# Elementos de Cálculo Numérico / Cálculo Numérico

Primer Cuatrimestre 2021

**Décimo ejercicio computacional**
**14/06/21 al Lunes 21/06/21**
**Recuerde subir el archivo en formato ejercicioX_NOMBREAPELLIDO.py**
**Recuerde al hacer consultas enviar su código**

En este ejercicio resolveremos una ecuación no-lineal con un método de Newton Raphson. Recordemos que en el de Newton-Rhapson buscamos los ceros de la función apoyandonos en la derivada de la función. Si buscamos ceros de $f(x)$, nuestra iteración será:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Para resolver este ejercicio, utilizaremos una herramienta de Python llamada "funciones lambda", que permite definir una función en una linea de código.

Por ejemplo, una función que evalúa el polinomio $p$ de grado 12

$$p(x) = \frac{1.3}{12} \sum_{i=1}^{12} (1-x)^i - 1$$

puede escribirse como:

```
p = lambda x: 1.3/12 * sum((1-x)**np.arange(1,13)) - 1
```

A Resuelva la ecuación
$$p(x) = 0$$
con una adivinanza inicial de $x_0 = 0.05$ utilizando la rutina *fsolve* del módulo *scipy.optimize*.

B Utilizando funciones lambda, construya la función $f(y) = y^2 - y^3$ y su derivada $f'(y)$.

C Escriba una función *NewtonRaphson(f,f_der,x0)*, que resuelva la ecuación $f(x) = 0$ con un criterio de convergencia de 8 dígitos correctos. Corrobore que su función obtiene el resultado correcto comparando contra *fsolve* en algún caso sencillo.

D Implemente el método de Euler implícito para el ejercicio 18 de la guía, donde hay que depejar a cada paso una ecuación no-lineal:
$$y_n = y_{n+1} - h(y_{n+1}^2 - y_{n+1}^3)$$
Para ello, utilice su rutina de Newton-Raphson tomando $y_n$ como estimación inicial. Para la EDO, utilice un valor inicial de $y_0 = 0.01$, y un paso $h = 2$.