Solucion de Ecuaciones No Lineales

Rubn Cuadra A01019102

***Abstract*—Manual de usuario: 'noLineal.m', el objetivo de este documento es documentar la implementacion del metodo Newton-Raphson y metodo de Secante como metodos numericos**

I. Introduccion

El codigo consiste en un archivo llamado de la misma manera que la funcion el cual recibe 4 argumentos y nos regresa 3 respuestas

```
function [x,Error,i]=noLineal(x0,eps,maxit,metodo)
```

x0 puede ser un arreglo de 2 posiciones o 1 numero. Esto depende de acuerdo al **metodo** que usemos, donde True(1) es Secante y False(0) es Newton-Raphson

eps se refiere a un numero, por lo general decimal, que sirve como criterio de convergencia, nos indica el rango que tiene el metodo para llegar a la solucion

maxIt es un numero entero que definira el numero de iteraciones en las que se debe obtener el resultado, en caso de que el codigo no llegue a la solucion basado en el eps nos devolvera un error

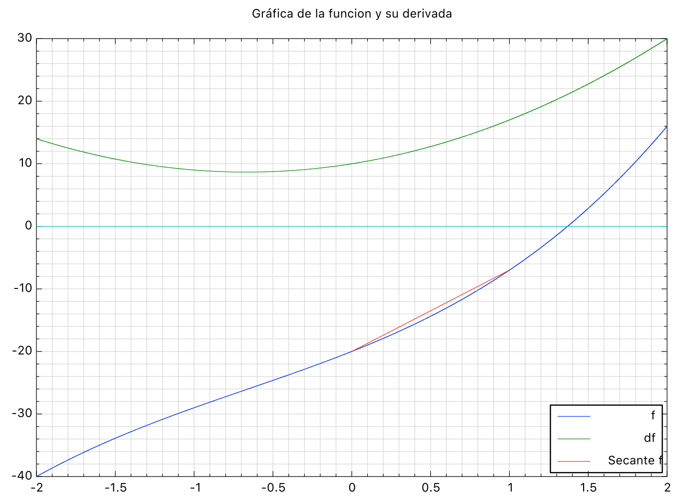

Fig. 1. Ecuaciones de f.m y df.m respectivamente, la secante se encuentra desde x0=0 a x1=1

Archivos .m respectivamente

```
function y = f(x)
        y = x^3+2*x^2+10*x-20;
end

function y = df(x)
        y = 3*x^2+4*x+10;
end
```

II. Comprension del algoritmo

A. Newton-Raphson

Se busca llevar la ecuacion $f(x) = 0$ a la forma $x = g(x)$ para que$g'(\bar{x}) = 0$

Entonces, en la funcion pasamos un valor inicial de X, un criterio de convergencia EPS y el numero de iteraciones maximas para llegar a la solucion, esto debido a la probabilidad de oscilar asi como de diverger. La formula usada es:

$$x = x_0 - \frac{F(x_0)}{F'(x_0)} \tag{1}$$

Paso 1
Inicializar valores a regresar con datos de Error
Paso 2
Iniciar ciclo en iter=0 hasta que iter=maxiters
Paso 3
Aplicar formula 1 en una variable designada temp
Paso 4
Comparar temp-x_0 Si es menor al criterio de convergencia, regresar exito y temp como respuesta
Paso 5
Si llegamos aqui convertir x_0 en temp y repetir paso 3, sumando 1 al iterador
Paso 6
Regresar valores de error si se acabaron las iteraciones

B. Secante

En el metodo de secante obtenemos 2 valores x_0 y x_1 asi como un EPS y de igual manera el numero maximo de iteraciones, con esto calculamos el proximo punto. Usamos la formula:

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \tag{2}$$

Paso 1
Inicializar valores a regresar con datos de Error
Paso 2
Iniciar ciclo en iter=0 hasta que iter=maxiters
Paso 3
Aplicar formula 1 en una variable designada next
Paso 4
Comparar next-x_1 Si es menor al criterio de convergencia, regresar exito y next como respuesta
Paso 5
Si llegamos aqui convertir $x_0 = x_1$ y $x_1 = next$ en temp y repetir paso 3, sumando 1 al iterador
Paso 6
Regresar valores de error si se acabaron las iteraciones

III. Ejemplo

Mandamos a llamar la funcion desde un archivo main.m. Asi mandamos a llamar la funcion con metodo secante si queremos usar de valores x iniciales 0 y 1, queriendo un EPS de 0.001 y un maximo numero de iteraciones de 10

```
[x,Error, i]=noLineal([0 1], 0.001, 10,true)
```

El resultado seria:

```
x =  1.36881288868898
Error = 0
i =  3
```

Si ahora queremos mandar a llamar la funcion con el metodo Newton-Raphson la llamariamos con los mismos valores pero la x inicial basta con ser un numero, en este caso 1

```
[x,Error, i]=noLineal(1, 0.001, 10,false)
```

y nos devolveria

```
x =  1.36933647058824
Error = 0
i =  2
```

Donde en ambos casos Error=0 significa que NO hay error. i = 2 — i = 3 nos define el numero de iteraciones respectivamente. x es la solucion

IV. Conclusion

No todos los metodos numericos son perfectos, las funciones pueden oscilar o diverger, ya que ambos son de segundo grado son bastante rapidos pero claro, Newton suele llegar mas rapido si logramos obtener la derivada de la funcion, que suele llevar mas procesador.

[0]Los acentos no se pudieron agregar por cuestion de la codificacion