

# Relatório do Trabalho: Simulação do Pêndulo Simples

Thiago Camerato

Tomás Lenzi (2220711)

Dezembro de 2025

---

## Introdução

Este trabalho aborda o estudo do movimento do pêndulo simples, um problema clássico da física que demonstra conceitos importantes sobre oscilações e equações diferenciais. O objetivo principal do projeto é utilizar o método numérico de Runge-Kutta de 4<sup>a</sup> ordem (RK4) para resolver a equação diferencial não linear que descreve o movimento do pêndulo.

A solução numérica obtida é então comparada com a solução analítica simplificada, que é válida apenas para pequenas amplitudes de oscilação. Para otimizar a precisão e a eficiência computacional, o método RK4 foi implementado tanto com passo constante quanto com uma estratégia de passo adaptativo, que ajusta o tamanho do passo para manter o erro local sob um limite pré-definido de  $\varepsilon = 10^{-5}$ .

O relatório apresenta o desenvolvimento do modelo matemático e dos métodos numéricos empregados, seguido por uma análise detalhada dos resultados computacionais, comparando a precisão, o custo computacional e a validade das diferentes abordagens para vários ângulos iniciais.

---

## Desenvolvimento

O movimento de um pêndulo simples, de comprimento  $L$  sob a ação da gravidade  $g$ , é descrito pela seguinte equação diferencial ordinária (EDO) de segunda ordem:

$$\frac{d^2\theta}{dt^2} + \frac{g}{L} \sin(\theta) = 0$$

Onde  $\theta$  é o ângulo em relação à vertical. Esta é uma EDO não linear devido ao termo  $\sin(\theta)$ . Para resolvê-la numericamente, primeiro a convertemos em um sistema de duas EDOs de primeira ordem, definindo  $\omega = d\theta/dt$  (velocidade angular):

1.  $\frac{d\theta}{dt} = \omega$
2.  $\frac{d\omega}{dt} = -\frac{g}{L} \sin(\theta)$

Este sistema foi resolvido utilizando o **método de Runge-Kutta de 4<sup>a</sup> Ordem**. A implementação foi feita em linguagem C, contemplando duas estratégias principais de integração:

- **Passo Fixo:** O método RK4 foi aplicado com três passos constantes diferentes ( $h = 0.01$  s,  $h = 0.001$  s e  $h = 0.0001$  s) para avaliar o impacto do tamanho do passo na precisão e no custo computacional.
- **Passo Adaptativo:** Foi implementada uma estratégia de passo adaptativo para otimizar a simulação. A cada passo, o erro local é estimado comparando o resultado de um passo  $h$  com dois passos  $h/2$ . Se o erro for maior que a tolerância  $\varepsilon = 10^{-5}$ , o passo é reduzido; se for muito menor, o passo é aumentado para o passo seguinte. Esta abordagem visa garantir a precisão desejada com o menor número de cálculos possível.

Para fins de comparação, também foi utilizada a **solução analítica aproximada**, derivada da linearização  $\sin(\theta) \approx \theta$  para pequenos ângulos. Esta aproximação transforma a EDO em um oscilador harmônico simples com solução:

$$\theta(t) = \theta_0 \cos\left(\sqrt{\frac{g}{L}} t\right)$$

cujo período é constante e dado por:

$$T \approx 2\pi \sqrt{\frac{L}{g}}.$$

O período  $T$  do ciclo foi calculado numericamente monitorando a mudança de sinal da velocidade angular  $\omega$ , utilizando interpolação linear para estimar o instante exato da inversão e, consequentemente, o meio período. Para maior precisão, o tempo de 10 períodos foi medido e então dividido por 10.

