PRACTICA 3

RAÍCES NUMÉRICAS NO LINEALES

Fernando Javier Nossa Chaparro

Carlos Rizo Maciá

Estudiantes MSC. computing graphics and video game development.

Universidad Rey Juan Carlos

2019

***Resumen*—El contenido de este documento resume nuestra experiencia en la práctica de raíces numéricas de funciones no lineales en el laboratorio. Esta práctica enseña la aplicación del método de la bisección y Newton-Raphson para encontrar raíces de funciones no lineales.**

I. INTRODUCCION

En el transcurso de la práctica crearemos un programa en Matlab que aplique los métodos de la bisección y Newton-Raphson a diversas funciones previamente proveídas por el profesor. Comparamos los resultados con los entregados por el profesor para confirmar la validez del código creado.

II. BISECCION

A. Primer caso

Siendo el intervalo dos valores denotados [x1,x2], en nuestro programa vamos a encontrar las raíces (valores en y en el que x=0) de la ecuación. Primeramente analizamos el caso más fácil, es decir, si alguno de los dos valores del intervalo evaluado en la función es igual a cero.

```
A = Fun(x1, ejNumber);
B = Fun(x2, ejNumber);
if(A == 0 || B ==0 )
    if(A==0)
        fprintf('la raiz de f(x) se encuentra en %f \n',  A);
    else
        fprintf('la raiz de f(x) se encuentra en %f \n',  B);
    end
end
```

Imagen 1: Validación de los intervalos en la raíz

B. Punto medio.

Seguidamente, si no encontramos la raíz evaluando los puntos que delimitan el intervalo, debemos calcular el punto medio entre ellos (xm). Este punto se consigue sumando y partiendo entre 2 los puntos del intervalo [x1,x2] y evaluando la función en este. De la misma manera, si este punto medio del intervalo evaluado en la función es igual a cero ahí se encuentra la raíz que buscamos.

C. Análisis de signo de la función

El siguiente paso es analizar el signo resultante del punto medio. Comparamos el signo con los puntos del intervalo [x1,x2] dados. Sabemos que la raíz se encuentra entre el punto medio y el intervalo con el que el signo no coincide. Este proceso se repetirá exhaustivamente hasta que se encuentre la raíz, o hasta que el error nos de un criterio de parada.

```
while( errorThreshold <= calculatedError )
    signoIzq = sign(A);
    signoDer = sign(B);
    if(signoIzq  ~= signoDer )

        xm = (x1+x2)/2;
        if(xm == 0)
            fprintf('el cero se encuentra en %f \n',  xm );
        else
        xmf = Fun(xm, ejNumber);
        sxm = sign(xmf);
            if(sxm == signoDer)
                x2 = xm;
            else
                x1 = xm;
            end
            A=Fun(x1, ejNumber);
            B=Fun(x2, ejNumber);
            calculatedError = log((x2-x1)/errorThreshold)/log(2);
        end
    end
    nIterations = nIterations+1;
end
```

Imagen 2: Iteración del punto medio evaluado y comparado con las funciones

III. NEWTON-RAPHSON

Este método tiene una convergencia cuadrática por tanto necesita menos iteraciones a la hora de hallar las raíces de una función. Lo primero que debemos tener es un punto inicial al cual le hallamos la recta tangente. Esta recta tangente la proyectamos hasta que sea secante con el eje x y en la intersección evaluamos ese valor. Repetiremos el proceso de trazar la tangente hasta que sea secante de nuevo, e iteramos hasta encontrar la raíz o hasta que el error nos de un criterio de parada. Esto nos lleva a la fórmula de la imagen 5 que obtiene una estimación de la raíz.

$$x_0^n = x_0^{n-1} - \frac{f\left(x_0^{n-1}\right)}{f'\left(x_0^{n-1}\right)}$$

Imagen 3: Fórmula final de Newton-Raphson

Siguiendo esta fórmula evaluamos el punto inicial guardado en x0 tanto la función como en la derivada de la función. Dividimos los resultados entre sí y restamos el resultado a nuestro x0. Tras el cálculo del error cometido el nuevo valor calculado pasa a almacenarse en x0 y repetiremos este proceso hasta que el error cometido sea menor a nuestro criterio de parada.

```
while (errorThreshold < calculatedError )
      fPrima=eval(subs(fPrima,x,x0));
      fx= Fun(x0,ejNumber);

      x1=x0-(fx/fPrima);
      x1=round(double(x1),4);

      calculatedError =abs(abs(x1)-abs(x0));

      x0=x1;
      nIterations = nIterations+1;
end
```

Imagen 4.: Cálculo de la pendiente iterando.

