Laboratorio No. 2

Resolución Numérica de Ecuaciones no Lineales

Como hemos visto en clase, los *métodos numéricos* para la resolución de *ecuaciones no lineales* pueden utilizarse para encontrar puntos en donde la primera derivada de una función f es igual a 0, i.e. encontrar *puntos estacionarios*. De acuerdo al teorema de Fermat, estos puntos estacionarios son "candidatos" para determinar los extremos locales de una función f. Más que determinar puntos estacionarios de una función unidimensional, es importante mencionar que, estos algoritmos para encontrar raíces son útiles para determinar los *step-sizes* (o *learning rates*) óptimos en algoritmos de optimización para el caso multidimensional.

En este laboratorio, deberá agregar a su librería dos algoritmos para resolver una ecuación de la forma:

$$F(x) = 0, \tag{1}$$

en donde, F es una función no lineal de una variable real. En particular, deberá implementar los siguientes algoritmos:

Algoritmo – Método de la Bisección

1. *Input:* una función $F : \mathbb{R} \to \mathbb{R}$ continua en un intervalo $[a, b]$ y que cambia de signo dentro de ese intervalo, número máximo de iteraciones $k_{\text{máx}}$ y tolerancia ε.
2. $k \leftarrow 0$
3. $x_k \leftarrow \dfrac{a+b}{2}$
4. **while** $k < k_{\text{máx}}$ **and** $|F(x_k)| > \varepsilon$ **do**
   4.1 **if** $F(a)F(x_k) < 0$ **then**
      $b \leftarrow x_k$
   else
      $a \leftarrow x_k$
   4.2 $k \leftarrow k + 1$
   4.3 $x_k \leftarrow \dfrac{a+b}{2}$
5. *Output:* x_k, una aproximación a una raíz de la ecuación (1) en el intervalo (a, b).

Algoritmo – Método de Newton-Raphson

1. *Input:* función diferenciable $F : \mathbb{R} \to \mathbb{R}$, solución inicial x_0, número máximo de iteraciones $k_{\text{máx}}$ y tolerancia ε.
2. $k \leftarrow 0$
3. **while** $k < k_{\text{máx}}$ **and** $|F(x_k)| > \varepsilon$ **do**
   3.1 $x_{k+1} \leftarrow x_k - \dfrac{F(x_k)}{F'(x_k)}$
   3.2 $k \leftarrow k + 1$
4. *Output:* x_k, una aproximación a una raíz de la ecuación (1).

Finalmente, aplique los dos algoritmos anteriores para resolver el siguiente problema de optimización convexo:

$$\min_{x\in\mathbb{R}} \; e^x + x^2. \tag{2}$$

¿Qué algoritmo converge más rápidamente? ¿qué ventajas y desventajas tiene el método de la bisección? ¿qué ventajas y desventajas tiene el método de Newton-Raphson?

Nota: recuerde que su programa debe desplegar una *tabla* que contenga los siguientes parámetros para cada una de las iteraciones:

a) k: número de la iteración,

b) x_k: aproximación a una de la soluciones de la ecuación (1) en la k-ésima iteración, y

c) $|F(x_k)|$: error en la k-ésima iteración.

En la Figura 1 se presenta un ejemplo de la tabla esperada.

Iter	Xn	Error
1.0000000000	4.0004800001	0.9995199999
2.0000000000	3.2015553130	0.7989246870
3.0000000000	2.5640997181	0.6374555949
4.0000000000	2.0582201284	0.5058795897
5.0000000000	1.6632929346	0.3949271938
6.0000000000	1.3698307570	0.2934621777
7.0000000000	1.1810673535	0.1887634035
8.0000000000	1.0990319490	0.0820354045
9.0000000000	1.0848524858	0.0141794633
10.0000000000	1.0844720383	0.0003804474
11.0000000000	1.0844717712	0.0000002671

Figura 1: Ejemplo de tabla con parámetros esperados.