Escuela Politécnica Nacional

Facultad de Sistemas en Ingeniería de Ciencias de la Computación

Metodos Numericos

Proyecto de sobrevuelo

Alumno:
Jorge Bosquez V.
Elliot Cazar M.
Stefany Espin S.
Jitala Stalin L.

Profesor:
Jonathan A. Zea

18 de junio de 2024

Índice

1. Objetivos

- Desarrollar un algoritmo robusto y eficiente capaz de calcular con precisión los puntos a lo largo de la trayectoria de una bomba lanzada desde un avión en movimiento, teniendo en cuenta su posible colisión con una montaña en su trayectoria.
- Mediante un análisis matemático, hallar la ecuación que describe el movimiento de la bomba desde su lanzamiento hasta su impacto, garantizando que los cálculos sean exactos y reflejen fielmente la física del problema.
- Crear una interfaz gráfica de usuario intuitiva que permita la entrada de las variables necesarias para el cálculo (como la distancia del avión y la altura de la montaña), y que visualice claramente la trayectoria de la bomba y el punto de impacto.

2. Introduccion

En la creación de este proyecto, se ha elegido abordar el problema de calcular la trayectoria de una bomba lanzada desde un avión en movimiento. El objetivo principal es graficar la trayectoria de la bomba desde el momento en que se suelta del avión hasta su colisión con una montaña situada en la trayectoria del avión.

El programa se compondrá de los siguientes elementos:

1. **Lógica de Cálculo de la Trayectoria:** Algoritmos que calcularán los puntos a lo largo de la trayectoria de la bomba.
2. **Función de Gráficación:** Una función dedicada a generar la gráfica de la trayectoria de la bomba.
3. **Interfaz Gráfica de Usuario (GUI):** Una interfaz que permitirá la entrada de las variables necesarias para el cálculo, tales como la distancia del avión, la altura del avión, la altura de la montaña, el ángulo de la pendiente de la montaña o la base de la montaña.

La interfaz gráfica no solo facilitará la entrada de datos, sino que también mostrará la gráfica de la trayectoria de la bomba, indicando claramente el punto de impacto con la montaña o, en su defecto, el punto donde la bomba impactaría en el suelo.

Para el desarrollo de este proyecto, se ha elegido el lenguaje de programación Python debido a su notable flexibilidad en el manejo de datos. Las bibliotecas utilizadas incluyen NumPy para la gestión de operaciones matemáticas, Matplotlib para la representación gráfica de la trayectoria de la bomba, así como la montaña y el punto de impacto, y finalmente, la biblioteca Tkinter, que ofrece herramientas útiles para la creación de interfaces gráficas.

3. Metodologia

3.1. Descripción de la solución.

El desarrollo del proyecto se centró en resolver un ejercicio de movimiento parabólico simple[1], ilustrado en la Figura 1, sin considerar la resistencia del aire. El problema a abordar era la trayectoria de una bomba que sigue un movimiento parabólico hasta impactar con una montaña. Para resolver este problema, se utilizó la ecuación del movimiento parabólico en función de x, y y se generó la gráfica de la trayectoria de la bomba.

Una vez obtenida la gráfica de la trayectoria, el siguiente paso fue determinar la ecuación que describe la forma de la montaña. Esta ecuación depende de variables como la altura y el ángulo de la montaña. Con ambas funciones —la de la trayectoria de la bomba y la de la montaña—, se procedió a calcular iterativamente los valores de y para diferentes valores de x y se compararon para encontrar el punto de intersección. Si los valores de y para ambas funciones son iguales en algún valor de x, ese es el punto de impacto de la bomba en la montaña. Si no se encuentra un valor de x en el que ambas funciones coincidan, se concluye que la bomba no impacta la montaña, sino que cae al suelo.

Para determinar el punto de impacto en el suelo, se aplican los principios del movimiento parabólico, que puede descomponerse en dos componentes: caída libre y movimiento rectilíneo uniforme. Utilizando la fórmula del movimiento de caída libre, se calcula el tiempo en el que la bomba llega al suelo. Luego, con el movimiento rectilíneo uniforme, se determina el valor de x en el que la bomba impacta el suelo.

