Taller de Aplicación de Modelos Numéricos

→|FÓRMULA RK4 VS...

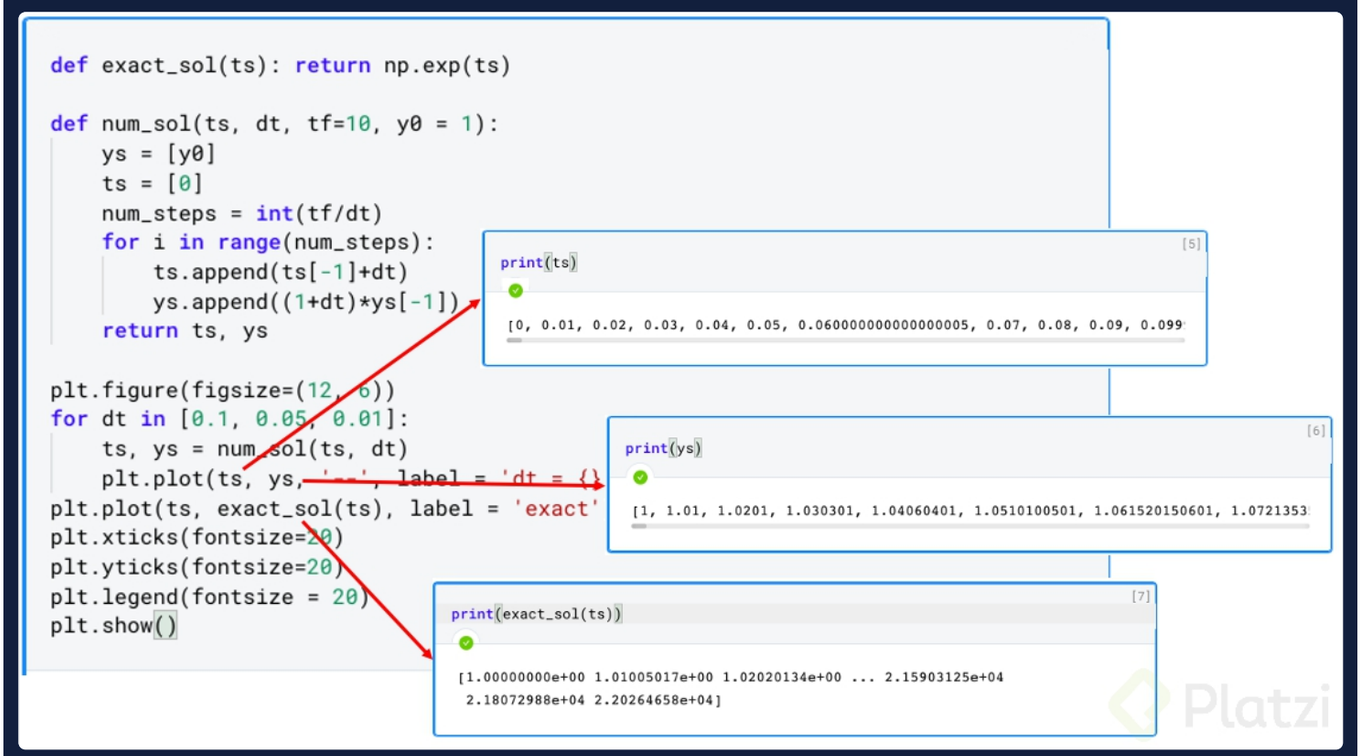
En la clase pasada resolvimos una ecuación diferencial muy sencilla con el método de Euler, pero mencionamos también que esa misma ecuación diferencial se puede resolver con métodos del cálculo y el álgebra tradicionales, por ejemplo, por medio de separación de variables, para obtener:

$$y(t) = e^t \quad [1]$$

Aquí podemos evidenciar la diferencia substancial entre una solución numérica y una solución exacta.

1. Por un lado, la solución exacta se obtiene siempre en términos de una función matemática que describe exactamente la relación entre las variables ***t*** e ***y***. Esta función puede calcularse para cualquier valor del tiempo.
2. Por otro lado, la solución numérica se obtiene como una lista o una tabla de valores discretos que permite conocer de forma aproximada la relación entre variables solamente para ciertos valores del tiempo (valores separados por un intervalo de tiempo que denotamos como Δt.

Sin embargo, siempre es posible convertir la solución exacta en una tabla de valores discretos (como hicimos en el notebook de la clase anterior) con la diferencia de que esta tabla contiene valores exactos del problema y no solamente aproximaciones, así podemos comparar al mismo nivel ambas soluciones.



Como ves en la imagen anterior (que es un extracto de la última celda del notebook de la clase pasada), puedes hacer un print() de cada lista de valores discretos del tiempo, que denotamos por $\mathbf{t}_s$, y de las soluciones numérica y exacta denotadas por $\boldsymbol{Y}_s$ y $\boldsymbol{exact\_sol(t_s)}$, respectivamente. De esta manera, podemos ver ambos tipos de soluciones a un sistema dinámico al mismo nivel pero con dos grandes advertencias:

1. Los métodos numéricos siempre tienen un error asociado por el carácter finito del intervalo de tiempo. Este error tiende a cero a medida que usamos intervalos $\boldsymbol{\Delta t}$ más pequeños.
2. Entre más pequeño es el intervalo de tiempo, más tiempo de cómputo se requiere para obtener la solución hasta un valor deseado.

Cuando hablamos de métodos numéricos, debemos siempre jugar con un balance entre rapidez y precisión. Si queremos una solución rápida, el error será grande, pero si queremos una solución muy precisa, entonces tomará más tiempo calcularla. Recordemos que en general consideramos una ecuación diferencial del tipo:

$$y' = f(t,y)$$

y podemos definir el **Error de Truncamiento Local (ETL)** como el error que se genera en una sola iteración para un tiempo $\boldsymbol{t_i}$. Además, después de un cierto número de iteraciones el error vendrá acumulado y a este error total se le denomina **Error de Truncamiento Global (ETG)**. Para el caso del método de Euler, el ETL se puede aproximar por la fórmula:

$$ETL(t_i) = \frac{1}{2}\Delta t^2 f'(t_i, y_i) \quad [2]$$

No te preocupes por el origen de esta fórmula, este es un truco que sale de varias cuentas matemáticas y por ahora no nos interesa conocerlas en detalle. Ahora, de la fórmula anterior podemos aproximar el **ETG(T_i)** considerando que se deben sumar todos los ETLs de las iteraciones previas al tiempo t_i. Entonces:

$$ETG(t_i) = ETL(t_1) + ETL(t_2) + ... + ETL(t_i)$$
$$ETG(t_i) = \frac{1}{2}\Delta t^2 \left(f(t_1, y_1) + f(t_2, y_2) + ... + f(t_i, y_i)\right)$$

Cada término con ***f(t,y)*** en general tiene un valor diferente para tiempos diferentes pero podemos tomar el máximo de todos esos como ***max(f)***, considerar que existen ***i*** términos en la suma y darnos cuenta de que la siguiente desigualdad es verdadera:

$$ETG(t_i) \leq \frac{1}{2}\Delta t^2 \, (i) \, max(f)$$

Sabemos que para cada paso de tiempo $t_i = i\Delta t$, y por simplicidad llamaremos al máximo de los valores $max(f(t, y)) = M$, entonces:

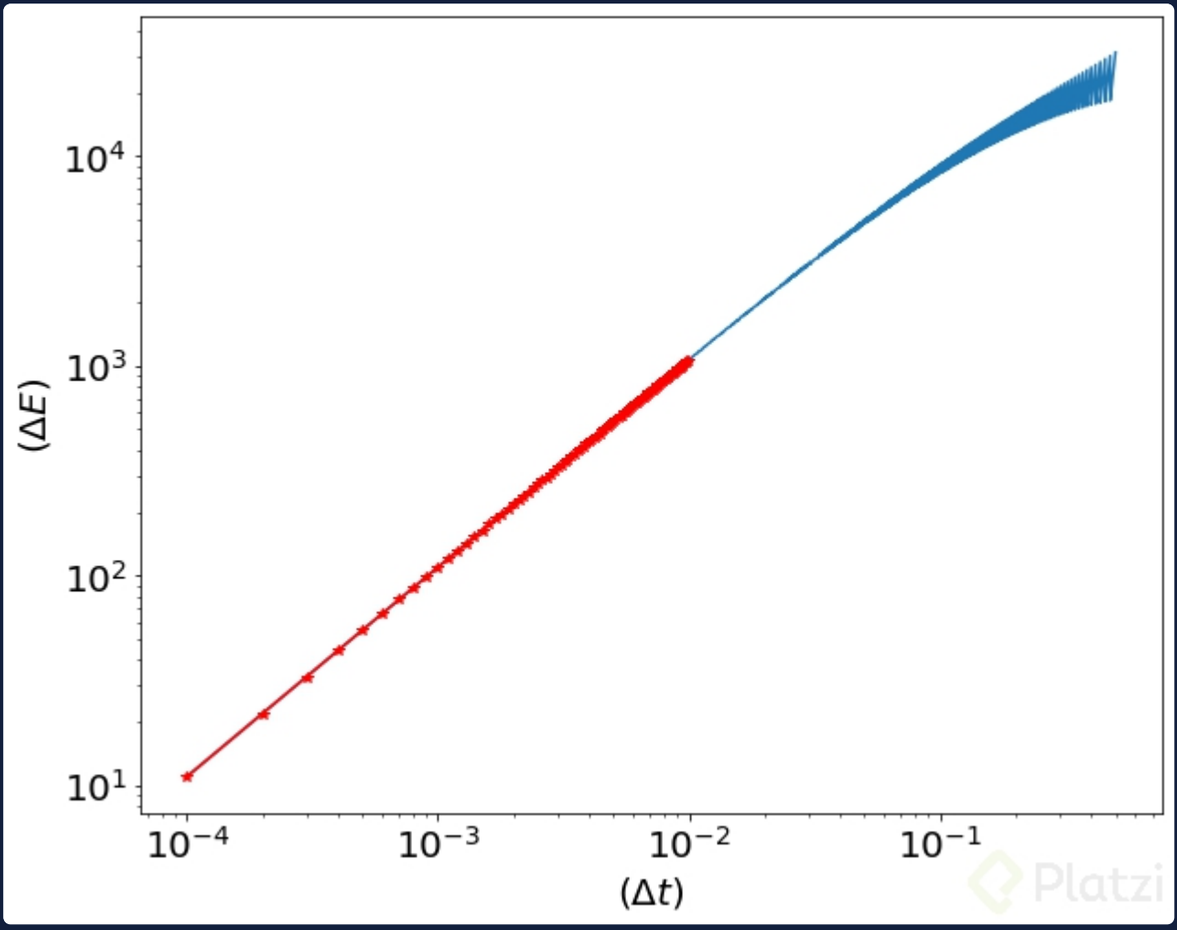
$$ETG(t_i) \leq \frac{1}{2}\Delta t^2 \left(\frac{Mt_i}{\Delta t}\right) = \Delta t \times \left(\frac{Mt_i}{2}\right) \quad [3]$$

Estas fórmulas, aunque son lo que uno normalmente encuentra en los libros, son solo cotas superiores que a menudo sobreestiman el error real en una simulación dada. En Python podemos calcular el ETG real del método de Euler para nuestro ejercicio previo así:

[116]

```
plot_text_size = 20
local_error = []
dt_arr = np.arange(0.0001, 0.5, 0.0001)
for dt in dt_arr:
    ts, ys_num = num_sol(ts, dt=dt)
    ys_ex = exact_sol(ts)
    local_error.append(np.abs(ys_num[-1] - ys_ex[-1]))
plt.figure(figsize=(10, 8))
plt.xscale('log')
plt.yscale('log')
plt.plot(dt_arr, local_error)
plt.plot(dt_arr[:-4900], local_error[:-4900], '-*r')
plt.xticks(fontsize=plot_text_size)
plt.yticks(fontsize=plot_text_size)
plt.xlabel(r'$(\Delta t)$', fontsize = plot_text_size)
plt.ylabel(r'$(\Delta E)$', fontsize = plot_text_size)
```

Truncated **Full output**



Donde ***local_error*** es un array() que guarda el error acumulado para cada simulación con un Δt específico, es decir, en este caso ese array guarda todos los ETG. La gráfica de la derecha muestra el resultado de graficar el ETG versus el intervalo Δt, colocando ambas variables en escala logarítmica. Los puntos rojos indican la parte de los datos donde se alcanza a percibir una tendencia lineal. Ahora, podemos hacer una regresión lineal simple

con Scikit Learn para verificar que los datos en rojo realmente describen una tendencia lineal:

```
[115]
from sklearn.linear_model import LinearRegression
x = np.log(dt_arr[:-4900]).reshape(-1, 1)
y = np.log(local_error[:-4900])
reg = LinearRegression().fit(x, y)
print(reg.score(x, y))
print(reg.coef_)
```

```
0.9999818383557955
[0.99261694]
```

Y vemos que el score de correlación es superior a 0.99, lo que indica que efectivamente los datos describen una relación lineal. El hecho de hacer una regresión con el logaritmo de los datos sucede porque en general suponemos que el error acumulado y el intervalo de tiempo tienen una relación de la forma:

$$ETG = K\Delta t^n$$

Entonces, al usar propiedades de los logaritmos resulta que:

$$log(ETG) = n \times log(\Delta t) + log(K)$$

Que se parece a la ecuación de una recta ***y = mx + b*** , donde ***y = log(ETG)***, ***m = n***, ***x = log(Δt)*** y ***b = log(K)***, siendo ***K*** y ***n*** valores constantes. A una relación de ese tipo se le conoce como una ley de potencias, y el propósito de esta relación es evidenciar los siguientes hechos:

1. Entre mayor es el número ***n***, menor será el error, ya que $\Delta t^n < \Delta t$ siempre que ***n > 1***
2. Entre menor sea ***K***, menor será ***log(K)*** y por lo tanto menor será el error.

En conclusión, un método numérico tendrá errores cada vez menores en la medida que ***n*** sea grande y ***K*** sea pequeño.

En la próxima clase veremos un método cuyo ETG en general es mucho menor que el generado por el método de Euler. El notebook de esta clase lo puedes encontrar en este link.