### Trechos em Pseudocódigo usados no desenvolvimento

- Conversão para sistema de 1ª ordem:

```
// estado = (theta, omega)
dtheta/dt = omega
domega/dt = - (g / L) * sin(theta)
```

- Passo RK4 (um passo):

```
k1 = f(t, y)
k2 = f(t + h/2, y + k1*h/2)
k3 = f(t + h/2, y + k2*h/2)
k4 = f(t + h, y + k3*h)
y_next = y + (h/6) * (k1 + 2*k2 + 2*k3 + k4)
```

- Integração com passo fixo:

```
t = t0
for i = 1..N:
    y = RK4_step(t, y, h)
    t = t + h
```

- Passo adaptativo (estimativa por duplicação de passo):

```
// tenta 1 passo h e 2 passos h/2; calcula erro e ajusta h
while t < tf:
    y1 = RK4_step(t, y, h)
    y2 = RK4_step(t, y, h/2)
    y2 = RK4_step(t + h/2, y2, h/2)
    error = max(|y2 - y1|)
    if error < eps:
        accept y = y2; t += h
        if error < eps/10: h *= 1.5
    else:
        h *= 0.5
```

- Cálculo do período (detecção de inversões em ):

```

inversions = []
for i = 1..n:
    if omega[i-1] * omega[i] <= 0:
        t_inv = interp_linear(time[i-1], time[i], omega[i-1], omega[i])
        append(inversions, t_inv)
// usando várias inversões: T = 2*(t_last - t_first)/(count - 1)
T = 2 * (inversions[last] - inversions[first]) / (count(inversions) - 1)

```

## Resultados e Análise

Esta seção apresenta os resultados obtidos através dos experimentos computacionais, focando na comparação entre a solução numérica e a analítica, na eficiência dos métodos e na dependência do período com a amplitude.

### Validade da Aproximação Linearizada

Os experimentos confirmam que a solução analítica simplificada só é precisa para ângulos iniciais pequenos. O erro da aproximação cresce rapidamente com o aumento do ângulo inicial  $\theta_0$ .

As Figuras 1, 2 e 3 abaixo ilustram a comparação entre a solução numérica (RK4) e a solução analítica linearizada para diferentes ângulos iniciais:

- **Figura 1:** Mostra o comportamento do pêndulo para  $\theta_0 = 10^\circ$ . As duas soluções praticamente coincidem, evidenciando a validade da aproximação linearizada para pequenos ângulos.
- **Figura 2:** Para  $\theta_0 = 45^\circ$ , observa-se uma diferença crescente entre as soluções, indicando o início da limitação da aproximação analítica.
- **Figura 3:** Com  $\theta_0 = 90^\circ$ , a divergência é clara, mostrando que a solução analítica não representa bem o sistema físico para grandes amplitudes.

