

Practica 2

González Borja, Miguel
Illescas Arizti, Rodrigo
Meyer Mañón, Juan Carlos
Rodríguez Orozco, Alejandro

28 / Abril / 2018

1 Introducción

A continuación los ejercicios de la práctica. Todos fueron realizados en Python 3.6 utilizando los paquetes *numpy* y *scipy.op*. También se incluye un cuaderno de IPython para cada ejercicio dentro de la carpeta Ejercicios, junto con los scripts *.py* Para obtener los resultados. Cada uno de estos utiliza los siguientes imports:

```
In [1]: import sys
        sys.path.append('../')
        import latexStrings as ls
        import numpy as np
        import matplotlib.pyplot as plt
        from math import *
        from IPython.display import Latex
        import odesolver
```

Se puede encontrar un repositorio del proyecto [aqui](#).

Ejercicio 1

Tenemos el PVI:

$$y' = y + 2e^{-t} \quad y(0) = 1 \quad (1)$$

Dado que el lado derecho de la EDO, $y + 2e^{-t}$ es de clase C^1 para todo $(t, y) \in \mathbb{R} \times \mathbb{R}$, y por lo tanto Lipschitz Continua en dicha region, sabemos que existe una solucion unica a este problema, que en general es dada por:

$$y(t) = (y_0 e^{-t_0} + e^{-2t_0} - e^{-2t}) e^t \quad (2)$$

Ahora apliquemos el metodo de Euler Explicito con paso $h = 0.1$:

```
In [1]: f = lambda t, y : y+2*np.exp(-t)
        exact = lambda t, t0, y0 : np.exp(t)*(y0*np.exp(-t0)+np.exp(-2*t0)-np.exp(-2*t))
        I = (0, 1)
```

```

y0 = 1

y1 = exact(1,0,1)

T, W = odesolver.solve(f,y0,I,10,method="Explicit Euler")

globalError = abs(W[0,10]-y1)
localErrors = [abs(W[0,i+1] - exact(T[i+1], T[i], W[0,i])) for i in range(10)]
maxLocalError = max(localErrors)

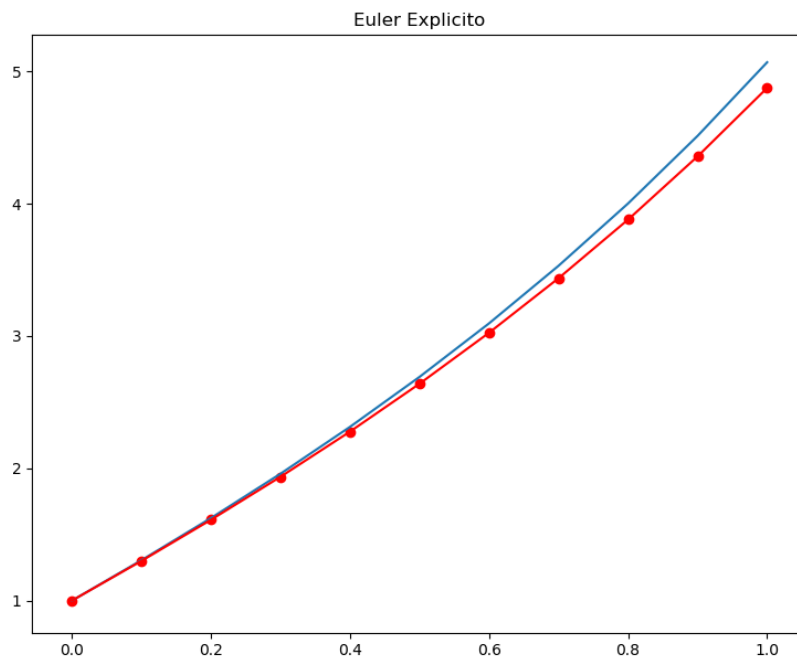
```

$w = \{1, 1.3, 1.6109675, 1.9358104, 2.2775551, 2.6393746,$
 $3.0246182, 3.4368423, 3.8798436, 4.3576938, 4.8747771\}$

Error Global: 0.1939071469339151

Maximo Error Local: 0.022668868433444622

Graficamente, observemos como se comporta la solucion exacta contra la aproximada:



Vemos que la solucion aproximada, aunque sigue la direccion general, siempre va por debajo de la solucion verdadera. Esto se debe a que nuestra solucion general (??) es convexa en el intervalo donde estamos buscando la solucion, y por lo tanto el metodo subestima la pendiente en sus aproximaciones.

