

Escuela Politécnica Nacional



Informe del Proyecto Bimestral I de Metodos Numéricos

Integrantes: Wellington Barros

Josué Mantuano

Hodalys López

Jonathan Paredes

Proyecto C Sobrevuelo 2

Fecha de entrega: 30 de Noviembre del 2024

1. Introducción

En el presente informe se muestra una simulación visual de la trayectoria de una bomba lanzada desde un avión. Dicha bomba impactara el suelo según los parámetros ingresados por una interfaz gráfica. Para el procesamiento matemático y visual del programa se utilizó la biblioteca matplotlib y para lograr una interfaz gráfica amigable se utilizó tkinter.

El objetivo de la simulación es analizar el movimiento parabólico, estudiando la trayectoria curva que describe un objeto cuando es lanzado desde otro en el cual la fuerza de gravedad será sumamente importante, mediante el uso de campos digitales se combinan la matemática, programación e implementación gráfica en un programa.

2. Metodología

2.1 Descripción de la solución

La solución incluye:

- **Una interfaz gráfica que permite ingresar los parámetros iniciales del problema.**

En el programa se implementa una interfaz donde el usuario puede ingresar los valores que van a servir para el comportamiento de la simulación. Mediante `frame_inputs` se registran los valores dispuestos por el usuario. Además, se implementan campos de texto con sus respectivas etiquetas que permiten diferenciar entre diferentes variables. Todos estos valores son registrados en un método de tabla `grid()` que asignara cada nombre de entrada al widget (cuadro de texto). Por último, para validar los valores introducidos, asegurándose de que cada valor sea numérico, se utiliza el comando `command=iniciar_simulacion`, que está configurado a un botón que inicia el cálculo y la animación.

- **Un cálculo matemático basado en la ecuación de caída libre y movimiento rectilíneo uniforme.**

El objeto en este caso la bomba que se deja caer desde una altura inicial y con una velocidad inicial de cero, logra poder combinar los conceptos matemáticos de la ecuación de caída libre y el Movimiento rectilíneo Uniforme, pues al hacer el análisis matemático, la bomba estará sujeta siempre a la expresión de caída libre $h = \frac{1}{2}gt^2$ y junto al MRU se describe el movimiento de un objeto que cae y se mueve al mismo tiempo, por una velocidad constante.

- **La representación gráfica de la trayectoria de la bomba y el entorno, incluyendo un cañón modelado como un trapecio.**

Para la elaboración de la gráfica de la trayectoria de la bomba y todo el entorno que se muestra se implementó la biblioteca matplotlib. En la cual se generó un espacio donde se ubicó el gráfico, dibujando cada uno de los elementos de la simulación.

El suelo de la pista de aterrizaje está representado con color marrón, luego para dibujar el cañón en forma de trapecio invertido se utilizó `ax.fill` en la cual se usó coordenadas específicas para cada uno de los vértices, en este sentido, el punto superior del trapecio identifica el ancho superior del cañón y la base del trapecio representa el ancho base del cañón. El cañón está representado con color blanco, para que, de la impresión, de que existe un hueco en el camino simulando al cañón. Por otro lado, el avión y la bomba están representados gráficamente como puntos redimensionados en la gráfica, pero con imágenes cargadas con la biblioteca Pillow, para que el estilo visual de la simulación sea más agradable. Al simular el movimiento horizontal del avión, este liberará la bomba al iniciar la simulación, y la trayectoria parabólica de esta será afectada directamente por los parámetros ingresados por el usuario, como la velocidad, haciendo uso de ecuaciones físicas de movimiento. Para animar esto se declaró la función `actualizar_animacion` con lo que se logra la representación visual de la trayectoria del avión y la bomba cuadro por cuadro, dicha función

actualiza las posiciones de estos dos objetos en todos los cuadros de la animación hasta llegar al impacto del suelo de la bomba.