IV. CALCULO DEL ERROR

A. Metodo de Biseccion

Considerando un valor de tolerancia del error se restan el punto [x2] menos [x1] de la iteración correspondiente, y se dividen por la tolerancia del error. La división de los logaritmos de este resultado y el logaritmo de 2 nos da el error cometido. Si este error es mayor a nuestra condición de límite seguiremos calculando

```
calculatedError = log((x2-x1)/errorThreshold)/log(2);
```

imagen5. Cálculo del error en Newton-Raphson

B. Newton-Raphson

El error calculado con este método, se calcula con la diferencia al cuadrado del punto actual menos el punto anterior.

```
calculatedError =abs(abs(x1)-abs(x0));
```

imagen6. Error dentro de la iteración.

V. RESULTADOS Y CONCLUSIONES

En las siguientes imágenes podemos observar los resultados de aplicar las funciones de la práctica a nuestros métodos

Por fzero, la raiz de x*sin(0.5 + x^2) + exp(-x) en 3.000000 está en 2.990249

Método de la bisección. Ej 1: El cero de la funcion se encuentra en el intervalo 1.076172 tras n = 9 iteraciones

Método de la bisección. Ej 2: El cero de la funcion se encuentra en el intervalo 0.641602 tras n = 12 iteraciones

Método de la bisección. Ej 2: El cero de la funcion se encuentra en el intervalo 0.641602 tras n = 9 iteraciones

Método de la bisección. Ej 3: El cero de la funcion se encuentra en el intervalo 0.364258 tras n = 10 iteraciones

Método de la bisección. Ej 3: El cero de la funcion se encuentra en el intervalo 1.920898 tras n = 11 iteraciones

Método de la bisección. Ej 3: El cero de la funcion se encuentra en el intervalo 3.563477 tras n = 11 iteraciones

Método de la bisección. Ej 4: El cero de la funcion se encuentra en el intervalo 0.713867 tras n = 12 iteraciones

Imagen 7: Raíces calculadas con bisección y el número de iteraciones necesarias para obtenerlas.

***Método de Newton-Raphson. Ej 1:** La raiz de la función se encuentra en 1.074400 tras n = 4 iteranciones.*

***Método de Newton-Raphson. Ej 2:** La raiz de la función se encuentra en 0.641700 tras n = 4 iteranciones.*

***Método de Newton-Raphson. Ej 2:** La raiz de la función se encuentra en 0.641700 tras n = 3 iteranciones.*

***Método de Newton-Raphson. Ej 3:** La raiz de la función se encuentra en 0.365200 tras n = 6 iteranciones.*

***Método de Newton-Raphson. Ej 3:** La raiz de la función se encuentra en 1.921800 tras n = 7 iteranciones.*

***Método de Newton-Raphson. Ej 3:** La raiz de la función se encuentra en 3.563700 tras n = 8 iteranciones.*

***Método de Newton-Raphson. Ej 4:** La raiz de la función se encuentra en 0.714200 tras n = 3 iteranciones.*

Imagen 8: Raíces calculadas con Newton-Raphson y el número de iteraciones necesarias para obtenerlas.

Estos valores corroboran que la velocidad de convergencia del método de Newton-Raphson es mayor que la del método de la bisección. El número de iteraciones para llegar a la raíz en una función es siempre menor para Newton-Raphson . Dependiendo de la función se ha aplicado un criterio de parada (error mínimo) distinto que se ha cumplido en todos los casos y se ha conseguido aproximar la raíz de la función que se nos pedía.