Universidad Alfonso X el Sabio
Martina García González, Lucía Mielgo Torres
Grado en Ingeniería Matemática
AMPLIACIÓN DE LOS MÉTODOS NUMÉRICOS

Taller 10: Sistemas de ecuaciones de orden superior

28 de enero de 2025

1. Introducción

1. El lanzamiento de un proyectil se puede modelar a partir de las leyes de Newton. Suponiendo que existe un coeficiente de rozamiento b, se pueden aproximar las posiciones x y y del proyectil como:

$$x'' = -bx'$$

$$y'' = -9{,}8 - by'$$

   Suponga que el proyectil se lanza desde una altura de 50 metros con una rapidez inicial de 100 m/s y un ángulo de lanzamiento de 30°. De este modo, las condiciones iniciales vienen dadas como: $x(0) = 0, \quad y(0) = 50, \quad x'(0) = 100\cos(30°), \quad y'(0) = 100\sin(30°)$. Aproxime la solución utilizando el método de Euler para el intervalo $t \in [0, 12]$ s utilizando un paso de $h = 0{,}1$. Tome el valor de $b = 0{,}04$. Grafique tanto las posiciones x e y, como las velocidades en cada dirección en función del tiempo. Además, responda a las siguientes preguntas:

   a) Altura máxima que alcanza el proyectil.

   b) Distancia máxima que recorre el proyectil.

2. Repita el ejercicio anterior utilizando el método Runge-Kutta de orden 4. Analice los resultados para distintos pasos: $h = 1$ y $h = 0{,}1$. Compare los resultados con los obtenidos en el problema 1.

2. Métodos y modelos matemáticos

Un sistema de ecuaciones diferenciales de orden superior puede ser reducido a un sistema equivalente de primer orden mediante la introducción de variables auxiliares.

Supongamos que tenemos un sistema de ecuaciones diferenciales de segundo orden de la forma:

$$x'' = f_1(x, y, x', y', t),$$
$$y'' = f_2(x, y, x', y', t).$$

Definimos las siguientes variables auxiliares para reducir el orden del sistema:

$$\begin{aligned} v_x &= x', \quad \text{(la velocidad en la dirección } x\text{)}, \\ v_y &= y', \quad \text{(la velocidad en la dirección } y\text{)}. \end{aligned}$$

Entonces, las ecuaciones originales se convierten en un sistema de cuatro ecuaciones de primer orden:

$$\begin{aligned} x' &= v_x, \\ y' &= v_y, \\ v_x' &= f_1(x, y, v_x, v_y, t), \\ v_y' &= f_2(x, y, v_x, v_y, t). \end{aligned}$$

Reducir el orden del sistema nos permitirá poder resolverlos mediante el uso de métodos numéricos diseñados para sistemas de primer orden, como Euler y Runge-Kutta. **Método de Euler**

El método de Euler es una técnica numérica para resolver ecuaciones diferenciales ordinarias de la forma:

$$\frac{dy}{dt} = f(t, y) \tag{1}$$

con una condición inicial $y(t_0) = y_0$. El método de Euler utiliza la derivada para aproximar la solución de la ecuación diferencial mediante una secuencia de pasos discretos. A partir de un valor inicial conocido, se calcula el siguiente valor de la función utilizando la fórmula:

$$y_{n+1} = y_n + hf(t_n, y_n) \tag{2}$$

donde:

- h es el tamaño del paso,
- $t_n = t_0 + nh$ es el tiempo en el paso n,
- y_n es la aproximación numérica de la solución en el tiempo t_n.

El método de Euler es explícito, lo que significa que el nuevo valor y_{n+1} se calcula directamente a partir del valor actual y_n y la pendiente $f(t_n, y_n)$. Este método es sencillo de implementar, pero puede ser ineficiente para obtener soluciones precisas cuando el tamaño de paso h es grande, ya que introduce errores numéricos acumulativos. Sin embargo, al reducir el valor de h, se mejora la precisión de la solución a costa de un mayor número de cálculos.