2.2 Desarrollo matemático

El cálculo implementado en el programa para modelar la trayectoria de la bomba lanzada desde el avión se realizó con varias formulas

Para calcular el tiempo de caída de la bomba con la altura H_a del avión hasta el suelo que se encuentra en un nivel de 0 en el plano se utilizo la formula del Tiempo de impacto

$$t_{\text{impacto}} = \sqrt{\frac{2H_a}{g}}$$

En donde:

H_a = altura del avion sobre el nivel del suelo

g = Gravedad en la Tierra aproximadamente de $9.81 \frac{m}{s^2}$

Dicha formula será implementada en el codigo como medio para determinar la duración de la simulación del movimiento de la bomba hasta llegar a tocar el suelo.

Luego para determinar la posición horizontal de la bomba en toda la trayectoria a medida que el tiempo transcurre y debido a que se tendrá un movimiento con una velocidad constante que hará que la bomba cambie constantemente de posición en el eje x se hará uso de la fórmula de x_{bomba}

$$x_{\text{bomba}} = x_{\text{suelta}} + V \cdot t$$

En donde

x_{bomba} = Posicion inicial de la bomba (eje x)

V = Velocidad Contaste del Avion (Parametro de usuario)

t = tiempo desde que la bomba fue lanzada

También se requiere una posición vertical de la bomba para ello se toma en cuenta el tiempo t , y la posición en el eje y que va a corresponder a la altura inicial del avión, parámetro indicado por el usuario, por lo que para indicar la posición vertical de la bomba teniendo en cuenta la gravedad de la tierra se tiene

$$y_{\text{bomba}} = H_a - \frac{1}{2}gt^2$$

En donde

H_a = Altura inicial de la bomba

g = Gravedad de la tierra

t = tiempo transcurrido desde que la bomba fue lanzada

Además, como en toda la trayectoria la bomba estará influenciada por la gravedad de la tierra, esta tendrá en un momento especifico un punto de impacto, en el cual la bomba tocará el suelo, esto se expresa así

$$y_{\text{bomba}} = 0$$

2.3 Diagrama de flujo /pseudocodigo

Inicio del Pseudocódigo

// Procedimiento principal para simular el vuelo de un avión y la caída de una bomba

// Paso 1: Obtener parámetros de entrada:

Leer la velocidad del avión (V)

Leer la altura del avión (Ha)

Leer el ancho superior del cañón

Leer la profundidad del cañón

Leer el ancho de la base del cañón

Leer la distancia desde donde se suelta la bomba desde el borde del cañón

// Paso 2: Verificar que los valores son válidos:

Si cualquier valor es menor o igual a cero, mostrar un mensaje de error y finalizar el procedimiento.

// Paso 3: Calcular la trayectoria de la bomba:

Llamar a la función CalcularTrayectoria(V, Ha, x_suelta)

Obtener la posición de la bomba (x_bomba, y_bomba, t)

// Paso 4: Mostrar la simulación en la ventana gráfica:

Configurar la ventana de simulación y los componentes gráficos.

Mostrar el cañón en el gráfico con las dimensiones proporcionadas.

Cargar las imágenes del avión y la bomba.

// Paso 5: Animar la simulación:

Llamar a la función AnimarAvionYBombas() para mover el avión y mostrar la bomba en su trayecto.

Función CalcularTrayectoria(V, Ha, x_suelta):

// Función para calcular la trayectoria de la bomba en función de la velocidad del avión y la altura

Calcular el tiempo de impacto (t_impacto):

$t_impacto = \sqrt{2 * Ha / g}$ // Usar la fórmula de caída libre

Definir el intervalo de tiempo:

t = intervalo de tiempo desde 0 hasta t_{impacto} (con un número adecuado de puntos)

// Calcular las posiciones de la bomba:

$x_{\text{bomba}} = x_{\text{suelta}} + V * t$

$y_{\text{bomba}} = H_a - (0.5 * g * t^2)$

// Regresar las posiciones (x_{bomba} , y_{bomba} , t) de la bomba.

Retornar (x_{bomba} , y_{bomba} , t)

// Paso 6: Función para animar el movimiento del avión y la caída de la bomba en la simulación

Definir Función AnimarAvionYBombas():

Mover el avión:

Mientras el avión no haya llegado al punto de suelta, mover el avión a una velocidad constante.

