Universidad de Los Andes
Facultad de Ingeniería y Ciencias Aplicadas

Guía de estudio
Laboratorio 2
Ecuaciones Diferenciales

Resolución de un problema de valor inicial por métodos numéricos

Recordemos que vimos algunos métodos numéricos para resolver problemas de la forma:

$$(PVI) \quad \begin{cases} y' = f(x,y)\,, & x \in [x_0, x_f] \\ y(x_0) = y_0 \end{cases}$$

donde dividimos el invervalo $[x_0, x_f]$ en N sub intervalos. Tenemos que el largo del intervalo es $L = x_f - x_0$, por lo que el largo de cada sub intervalo será $h = \frac{L}{N}$ (si dividimos de manera homogénea). Con esto, tendremos N sub intervalos de la forma $[x_j, x_{j+1}]$, donde

$$x_n = x_0 + nh\,,\ n = 0, 1, 2, \ldots, N$$

Los métodos numéricos nos dan valores y_n, los cuales son las aproximaciones de los términos $y(x_n)$, para cada n. Por ejemplo, vimos que el método de *Euler Progresivo* nos entrega los valores de y_n mediante la siguiente fórmula:

$$y_{n+1} = y_n + hf(x_n, y_n)\,,\ n = 0, 1, 2, \ldots N-1$$

Podemos programar esto en Python, para lo cual será necesaria la libreria Numpy. Por ejemplo, tomemos el problema:

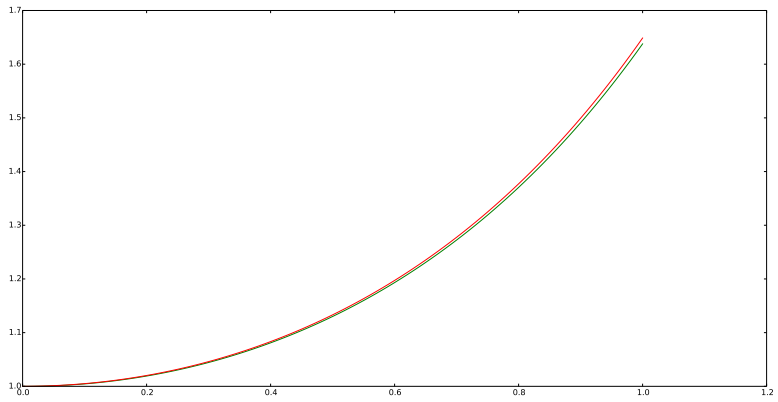
$$(\star) \quad \begin{cases} y' = xy\,, & x \in [0,1] \\ y(0) = 1 \end{cases}$$

La idea es rescatar los valores de cada par $(x_n, y_n))$, con $n = 0, 1, \ldots, N$, para así poder graficar. Para resolver el problema $(\star)$ con el método de Euler Progresivo, usaremos el siguiente código:

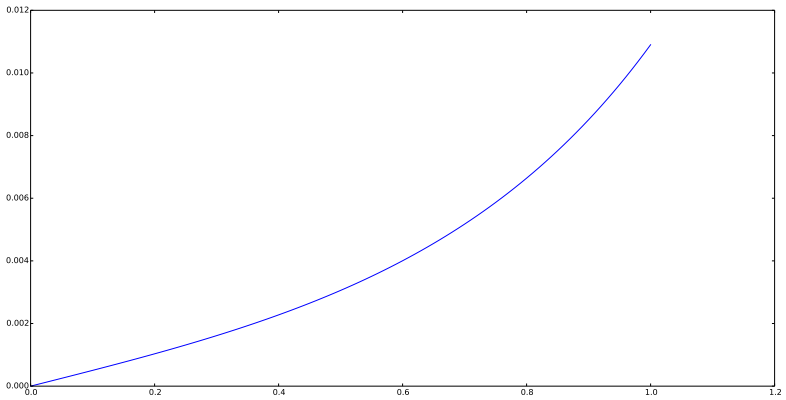
```
import numpy as np

x0 = 0.0
xf = 1.0
y0 = 1.0
L = xf-x0
N = 100
h = L/N

X = np.zeros(N+1)
Y = np.zeros(N+1)

X[0] = x0
Y[0] = y0


for n in range(N):
```

```
if n==0:
    f = x0*y0
    y = y0 + h*f
    x = x0 + h
    X[n+1] = x
    Y[n+1] = y
else:
    f = x*y
    y = y + h*f
    x = x + h
    X[n+1] = x
    Y[n+1] = y
```

donde tenemos que `X` e `Y` son los vectores donde guardaremos cada valor de x_n e y_n respectivamente. Notemos que usamos 100 sub intervalos. Además, podemos comparar la solución numérica con la exacta, donde esta última es $y(x) = e^{x^2/2}$. Graficamente, esta comparación es la siguiente:



Tenemos en verde a la solución aproximada y en rojo a la solución exacta. Para finalizar, también podemos graficar el error $e_n = y_n - y(x_n)$:



Notar que, independiente de la forma de la función que define el error, este se mantiene siempre cercano a 0. Como estudio, se recomienda hacer este mismo ejercicio para los demás métodos numéricos vistos en clases y hacer comparaciones (entre una aproximación con la exacta o bien entre aproximaciones), así también probar otros ejemplos.