



ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

MÉTODOS NUMÉRICOS (ICCD412)

PROYECTO IB

Alumnos:

Adrián Francisco Correa A.
Ian Andrés Oñate G.
Stiv Stalyn Quishpe C.
Jorge Luis Yánez S.

PROFESOR: Jonathan Alejandro Zea G.

Tabla de contenido

1. OBJETIVOS.....	3
2. INTRODUCCIÓN.....	3
3. METODOLOGÍA	4
3.1. Cálculo de la trayectoria.....	4
3.2. Generación de la visualización.....	5
3.3. Interfaz gráfica.....	5
3.4. Estructura del terreno	5
3.5. Desafíos encontrados	6
3.6. Diagrama de flujo del código	7
4. RESULTADOS	8
4.1. Trayectoria larga con impacto en el suelo más allá del cráter	8
4.2. Trayectoria que impacta el fondo del cráter	8
5. CONCLUSIONES	10
6. RECOMENDACIONES.....	10
7. REFERENCIAS BIBLIOGRAFICAS.....	11

1. OBJETIVOS

- o Describir y modelar matemáticamente el movimiento parabólico de la bomba en función de las condiciones iniciales (velocidad, altura y distancia).
- o Incorporar las características geométricas del cañón en el análisis para determinar la posición del impacto con respecto a su base.
- o Graficar la trayectoria de la bomba y mostrar el punto donde impacta, ya sea este dentro o fuera del cráter.
- o Validar los resultados del modelo matemático con simulaciones o representaciones gráficas.
- o Explorar cómo los parámetros del problema (velocidad inicial, altura, ángulo del cañón) afectan la trayectoria de la bomba y su punto de impacto.

2. INTRODUCCIÓN

Entender cómo se mueve un objeto bajo la influencia de la gravedad es clave para analizar fenómenos físicos en distintos contextos. Este informe presenta una herramienta interactiva desarrollada en Python, diseñada para simular y visualizar la trayectoria de una bomba lanzada desde un avión en movimiento, considerando un terreno con un cráter representado de forma geométrica. Usando bibliotecas como matplotlib para crear gráficos dinámicos y tkinter para construir una interfaz intuitiva, la aplicación permite no solo observar la trayectoria, sino también evaluar el impacto de la bomba al chocar con el cráter o el suelo.

El objetivo de este proyecto es explicar cómo se diseñó e implementó esta herramienta, mostrar los resultados obtenidos y reflexionar sobre su utilidad para entender mejor los movimientos físicos. Gracias a la posibilidad de experimentar con parámetros como la altura inicial del avión, la velocidad de la bomba o las dimensiones del cráter, el usuario puede observar de manera clara cómo estos factores influyen en el comportamiento del sistema.

Python fue elegido para este proyecto debido a su flexibilidad y su robusto ecosistema de bibliotecas científicas. En este caso, matplotlib se usa para representar de forma gráfica el terreno y la trayectoria, mientras que numpy permite realizar los cálculos matemáticos necesarios. Además, se utiliza tkinter para crear una interfaz simple y fácil de usar, que permite a los usuarios configurar las condiciones de la simulación sin complicaciones. Este enfoque logra combinar precisión matemática con una presentación visual clara y atractiva.

En este informe se detalla cómo funciona el código, explicando las partes principales y su lógica. La herramienta simula el movimiento parabólico de la bomba bajo la fuerza de la gravedad y detecta con precisión los choques contra el cráter o el suelo. También se cuida que la simulación sea fluida y visualmente agradable, con elementos como el movimiento continuo del avión y la representación progresiva de la trayectoria.

Finalmente, se analizan diferentes escenarios simulados, cambiando parámetros como la altura inicial del avión o las dimensiones del cráter. Se destacan las fortalezas y limitaciones de la herramienta, además de posibles aplicaciones en la enseñanza de física o en simulaciones más complejas. Este informe concluye con sugerencias para mejorar la herramienta y extender sus funcionalidades, subrayando la importancia de usar recursos interactivos y visuales para facilitar la comprensión de fenómenos físicos.

3. METODOLOGÍA

Este proyecto se basa en calcular y visualizar la trayectoria de una bomba lanzada desde un avión en movimiento. Para lograrlo, usamos ecuaciones físicas básicas y herramientas gráficas interactivas.