Cuando el avión llega al punto de suelta:

Iniciar el cálculo de la posición de la bomba en función del tiempo de caída.

//Paso 7: Actualizar la posición de la bomba:

Cada vez que se actualiza la animación, recalcular la posición (x_{bomba} , y_{bomba}) de la bomba.

Dibujar la bomba en la pantalla:

Dibujar la bomba en la posición calculada en el gráfico.

Repetir la animación hasta que la bomba impacte el suelo.

Fin de la animación: - La animación termina cuando la bomba llega al suelo (cuando y_{bomba} alcanza el valor de 0 o es menor que 0).

Fin del pseudocódigo

Explicación del Pseudocódigo

SimulacionBombaAvion()

- Obtener velocidad (V), altura (H_a) y dimensiones del cañón.
- Verificación de validades de valores mostrando mensajes de error.
- Llamada función CalcularTrayectoria para saber las posiciones de la bomba.

- Dibujar todos los objetos, cañon, avión y bomba en ventana grafica.
- Llamada función AnimarAvionYBombas para mover el avión y bomba hasta llegar al punto de impacto.

CalcularTrayectoria(V, Ha, x_suelta)

- Utilización de la fórmula de caída libre
- Cálculo de posición horizontal y vertical de la bomba

AnimarAvionYBombas()

- Movimiento del avión con velocidad constante desde el punto donde se suelta la bomba
- Caída de la bomba al momento de ser lanzada por el avión en el principio de la animación
- Actualización de la animación dependiendo de la trayectoria calculada hasta llegar al suelo.

2.4 Detalles importantes de implementación

Importación de Bibliotecas:

- Tkinter → Utilizado para la construcción de la interfaz grafica del usuario. Permitiendo el uso de diferentes Widgets como ventanas, botones y cuadros de texto.
- Matplotlib → Creación de los gráficos en 2D y animación en tiempo real como el entorno de la simulación, cañon, avión y bomba.
- Pillow (PIL) → Manejo de imágenes en la ventana grafica.
- NumPy → Cálculos matemáticos.
- Matplotlib.backends.backend_tkagg → Integración de gráficos generados por la biblioteca Matplotlib en la interfaz de Tkinter para poder visualizarlos en una ventana.

Funcion Calcular Trayectoria:

- Cálculo de trayectoria de la bomba al momento de ser lanzada del avión, utilizando formulas que estan totalmente influenciadas por la gravedad.
- Calculo del tiempo de impacto de la bomba.
- Generacion del plano con coordenadas x,y para la trayectoria de la bomba.

Funcion animarAvionYBombas:

- Actualización cuadro por cuadro de la animación del avión y la bomba.
- Avión con velocidad constante.
- Trayectoria de la bomba en caída libre con movimiento dependiente de valores calculados anteriormente.

Generacion del Grafico:

- Fondo marrón para el suelo y blanco para representar la forma de trapecio invertido del cañon ajustando la altura y profundidad.


Integración Grafica Tkinter:

- Cuadros de texto para el ingreso de los parámetros importantes
- Implementación de un botón para el inicio de la simulación y lectura de los datos ingresados por el usuario.
- La interfaz gráfica se actualiza con los datos del grafico implementado por Matplotlib.

3. Resultados

Se establecieron los parámetros para ajustar ciertas condiciones iniciales al comenzar la simulación, se pudo observar que la bomba sigue una trayectoria parabólica, la cual está influenciada por las ecuaciones físicas del movimiento rectilíneo uniforme y la caída libre de la bomba.

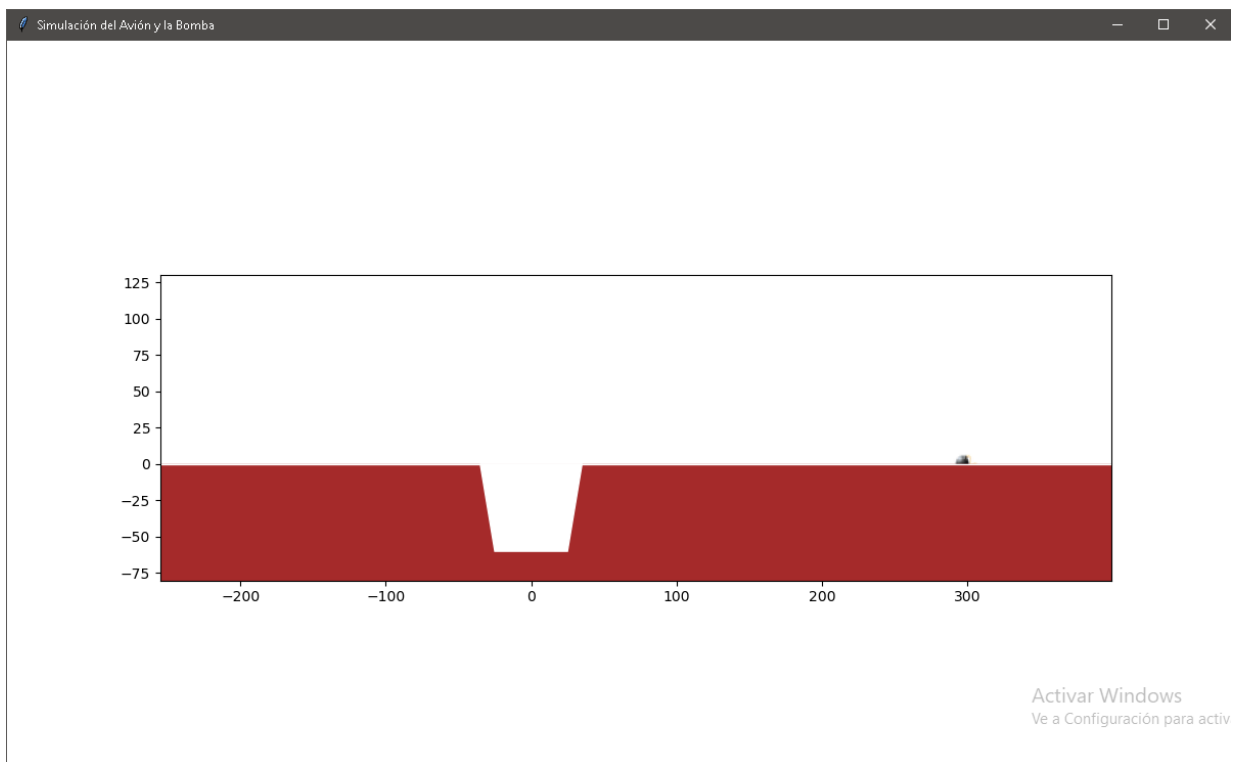
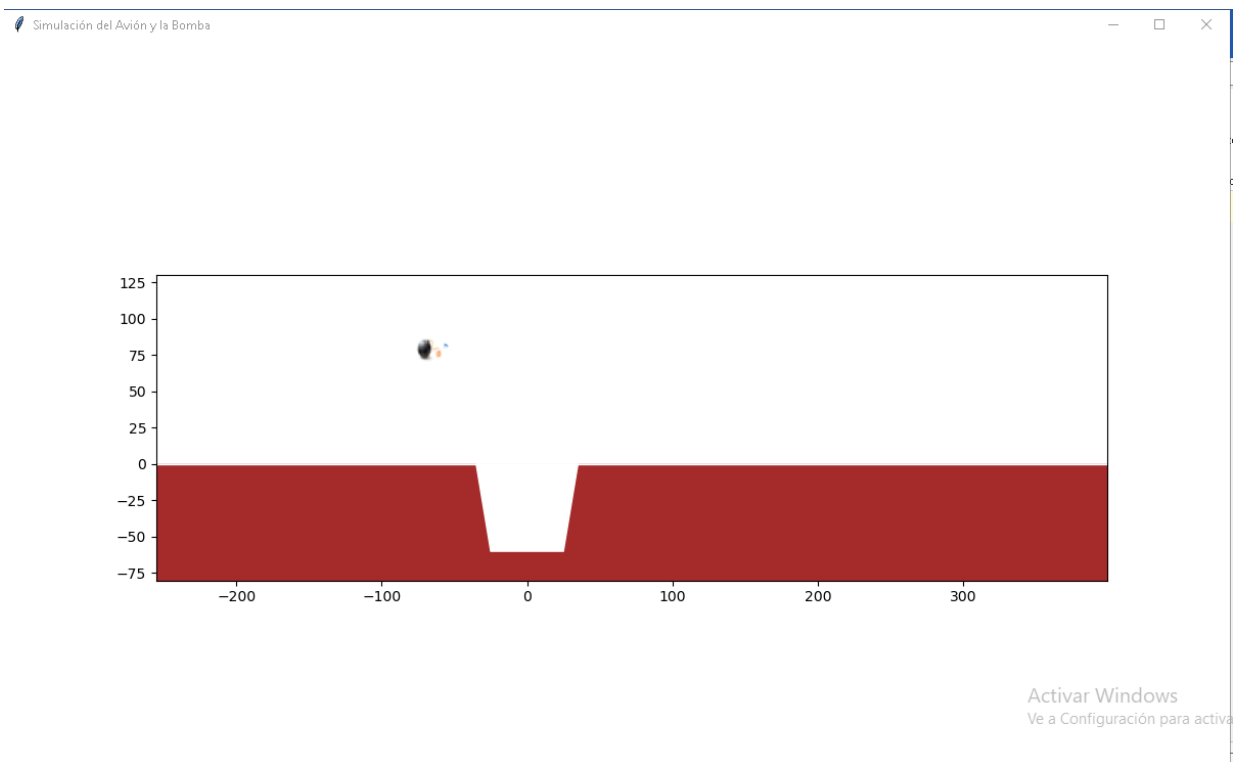
Durante la simulación se puede observar al avión desplazarse horizontalmente desde la izquierda por el plano a una velocidad constante dispuesta en los cuadros de texto. Luego de que el avión suelta la bomba, esta se ve afectada por la gravedad de la tierra, lo que provoca un movimiento curvo simulando una parábola hasta llegar al punto de impacto



