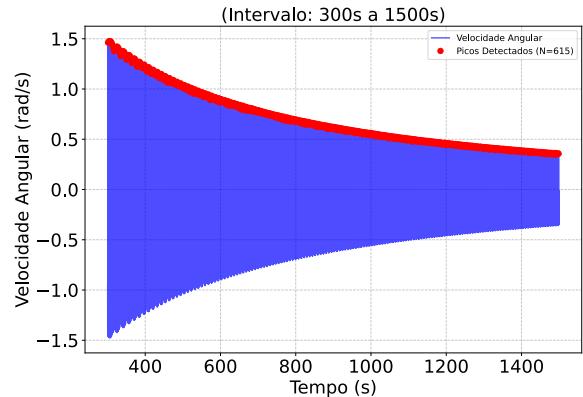
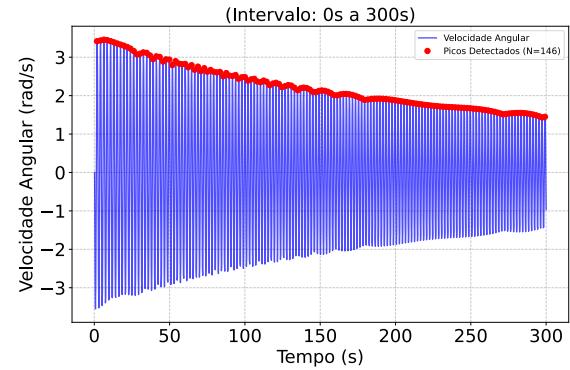
# ----- converte em distância mínima entre picos -----
distance_samples = int(0.6 * period_est / dt)

# ----- 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

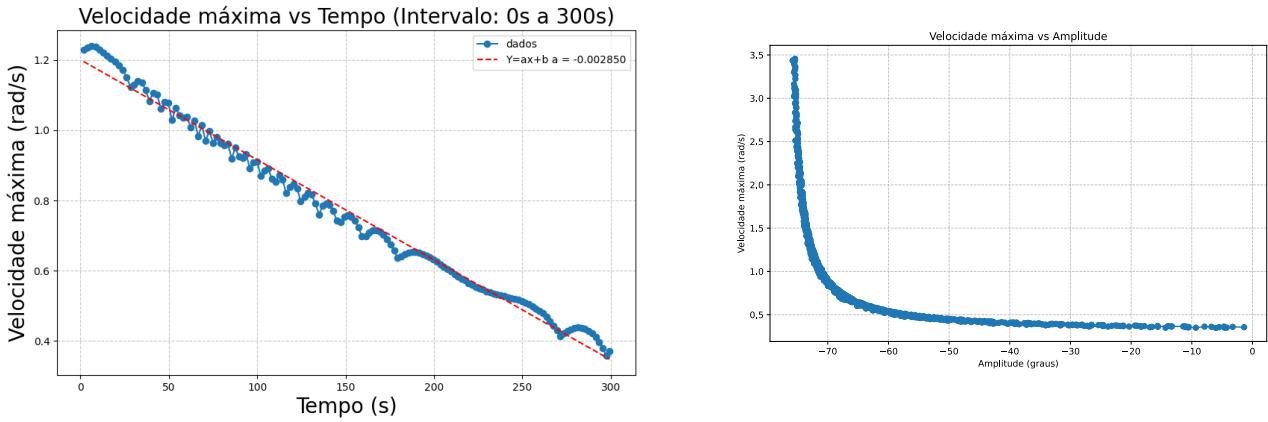


Figura 1: regime de amortecimento 'forte'

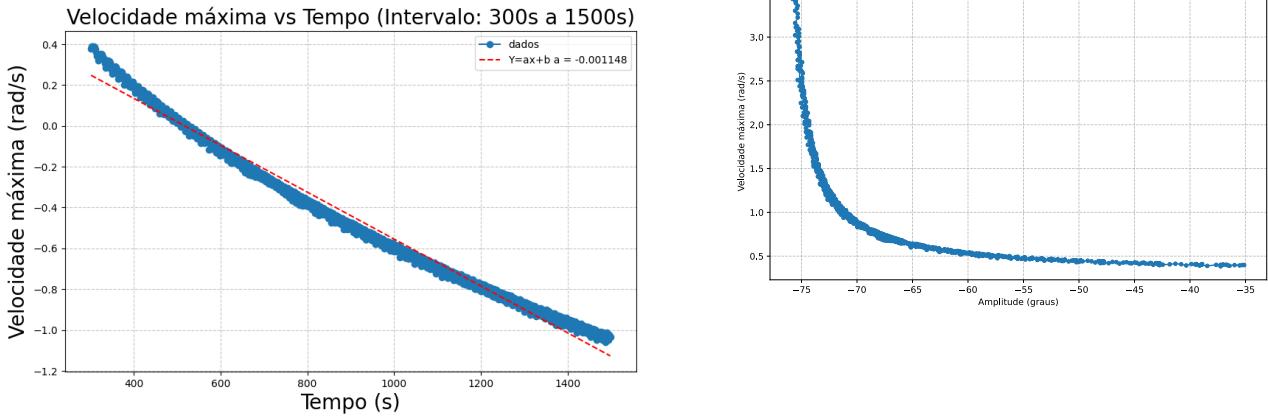


Figura 2: regime de amortecimento 'fraco'

