Raices

Martínez Navallez Daniel Isac

Octubre 2018

1 Introducción

En este artículo veremos dos diferentes métodos para encontrar raices de una función; método de bisección y Newton-Raphson.

Se presentarán dos algoritmos en Fortran90 para correr ambos métodos.

2 Método de Bisección

Este es un método que utiliza un algoritmo para encontrar las raices de una función basado en el teorema del valor intermedio. Para ésto, se toma que la función es continua en un intervalo cerrado definido como [a,b]. La función toma los valores f(a) y f(b), si estos dos tienen signos opuestos existe un valor cero, para un punto c, que está intermedio de este intervalo [a,b], por lo que f(c)=0, garantizando la existencia de almenos una raiz para esta función.

Para esto se deben de asegurar ciertos puntos:

1.Se debe de tener la seguridad de que la función es continua en el intervalo [a,b].
2.Se verifica que f(a)f(b)<0
3.Se calcula el punto medio c en el intervalo [a,b] y se evalua f(c), si el resultado es igaul a cero se ha encontrado la raiz. En caso de que no, se verifica si hubo cambio con f(a) o f(b).
4.Se redefine el intervalo como [a,c] o [c,b] dependiendo de los cambios de signo con el intervalo original.
5.Con el nuevo intervalo se procede a realizar el mismo procedimiento cuantas veces sea necesario dependiendo de tu error requerido.

A continuación se muestra un programa escrito en fortran para el método de bisección.

```
program biseccion

implicit none
real:: a,b,c,error,f
error=1.0e-06
write(*,*)"Proporciona dos números en donde se encuentre la raiz"

10 read(*,*) a,b
15 if (f(a)*f(b) .lt. 0) then
c=(a+b)/2.0
else
write(*,*)"Ingresa otro valor donde se encuentren las raices"
goto 10
end if
if (f(a)*f(c) .lt. 0) then
b=c
else
a=c
end if
if (abs(b-a) .gt. error) goto 15

write(*,*)"La raiz es:",c
end program

real function f(x)
implicit none
real::x
f=x**3-x-2
end function
```

En el siguiente programa se declaran cinco variables: "a" y "b" para el intervalo cerrado; "c" será el punto intermedio entre dichon intervalo; a la variable "error" se le asignará el valor que se quiere para éste; "f" se usará ára declarar la función a la que se le quiere encontrar su raiz.

Se le pide al usario ingrese el intervalo [a,b] en el que se encuentre una raiz, para cumplir los puntos anteriores se debe verificar que al evaluar f en a y b, su multiplicació debe ser menor a cero, garantizando que que tienen signos opuestos, garantizado esto obtenemos el valor intermedio c=(a+b)/2, de no cumplir a y b con los requerimientos se le pide al usuario que ingrese unos valores correctos donde se encuentre la raiz.

Con los valores correctos de a y b se sigue con el procedimiento ahora, obtenido el punto intermedio asignado a "c", procedemos a realizar la siguiente multiplicación f(a)*f(c)<0, de ser esto cierto, entonces, ahora b=c, de ser esta expresión falsa, entonces, ahora a=c.

Terminando con el proceso comprovamos el error comprovando que el valor absoluto de (b-) sea menor que el error, de lo contrario se repetite el proceso de bisección hasata que se cumpla dicha desigualdad.

Para ya termianr el programa creamos una función f(x), para este caso la función será: $x^3 - x - 2$

2.1 Método de Newton Raphson

Este método resulta ser más efectivo que Bisección y fue propuesto por Isaac Newton y Joseph Raphson, y al igual que el método de bisección sirve para aproximar por lo menos una de las raices de una funcón detrminanda.

Para inciar con el método se neceita tener a la función f(x), su derivada f'(x) y un punto cercano a una de las raices x0, teniendo ésto lo siguiente es fácil. Obtendremos un nuevo valor que es una mejor aproximación a la raiz de la siguiente manera:

$$x = x0 - \frac{f(x0)}{f'(x0)} \tag{1}$$

Para cada nuevo valor que se genere realizaremos otra iteracción para obtener un valor cada vez más cercano al de la raiz.

A continuación se muestra un código para el método de Newton-Raphson en FORTRAN 90:

```
Program NewtonRap

  IMPLICIT NONE

  !Declaramos las siguientes variables
  !x: será una mejor a proximación a la raiz
  !x1: un punto en el eje x cercano a la raiz
  !error: corresponde a un porcentaje deseado para la exactitud de la raiz
  !f: es para la función que se desea aproximar su raiz
  !f1: es la derivada de f

  REAL:: x,x1,error,f,f1

  !Pedimo al usario ingrese nuestro punto cercano a la raiz x1 y lo leemos
  WRITE(*,*)'Ingresa un punto cerca de la raiz'
  READ(*,*) x1

  !Usaremos un loop para llevar acabo la operación las veces necesarias
  !El loop será deteneido por un if en donde se muestra que el error debe de ser
```

```
!menor a un porcentaje ya señalado
DO
   !Se evalua la función y su derivada en x1
   f=x1**3-x1-2
   f1=3*x1**2-1
   !Aquí tenemos el método de Newton que nos da una mejor aproximación a
   !nuestra raiz con la siguiente operación
   x=x1-(f/f1)
   !Definimos el error
   error= abs((x-x1)/x)*100
   x1=x
   !Si este if cumple con lo señalado entonces se detendrá el loop, de otra
   !forma sería infinito
   if(error<0.0000001)exit
END DO

!Imprimimos en pantanlla el valor de x correspondiente a la raiz
PRINT*,'La raiz es:',x

END PROGRAM NewtonRap
```

3 Conclusión

Se presentaron dos métodos, de los más utilizados, para la aproximación de raices, de los dos encontramos una forma de ejecutarlos por medio de fortran, encontrando una manera más sencilla y rápida de aproximar a la raiz de una función dada.