3.1. Cálculo de la trayectoria

El movimiento de la bomba se calcula utilizando las ecuaciones del movimiento parabólico bajo la influencia de la gravedad. La ecuación base considera la altura (y) de la bomba en función del tiempo (t) y la distancia horizontal (x):

$$y = y_0 - \frac{1}{2}gt^2$$

Donde:

y es la altura inicial del avión.

g es la aceleración de la gravedad (9.81 m/s^2).

t es el tiempo, calculado a partir de la distancia horizontal recorrida (x) y la velocidad inicial en esa dirección (V_{0x}).

```
# Constante de gravedad
g = 9.81

# Función de la trayectoria de la bomba
def trayectoria(x, x0, y0, v0x, v0y):
    t = (x - x0) / v0x
    return y0 - 0.5 * g * t**2
```

Ilustración 1 Ecuación para calcular la trayectoria

Además, incluimos detección de colisiones para que la simulación sea más realista. Esto significa que el programa verifica si la bomba impacta contra las paredes o el fondo del cráter. Para esto, se usan las coordenadas del terreno y se comparan con la posición de la bomba.

3.2. Generación de la visualización

La visualización se crea con matplotlib, una biblioteca de Python que permite dibujar gráficos. En esta simulación, mostramos:

- **Cráter:** Representado como un trapecio invertido que depende de las dimensiones y ángulos que define el usuario.
- **Suelo:** Se extiende a ambos lados del cráter para simular un terreno continuo.
- **Trayectoria de la bomba:** Una curva que se dibuja en tiempo real mientras la bomba cae.
- **Avión:** Representado con una imagen que se mueve horizontalmente, simulando el vuelo.

El gráfico se va actualizando dinámicamente conforme avanza la simulación, lo que lo hace más interactivo y fácil de entender.

3.3. Interfaz gráfica

Para que cualquiera pueda usar la simulación sin saber programar, diseñamos una interfaz gráfica con tkinter. Es sencilla y tiene campos donde se ingresan parámetros como:

- Altura inicial del avión.
- Velocidad inicial de la bomba.
- Dimensiones del cráter (longitud, altura y ángulo de las paredes).
- Posición del cráter en el terreno.

Además, hay un botón que inicia la simulación después de configurar los parámetros. Cuando el usuario hace clic, la simulación se ejecuta y muestra el resultado visualmente.

3.4. Estructura del terreno

El terreno simulado está compuesto por:

- **Cráter:** Tiene forma de trapecio invertido. Las coordenadas de sus vértices se calculan con base en la longitud, altura y ángulo que define el usuario.
- **Suelo:** Es una superficie plana que rodea al cráter, extendiéndose a los lados para completar la simulación.

Para construirlo usamos numpy, que nos permite hacer cálculos matemáticos precisos, y matplotlib, que se encarga de dibujar el terreno y los demás elementos.

3.5. Desafíos encontrados

- **Selección de la fórmula correcta:** Uno de los primeros desafíos fue elegir la fórmula adecuada para calcular la trayectoria de la bomba. Analizamos diversas ecuaciones relacionadas con el movimiento parabólico y, finalmente, optamos por una que consideraba únicamente la gravedad y la velocidad del avión, ya que era la que mejor se ajustaba a las necesidades del proyecto.
- **Aprendizaje de librerías:** Trabajar con librerías como Matplotlib y Tkinter fue un reto al inicio debido a la amplia variedad de funciones que ofrecen y la necesidad de identificar cuáles eran las más adecuadas para cada contexto. Matplotlib resultó indispensable para la visualización gráfica de la simulación, mientras que Tkinter facilitó la creación de una interfaz interactiva. Dominar estas herramientas requirió tiempo, práctica y la consulta de tutoriales y documentación.
- **Integración de componentes:** Un desafío significativo fue integrar los distintos componentes del código, como la simulación y la interfaz gráfica de usuario. Fue crucial asegurarse de que todos los elementos funcionaran de manera fluida y que la interfaz fuera clara y fácil de usar para el usuario final. Este proceso requirió varias iteraciones y ajustes para lograr una solución funcional y eficiente.

