

FÍSICA – PRÁTICAS EXTENSIVAS

Ricardo Dalke Meucci

Grupo: Júlia Almeida Leite, Maria Eduarda Coelho e Samuel Sampaio

LANÇAMENTO DE UM PROJÉTIL

O código simula o lançamento de um projétil considerando resistência do ar e exibe os resultados com uma animação interativa em um gráfico. Além disso, ele possui uma interface gráfica criada com Tkinter para entrada dos parâmetros.

CONVERSÃO DO ÂNGULO

No início da função `simular_lancamento`, o ângulo fornecido em graus é convertido para radianos:

```
angulo_0 = math.radians(angulo_0)
```

Isso é necessário porque as funções trigonométricas em Python (`math.sin` e `math.cos`) trabalham com ângulos em radianos. A fórmula usada é:

$$radianos = graus \cdot \frac{\pi}{180}$$

CÁLCULO DAS COMPONENTES INICIAIS DA VELOCIDADE

Após a conversão do ângulo, calcula-se a velocidade inicial nas direções horizontal (v_{0x}) e vertical (v_{0y}):

```
v_zero_x = v_zero * math.cos(angulo_0)
v_zero_y = v_zero * math.sin(angulo_0)
```

Fórmulas:

$$v_{0x} = v_0 \cos(\theta_0) \qquad v_{0y} = v_0 \sin(\theta_0)$$

Esses valores inicializam as velocidades nas direções x e y .

FORÇA DE ARRASTO

Durante a simulação, a força de arrasto (F_a) é recalculada a cada instante:

```
Fa = 0.5 * c * A * p * v**2
```

Fórmula:

$$F_a = \frac{1}{2} c A \rho v^2$$

Aqui:

- $c \rightarrow$ Coeficiente de arrasto.
- $A \rightarrow$ Área frontal do projétil.
- $\rho \rightarrow$ Densidade do ar.
- $v \rightarrow$ Velocidade total do projétil no instante.

A força depende do quadrado da velocidade, o que significa que aumenta muito conforme o projétil vai mais rápido.

VELOCIDADE TOTAL

A velocidade total v combina as componentes v_x e v_y :

```
v = math.sqrt(v_x[-1]**2 + v_y[-1]**2)
```

Fórmula:

$$v = \sqrt{v_x^2 + v_y^2}$$

Essa velocidade é usada no cálculo da força de arrasto e para determinar a direção em que essa força age.

ACELERAÇÕES

As acelerações são calculadas considerando a força de arrasto. Para a direção horizontal (a_x):

```
aceleracao_x = -Fa * v_x[-1] / (m * v)
```

Fórmula:

$$a_x = -\frac{F_a v_x}{mv}$$

E para a direção vertical (a_y):

```
aceleracao_y = -ace - Fa * v_y[-1] / (m * v)
```

Fórmula:

$$a_y = -g - \frac{F_a v_y}{mv}$$

Aqui:

- $-g \rightarrow$ Aceleração da gravidade atua para baixo.
- $-\frac{F_a v_y}{mv} \rightarrow$ Componente vertical da força de arrasto.

Essas acelerações são recalculadas a cada intervalo de tempo (Δt / Delta t).

ATUALIZAÇÃO DAS VELOCIDADES

Com as acelerações calculadas, as velocidades são atualizadas para o próximo instante usando:

```
v_x.append(v_x[-1] + aceleracao_x * dt)
v_y.append(v_y[-1] + aceleracao_y * dt)
```

Fórmulas:

$$v_x(t + \Delta t) = v_x(t) + a_x \cdot \Delta t$$

$$v_y(t + \Delta t) = v_y(t) + a_y \cdot \Delta t$$

Isso é uma aplicação do método de Euler, uma técnica numérica para integrar equações diferenciais.

ATUALIZAÇÃO DAS POSIÇÕES

As novas posições do projétil (x e y) são calculadas com as velocidades atualizadas:

```
pos_x.append(pos_x[-1] + v_x[-1] * dt)
pos_y.append(pos_y[-1] + v_y[-1] * dt)
```

Fórmulas:

$$x(t + \Delta t) = x(t) + vx(t + \Delta t) \cdot \Delta t$$

$$y(t + \Delta t) = y(t) + vy(t + \Delta t) \cdot \Delta t$$

Essas posições são usadas para atualizar o gráfico da trajetória.

VERIFICAÇÃO DE CONDIÇÃO DE PARADA

Se o projétil atinge o solo ($y < 0$), a simulação para:

```
if pos_y[-1] < 0:
    return line, # Retorna sem atualizar
```

CÁLCULO DE VALORES MÁXIMOS

Os valores máximos (altura, distância, tempo total) são encontrados durante a simulação:

```
print(f"Distância máxima atingida: {max(pos_x):.2f} m")
print(f"Altura máxima atingida: {max(pos_y):.2f} m")
print(f"Tempo total da trajetória: {max(tempo):.2f} s")
```

- Distância máxima (x_{max}): O maior valor de `pos_x`.
- Altura máxima (y_{max}): O maior valor de `pos_y`.
- Tempo total: O maior valor do vetor `tempo`.

FUNÇÃO DE SIMULAÇÃO E ANIMAÇÃO

```
def simular_lancamento(v_zero, angulo_0, altura_inicial, ace,
r, A, p, c, m, dt):
```

- **Inicialização das variáveis**
Vetores são criados para armazenar posições (`pos_x`, `pos_y`), velocidades (`v_x`, `v_y`) e tempos (`tempo`).
- **Configuração do gráfico**
Um gráfico é criado usando `matplotlib`, com:
 - Imagem de fundo carregada via `mpimg.imread`.
 - Limites ajustados para representar a trajetória.
 - Linha para representar a trajetória do projétil (`line`).
- **Função de atualização**

```
def update(frame)
```

Em cada "frame" da animação:

1. Calcula-se a força de arrasto;
2. Determinam-se as acelerações horizontais e verticais;
3. Atualizam-se as velocidades e posições.
4. Caso o projétil atinja o solo, a animação para.

- **Execução da animação**

A animação é gerada com `FuncAnimation`, que atualiza o gráfico para cada intervalo de tempo.

- **Exibição dos resultados**

- Distância máxima
- Altura máxima
- Tempo total

FUNÇÃO PARA INICIAR A SIMULAÇÃO

```
def iniciar_simulacao():
```

- Lê os valores das entradas fornecidas na interface gráfica.
- Chama a função `simular_lancamento` com esses valores.

INTERFACE GRÁFICA COM TKINTER

- `tk.Tk()`: Cria a janela principal da interface gráfica, onde você define o título e o tamanho da janela usando `title()` e `geometry()`.
- `Image.open`: Carrega e manipula imagens para serem usadas na interface. No caso do código, a imagem de fundo é carregada com essa função, redimensionada para se ajustar ao tamanho da janela, e convertida para um formato compatível com Tkinter usando `ImageTk.PhotoImage()`.
- `ttk.Label()`: Cria rótulos (labels) explicativos que são usados para fornecer informações ou instruções sobre o que o usuário deve inserir em cada campo de entrada.
- `ttk.Entry()`: Cria campos de entrada (caixas de texto) onde o usuário pode digitar valores. No código, esses campos são pré-preenchidos com valores padrão, utilizando o método `insert()`.
- `ttk.Button()`: Cria um botão na interface gráfica que, quando clicado, chama a função `iniciar_simulacao()`, que executa a simulação do lançamento do projétil.

EXECUÇÃO DO PROGRAMA

Por último, o programa chama `root.mainloop()` para iniciar a interface gráfica e aguardar interações do usuário.