Método de Euler

Brian David Leiva - 2005 106 31 - fisica.cn28c@slmail.me

6 de septiembre de 2024

Resumen

En el siguiente documento presentamos una implementación del método de Euler para analizar el comportamiento de un proyectil lanzado con una velocidad inicial dada en un medio con fuerza de arrastre asociada.

Introducción

El método de Euler es una técnica numérica simple que nos ayuda a aproximar soluciones a problemas que involucran ecuaciones diferenciales. Aunque no es el método más preciso, es fácil de entender y aplicar, lo que lo hace ideal para este tipo de simulaciones.

En esta investigación, implementamos el método de Euler para calcular la trayectoria de un objeto en caída libre, considerando tanto la gravedad como la resistencia del aire. Nuestro objetivo es mostrar cómo se puede usar este método para simular fenómenos físicos y evaluar su precisión y limitaciones.

Antecedentes

El método de Euler se basa en la idea de aproximar la solución de una ecuación diferencial mediante una serie de pasos discretos.

En el contexto de la simulación de fenómenos físicos, el método de Euler ha sido utilizado debido a su capacidad para proporcionar una primera aproximación rápida y sencilla. En particular, ha sido aplicado en la simulación de movimientos bajo la influencia de fuerzas constantes, como la gravedad, y en la modelación de sistemas dinámicos simples.

Metodología

Utilizamos un python notebook [2] en el cual programamos varias funciones que implementan el método de Euler.

Utilizamos la función proyectil para modelar el sistema.

Esta función toma los parámetros requeridos y les da los valores por defecto que se pidieron.

- b2_m (float): Coeficiente de arrastre dividido por la masa del proyectil.
- g (float): Aceleración debida a la gravedad.
- v_0 (float): Velocidad inicial del proyectil.
- theta (float): Ángulo de lanzamiento del proyectil.
- N (int): Número máximo de iteraciones.
- delta_t (float): Tamaño del paso de tiempo para la simulación.
- stop_fn (función): Función opcional para determinar cuándo detener la simulación.

Luego se inicializan las posiciones (x, y) y las componentes de la velocidad (v_x, v_y) del proyectil. Las velocidades iniciales se calculan a partir de la velocidad inicial y el ángulo de lanzamiento.

La función utiliza un bucle while para iterar hasta que se cumpla la condición de parada definida por stop_fn. En cada iteración, se realizan los siguientes pasos:

1. Se calcula la magnitud de la velocidad a partir de sus componentes.
2. Se actualizan las componentes de la velocidad utilizando las ecuaciones del método de Euler, que consideran la fuerza de arrastre y la gravedad.
3. Se actualizan las posiciones del proyectil.
4. Se almacenan las posiciones y velocidades en listas para su posterior análisis.
5. Se incrementa el contador de iteraciones.

Finalmente la función nos devuelve una tupla con cuatro elementos, las cuales son listas con los valores en x, los valores en y, y las componentes en x y y de la velocidad en cada momento. De esta manera, cada índice i de cada una de las listas corresponde al valor de esa variable en el momento $\Delta t = i$. Es decir $xs[5]$ es el valor de la coordenada x en el tiempo $t = 5$. Esto nos facilita utilizar estos valores con el software de graficado (matplotlib).

Funciones auxiliares

La función v calcula la magnitud de la velocidad a partir de sus componentes, mientras la función es_x_maximo permite detener la simulación cuando el proyectil alcanza su altura máxima. Esta última es una de las posibles funciones de detenido que utilizaremos más adelante para encontrar el alcance horizontal.

Cálculo de la trayectoria

Para calcular la trayectoria utilizamos la función proyectil definida anteriormente. Utilizamos b2_m=0 para el caso sin resistencia del aire y el valor dado por defecto para el caso con resistencia. La función proyectil nos devuelve una tupla con cuatro listas, de las cuales tomamos las dos primeras, es decir, la lista de todas las posiciones en x y la de las posiciones en y.