Comparação de Soluções -  $\theta_0 = 10^\circ$

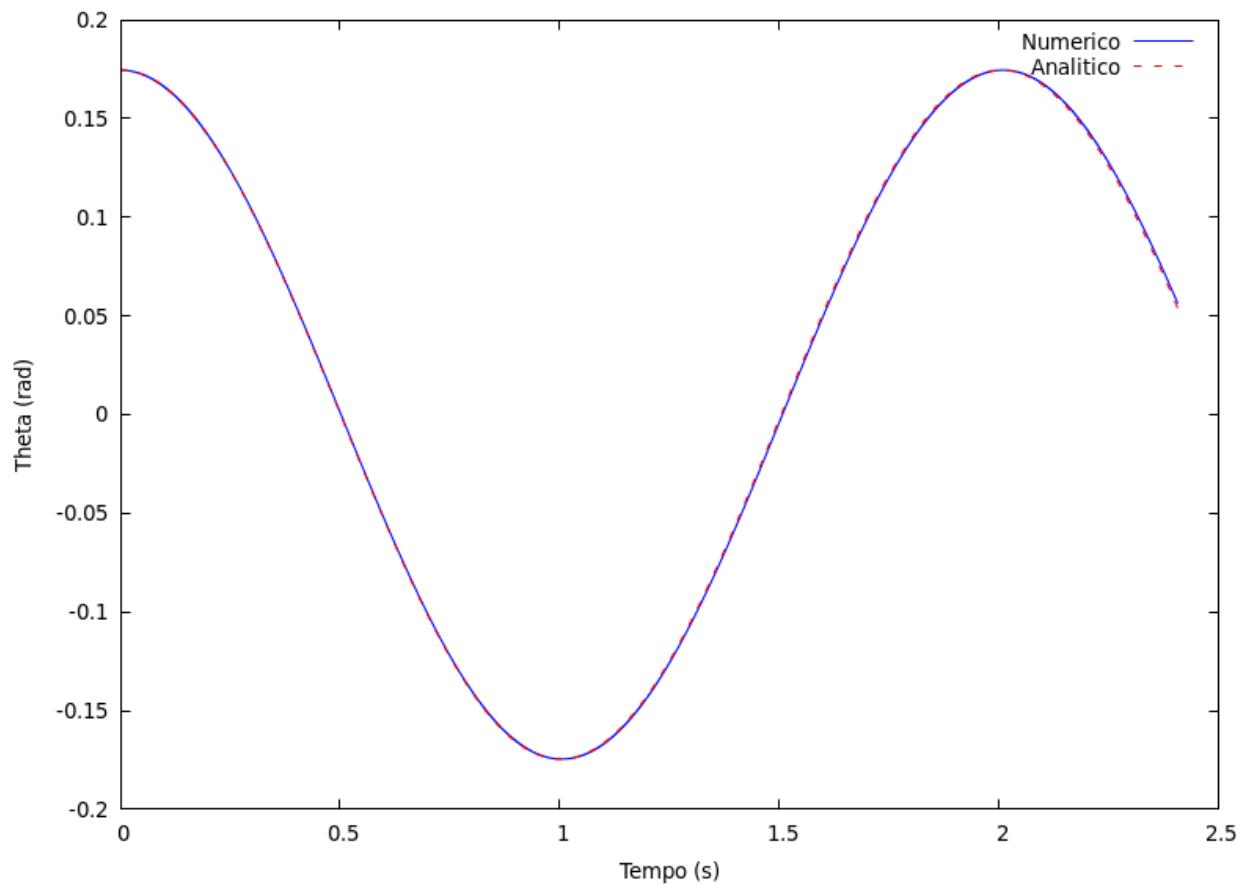


Figura 1: Comparação entre a solução numérica (azul) e a analítica (vermelho tracejado) para um ângulo inicial de  $10^\circ$ .

