



ESCUELA POLITÉCNICA NACIONAL

# MÉTODOS NUMÉRICOS

JIMÉNEZ JARAMILLO YASID GABRIEL

# [Tarea 12] Ejercicios Unidad 05-A | ODE Método de Euler

```
%load_ext autoreload
import numpy as np
import math
from src import ODE_euler, graphics, ODE_euler_nth
```

The autoreload extension is already loaded. To reload it, use:  
%reload\_ext autoreload

---

## EJERCICIO TRES

Utilice el método de Euler para aproximar las soluciones para cada uno de los siguientes problemas de valor inicial.

### PARTE A

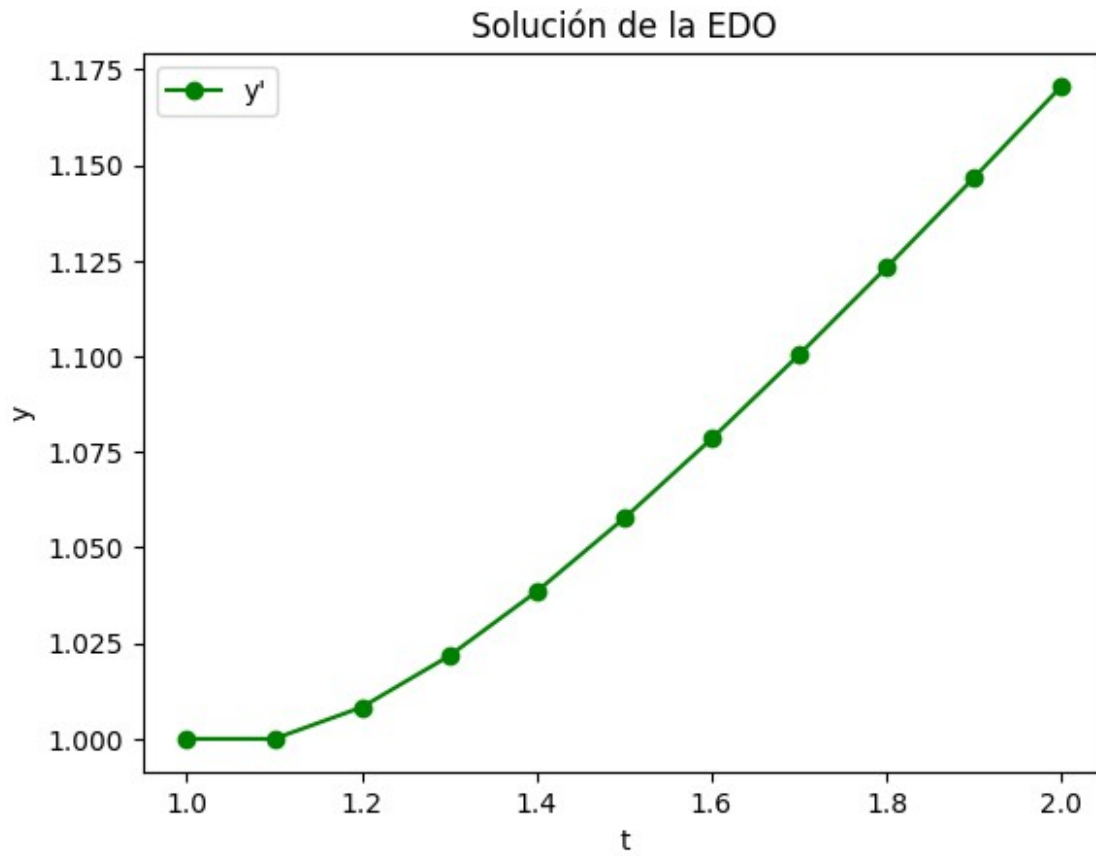
$$\begin{cases} y' = \frac{y}{t} - \left(\frac{y}{t}\right)^2, & 1 \leq t \leq 2 \\ y(1) = 1, & h = 0.1 \end{cases}$$

```
y_der = lambda t, y: y/t - (y/t)**2
y_init = 1

ys2a, ts2a, h = ODE_euler(a = 1, b = 2, f = y_der, y_t0 = y_init, N =
10)

print(f"El valor de h es: {h}")
graphics(ts2a, ys2a)

El valor de h es: 0.1
```