3.6. Diagrama de flujo del código

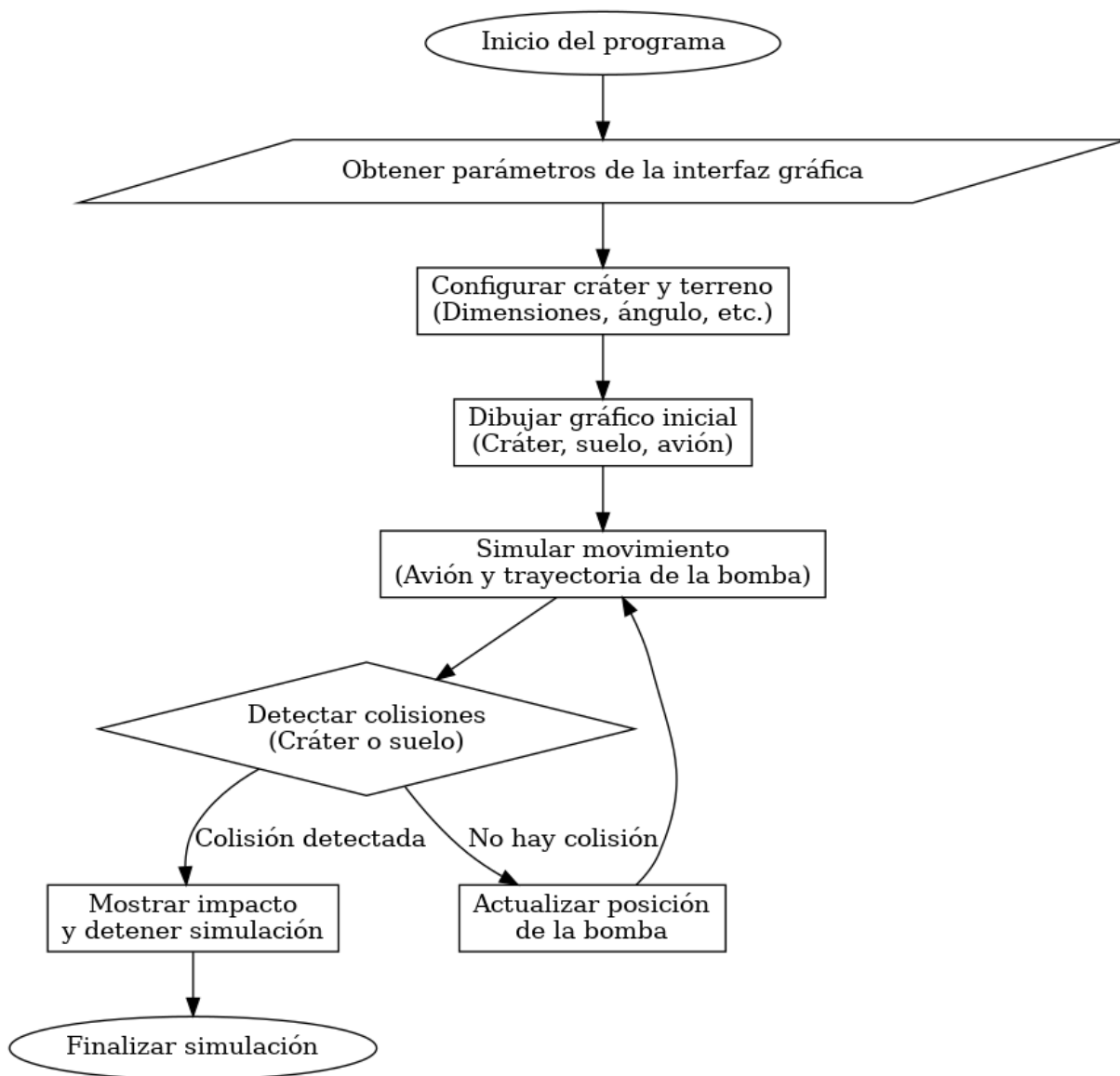
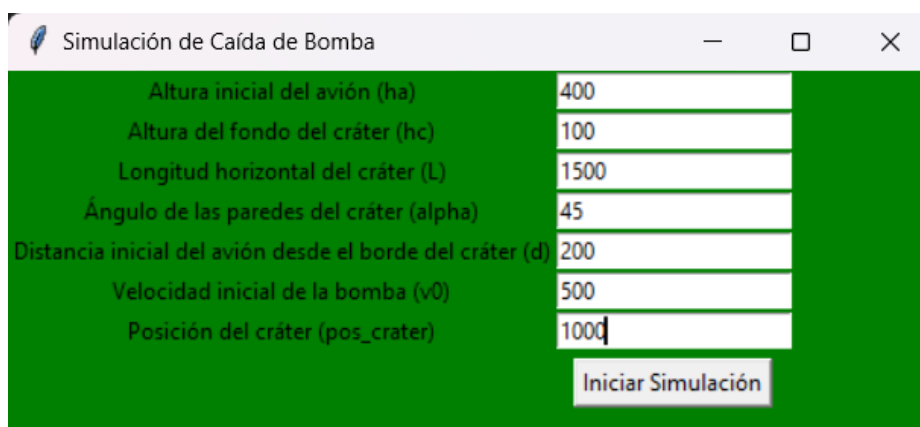


