

ANÁLISIS NUMÉRICO I – 2014

Trabajo de Laboratorio N^o 2

1. Escriba una función en OCTAVE que implemente el método de bisección para hallar una raíz de $f : \mathbb{R} \rightarrow \mathbb{R}$ en el intervalo $[a, b]$. La función debe llamarse “**rbisec**”, y tener como entrada los argumentos (fun, I, e, m) , donde “**fun**” es una función que dado x retorna $f(x)$, $I = [a, b]$ es un intervalo en \mathbb{R} , e es la tolerancia deseada del error y m es el número máximo de iteraciones permitidas. El algoritmo debe finalizar en la k -ésima iteración si $|f(x_k)| < e$ o $k \geq m$. La salida debe ser $[hx, hf]$ donde $hx = [x_1, \dots, x_N]$ es el histórico de puntos medios y $hf = [f(x_1), \dots, f(x_N)]$ el histórico de los respectivos valores funcionales.
2. Utilice la función del ejercicio anterior para:
 - (a) encontrar la menor solución positiva de la ecuación $2x = \tan(x)$ con un error menor a 10^{-5} en menos de 100 iteraciones. ¿Cuántas iteraciones son necesarias cuando comenzamos con el intervalo $[0.8, 1.4]$? Usar la siguiente sintaxis:
`octave> [hx, hy] = ej2a(@fun_ej2a, I, e, m)`
 - (b) Encontrar una aproximación a $\sqrt{3}$ con un error menor a 10^{-5} . Para esto, considere la función $f(x) = x^2 - 3$ (que debe llamarse “**fun_ej2b**”).
`octave> [hx, hy] = ej2b(@fun_ej2b, I, e, m)`
 - (c) Graficar conjuntamente f y los pares $(x_k, f(x_k))$ para las dos funciones anteriores y con al menos dos intervalos iniciales distintos para cada una.
3. Escriba una función en OCTAVE que implemente el método de Newton¹ para hallar una raíz de $f : \mathbb{R} \rightarrow \mathbb{R}$ partiendo de un punto inicial x_0 . La función debe llamarse “**rnewton**”, y tener como entrada (fun, x_0, e, m) donde “**fun**” es una función que dado x retorna $f(x)$ y $f'(x)$, x_0 es un punto inicial en \mathbb{R} , e es la tolerancia deseada del error y m es el número máximo de iteraciones permitidas. El algoritmo debe finalizar en la k -ésima iteración si vale alguna de las siguientes:

$$\frac{|x_k - x_{k-1}|}{|x_k|} < e, \quad |f(x_k)| < e, \quad k \geq m.$$

La salida debe ser $[hx, hf]$ donde $hx = [x_1, \dots, x_N]$ es el histórico de puntos generados y $hf = [f(x_1), \dots, f(x_N)]$ el histórico de los respectivos valores funcionales.

4. Escriba una función en OCTAVE que aproxime $\sqrt[3]{a}$ con un error menor a 10^{-6} . Para ello, haga una variable global a (use el comando `global`) y llame al método de Newton del ejercicio anterior para la función $f(x) = x^3 - a$. Utilice las funciones “**rnewton**” y “**fun_ej4**”.
5. Escriba una función en OCTAVE que implemente el método de Iteración de Punto Fijo para hallar un punto fijo de $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ partiendo de un punto inicial x_0 . La función debe llamarse “**ripf**”, y tener como entrada (fun, x_0, e, m) donde “**fun**” es una función que dado

¹Creado por I. Newton en 1669 para polinomios de tercer grado, usado por J. Raphson en 1690 para polinomios de cualquier grado y reescrito por T. Simpson en 1740 para funciones generales.

x retorna $\varphi(x)$, x_0 es un punto en \mathbb{R} , e es la tolerancia deseada del error y m es el número máximo de iteraciones permitidas. El algoritmo debe finalizar en la k -ésima iteración si $|x_k - x_{k-1}| < e$ o $k \geq m$. La salida debe ser hx donde $hx = [x_1, \dots, x_N]$ es el histórico de puntos generados.

6. Escriba una función con las mismas entradas que la del ejercicio anterior, que llame al método de iteración de punto fijo y que retorne el último valor generado x_N , la cantidad de iteraciones N y abra una ventana de gráfico donde aparezcan con distintos colores la recta x , la función $\varphi(x)$ y la poligonal generada por los pares (x_0, x_1) , (x_1, x_1) , (x_1, x_2) , (x_2, x_2) , \dots , (x_{N-1}, x_{N-1}) , (x_{N-1}, x_N) , (x_N, x_N) .
7. Se desea conocer la gráfica de una función u definida implícitamente: $u(x) = y$ donde y es solución de

$$y - e^{-(1-xy)^2} = 0.$$

Implemente tres versiones de esta función, hallando el valor de y con los métodos de los ejercicios 1, 3 y 5. Los valores iniciales y tolerancias usadas por los distintos métodos deben ser escogidos de manera que cualquier usuario pueda graficar u en el intervalo $[0, 10]$ sin inconvenientes.