## PARTE B

$$\begin{cases} y' = 1 + \frac{y}{t} + \left(\frac{y}{t}\right)^2, & 1 \leq t \leq 3 \\ y(1) = 0, & h = 0.2 \end{cases}$$

```

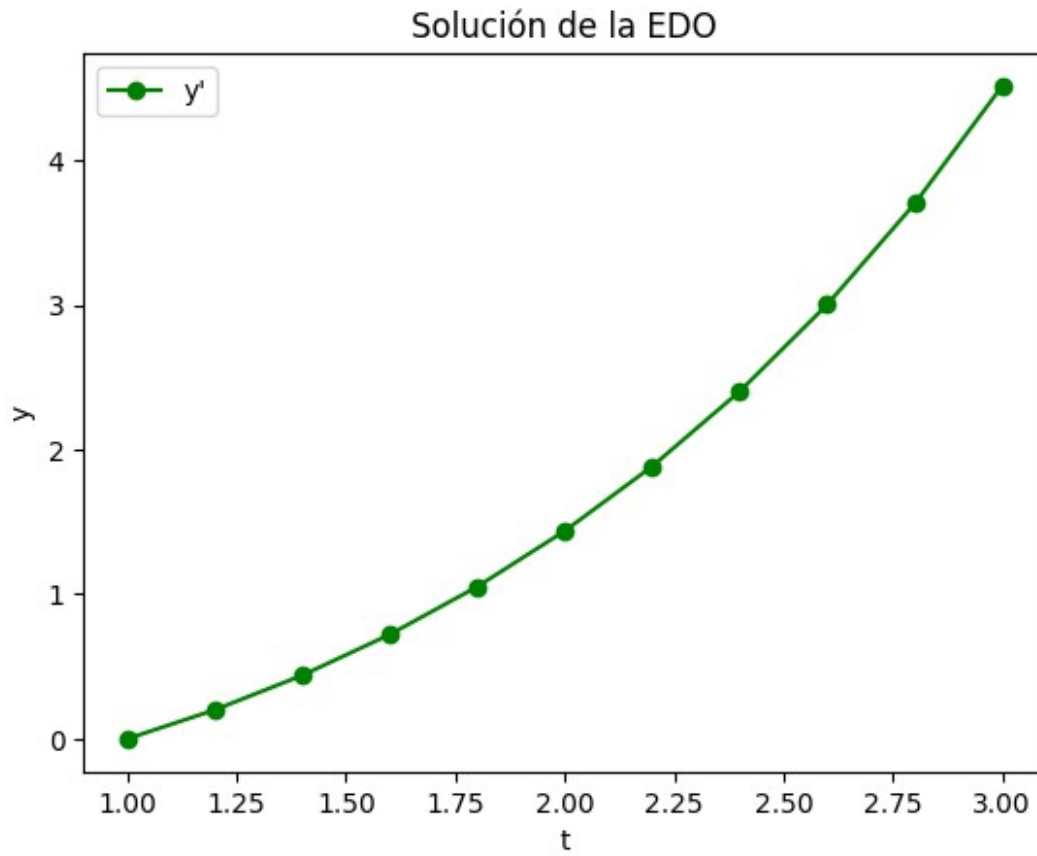
y_der = lambda t, y: 1 + y/t + (y/t)**2
y_init = 0

ys2b, ts2b, h = ODE_euler(a = 1, b = 3, f = y_der, y_t0 = y_init, N =
10)

print(f"El valor de h es: {h}")
graphics(ts2b, ys2b)

El valor de h es: 0.2