Comparação de Soluções -  $\theta_0 = 45^\circ$

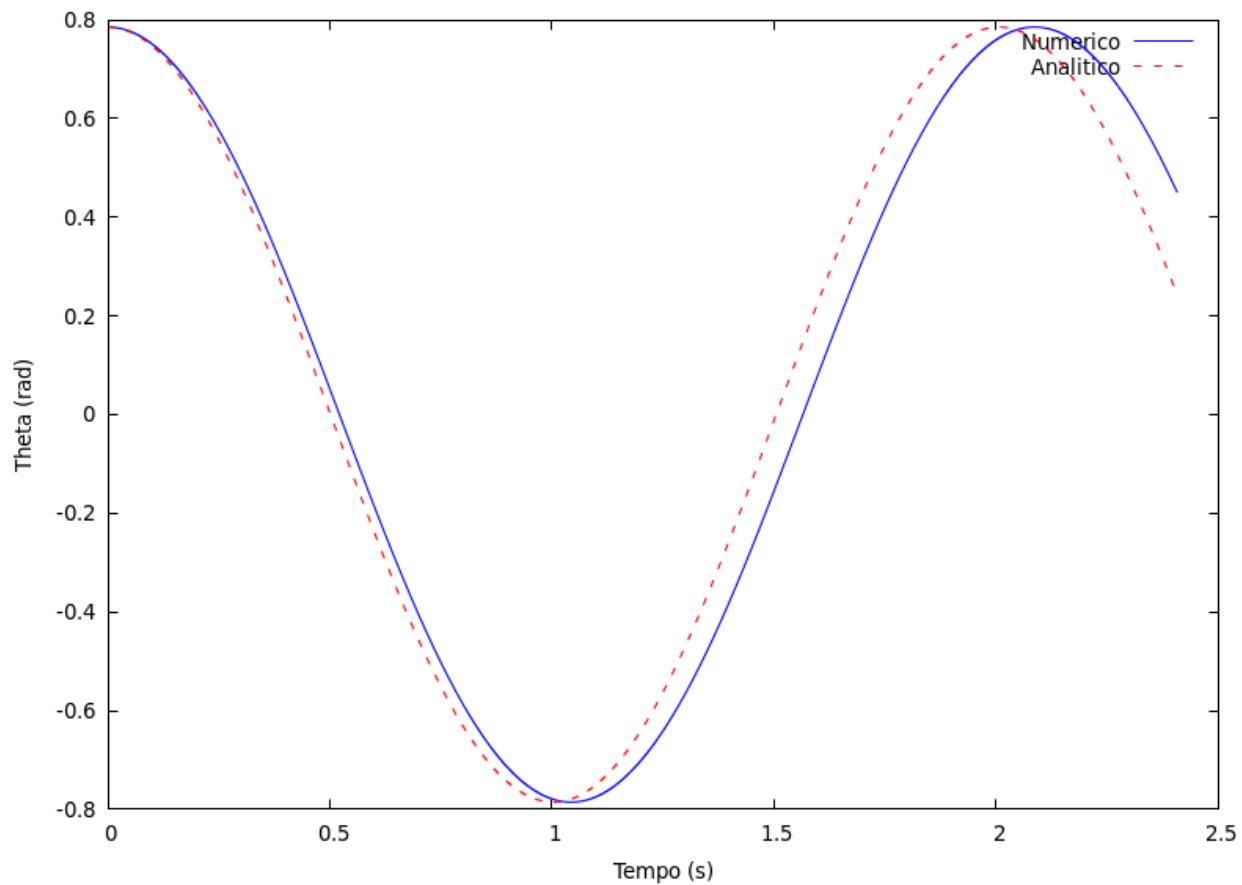
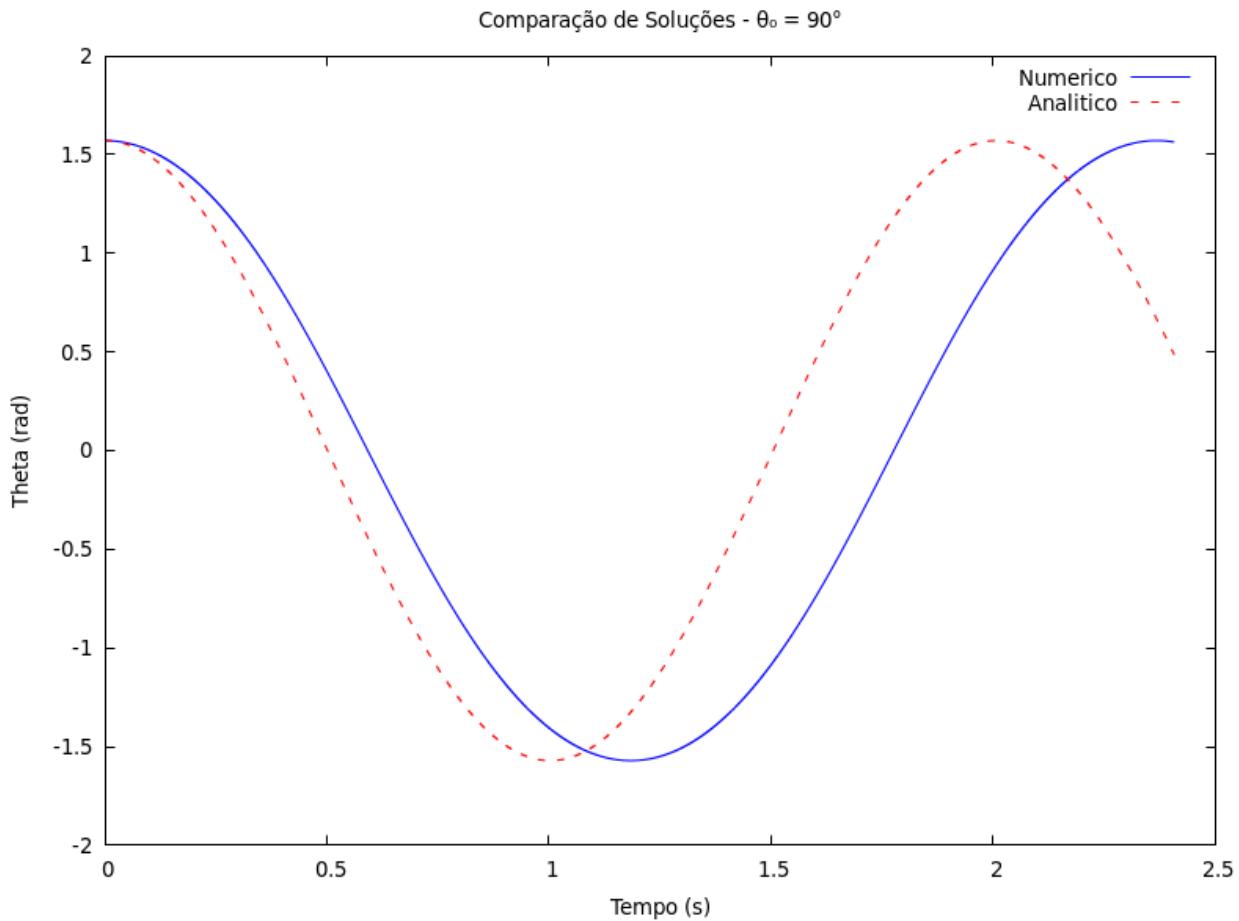