Finalmente, una vez obtenido el valor de x en el punto de impacto, se grafica la trayectoria de la bomba hasta ese punto, mostrando el punto de impacto (x, y) ya sea en la montaña o en el suelo.

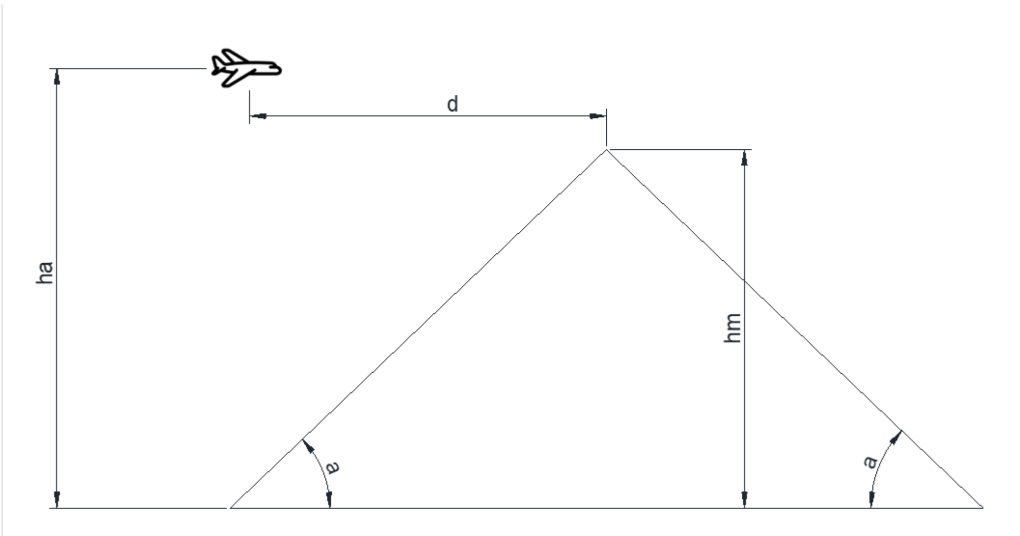

Figura 1: Ejercicio del proyecto

3.2. Desarrollo matemático.

Para graficar la trayectoria de la bomba, primero necesitamos obtener su ecuación de movimiento parabólico. Utilizamos la fórmula del movimiento parabólico para determinar la función de la trayectoria de la bomba[2]:

$$F_{bomba}(x) = y_0 - \frac{g}{2v^2}x^2$$

donde:

- g es la aceleración debida a la gravedad, que en este caso consideramos $g = 9{,}81 m/s^2$
- v es la velocidad inicial de la bomba, la cual es igual a la velocidad constante del avión.
- y_0 es la altura desde la cual la bomba comienza a caer, correspondiente a la altura del avión en ese punto.
- x representa la distancia horizontal.

Con esta ecuación, podemos graficar la trayectoria de la bomba en función de (x, y).

A continuación, se debe determinar la ecuación para graficar la montaña. Según la ilustración del ejercicio, la montaña tiene la forma de un triángulo isósceles. La base de la montaña comienza a una cierta distancia d_2 desde el origen de coordenadas. El dato proporcionado es la distancia horizontal desde el origen hasta el pico de la montaña. Para calcular d_2, restamos la mitad de la base de la montaña desde la distancia del origen hasta el pico. La mitad de la base b de la montaña se calcula utilizando la fórmula:

$$b = hm/tan\alpha$$

donde:

- b es la mitad de la base de la montaña.
- hm es la altura de la montaña.
- α es el ángulo de inclinación de la pendiente.

Una vez calculada la mitad de la base, obtenemos la distancia d_2 donde comienza a elevarse la montaña. Para graficar la montaña, usamos la ecuación de la recta de punto-pendiente. La pendiente está dada por $tan\alpha$. La función que describe la pendiente ascendente de la montaña es:

$$F_{acenso}(x) = (x - d_2)tan\alpha$$

donde $x - d_2$ desplaza la gráfica horizontalmente en d_2 unidades.

Para la pendiente descendente, usamos una función similar pero ajustada para empezar desde la altura hm:

$$F_{descenso}(x) = hm - (x - d_2)tan\alpha$$

Esto asegura que la gráfica comience desde la altura hm, evitando que decrezca hacia valores negativos. Combinando estas dos funciones, obtenemos la función completa de la montaña:

donde b es la mitad de la base de la montaña, hm es la altura de la montaña y α es el angulo. una vez calculada la base de la mostaña se obtiene la distancia x donde empieza a crecer la montaña, la cual se denominara d_2. Para graficar la montaña se uso la funcion linal punto pendiente, donde el valor de la pendiente esta dado por $tan\alpha$, quedando la funcion de creciemitno como $(x-d_2)tan\alpha$ donde $x-d_2$ es necesario para desplazar la grafica un d_2 distancia hacia el lado positivo de los x. ahora se neceita la funcion para que la montaña decresca, para ello usando la misma funcion punto pendiente pero cambiado de signo quedando $hm-(x-d_2)tan\alpha$ donde hm permite que la grafica empieze desde un valor hm hacia arriba, de lo contrario la grafica empezaria a decrecer hacia los valores negativos, por ultimo solo es unir las funcinoes en uno sola quedando

$$f_{montaña}(x) = \begin{cases} 0 & \text{si } x < d_2 \\ (x-d_2)tan\alpha & \text{si } d_2 < x < d \\ hm-(x-d_2)tan\alpha & \text{si } d < x < d+b \\ 0 & \text{si } x > d+b \end{cases}$$

Esta función permite graficar la montaña con valores de 0 desde el origen hasta el punto donde comienza a elevarse (en d_2). Luego, la montaña crece linealmente hasta alcanzar su pico. Después del pico, la montaña desciende simétricamente hasta el suelo, y más allá de este punto, la función vuelve a tener valores de 0.

Con las funciones de la trayectoria de la bomba y la forma de la montaña definidas, procedemos a iterar sobre los valores de x para encontrar el punto donde ambas funciones se intersectan. El valor de x en el que las funciones coinciden es el punto de impacto de la bomba en la montaña. Este proceso se describirá en detalle en la siguiente sección.

En caso de que la bomba no impacte en la montaña, calculamos su punto de impacto en el suelo utilizando las fórmulas de la caída libre y el movimiento rectilíneo uniforme. Primero, determinamos el tiempo de impacto $t_{impacto}$ con la siguiente ecuación:

$$t_{impactp} = \sqrt{\frac{2ha}{g}}$$

donde:

- ha es la altura del avión.
- g es la aceleración debida a la gravedad.

Luego, utilizando el movimiento rectilíneo uniforme, calculamos la distancia horizontal $x_{impacto}$ en el momento del impacto:

$$x_{impacto} = v\sqrt{\frac{2ha}{g}}$$

donde v es la velocidad constante del avión.

inalmente, se grafica la trayectoria completa de la bomba hasta su punto de impacto, sea en la montaña o en el suelo, representando claramente el punto (x, y) de impacto.

3.3. Diagrama de flujo / pseudocódigo.

3.3.1. Funciones Bomba y montaña

Para implementar las funciones de la trayectoria de la bomba y la montaña, se codificaron mediante funciones en Python. Previamente, se calcularon los valores de b y d_2 necesarios para definir la función de la montaña. Además, se calculó el x de impacto de la bomba sin considerar la montaña. Esto es crucial para establecer un rango de valores de x sobre el cual graficar tanto la trayectoria como la montaña.

```
// Calcular la mitad de la base de la montaña usando la altura máxima hm y el ángulo de
la pendiente a
b = hm / tan(radians(a))
// Calcular la distancia desde el origen hasta el inicio de la montaña (d2)
d2 = d - b
// Definir la función que describe la forma de la montaña
Función mountain(x):
    Si x es menor que d2 entonces:
        retornar 0  // No hay montaña antes de d2
    Si no, si x está entre d2 y d inclusive entonces:
        retornar (x - d2) * tan(radians(a))  // Subida de la montaña
    Si no, si x está entre d y d + b inclusive entonces:
        retornar hm - (x - d) * tan(radians(a))  // Bajada de la montaña
    De lo contrario:
        retornar 0  // Después de la montaña, la altura es 0
// Definir la función que describe la trayectoria del avión
Función trayectoria(x, g, v, ha):
    retornar ha - (g / (2 * v^2)) * x^2
// Calcular el tiempo de vuelo hasta que la trayectoria toque el suelo (donde y = 0)
tiempo_impacto_suelo = sqrt(2 * ha / g)
// Calcular la posición horizontal del impacto con el suelo usando la velocidad v y el
tiempo de impacto
x_impacto_suelo = v * tiempo_impacto_suelo
```

