Escuela Politécnica Nacional

Métodos Numéricos

Paralelo: GR1CC

Proyecto

Primer Bimestre

Ozzy Loachamín
Miguel Muzo
Diego Suquillo
Felipe Quirola

2024 – A

Índice

1. Objetivos

- Modelar la trayectoria de la bomba utilizando métodos numéricos.
- Implementar un programa en Python para resolver las ecuaciones de movimiento.
- Visualizar la trayectoria de la bomba y el punto de impacto en gráficos adecuados.
- Analizar los resultados obtenidos y comparar con soluciones teóricas cuando sea posible.

2. Introducción

Este proyecto se enfoca en desarrollar un algoritmo robusto y eficiente para calcular con precisión la trayectoria de una bomba lanzada desde un avión en movimiento, considerando la posibilidad de colisión con una montaña. El objetivo principal es asegurar que los cálculos reflejen fielmente la física del problema, desde el momento del lanzamiento hasta el eventual impacto [1].

El algoritmo utilizado divide el tiempo en pequeños pasos y utiliza fórmulas matemáticas para entender cómo se mueve la bomba bajo la influencia de la gravedad y la forma de la montaña. Se utiliza una técnica llamada método de Newton-Raphson para determinar el momento exacto en que la bomba podría chocar con la montaña, asegurando que los resultados sean lo más precisos posible [2].

3. Desarrollo

3.1. Desarrollo matemático

La trayectoria de la bomba está descrita por las siguientes ecuaciones de movimiento en los ejes x y y, respectivamente:

$$x(t) = x_0 + v \cdot t$$

$$y(t) = h_a - \frac{1}{2}gt^2$$

donde:

- x_0 es la posición inicial en x (el punto de suelta de la bomba).
- v es la velocidad del avión.
- h_a es la altura inicial del avión.
- g es la aceleración debida a la gravedad (9.81 m/s^2).

- t es el tiempo.

La montaña se representa como un triángulo isósceles con su vértice superior centrado en d. Las ecuaciones de los lados de la montaña son:

$$y = \frac{h_m}{\text{half\_base}} \cdot (d - x) \quad \text{(lado izquierdo)}$$
$$y = \frac{h_m}{\text{half\_base}} \cdot (x - d) \quad \text{(lado derecho)}$$

donde:

- h_m es la altura de la montaña.
- half_base es la mitad de la base de la montaña, calculada como $h_m \cdot \tan(\text{angle})$.

Para encontrar la intersección entre la trayectoria de la bomba y los lados de la montaña, igualamos las ecuaciones:

$$h_a - \frac{1}{2}gt^2 = \frac{h_m}{\text{half\_base}} \cdot (d - (x_0 + vt)) \quad \text{(lado izquierdo)}$$
$$h_a - \frac{1}{2}gt^2 = \frac{h_m}{\text{half\_base}} \cdot ((x_0 + vt) - d) \quad \text{(lado derecho)}$$

Estas son ecuaciones cuadráticas en t, que podemos resolver usando la fórmula cuadrática:

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

donde:

- $a = \frac{1}{2}g$
- $b = \pm v \frac{h_m}{\text{half\_base}}$ (positivo para el lado izquierdo y negativo para el lado derecho)
- $c = h_a - h_m - \frac{h_m}{\text{half\_base}} \cdot (d - x_0)$ (para el lado izquierdo) o $c = h_a - h_m - \frac{h_m}{\text{half\_base}} \cdot (x_0 - d)$ (para el lado derecho)

Intersección con el Suelo

Si la bomba no impacta con la montaña, calculará el tiempo de impacto con el suelo, que ocurre cuando $y(t) = 0$:

$$0 = h_a - \frac{1}{2}gt^2$$

Resolviendo para t, tenemos:

$$t = \sqrt{\frac{2h_a}{g}}$$

El punto de impacto en el suelo se calcula con:

$$x_{\text{impacto}} = x_0 + v \cdot t$$

y la coordenada y será 0 en el impacto con el suelo.

3.2. Descripción del Proceso

Este diagrama de flujo proporciona una vista estructurada del proceso y las decisiones tomadas durante la ejecución del código, permitiendo entender la secuencia lógica y las interacciones entre las funciones y cálculos. A continuación, se describe el proceso:

1. **Inicio y Configuración de Widgets:** El código comienza configurando los widgets interactivos con ipywidgets.
2. **Función Principal:** `dibujar_triangulo_isosceles` es la función principal que se llama cuando se ajustan los valores de los widgets.
3. **Validación:** Comprueba si la altura del avión es mayor que la altura de la montaña.
4. **Cálculos Iniciales:** Calcula la mitad de la base del triángulo de la montaña y las coordenadas del triángulo.
5. **Creación de Figura y Ejes:** Configura el espacio de dibujo usando matplotlib.
6. **Dibujo de la Montaña:** Dibuja la montaña como un triángulo.
7. **Posición del Avión y caída de la Bomba:** Calcula las posiciones iniciales y los puntos de caída.
8. **Función para Trayectoria de la Bomba:** `calcular_trayectoria_bomba` calcula el tiempo y la posición de impacto usando el método de Newton-Raphson.
9. **Iteración de Newton-Raphson:** Encuentra el tiempo de impacto resolviendo la ecuación de movimiento.
10. **Dibujo de Trayectorias:** Dibuja las trayectorias del avión y de la bomba.
11. **Ajustes y Finalización:** Ajusta los límites del gráfico, añade cuadrícula, leyenda y muestra el gráfico final.