Figura 2: Comparação para  $45^\circ$ .



*Figura 3: Comparação para  $90^\circ$ .*

Os gráficos de diferença, calculados como

$$|\theta * \text{num} - \theta * \text{anal}|,$$

quantificam o erro ao longo do tempo.

As Figuras 4, 5 e 6 apresentam a diferença absoluta entre as soluções numérica e analítica ao longo do tempo para os mesmos ângulos iniciais:

- **Figura 4:** Para  $\theta_0 = 10^\circ$ , a diferença permanece próxima de zero, confirmando a precisão da solução analítica nesse regime.
- **Figura 5:** Em  $\theta_0 = 45^\circ$ , a diferença cresce com o tempo, evidenciando o erro acumulado da aproximação linearizada.
- **Figura 6:** Para  $\theta_0 = 90^\circ$ , o erro é significativo durante todo o ciclo, mostrando que a solução analítica não é adequada para grandes amplitudes.

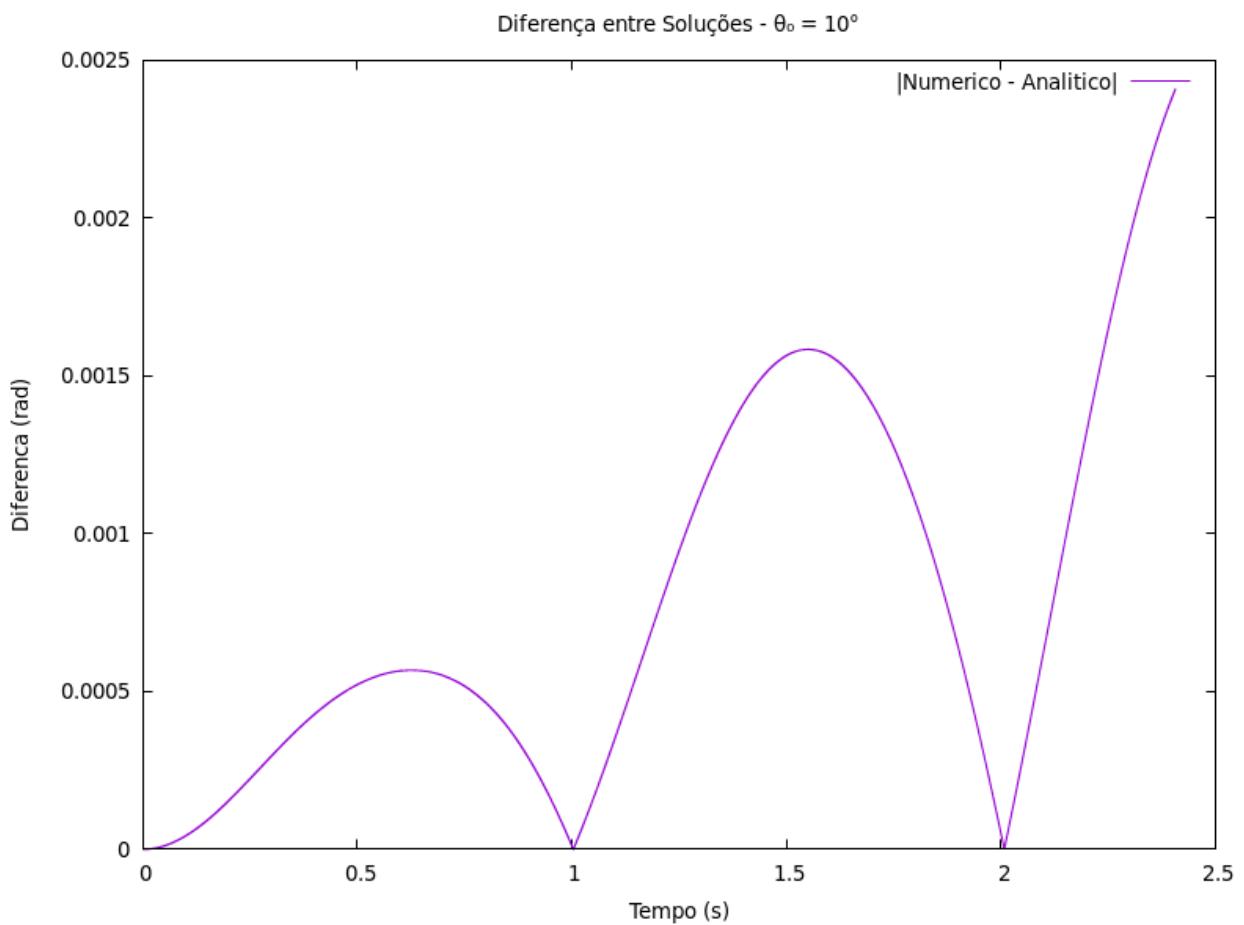


Figura 4: Diferença entre solução numérica e analítica para  $\theta_0 = 10^\circ$ .

