

ELEMENTOS DE CÁLCULO NUMÉRICO / CÁLCULO NUMÉRICO

Primer Cuatrimestre 2020

Tercer ejercicio computacional

Lunes 27/04/20 al Lunes 04/04/20

Recuerde subir el archivo en formato `ejercicioX_NOMBREAPELLIDO.py`

Recuerde al hacer consultas enviar su código

Un ejemplo de aplicación de Euler puede encontrarse en la lista de reproducción de videos de la parte computacional.

a) Partiendo de la ecuación:

$$\begin{aligned}y' &= -2y + t^2 \\ y(0) &= 1\end{aligned}$$

con solución

$$y(t) = \frac{3}{4}e^{-2t} + \frac{1}{2}(t^2 - t + \frac{1}{2})$$

Resuelvala empleando el método de Euler entre $t = 0$ y $t = 10$. Considere los números de pasos N con valores 10,50,100,500,1000. Grafique $y(t)$ para los distintos valores de N y compare con la solución exacta. ¿Cómo cambia el error en $t = 10$ en función de $1/N$?

Ayuda: construya una función `resuelve_y(y0,t0,T,N)` donde y_0 es el valor de y en t_0 , T es el tiempo final de cálculo, y N es el número de pasos o iteraciones. Esta función debería devolver dos listas (o dos arrays), `t` e `y`, con los tiempos y el valor calculado de y en los mismos:

```
def resuelve_y(y0,t0,T,N):  
    # CODIGO  
    return([t,y]) # Notar que retorno una lista con dos listas
```

b) Partiendo de la ecuación:

$$\begin{aligned}y'' &= -2y \\ y(0) &= 0 \\ y'(0) &= 1\end{aligned}$$

con solución

$$y(t) = \frac{1}{\sqrt{2}} \sin(\sqrt{2}t)$$

Resuelvala entre $t = 0$ y $t = 10$. Para esto, transforme la ecuación en un sistema de ecuaciones, usando el reemplazo $y_1 = y$, $y_2 = y'$, empleando el método de Euler en cada ecuación individual. Compare el resultado de actualizar primero y_1 y primero y_2 con la solución exacta:

A

$$\begin{aligned}y_2(t + dt) &= y_2(t) + F(t, y_1(t), y_2(t)) * dt \\ y_1(t + dt) &= y_1(t) + F(t, y_1(t), y_2(t)) * dt\end{aligned}$$

B

$$\begin{aligned}y_2(t + dt) &= y_2(t) + F(t, y_1(t), y_2(t)) * dt \\ y_1(t + dt) &= y_1(t) + F(t, y_1(t + dt), y_2(t)) * dt\end{aligned}$$

C

$$\begin{aligned}y_1(t + dt) &= y_1(t) + F(t, y_1(t), y_2(t)) * dt \\ y_2(t + dt) &= y_2(t) + F(t, y_1(t + dt), y_2(t)) * dt\end{aligned}$$

Habiendo elegido un método (A, B o C), calcule el error en función del tiempo para distinto número de pasos N como en el problema anterior. ¿Cómo aumenta el error en función del número de ciclos, para cada N? ¿Respeto el periodo la solución?

- c) Extra: usando la función de python `eval()` que recibe el nombre de un función (como string) y retorna la función, modifique la función que calcula y a partir de Euler del punto 1, para que reciba también una función que calcula y' y pueda ser reutilizada para distintas ecuaciones diferenciales.

Ejemplo de uso de `eval()`:

```
def suma_dos(x):  
    return(x+2)  
eval('suma_dos')(10) # Esto da 12
```