Encontrando el alcance horizontal

Para calcular el alcance horizontal utilizamos la función de detenido es_x_maximo en el parámetro stop_fn de la función proyectil. Esto detendrá la simulación en el momento en que el valor de y sea igual o menor a cero, devolviendo el último (y máximo) valor de x en xs[-1].

Encontrando el ángulo óptimo para alcance horizontal máximo

Para verificar el ángulo óptimo de 45 grados, definimos la función angulos_a_alcance, la cual devuelve un diccionario cuyas claves son los ángulos y sus valores son los alcances horizontales correspondientes.

La función angulos_a_alcance toma los siguientes parámetros:

- delta_theta (int): Incremento en grados para variar el ángulo de lanzamiento.
- b2_m (float): Coeficiente de arrastre dividido por la masa del proyectil (por defecto es 0, ignorando la resistencia del aire).

Se inicializa un diccionario resultado para almacenar el alcance horizontal máximo para cada ángulo. La variable grados se inicializa en 0 para comenzar desde un ángulo de 0 grados.

La función utiliza un bucle while para iterar sobre diferentes ángulos de lanzamiento, desde 0 hasta 90 grados, incrementando el ángulo en delta_theta grados en cada iteración. En cada iteración, se convierte el ángulo actual de grados a radianes utilizando la función radianes_desde. Luego, se llama a la función proyectil con el ángulo actual y el coeficiente de arrastre especificado. La simulación se detiene cuando el proyectil alcanza su alcance horizontal máximo utilizando la función es_x_maximo. El alcance horizontal máximo (xs[-1]) se almacena en el diccionario resultado con la clave correspondiente al ángulo en grados. Finalmente, se incrementa el ángulo en delta_theta grados para la siguiente iteración.

Para encontrar el ángulo óptimo en este caso, utilizamos la misma estrategia pero esta vez con fuerza de arrastre. Dibujamos líneas verticales tanto en 45 como en el ángulo óptimo para ver la diferencia.

Resultados

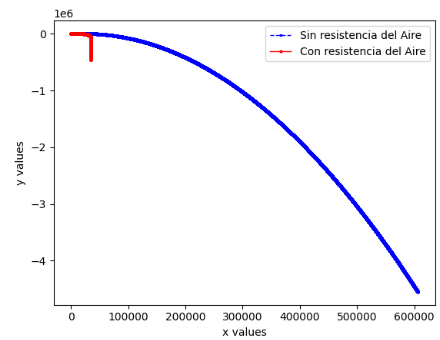

Figura 1:

Como se puede observar en la Figura 1, las trayectorias obtenidas muestran que en el caso con resistencia del aire, el proyectil deja de moverse por completo en la dirección x. Esto se debe a que la resistencia del viento representa una fuerza constante en la dirección -x, llevando la velocidad a cero. Mientras en la dirección y, aunque no se detiene, avanza mucho menos debido a que se establece una velocidad tal que la fuerza de arrastre es equivalente a la fuerza de gravedad, es decir la velocidad terminal.

Al utilizar la función de detenido es_x_maximo, se obtiene el alcance horizontal máximo del proyectil. La gráfica de la trayectoria (Figura 2) confirma que el valor de x en xs[-1] es el alcance máximo.

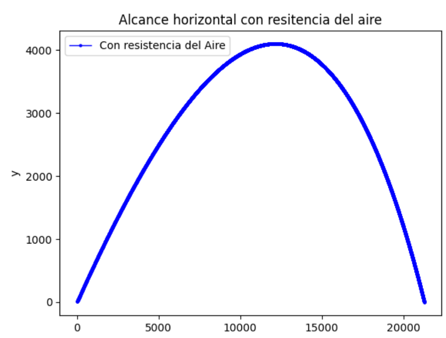

Figura 2:

La función angulos_a_alcance muestra que el ángulo óptimo sin resistencia del aire es de 45 grados. La gráfica del ángulo contra el alcance horizontal (Figura 3) nos confirma este resultado.