Diferença entre Soluções -  $\theta_0 = 45^\circ$

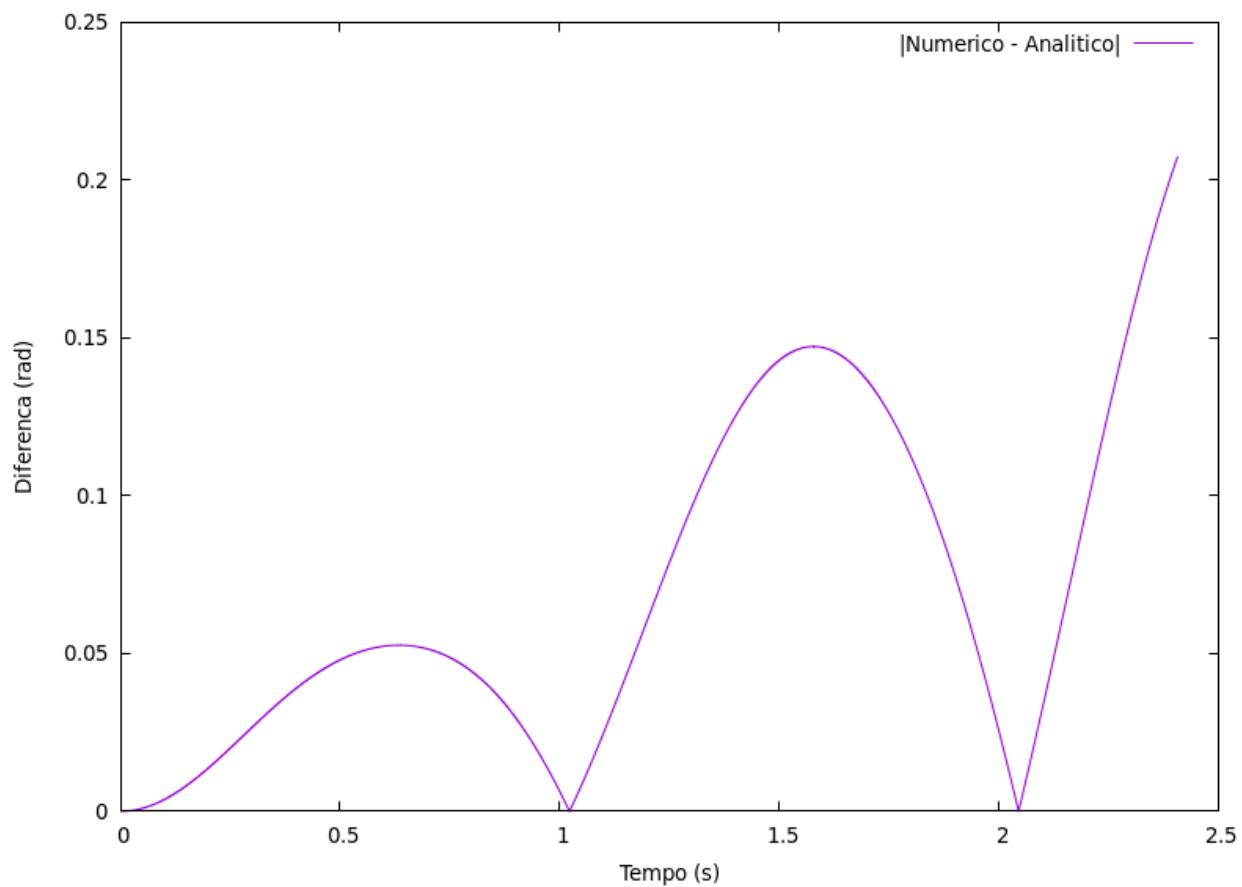


Figura 5: Diferença para  $\theta_0 = 45^\circ$ .