```



## PARTE C

$$\begin{cases} y' = -(y+1)(y+3), & 0 \leq t \leq 2 \\ y(0) = -2, & h = 0.2 \end{cases}$$

```

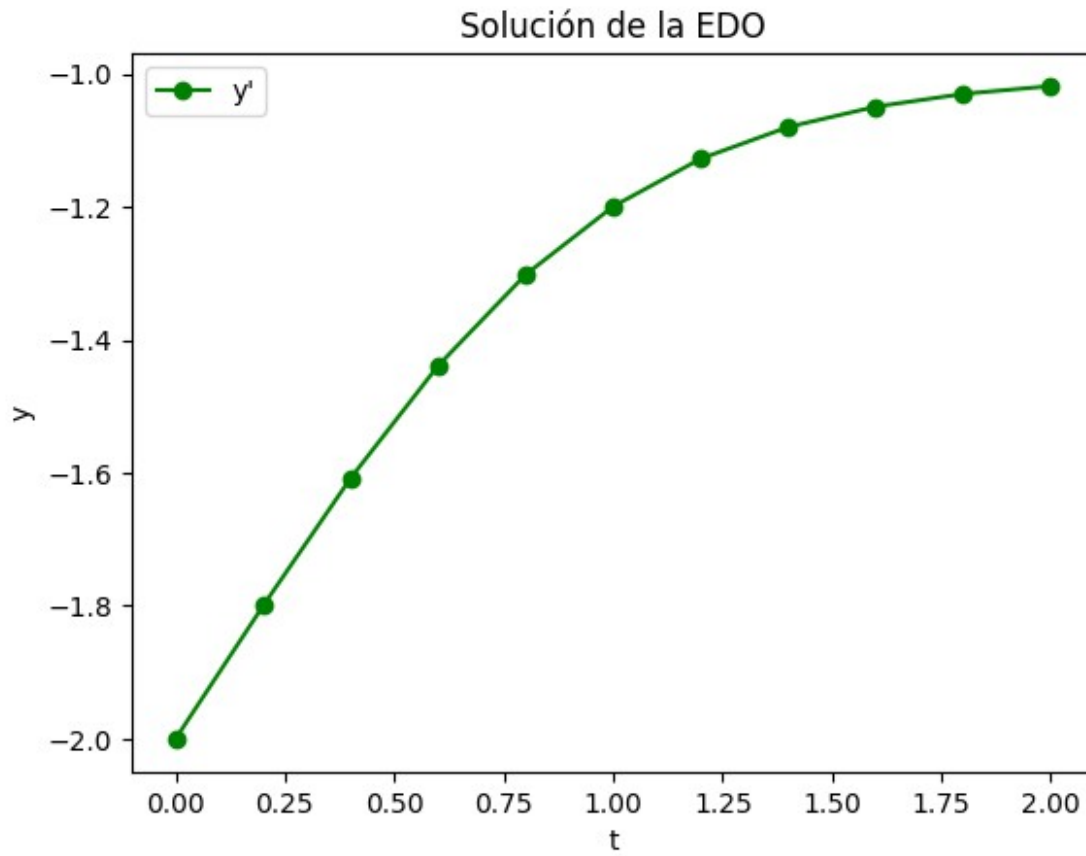
y_der = lambda t, y: -(y + 1)*(y + 3)
y_init = -2

ys2c, ts2c, h = ODE_euler(a = 0, b = 2, f = y_der, y_t0 = y_init, N =
10)

print(f"El valor de h es: {h}")
graphics(ts2c, ys2c)

El valor de h es: 0.2

```



## PARTE D

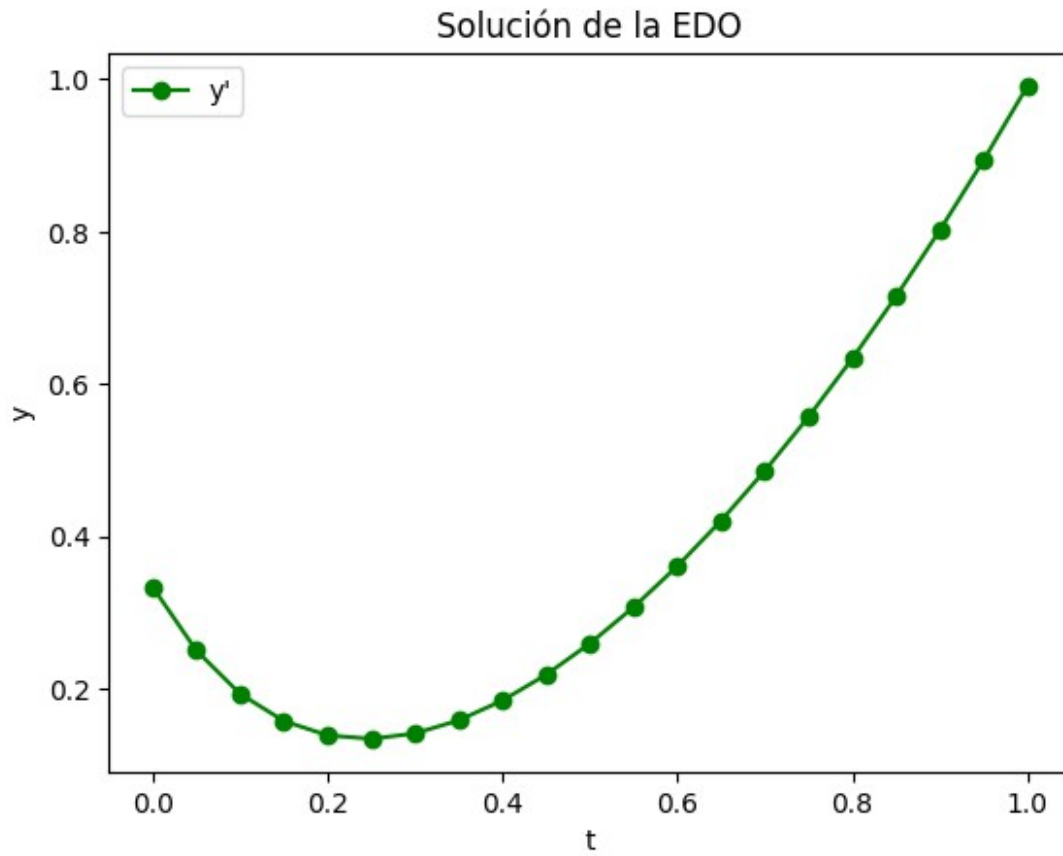
$$\begin{cases} y' = -5y + 5t^2 + 2t, & 0 \leq t \leq 1 \\ y(0) = \frac{1}{2}, & h = 0.1 \end{cases}$$

```
y_der = lambda t, y: -5*y + 5*t**2 + 2*t
y_init = 1/3
```

```
ys2d, ts2d, h = ODE_euler(a = 0, b = 1, f = y_der, y_t0 = y_init, N = 20)
```

```
print(f"El valor de h es: {h}")
graphics(ts2d, ys2d)
```

```
El valor de h es: 0.05
```