Método de Runge-Kutta de Orden 4

El método de Runge-Kutta de orden 4 (RK4) es un método numérico utilizado para resolver ecuaciones diferenciales ordinarias de la forma $\frac{dy}{dt} = f(t, y)$

Dado un intervalo $[t_0, t_f]$ y un valor inicial $y(t_0) = y_0$, el método RK4 calcula una aproximación de $y(t_f)$ siguiendo estos pasos:

- Definir el tamaño del paso h:

  El intervalo se divide en N pasos, donde h se calcula como:

$$h = \frac{t_f - t_0}{N} \tag{3}$$

- Iterar para cada paso:

  Para cada paso i desde 0 hasta $N - 1$:

  - Calcular el valor actual de t:

$$t_i = t_0 + i \cdot h \tag{4}$$

  - Calcular las pendientes:

$$k_1 = h \cdot f(t_i, y_i) \tag{5}$$

$$k_2 = h \cdot f\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \tag{6}$$

$$k_3 = h \cdot f\left(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \tag{7}$$

$$k_4 = h \cdot f(t_i + h, y_i + k_3) \tag{8}$$

- Actualizar el valor de y:

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{9}$$

- Al finalizar la iteración, el valor y_N será la aproximación de la solución $y(t_f)$.

3. Resultados

El problema describe el movimiento de un proyectil con resistencia al aire, modelado por el siguiente sistema de ecuaciones diferenciales de segundo orden:

$$x'' = -bx',$$

$$y'' = -9{,}8 - by',$$

donde:

- $x(t)$ y $y(t)$ son las posiciones en las direcciones horizontal y vertical, respectivamente.
- b es el coeficiente de rozamiento.
- $g = 9{,}8\,\mathrm{m/s^2}$ es la aceleración debida a la gravedad.

Las condiciones iniciales son:

$$x(0) = 0, \quad y(0) = 50, \quad x'(0) = v_0 \cos(30°), \quad y'(0) = v_0 \sin(30°),$$

con $v_0 = 100\,\mathrm{m/s}$ como la velocidad inicial.

Para resolver este sistema, reducimos las ecuaciones de segundo orden a un sistema de ecuaciones diferenciales de primer orden mediante la introducción de variables auxiliares.

Soluciones analítica

4. Solución Analítica y Discusión

Ecuaciones para $x(t)$ e $y(t)$

La trayectoria del proyectil se rige por el siguiente sistema de ecuaciones diferenciales:

$$x'' = -bx', \quad y'' = -g - by',$$

y las condiciones iniciales:

$$x(0) = 0, \quad y(0) = 50, \quad x'(0) = v_0 \cos(30°), \quad y'(0) = v_0 \sin(30°).$$

Cálculo de $x(t)$ La ecuación para $x(t)$ se deriva integrando la ecuación diferencial:

$$x'' = -bx'.$$

Primera integración:

$$x'(t) = x'(0)e^{-bt},$$

donde $x'(0) = v_0 \cos(30°)$.

Segunda integración:

$$x(t) = \frac{x'(0)}{b}\left(1 - e^{-bt}\right),$$

donde $x(0) = 0$. Sustituyendo los valores iniciales:

$$x(t) = \frac{v_0 \cos(30°)}{b}\left(1 - e^{-bt}\right).$$

Cálculo de $y(t)$ De manera similar, para $y(t)$:

$$y'' = -g - by'.$$

Primera integración:

$$y'(t) = \left(y'(0) + \frac{g}{b}\right) e^{-bt} - \frac{g}{b},$$

donde $y'(0) = v_0 \sin(30°)$.

Segunda integración:

$$y(t) = \frac{y'(0) + \frac{g}{b}}{b} \left(1 - e^{-bt}\right) - \frac{g}{b}t + y(0),$$

donde $y(0) = 50$. Sustituyendo los valores iniciales:

$$y(t) = \frac{v_0 \sin(30°) + \frac{g}{b}}{b} \left(1 - e^{-bt}\right) - \frac{g}{b}t + 50.$$

Cálculo Analítico de Altura y Distancia Máxima

Altura Máxima

La altura máxima se alcanza cuando $y'(t) = 0$. Resolviendo esta condición, el tiempo se calcula como:

$$t_{\text{máx}} = \frac{1}{b} \ln\left(\frac{g + bv_0 \sin(30°)}{g}\right).$$

Sustituyendo los valores ($b = 0{,}04$, $v_0 = 100\,\text{m/s}$, $g = 9{,}8\,\text{m/s}^2$, $y_0 = 50\,\text{m}$):

$$t_{\text{máx}} = 4{,}64\,\text{s}.$$

La altura máxima se obtiene sustituyendo $t_{\text{máx}}$ en $y(t)$:

$$y_{\text{máx}} = 162{,}48\,\text{m}.$$

Distancia Máxima

La distancia máxima se alcanza cuando el proyectil toca el suelo, es decir, $y(t) = 0$. Resolviendo $y(t) = 0$ para t, se encuentra el tiempo de impacto:

$$t_{\text{impacto}} = 10{,}64\,\text{s}.$$

La distancia horizontal máxima se calcula sustituyendo t_{impacto} en $x(t)$:

$$x_{\text{máx}} = 750{,}61\,\text{m}.$$

Interpretación Física de la Trayectoria

El proyectil sigue una trayectoria modificada debido al rozamiento con el aire, representado por el coeficiente b. Este rozamiento desacelera tanto el movimiento horizontal como el vertical.

- La altura máxima de $162{,}48\,\text{m}$ ocurre en $t_{\text{máx}} = 4{,}64\,\text{s}$.

- La distancia máxima horizontal de $750{,}61\,\text{m}$ se alcanza en $t_{\text{impacto}} = 10{,}64\,\text{s}$, cuando el proyectil toca el suelo.

Gráfica de la Trayectoria

La siguiente gráfica muestra la trayectoria del proyectil calculada analíticamente, recortada en $y = 0$:

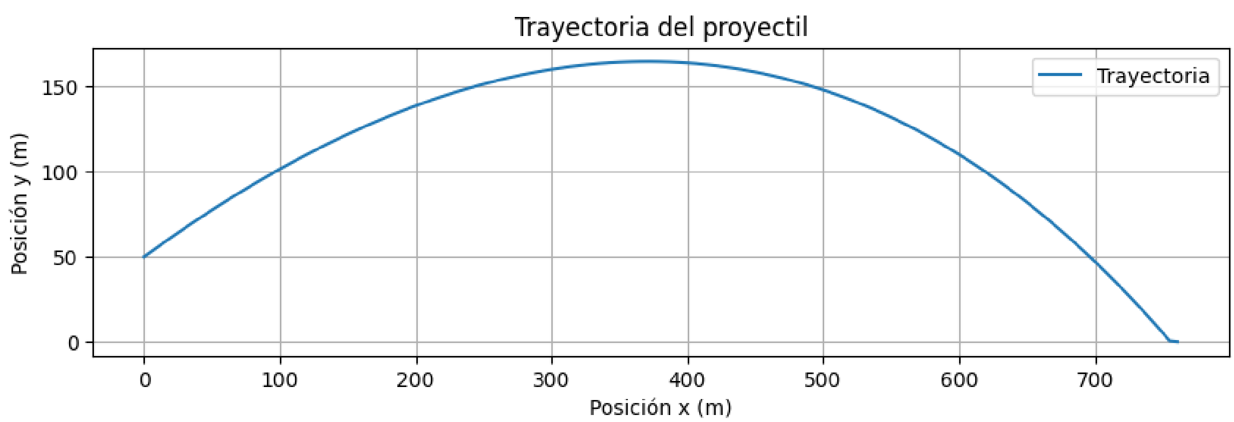

Figura 1: Trayectoria del proyectil según la solución analítica.

Definición de variables auxiliares

Definimos las velocidades en las direcciones x y y y como:

$$v_x = x', \quad v_y = y'.$$

Por lo tanto:

$$x'' = v'_x, \quad y'' = v'_y.$$

El sistema original queda transformado en:

$$x' = v_x,$$

$$y' = v_y,$$

$$v'_x = -bv_x,$$

$$v'_y = -9{,}8 - bv_y.$$

Sistema resultante

El sistema ahora consta de cuatro ecuaciones diferenciales de primer orden, sustituyendo b=0,04:

$$\begin{aligned} x' &= v_x, \\ y' &= v_y, \\ v'_x &= -0,04v_x, \\ v'_y &= -9{,}8 - 0,04v_y. \end{aligned}$$

Condiciones iniciales

Las condiciones iniciales se transforman como:

$$x(0) = 0, \quad y(0) = 50, \quad v_x(0) = v_0 \cos(30^\circ) \approx 86{,}60 m/s, \quad v_y(0) = v_0 \sin(30^\circ) = 50 m/s,$$

Estos gráficos muestran los resultados obtenidos mediante el método de Euler con paso h=0,1, nos sale que la altura máxima del proyectil es de 164,77m y la distancia máxima de 760,70m.