Ilustración 2 Diagrama de flujo del código

4. RESULTADOS

4.1. Trayectoria larga con impacto en el suelo más allá del cráter

La simulación muestra la trayectoria parabólica que sigue la bomba desde el momento en que es lanzada por el avión hasta que impacta en el suelo más allá del cráter. En este caso, la bomba atraviesa el cráter sin colisionar con sus paredes o su fondo, y continúa su movimiento hasta alcanzar el terreno plano, donde se detiene y finaliza la simulación.



Simulación de Caída de Bomba

Altura inicial del avión (h_a)	400
Altura del fondo del cráter (h_c)	100
Longitud horizontal del cráter (L)	1500
Ángulo de las paredes del cráter (α)	45
Distancia inicial del avión desde el borde del cráter (d)	200
Velocidad inicial de la bomba (v_0)	500
Posición del cráter (pos_crater)	1000

Iniciar Simulación

Ilustración 3 Datos ingresados para ejemplo 4.1

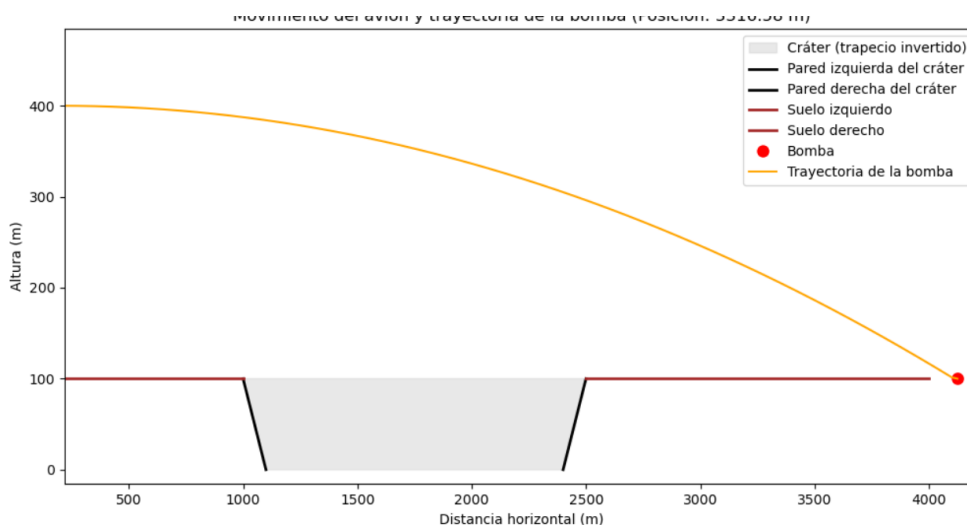


Ilustración 4 Imagen resultante del ejemplo 4.1

4.2. Trayectoria que impacta el fondo del cráter

La simulación muestra la trayectoria parabólica que sigue la bomba desde el momento

en que es lanzada por el avión hasta que impacta en el cráter. La bomba colisiona con

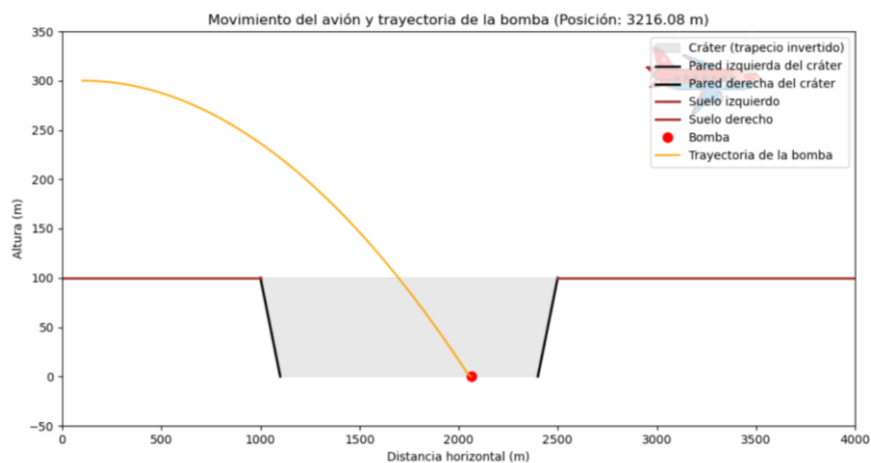



Ilustración 6 Imagen resultante del ejemplo 4.2.

el cráter, según los datos que el usuario ingrese, lo que detiene su movimiento y finaliza la simulación.