The image shows a software window titled "Simulación de Trayectoria" with a dark blue background. Inside, there is a central panel titled "Parámetros de Entrada" (Input Parameters) with a light blue border. This panel contains six input fields, each with a label and a numerical value:

- Velocidad del Avión (V) [m/s]: 100
- Altura del Avión (Ha) [m]: 80
- Ancho Superior del Cañón [m]: 70
- Profundidad del Cañón [m]: 60
- Ancho de la Base del Cañón [m]: 50
- Distancia de Suelta desde el Borde del Cañón [m]: 70

Below the input panel, there is a large green button with the text "Iniciar Simulación" (Start Simulation).



Además, visualmente se pudo lograr una representación del avión y bomba de modo que el programa sea más llevadero para el usuario, también de que el trapecio invertido pintado de blanco da la impresión de que en realidad existe un cañón en el cual la bomba puede caer en algún momento con parámetros específicos.

Esta simulación combina los conceptos gráficos, matemáticos y de programación para que el programa represente claramente el fenómeno físico, comprobando además el buen uso de las ecuaciones de caída libre y MRU para la trayectoria.

4. Conclusiones

- Se pudo comprobar que la simulación de la trayectoria de la bomba lanzada desde el avión implementa conceptos importantes en la física, como el MRU y caída libre para lograr el un efecto parabolico. Junto con una programacion que implementa matemáticamente la serie de instrucciones necesarias para graficar a través de Tkinter y Matplotlib, se implementa de manera precisa y dinámica el movimiento de manera que sea intuitivo y fácil para el usuario el manejo y entendimiento del programa.
- Al utilizar la interfaz gráfica se resaltó la importancia que tienen las diferentes variables en la trayectoria de la bomba, y como esto puede ser representado satisfactoriamente a través de la programación.
- Al calcular el tiempo de impacto de la bomba y determinar la posición vertical y horizontal de esta durante toda la simulación, son factores precisos para representar en tiempo real la animación fluida y coherente en la teoría de la física.
- El manejo adecuado de excepciones al momento del ingreso de los datos previene posibles errores en la simulación y ejecución de programa. Esto mejora al programa es aspectos como la robusticidad y estabilidad.