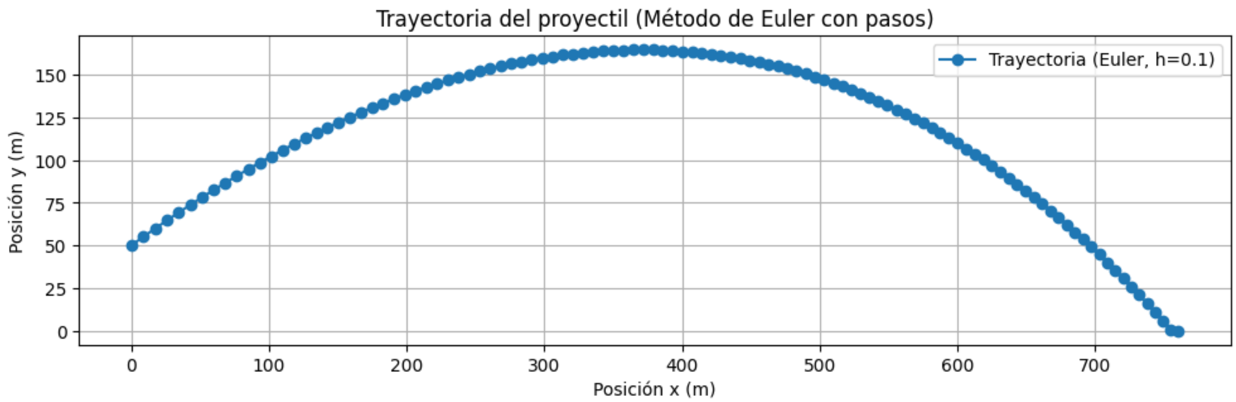

Figura 2: Trayectoria del proyectil con Euler h=0.1

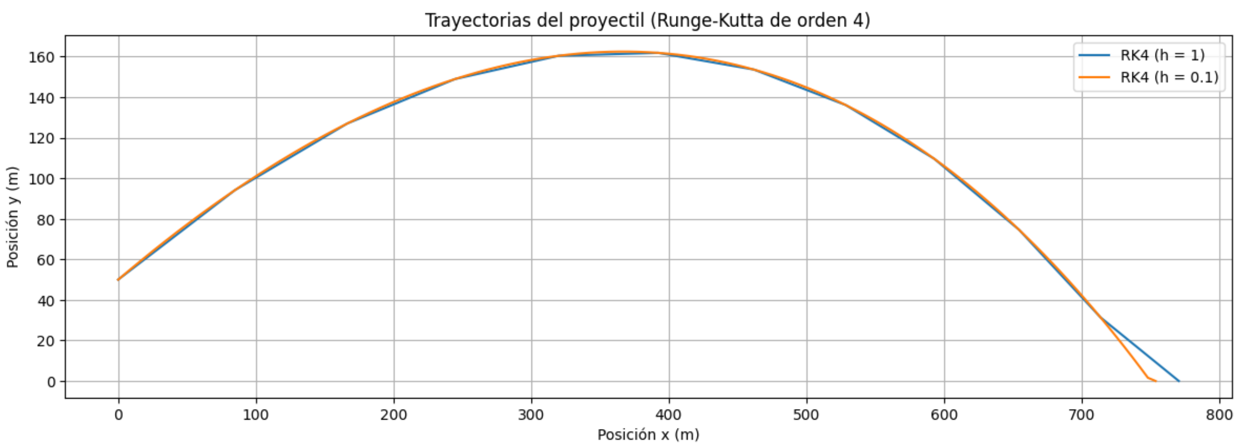

Figura 3: Trayectoria del proyectil con RK-4, para h=1 y h=0,1

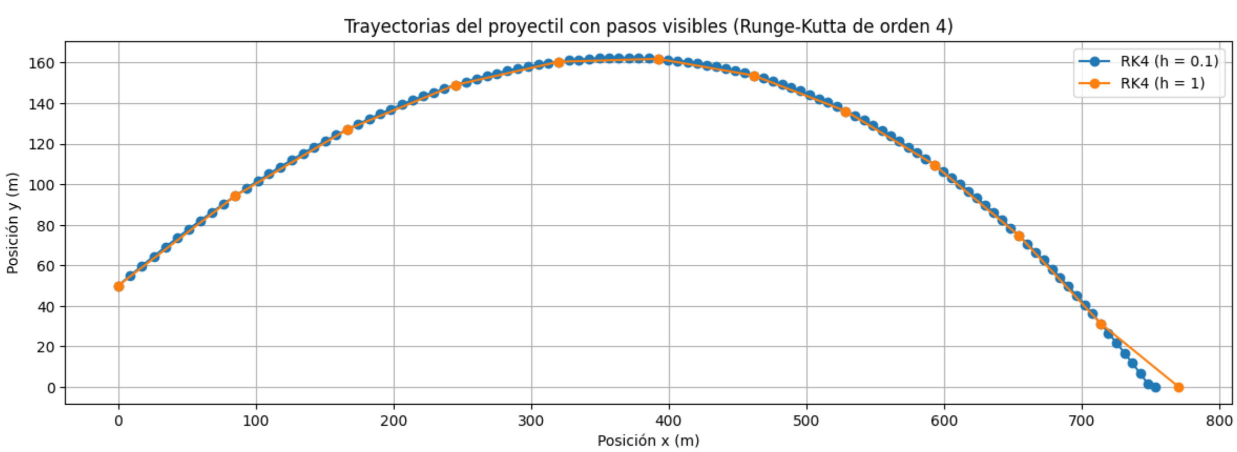

Figura 4: Trayectoria del proyectil con RK-4, con pasos

Estos gráficos muestran los resultados obtenidos mediante el método de Runge-Kutta de Orden 4, nos sale que con h=1, la altura máxima del proyectil es de 161,86 m y la distancia máxima de 770,68 m. Para h=0,1 la altura máxima nos da que es de 162,47 m y la distancia máxima de 753,85 m.

5. Discusión

Método de Euler

Los resultados obtenidos con el método de Euler ($h = 0{,}1$) se comparan con la solución analítica en la siguiente tabla:

Método	Altura Máxima (m)	Error Altura (m)	Distancia Máxima (m)	Error Distancia (m)
Solución Analítica	162.48	-	750.61	-
Euler ($h = 0{,}1$)	164.77	2.29	760.70	10.09

Cuadro 1: Resultados y errores para el método de Euler.

El método de Euler es sencillo y fácil de implementar, pero introduce errores relativos significativos incluso con pasos pequeños ($h = 0{,}1$). En este caso:

- El error absoluto en la altura máxima es de 2,29 m.
- El error absoluto en la distancia máxima es de 10,09 m.

Esto demuestra que el método de Euler es adecuado para cálculos rápidos o estimaciones iniciales.

Método de Runge-Kutta (RK4)

Los resultados obtenidos con el método de Runge-Kutta de orden 4 (RK4) para diferentes pasos ($h = 1$ y $h = 0{,}1$) se presentan en la siguiente tabla:

Método	Altura Máxima (m)	Error Altura (m)	Distancia Máxima (m)	Error Distancia (m)
Solución Analítica	162.48	-	750.61	-
RK4 ($h = 1$)	161.86	0.62	770.68	20.07
RK4 ($h = 0{,}1$)	162.47	0.01	753.85	3.24

Cuadro 2: Resultados y errores para el método de Runge-Kutta (RK4).

El método de Runge-Kutta de orden 4 (RK4) ofrece una mayor precisión en comparación con Euler, especialmente cuando se utiliza un paso pequeño ($h = 0{,}1$). En este caso:

- Con $h = 0{,}1$, el error absoluto en la altura máxima es de $0{,}01\,\mathrm{m}$ y en la distancia máxima es de $3{,}24\,\mathrm{m}$.
- Incluso con un paso grande ($h = 1$), el error absoluto en la altura máxima es de $0{,}62\,\mathrm{m}$ y en la distancia máxima es de $20{,}07\,\mathrm{m}$.

Esto muestra que RK4 es robusto incluso con pasos más grandes, lo que lo hace más eficiente y confiable para problemas complejos.

6. Conclusiones

1. **Orden de los errores:**
    - El método de Euler es de primer orden, con un error global de $O(h)$. Para mejorar la precisión, se requieren pasos más pequeños, lo que aumenta el número de iteraciones y el costo computacional.
    - El método Runge-Kutta de orden 4 (RK4) es de cuarto orden, con un error global de $O(h^4)$. Esto le permite alcanzar una mayor precisión utilizando pasos más grandes, siendo más eficiente en problemas que exigen alta exactitud.
2. **Precisión y estabilidad:**

    RK4 es mucho más preciso que Euler para un mismo tamaño de paso. Esto lo hace adecuado para problemas complejos, como trayectorias no lineales o simulaciones físicas.
3. **Costo computacional:**
    - Euler es más sencillo computacionalmente, ya que solo requiere calcular una pendiente por paso. Esto lo hace útil para cálculos rápidos y problemas simples.
    - RK4, aunque más costoso debido al cálculo de cuatro pendientes (k_1, k_2, k_3, k_4) por paso, compensa este esfuerzo al permitir el uso de pasos más grandes sin comprometer la precisión.
4. **Elección del método:**
    - El método de Euler es ideal para obtener soluciones rápidas o aproximaciones iniciales en problemas simples. Sin embargo, su acumulación de errores lo hace

menos adecuado para simulaciones largas o problemas donde la precisión es crítica.

- RK4 es la mejor opción en situaciones donde se necesita alta precisión y estabilidad, como en trayectorias balísticas o sistemas dinámicos complejos.

7. Referencias

Douglas J. Faires and Richard L. Burden, *Análisis Numérico*, Cengage Learning, 1998.

8. Anexos

Las funciones utilizadas para realizar los ejercicios son las siguientes.

Función del Método de Euler

```
def euler_method(h, n_steps, vx, vy, x, y):
    for i in range(1, n_steps):
    vx[i] = vx[i - 1] - h * b * vx[i - 1]
    vy[i] = vy[i - 1] - h * (g + b * vy[i - 1])
    x[i] = x[i - 1] + h * vx[i - 1]
    y[i] = y[i - 1] + h * vy[i - 1]

    # Detener si el proyectil toca el suelo
    if y[i] < 0:
        y[i] = 0
        break
```

Función del Método de Runge-Kutta

```
def runge_kutta_4(h, t_max):
    n_steps = int(t_max / h) + 1
    t = np.linspace(0, t_max, n_steps)
    x = np.zeros(n_steps)
    y = np.zeros(n_steps)
    vx = np.zeros(n_steps)
    vy = np.zeros(n_steps)
```

```
# Condiciones iniciales
x[0], y[0], vx[0], vy[0] = x0, y0, vx0, vy0

for i in range(1, n_steps):
    k1_x, k1_y, k1_vx, k1_vy = sistema(t[i - 1], x[i - 1], y[i - 1], vx[i - 1], vy[i
    k2_x, k2_y, k2_vx, k2_vy = sistema(
        t[i - 1] + h / 2,
        x[i - 1] + h / 2 * k1_x,
        y[i - 1] + h / 2 * k1_y,
        vx[i - 1] + h / 2 * k1_vx,
        vy[i - 1] + h / 2 * k1_vy,
    )
    k3_x, k3_y, k3_vx, k3_vy = sistema(
        t[i - 1] + h / 2,
        x[i - 1] + h / 2 * k2_x,
        y[i - 1] + h / 2 * k2_y,
        vx[i - 1] + h / 2 * k2_vx,
        vy[i - 1] + h / 2 * k2_vy,
    )
    k4_x, k4_y, k4_vx, k4_vy = sistema(
        t[i - 1] + h,
        x[i - 1] + h * k3_x,
        y[i - 1] + h * k3_y,
        vx[i - 1] + h * k3_vx,
        vy[i - 1] + h * k3_vy,
    )

    x[i] = x[i - 1] + h / 6 * (k1_x + 2 * k2_x + 2 * k3_x + k4_x)
    y[i] = y[i - 1] + h / 6 * (k1_y + 2 * k2_y + 2 * k3_y + k4_y)
    vx[i] = vx[i - 1] + h / 6 * (k1_vx + 2 * k2_vx + 2 * k3_vx + k4_vx)
    vy[i] = vy[i - 1] + h / 6 * (k1_vy + 2 * k2_vy + 2 * k3_vy + k4_vy)

    # Detener si el proyectil toca el suelo
    if y[i] < 0:
        y[i] = 0
```

```
        return t[:i+1], x[:i+1], y[:i+1], vx[:i+1], vy[:i+1]

return t, x, y, vx, vy
```