Ahora apliquemos el metodo de Euler Explicito con pasos $h = 0.1 \times 2^{-k}$ para $k \in \{0, 1, \dots, 5\}$ en el intervalo $[0, 1]$:

```

In [2]: res = []
        globalErrors=[]
        steps = []

        for k in range(6):

            h = 0.1*(2**(-k))
            m = int((I[1]-I[0])/h)

            T, W = odesolver.solve(f, y0, I, m, method="Explicit Euler")

            g = abs(W[0,m]-y1)
            globalErrors.append(g)

            localErrors = [abs(W[0,i+1] - exact(T[i+1], T[i], W[0,i])) for i in range(m)]
            maxLocalError = max(localErrors)

            eoc = "NaN" if k==0 else log(g/prevg)/log(h/prevh)
            res.append([k, h, maxLocalError, eoc])

            prevh = h
            prevg = g

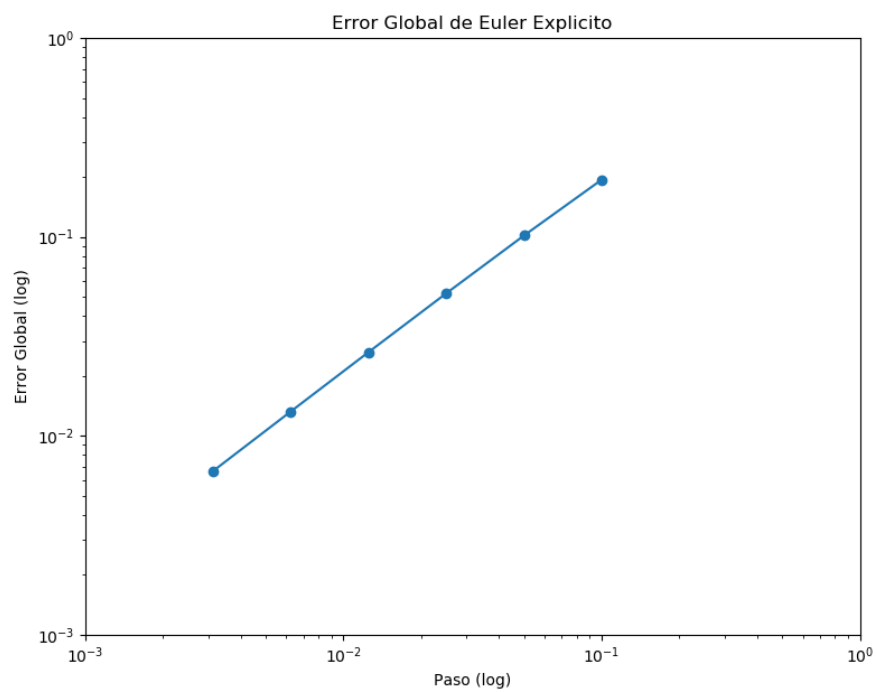
```

Observemos el comportamiento del error del metodo en relacion con el tamaño del paso:

k	Paso	Error Local Maximo	eoc
0	0.1	0.022668868433444622	NaN
1	0.05	0.005982387299206415	0.933587546963545
2	0.025	0.0015384740393056262	0.9656406382261543
3	0.0125	0.00039021932959482086	0.9825161515419355
4	0.00625	9.827082448321534e-05	0.9911799106800493
5	0.003125	2.4658225118656674e-05	0.9955701393945221

Sabemos que en teoria, los pasos de Euler Explicito tienen un error del orden de $O(h^2)$, mientras que el error global es de orden $O(h)$. En efecto, vemos que el error local maximo siempre cumple la cota deseada, pues en el peor de los casos, cuando $h = 0.1$, el error es aproximadamente $0.022 = 2.2h^2$, y conforme disminuye el tamaño de los pasos se sigue cumpliendo la cota.

Por otro lado, observamos que la razon de convergencia experimental (eoc) empieza siendo 0.93 pero rapidamente se aproxima a 1 conforme disminuye el tamaño de los pasos utilizados en su calculo. Esto nos lleva a la conclusion de que el error global en efecto es de orden 1, lo cual es aun mas evidente al observar el comportamiento del error en terminos del tamaño del paso, como se muestra en la siguiente grafica:



Podemos observar que, en escala logaritmica, el error global y el paso tienen una relacion lineal, que es lo esperado de un metodo con error global de orden $O(h)$.