 Simulación de Caída de Bomba

Altura inicial del avión (h_a)	300
Altura del fondo del cráter (h_c)	100
Longitud horizontal del cráter (L)	1500
Ángulo de las paredes del cráter (α)	45
Distancia inicial del avión desde el borde del cráter (d)	100
Velocidad inicial de la bomba (v_0)	250
Posición del cráter (pos_crater)	1000

Iniciar Simulación

Ilustración 5 Datos ingresados para el ejemplo 4.2.

5. CONCLUSIONES

- El programa modela correctamente cómo se mueve la bomba utilizando ecuaciones simples, calculando con precisión su trayectoria según la velocidad, altura y distancia iniciales.
- Se incorpora la forma del cráter al análisis, lo que permite identificar si la bomba golpea las paredes, el fondo del cráter o el suelo más allá.
- Los gráficos muestran claramente la trayectoria de la bomba y dónde impacta, haciendo que los resultados sean fáciles de interpretar.
- Las simulaciones confirman que los cálculos matemáticos están bien hechos, ya que los resultados obtenidos coinciden con lo que se esperaba según las ecuaciones.
- Cambiar parámetros como la velocidad, la altura o el ángulo del cráter genera trayectorias muy diferentes, mostrando cómo estos factores influyen directamente en el movimiento de la bomba.
- La simulación no solo sirve para calcular trayectorias, sino también para experimentar con diferentes configuraciones y entender mejor cómo afectan los resultados.
- El programa combina cálculos matemáticos y gráficos dinámicos, haciendo que sea práctico y visualmente intuitivo tanto para estudiar como para aplicar en otros escenarios.
- Los objetivos del proyecto se cumplieron completamente, ya que el código desarrollado permite analizar, calcular y visualizar de manera clara y precisa el movimiento de la bomba y su interacción con el terreno.

6. RECOMENDACIONES

- **Instala lo necesario antes de empezar**

Asegúrate de instalar todas las bibliotecas necesarias ejecutando el comando:

```
pip install -r requirements.txt
```

Esto evitará errores por falta de dependencias al correr el programa.

- **Configura correctamente la interfaz gráfica**

Antes de iniciar la simulación, ajusta los tamaños de la ventana y verifica que los campos de texto sean visibles y fáciles de llenar. Esto te permitirá ingresar los datos sin complicaciones.

- **Revisa los valores que introduces**

Introduce números realistas. Por ejemplo, la altura inicial del avión debe ser mayor a la altura del cráter, y los ángulos deben estar entre 0° y 90° . Esto asegura que la simulación funcione correctamente.

- **Experimenta con diferentes parámetros**
Prueba distintas combinaciones de altura, velocidad y ángulo. Esto te ayudará a entender cómo influyen en la trayectoria y cómo interactúa la bomba con el cráter o el suelo.
- **Ten en cuenta las limitaciones del programa**
Recuerda que este programa no incluye factores como la resistencia del aire o el viento, por lo que los resultados son aproximaciones basadas en un modelo ideal.
- **Reinicia la simulación si lo necesitas**
Si quieres probar otros valores, simplemente cierra la gráfica generada y ejecuta el programa de nuevo. Esto te permitirá experimentar fácilmente.
- **Guarda las gráficas generadas**
Las gráficas son excelentes para analizar lo que ocurre en la simulación. Captura imágenes de los resultados para usarlas en reportes o análisis posteriores.
- **Aprende mientras experimentas**
Este programa no solo es útil para hacer simulaciones, sino también para entender cómo funcionan las trayectorias parabólicas y cómo interactúan con el entorno. Úsalo como una herramienta educativa para visualizar conceptos físicos.

7. REFERENCIAS BIBLIOGRAFICAS

- tkinter — Python interface to Tcl/Tk. (s/f). Python documentation. Recuperado el 30 de noviembre de 2024, de <https://docs.python.org/es/3/library/tkinter.html>
- Sumich, A., & Melnikov, E. (2024, abril 12). Biblioteca NumPy en Python. Diveintopython.org. <https://diveintopython.org/es/learn/data-science/numpy>
- Guía definitiva para dominar Matplotlib. (2022, febrero 17). Platzi. <https://platzi.com/blog/matplotlib/>