Listing 1: Creacion de la funcion montaña y trayectoria, ademas del calculo del x de impacto

3.3.2. Iteracion para encontrar el x de impacto

Una vez definidas las funciones para la montaña y la trayectoria de la bomba, se iterarán en un rango de valores decimales desde 0 hasta $d + b$. Si los valores de ambas funciones coinciden en algún punto, se identifica el x de impacto en la montaña. En caso contrario, se asigna a la variable x de impacto en la montaña el valor previamente calculado, indicando que la bomba impactó en el suelo.

```
// Inicializar la variable de x_vlas con un rango de valores
x_vals = valores reales entre 0 y d + b
// Recorrer cada valor en x_vals
Para cada x en x_vals hacer:
    // Calcular la altura de la trayectoria en el punto x
    altura_trayectoria = trayectoria(x, g, v, ha)
    // Calcular la altura de la montaña en el punto x
    altura_montana = mountain(x)
    // Comparar la altura de la trayectoria con la de la montaña
    Si altura_trayectoria es menor o igual que altura_montana entonces:
        // Encontramos el punto de impacto con la montaña
        x_impacto_montana = x
        y_impacto_montana = altura_montana
        // Salir del bucle ya que encontramos el punto de impacto
        Romper bucle
// Verificar si no hubo impacto con la montaña
Si x_impacto_montana es None entonces:
    // El impacto es con el suelo
    x_impacto_montana = x_impacto_suelo
    y_impacto_montana = 0
```

Listing 2: Encontrar x de impacto

3.3.3. Diagrama de flujo

La ejecución del código se ilustra en la Figura 2, que muestra el flujo lógico del programa. A continuación, se describe el proceso:

1. **Entrada de Variables Iniciales:** El usuario proporciona las variables iniciales necesarias para la simulación.

2. **Cálculo de la Función Montaña:** Se calculan los valores necesarios para definir la función que describe la montaña, y se implementa esta función en el programa.

3. **Definición y Cálculo de la Trayectoria de la Bomba:** Se define la función que describe la trayectoria de la bomba y se calcula el punto de impacto en el eje x Estos cálculos permiten establecer el rango de valores necesario para graficar las funciones.

4. **Iteración para Determinar el Impacto:** Se implementa un proceso iterativo para encontrar el punto exacto en el eje x donde la bomba impacta la montaña. Además, se verifica si realmente se produce el impacto. Si no hay impacto, el valor de x calculado previamente se asigna como el punto de impacto en la montaña.

5. **Generación del Rango de Valores para la Gráfica:** Se ajusta el rango de valores para la gráfica de la trayectoria de la bomba. Esto se hace de manera que se pueda graficar toda la trayectoria si la bomba no impacta con la montaña, o solo hasta el punto de impacto si este ocurre.

6. **Graficación Final:** Finalmente, se grafican las dos funciones: la trayectoria de la bomba y el perfil de la montaña.