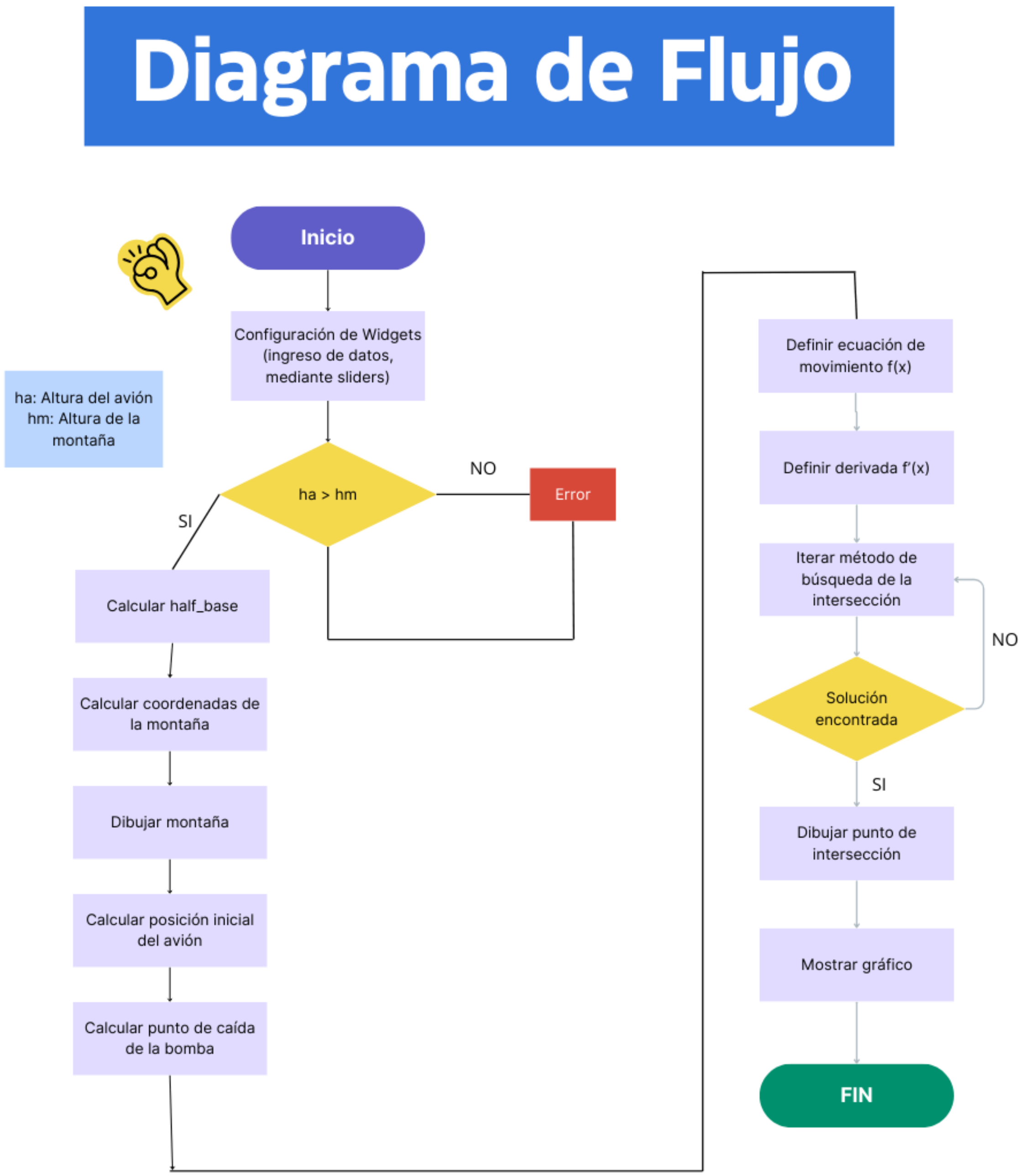

Figura 1: Diagrama de flujo del proceso

3.3. Implementación y Visualización en Python

El código implementado en Python se estructura utilizando NumPy para cálculos matemáticos y Sympy para manejo de ecuaciones simbólicas, asegurando así la precisión y eficiencia computacional del algoritmo. La integración con IPython Widgets permite la interactividad mediante controles deslizantes, lo que facilita la experimentación y comprensión del impacto de diferentes parámetros como la velocidad del avión, alturas y distancias relevantes.

Para la representación gráfica, se utiliza la biblioteca Matplotlib de Python, que permite visualizar de manera interactiva la trayectoria de la bomba, la ruta del avión, el punto de suelta de la bomba y el punto de impacto estimado. Esta visualización proporciona una comprensión clara y detallada del comportamiento físico del sistema.

La función `dibujar_triangulo_isosceles` se encarga de generar gráficos dinámicos que muestran la trayectoria calculada de la bomba y su interacción con la montaña. Esto no solo proporciona una herramienta educativa efectiva, sino que también valida la exactitud del modelo matemático implementado.

4. Conclusiones

El método de Newton-Raphson es muy eficiente y preciso para calcular el tiempo de impacto de un proyectil. Su rapidez en encontrar soluciones lo hace ideal para simulaciones que requieren cálculos rápidos y exactos.

El proyecto demuestra que Newton-Raphson es útil no solo para cálculos simples, sino también en escenarios complejos como impactos en montañas. Su capacidad para manejar obstáculos y condiciones reales lo hace muy versátil.

Referencias

[1] Gadella M, Nieto LM. Métodos matemáticos avanzados para ciencias e ingenierías. Secretariado de Publicaciones e Intercambio Editorial, Universidad de Valladolid; 2000.

[2] Fink KD. Métodos numéricos; 1995.