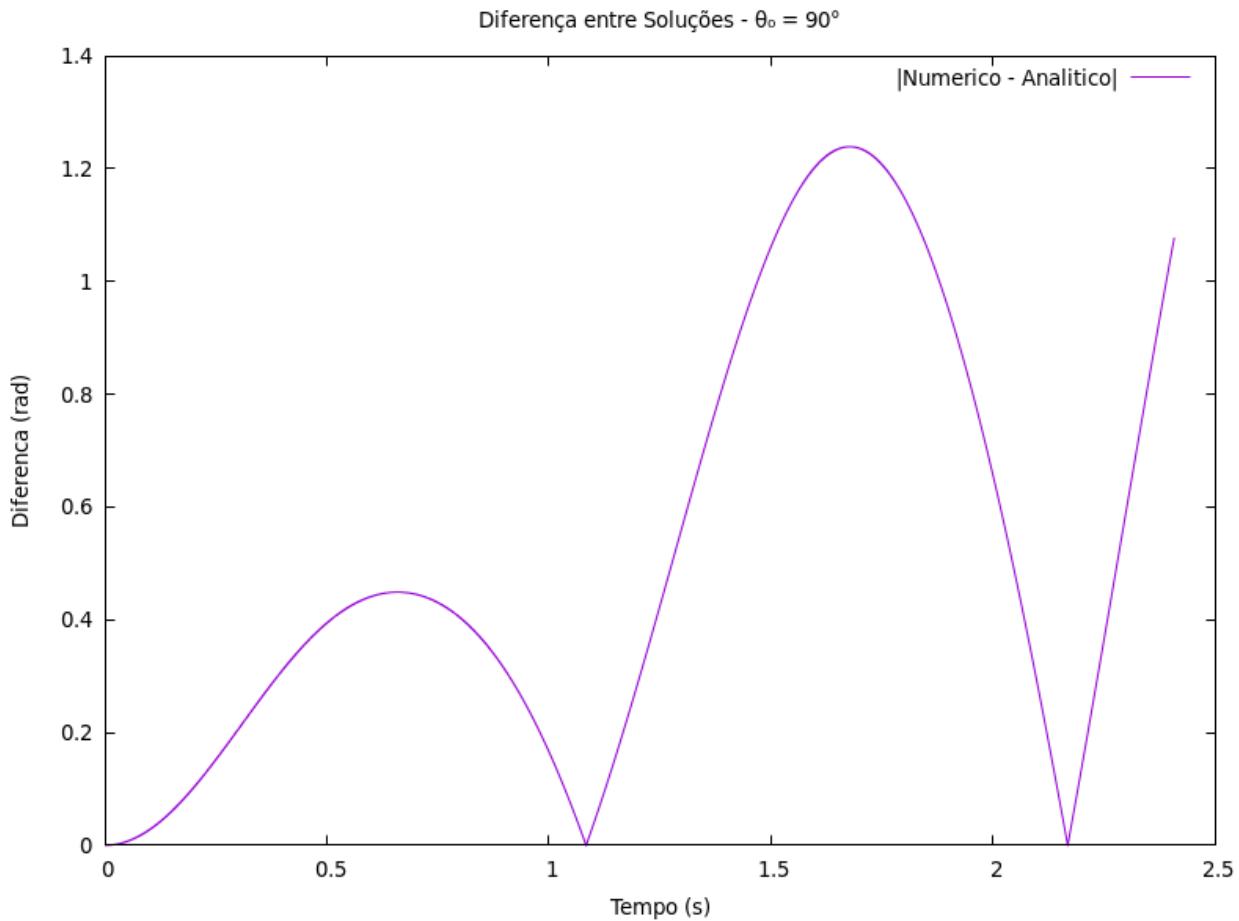


Figura 6: Diferença para  $\theta_0 = 90^\circ$ .

### Eficiência dos Métodos e Análise do Período

A análise do período e do número de passos revela a superioridade do método adaptativo.

Ângulo Inicial ( $\theta_0$ )	Período (Adaptativo)	Passos ( $h = 0,0001$ )	Passos (Adaptativo)	Redução
45°	2.086253 s	300910	473	~636x
90°	2.367843 s	300910	501	~600x
120°	2.754091 s	300910	548	~550x

Além disso, ao contrário do modelo linearizado que prevê período constante ( $T \approx 2,006$  s), os resultados numéricos mostram que o **período aumenta com a amplitude inicial**. Exemplos:

- $\theta_0 = 5^\circ$ :  $T \approx 2,007$  s
- $\theta_0 = 150^\circ$ :  $T \approx 3,535$  s

### Análise de Tempo de Execução

Para  $\theta_0 = 90^\circ$ , a simulação de 10 períodos ( 24 s reais) levou apenas 0,11 ms — mais de **20.000× mais rápido** que o tempo físico.

## Conclusão

Este trabalho realizou com sucesso a simulação do pêndulo simples utilizando RK4 com passo fixo e adaptativo. A abordagem adaptativa foi significativamente mais eficiente. Confirmou-se também que a aproximação linearizada só é válida para pequenas amplitudes e que o período real aumenta conforme  $\theta_0$  cresce.

---

## Referências

- [1] Press, W. H., et al. *Numerical Recipes in C*.
- [2] Butcher, J. C. *Numerical Methods for ODEs*.
- [3] INF1608 — Enunciado do trabalho.