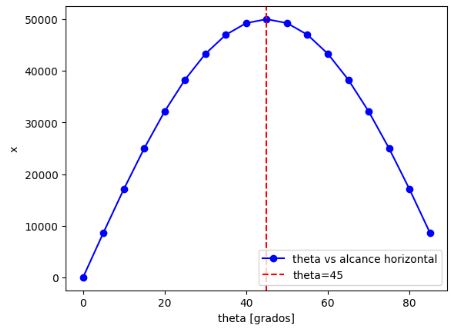

Figura 3:

La figura 4 muestra que el ángulo óptimo con fuerza de arrastre es de 38.5 grados, en contraste con los 45 grados sin resistencia del aire.

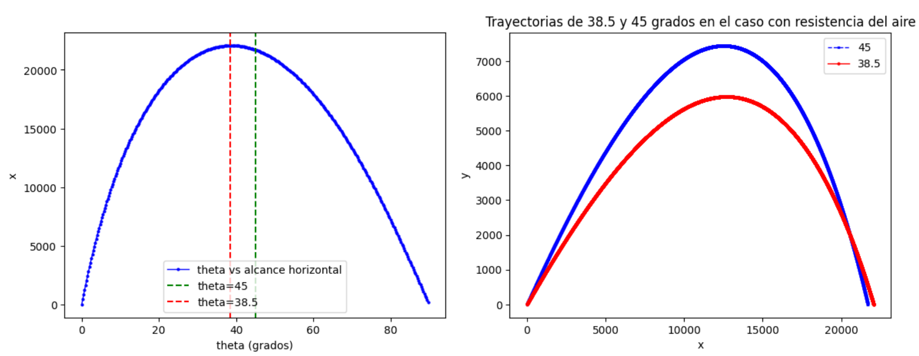

Figura 4:

Discusión

La resistencia del aire tiene un impacto significativo en la trayectoria del proyectil. Observamos que la fuerza de arrastre en la dirección -x reduce la velocidad horizontal a cero, mientras que en la dirección y se alcanza una velocidad terminal donde la fuerza de arrastre equilibra la gravedad.

Mostramos que la función de detenido es_x_maximo es efectiva para determinar el alcance horizontal máximo. Y pudimos realizar la verificación de que el ángulo óptimo de 45 grados sin resistencia del aire es consistente con la teoría.

La resistencia del aire cambia el ángulo óptimo de lanzamiento a 38.5 grados. Este resultado es importante para aplicaciones prácticas donde la resistencia del aire no puede ser ignorada. La comparación de las trayectorias para 45 y 38.5 grados muestra claramente la diferencia en el alcance horizontal debido a la fuerza de arrastre.

Conclusiones

El estudio del tiro parabólico con y sin resistencia del aire nos ha permitido comprender mejor cómo las fuerzas externas afectan la trayectoria de un proyectil. En ausencia de resistencia del aire, el ángulo óptimo de lanzamiento para alcanzar la máxima distancia horizontal es de 45 grados, lo cual es consistente con la teoría clásica del movimiento parabólico. Sin embargo, cuando se introduce la resistencia del aire, el ángulo óptimo de lanzamiento se reduce significativamente. En nuestro caso, encontramos que el ángulo óptimo es de 38.5 grados para un coeficiente de arrastre específico. Esto demuestra que la resistencia del aire no solo reduce la distancia horizontal alcanzada, sino que también modifica el ángulo de lanzamiento necesario para maximizar dicha distancia.

La metodología empleada, basada en el método de Euler y funciones auxiliares para detener la simulación en puntos clave, ha demostrado ser efectiva y precisa para modelar el comportamiento del proyectil bajo diferentes condiciones. Las simulaciones y gráficas obtenidas validan la precisión de nuestro enfoque y proporcionan una herramienta útil para futuros estudios y aplicaciones prácticas.

Además, la comparación entre las trayectorias con y sin resistencia del aire resalta la importancia de considerar factores externos en el diseño y análisis de sistemas de lanzamiento. En aplicaciones prácticas, como el lanzamiento de proyectiles en deportes o en ingeniería militar, es crucial tener en cuenta la resistencia del aire para optimizar el rendimiento y la precisión.

Referencias

[1] Juan Diego Chang. Fisica Computacional.

[2] Brian D. Leiva. El Metodo de Euler.