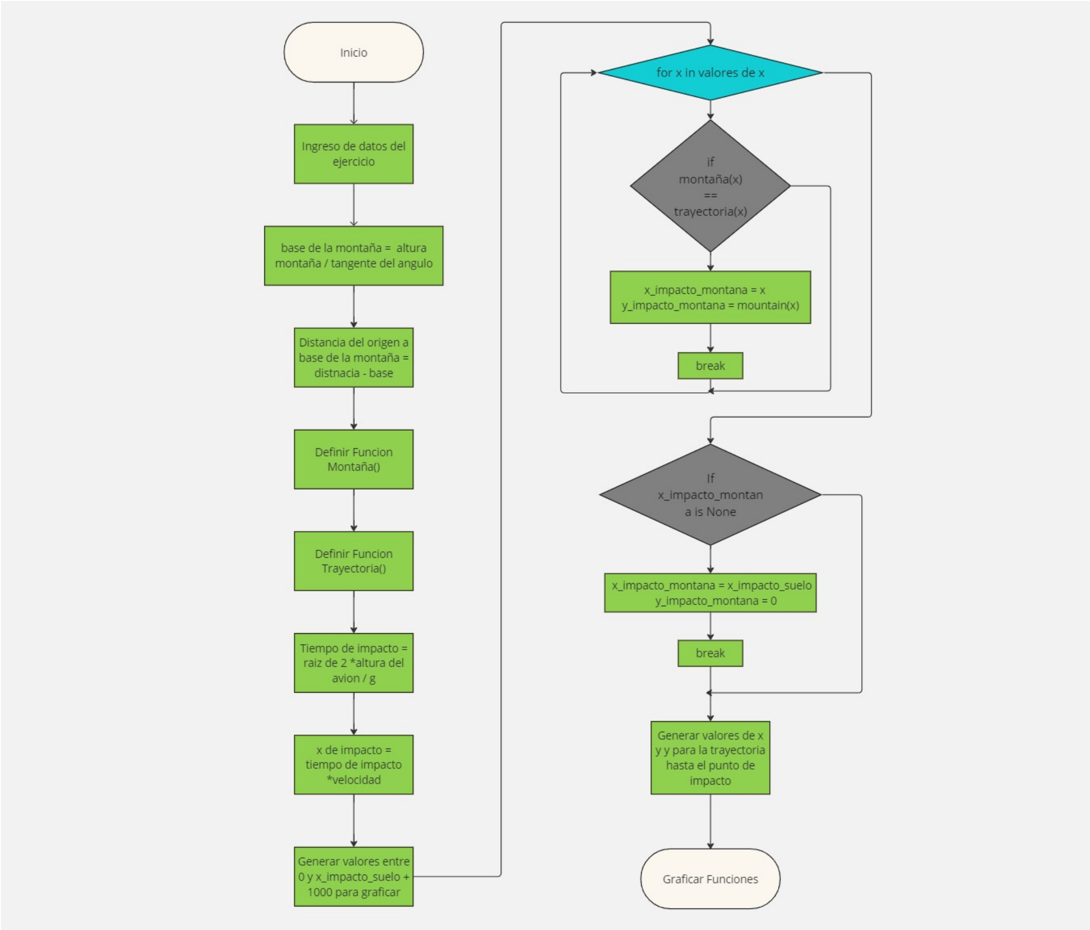

Figura 2: Información de la imagen

3.4. Detalles importantes de la implementación.

3.4.1. Claururas en la entrada de valores

3.4.2. Interfaz grafica

4. Resultados

5. Conclusiones

- El algoritmo desarrollado mostró una alta precisión y eficiencia en el cálculo de la trayectoria de la bomba, validado mediante múltiples pruebas con diferentes velocidades y distancias de lanzamiento, demostrando que estos parámetros influyen significativamente en la trayectoria.

- Mediante un análisis matemático, se logró derivar la ecuación que rige el movimiento de la bomba hasta su impacto, lo que permitió realizar cálculos precisos y consistentes con la realidad física.

- La interfaz gráfica de usuario implementada demostró ser altamente funcional e intuitiva, facilitando la entrada de datos y la visualización de los resultados, mejorando así la interacción del usuario con el programa y permitiendo una interpretación clara y comprensible de la trayectoria de la bomba en diversos escenarios.

Bibliografía

[1] *Bombardear un blanco móvil desde un avión*. URL: http://www.sc.ehu.es/sbweb/fisica/cinematica/bombardeo/bombardeo.htm.

[2] *Movimiento parabólico*. URL: https://fisiquimicamente.com/recursos-fisica-quimica/apuntes/1bach/movimiento-parabolico/.