

ANÁLISIS NUMÉRICO I/ ANÁLISIS NUMÉRICO
Licenciatura en Matemática/ Ciencias de la Computación
FAMAF, UNC — Año 2017

TRABAJO DE LABORATORIO N° 2

1. Escribir 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 si $k \geq m$. 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. Utilizar 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] = rbisec(@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] = rbisec(@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. Escribir una función en OCTAVE que implemente el método de Newton 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. Escribir 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 $f(x) = x^3 - a$. Utilizar las funciones “**rnewton**” y “**fun_ej4**”.

5. Escribir 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 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. Se quiere usar la fórmula de iteración $x_{n+1} = 2^{x_n-1}$ para resolver la ecuación $2x = 2^x$. Utilizar la función del ejercicio anterior para investigar si converge; y en caso afirmativo, estudiar hacia qué valores lo hace para distintas elecciones de x_0 , tomando un número máximo de 100 iteraciones y un error menor 10^{-5} . Usar la siguiente sintaxis:

```
octave>[hx] = ripf(@fun_ej6, x0, e, m)
```

7. Se desea conocer la gráfica de una función u definida implícitamente: $u(x) = y$ donde y es solución de

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

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