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

V. CORREÇÃO DOS GRÁFOS DE VELOCIDADE MÁXIMA VS AMPLITUDE

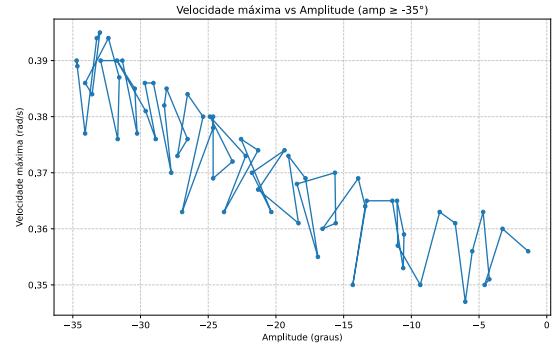
Ajustei o sentido de leitura do vetor da coluna de velocidade angular, medido pelo Pasco. O erro era um simples descuido na hora de plotar os dados no gráfico, o que anteriormente resultava numa interpretação física inversa à esperada.

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

VI. REUNIAO 9DEZ

A. Espaço de fase no manim: estudo incial

Pelo fato de não termos conseguido fazer o ajuste ainda. Achei coerente usar um modelo de pêndulo



não-linear com amortecimento viscéo para criar um cenário hipotético que reproduza qualitativamente as features observadas (grandes amplitudes, decaimento de energia). Gerei 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 permite 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.

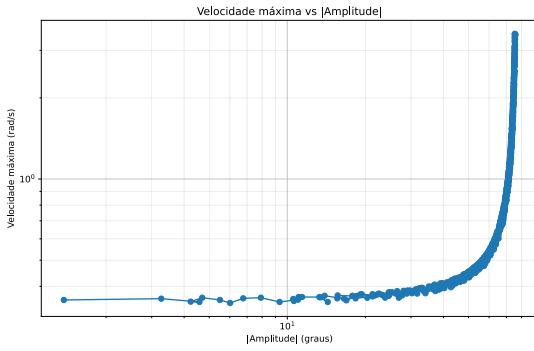


Figura 3: O comportamento não é linear: em pequenas amplitudes, a velocidade máxima varia pouco, mantendo a energia praticamente constante. À medida que a amplitude aumenta, a velocidade máxima cresce muito mais rapidamente do que prevê a aproximação linear. Pode-se concluir que, para grandes amplitudes,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

```
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)
```

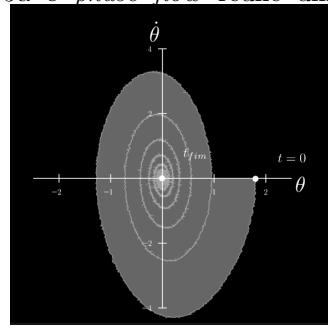
e também a resolução numérica da EDO:

```
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)
```

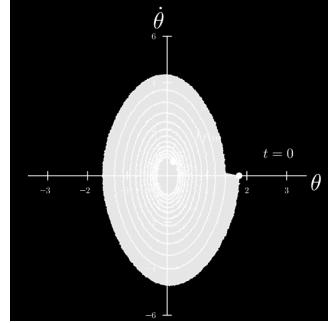
B. Phase flow

De modo geral, ao escolhermos um ponto no espaço de fase observamos a órbita correspondente — sua evolução

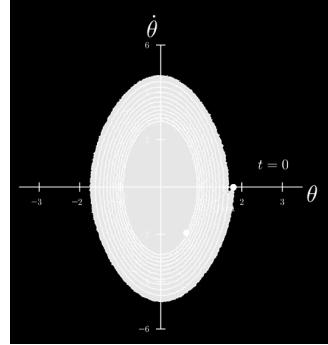
no tempo. Já o *phase flow* reúne um conjunto de



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



(b) Amortecimento médio ($b = 0.15$)



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

Figura 4: Resultados das simulações.

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)]
```

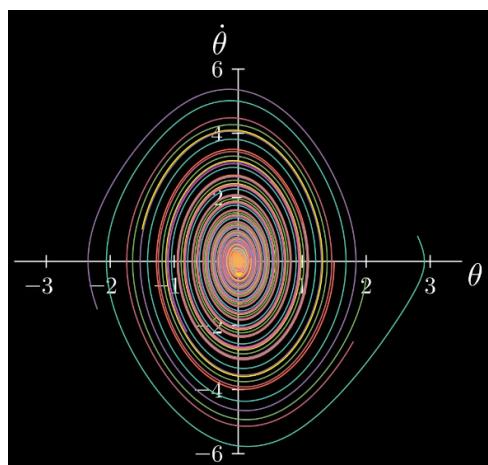


Figura 5: simulação de *phase flow* com conjunto de 10 condições iniciais diferentes.