# Método de Euler

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

August 16, 2024

**Abstract**

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.

## 1 Modelado del sistema

Utilizamos la función proyectil para modelar el sistema. esta función toma los parametros requeridos y les da los valores por defecto que se pidieron.

**Parámetros de la Función** La función ‘proyectil‘ toma los siguientes parámetros:
- ‘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.

**Variables Iniciales** 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.

**Iteración del Método de Euler** 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. **Cálculo de la Velocidad Actual**: Se calcula la magnitud de la velocidad a partir de sus componentes.
2. **Actualización de las Velocidades**: Se actualizan las componentes de la velocidad utilizando las ecuaciones del método de Euler, que consideran la fuerza de arrastre y la gravedad.

3. **Actualización de las Posiciones**: Se actualizan las posiciones del proyectil.

4. **Almacenamiento de Resultados**: Se almacenan las posiciones y velocidades en listas para su posterior análisis.

5. **Incremento del Contador de Iteraciones**: Se incrementa el contador de iteraciones.

**Valores de retorno** 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. La función ‘es_x_maximo‘ permite detener la simulación cuando el proyectil alcanza su altura máxima, esta es una de las posibles funciones de detenido que utilizaremos más adelante para encontrar el alcance horizontal.

## 2 Cáculo de la trayectoria

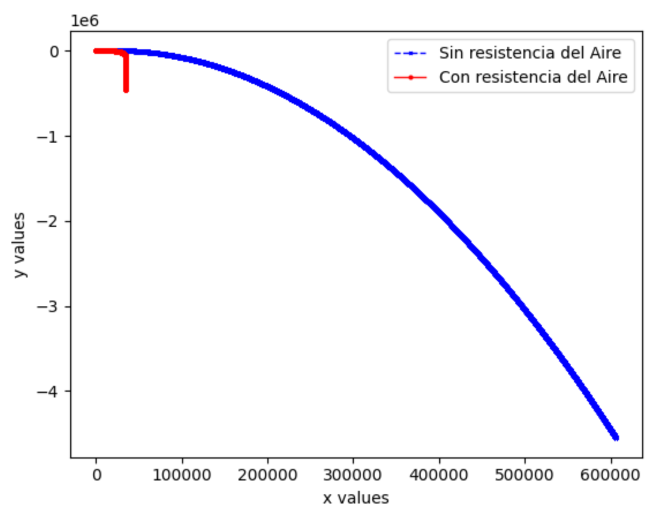


Figure 1:

Para calcular la trayectoria utilizamos la función proyectil definida anteriormente, utilizamos $b_2/m = 0$ 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 $y$.

Las graficamos para obtener la figura 1 en la cual podemos observar que dos de las propiedades más sobresalientes son que la trayectoria con resistencia del aire deja de moverse por completo en la dirección $x$, lo cual podemos explicar porque la resistencia del viento representa una fuerza constante en la dirección $-x$ y termina por llevar la velocidad a cero. Lo segundo es que a pesar de que no se detiene en la dirección $y$ si avanza mucho menos en la misma debido a que se establece en la una velocidad tal que la fuerza de arrastre es equivalente a la fuerza de gravedad. Velocidad que ya conocemos como velocidad terminal.

# 3 Cálculo del Alcance Horizontal

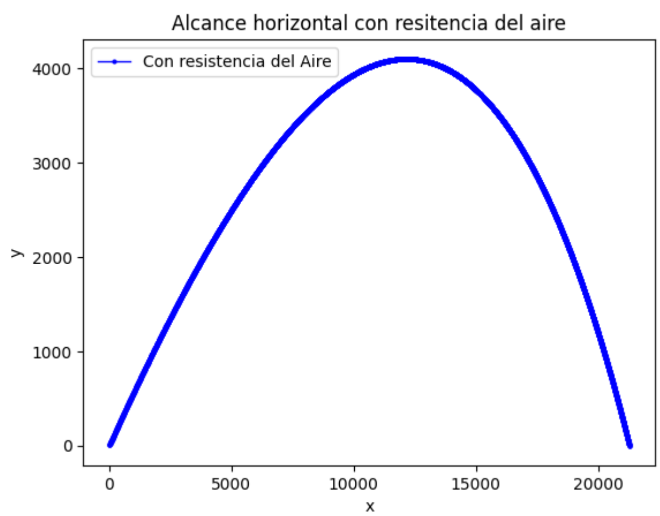


Figure 2:

Para calcular el alcance horizontal utilizamos la función de detenido es_x_maximo en el parámetro stop_fn de la función proyectil. Lo que detendrá la simulación en el momento en que el valor de $y$ sea igual o menor a cero. Lo cual devolverá el último (y máximo) valor de $x$ en xs[-1]. Para verificarlo, graficamos la trayectoria de un tiro de proyectil obteniendo la figura 2.

# 4 Verificación del Ángulo inicial Óptimo sin Fuerza de Arrastre

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

**Parámetros de la Función** 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).

**Variables Iniciales** 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.

**Iteración para Variar el Ángulo** 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.

Utilizamos esta función sin resistencia del aire, graficamos el resultado, es decir el ángulo contra el alcance horizontal correspondiente al mismo y dibujamos una línea vertical en $\theta = 45$. Obteniendo así la figura 3.

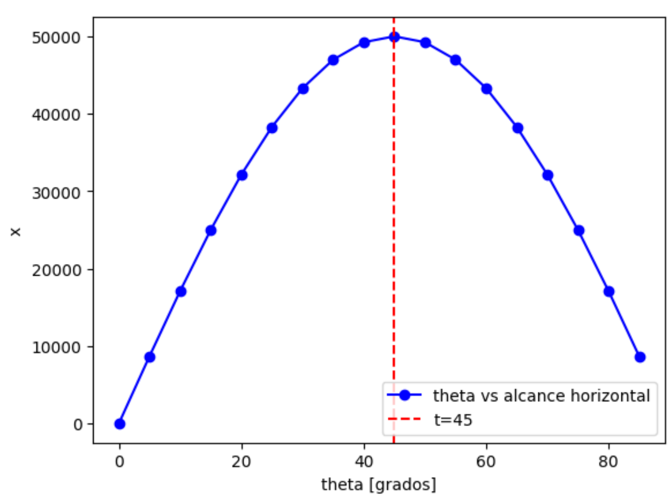


Figure 3:

# 5 Encontrando el Ángulo Óptimo con Fuerza de Arrastre

Para encontrar el ángulo óptimo en este caso sencillamente 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. El resultado es la figura 4. Que nos muestra a la izquierda el ángulo vs el alcance horizontal para cada variación de 0.25 grados y a la derecha las trayectorias para 45 y 38.5 grados, el cuál es el ángulo óptimo para una fuerza de arrastre de $\frac{b_2}{m} = 0.00004$.

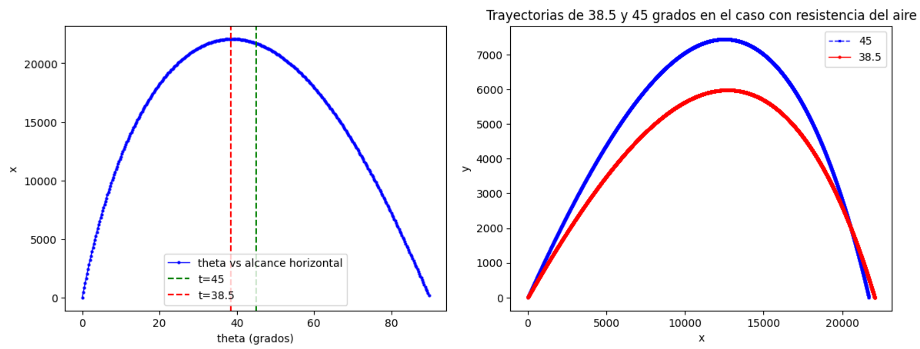


Figure 4: