1. Introducción

1. Desarrolle una función en Python para calcular la integral de una ecuación $\frac{dy}{dt} = f(t)$ donde $t \in [t_0, t_f]$ con $y(t_0) = y_0$ mediante el método de Heun, dando como variables $f(t)$, t_0, t_f, y_0.

2. Implemente computacionalmente un programa que resuelva la ecuación diferencial $\frac{dy}{dt} = 4e^{0.8t} - 0.5y$, donde $t \in [0, 4]$, $y(0) = 2$, mediante el método del trapecio utilizando los números de paso de $h = 0.1$, $h = 0.01$ y $h = 0.001$. Compare los resultados con la solución analítica y el método de Euler.

2. Métodos y modelos matemáticos

El método de Heun es una técnica de integración numérica utilizada para resolver ecuaciones diferenciales ordinarias de la forma:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0 \tag{1}$$

Es una mejora del método de Euler, ya que utiliza una aproximación más precisa del incremento en cada paso.

El método se basa en dos etapas:

1. Primero, se calcula una predicción del valor de y en el siguiente paso usando el método de Euler:

   $$y_{\text{pred}} = y_n + hf(t_n, y_n) \tag{2}$$

   donde h es el tamaño del paso, t_n es el tiempo actual, y y_n es el valor de la solución en t_n.

2. Luego, se ajusta este valor calculando el promedio entre la pendiente en t_n y la pendiente en $t_n + h$:

   $$y_{n+1} = y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_n + h, y_{\text{pred}})\right) \tag{3}$$

$$\frac{dy}{dt} = 4e^{0.8t} - 0.5y$$

con la condición inicial $y(0) = 2$, utilizando los métodos numéricos de Euler y Heun, y se compara con la solución analítica de la ecuación.

Solución Analítica

La solución analítica de la ecuación diferencial es:

$$y(t) = \frac{4}{1.3}\left(e^{0.8t} - e^{-0.5t}\right) + 2e^{-0.5t}$$

Resultados estimados con métodos Heun y Euler

A continuación se presenta una tabla comparativa de los valores obtenidos para las primeras 5 iteraciones, utilizando un tamaño de paso de $h = 0.1, 0.001, 0.0001$, con el método de Euler, el método de Heun y la solución analítica.

Table 1: Comparación de los Métodos para $h = 0.1$

t	Solución Analítica	Método de Euler	Método de Heun
0.000000	2.000000	2.000000	2.000000
0.100000	2.308790	2.300000	2.309157
0.200000	2.636362	2.618315	2.637113
0.300000	2.984620	2.956803	2.985770
0.400000	3.355606	3.317463	3.357177

Table 2: Comparación de los Métodos para $h = 0.01$

t	Solución Analítica	Método de Euler	Método de Heun
0.000000	2.000000	2.000000	2.000000
0.010000	2.023089	2.023150	2.023089
0.020000	2.046479	2.046620	2.046481
0.030000	2.070174	2.070406	2.070177
0.040000	2.094182	2.094516	2.094187

Table 3: Comparación de los Métodos para $h = 0.001$

t	Solución Analítica	Método de Euler	Método de Heun
0.000000	2.000000	2.000000	2.000000
0.001000	2.002308	2.002308	2.002308
0.002000	2.004621	2.004621	2.004621
0.003000	2.006939	2.006939	2.006939
0.004000	2.009261	2.009261	2.009261

4. Discusión

El método de Heun obtiene una mejor aproximación que el método de Euler, como se observa en las tablas. Esto se debe a que Heun promedia la pendiente en los puntos consecutivos, lo que corrige la predicción inicial y mejora la precisión en comparación con Euler.

Sin embargo, cuando el tamaño del paso h h es muy pequeño, la diferencia entre ambos métodos se reduce significativamente. Esto ocurre porque, con pasos más pequeños, la pendiente no varía tanto entre los puntos consecutivos, por lo que la predicción de Euler se aproxima bastante a la realidad.