## EJERCICIO CUATRO

Aquí se dan las soluciones reales para los problemas de valor inicial en el ejercicio 3. Calcule el error real en las aproximaciones del ejercicio 3.

### PARTE A

$$y(t) = \frac{t}{1 + \ln t}$$

```
def y1(t):
    return t/(1 + math.log(t))

errorReal = np.mean([abs(y(t) - y_aprox) / abs(y(t)) for y_aprox, t in
zip(ys2a, ts2a)])
print(f"El error real es: {errorReal}")

El error real es: 0.9109203179090191
```

## PARTE B

$$y(t) = t \tan(\ln t)$$

```
def y2(t):  
    return t*math.tan(math.log(t))  
  
errorReal = np.mean([abs(y(t) - y_aprox) / abs(y(t)) for y_aprox, t in  
zip(ys2b, ts2b)])  
print(f"El error real es: {errorReal}")  
  
El error real es: 0.9884942989570799
```

## PARTE C

$$y(t) = -3 + \frac{2}{1 + e^{-2t}}$$

```
def y3(t):  
    return - 3 + 2/(1 + math.exp(-2*t))  
  
errorReal = np.mean([abs(y(t) - y_aprox) / abs(y(t)) for y_aprox, t in  
zip(ys2c, ts2c)])  
print(f"El error real es: {errorReal}")  
  
-----  
-----  
ZeroDivisionError                                Traceback (most recent call  
last)  
Cell In[32], line 4  
      1 def y3(t):  
      2     return - 3 + 2/(1 + math.exp(-2*t))  
----> 4 errorReal = np.mean([abs(y(t) - y_aprox) / abs(y(t)) for  
y_aprox, t in zip(ys2c, ts2c)])  
      5 print(f"El error real es: {errorReal}")  
  
ZeroDivisionError: float division by zero
```

## PARTE D

$$y(t) = t^2 + \frac{1}{3}e^{-5t}$$

```
def y4(t):  
    return t**2 + (1/3)*math.exp(-5*t)  
  
errorReal = np.mean([abs(y(t) - y_aprox) / abs(y(t)) for y_aprox, t in  
zip(ys2d, ts2d)])  
print(f"El error real es: {errorReal}")
```

El error real es: 0.7773952281750381

---

## EJERCICIO CINCO

Utilice los resultados del ejercicio 3 y la interpolación lineal para aproximar los siguientes valores de  $y(t)$ . Compare las aproximaciones asignadas para los valores reales obtenidos mediante las funciones determinadas en el ejercicio

### PARTE A

$$y(0.25) \text{ y } y(0.93)$$

```
res = y1(0.25)
print(res)

res = y1(0.93)
print(res)

-0.6471748623905226
1.0027718477462106
```

### PARTE B

$$y(t) = y(1.25) \text{ y } y(1.93)$$

```
res = y2(1.25)
print(res)

res = y2(1.93)
print(res)

0.2836531261952289
1.4902277738186658
```

### PARTE C

$$y(2.10) \text{ y } y(2.75)$$

```
res = y3(2.1)
print(res)

res = y3(2.75)
print(res)

-1.0295480633865461
-1.008140275431792
```



## PARTE D

$$y(t) = y(0.54)y(0.94)$$

```
res = y4(0.54)  
print(res)
```

```
res = y4(0.94)  
print(res)
```

```
0.3140018375799166  
0.8866317590338986
```



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

---

**REPOSITORIO:**

[https://github.com/ImYasid/METODOS\\_NUMERICOS.git](https://github.com/ImYasid/METODOS_NUMERICOS.git)

**REFERENCIAS BIBLIOGRÁFICAS:**

- [1] Richard L. Burden, 2017. Análisis Numérico. Lugar de publicación: 10ma edición. Editorial Cengage Learning.

**DECLARACIÓN DEL USO DE INTELIGENCIA ARTIFICIAL**

Se utilizó IA para la optimización de código adicional al mejoramiento de la gramática del texto para un mejor entendimiento.