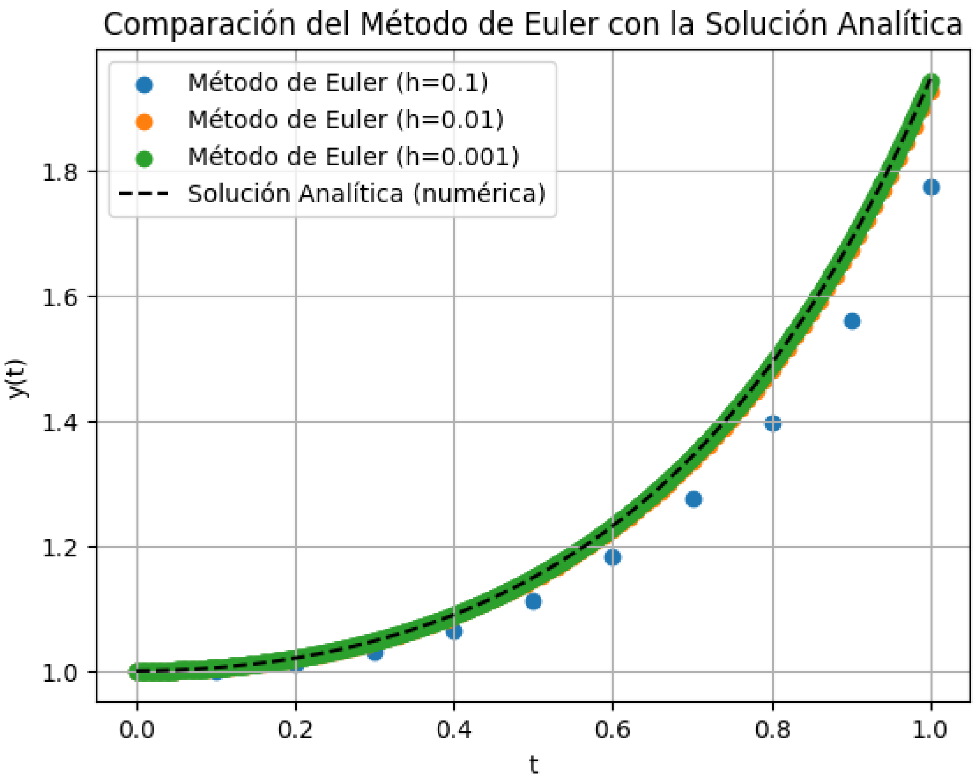
En resumen, mientras que Heun es superior para pasos grandes, en pasos muy pequeños ambos métodos tienden a dar resultados muy similares, ya que ambos siguen de cerca la pendiente real de la solución en cada iteración.

Representación gráfica

Para visualizar las diferencias, se ha representado la solución analítica junto con las aproximaciones de los métodos de Euler y Heun . Se observa que Heun sigue más de cerca la solución exacta.

5. Conclusiones

En resumen, el método de Heun es más preciso que el de Euler, ya que promedia la pendiente en los puntos consecutivos, corrigiendo los errores y proporcionando una aproximación más cercana a la solución analítica.



Sin embargo, cuando hacemos el h muy pequeño, aunque los resultados de ambos métodos se vuelven muy similares, surge otro problema: el riesgo de que error de redondeo se dispare. Esto sucede porque, al reducir el tamaño del paso, el número de iteraciones aumenta considerablemente, y aunque cada paso tenga un error pequeño, estos errores pueden sumarse y, debido a las limitaciones numéricas de los cálculos, llevar a una pérdida de precisión.

6. Referencias

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

7. Anexos

Las funciones utilizadas para realizar los ejercicios son las siguientes.

Función del Método de Heun

```
def heun(f, t0, tf, y0, h):
    n = int((tf - t0) / h)  # Número de pasos
    y = y0
    t = t0

    # Listas para guardar los valores de t e y
    t_values = [t0]
    y_values = [y0]

    # Bucle para realizar los pasos del método de Heun
    for _ in range(n):
        y_euler = y + h * f(t, y)  # Paso de predicción (Euler)
        y = y + h / 2 * (f(t, y) + f(t + h, y_euler))  # Paso de corrección (Heun)
        t = t + h
        t_values.append(t)
        y_values.append(y)

    return t_values, y_values
```

Función del Método de Euler

```
def euler(f, t0, tf, y0, h):
    n = int((tf - t0) / h)  # Número de pasos
    y = y0
    t = t0

    # Listas para guardar los valores de t e y
    t_values = [t0]
    y_values = [y0]

    # Bucle para realizar los pasos del método de Euler
    for _ in range(n):
        y = y + h * f(t, y)  # Actualización de Euler
        t = t + h            # Avanzamos el tiempo
```

```
        t_values.append(t)
        y_values.append(y)

    return t_values, y_values
```