

Actividad 3

Jorge Iván García Bojorquez

October 2018

1 Método de Bisección

Es uno de los métodos más sencillos y utilizados para encontrar raíces de ecuaciones de una sola variable. El método consiste en tomar un intervalo cerrado $[a, b]$ donde evaluamos la función en sus extremos, si $f(a) \cdot f(b) < 0$ entonces el valor 0 es un valor que se encuentra entre $f(a)$ y $f(b)$.

1.1 Algoritmo

Tomamos dos puntos en un intervalo cerrado $[a, b]$.

Si $f(a) \cdot f(b) < 0$. Buscamos el punto medio entre ambos números:

$$n = \frac{a + b}{2} \quad (1)$$

Evalúamos n en la función para obtener $f(n)$.

Comparamos $f(n)$ con $f(a)$ y $f(b)$:

Si $f(a) \cdot f(n) < 0$ redefinimos el intervalo haciendo $b=n$, y repetimos el proceso.

Si $f(b) \cdot f(n) < 0$ redefinimos el intervalo haciendo $a=n$, y repetimos el proceso.

De esta manera buscaremos los puntos medios de cada intervalo que vamos definiendo hasta converger en una raíz.

1.2 Programa

Este es el código del programa que escribí para el método de bisección, el código es para la ecuación $x^3 - x - 2$.

```
!Funcion a resolver
FUNCTION F(i)
  real, intent(in) :: i
  real :: F
  f=(i**3)-i-2.0
end function F
```

```

program biseccion
  implicit none
  !a y b extremos del intervalo n es el punto medio,
  !fn,fa,fb funcion valuada en n,a,b
  real::F,fn,fa,fb,n,a,b

  print*,"Introduzca los extremos [a,b] para encontrar raiz de x^3 -x -2"
  read*,a
  read*,b
  fa=f(a)
  fb=f(b)

  if(fa*fb<0)then
    do
      n=(a+b)/2.0
      fn=f(n)
      !reasignacion de fronteras
      if(fn*fa<0) then
        b=n
        print*,n
        if(abs(b-a)<.0001)exit !criterio de paro
      else
        a=n
        fa=fn
        print*,n
        if(abs(b-a)<.0001)exit
      end if
    end do

    print *, "n=",n,"f(n)=",f(n)
  else
    print*,"Los extremos tienen el mismo signo favor de correr el
    programa con distintos valores"
  end if
end program

```

Este programa usa como criterio de paro que la distancias entre las fronteras sea menor a .0001.

Si se deseara resolver otra funcion solo se deberia cambiar la escrita en FUNCTION F(n).

2 Newton Raphson

Es un metodo abierto que no garantiza la convergencia global. La unica manera de garantizar la convergencia es seleccionando un numero cercano a la raíz buscada. El metodo de newton consiste en la iteracion de la siguiente función:

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

De donde con el nuevo valor obtenido (x_{n+1}) calculamos una nueva x y así sucesivamente hasta que converja a un valor.

2.1 Programa

Aquí presento el programa que escribí para encontrar raíces utilizando este método. El criterio de paro utilizado fue un error estimado que calculábamos con cada iteración que hacía el programa. Se calculó de la siguiente manera: $err = \frac{x_{n+1} - x_n}{x_{n+1}}$.

De igual manera al programa anterior en Function F(n) está la función que resuelve este programa, el único cambio es que agregamos otra función llamada Fp(n) que es la derivada de la función que resuelve este programa, si desea cambiar la función no le pide cambiar la derivada también.

```
function F(n) !Funcion
    implicit none
    real,intent(in)::n
    real::F
    F=n**3 - n -2
end function F
function Fp(n) !Derivada
    implicit none
    real,intent(in)::n
    real::Fp
    Fp=3*n**2 - 1
end function Fp
program nr
    implicit none
    !x buscada, xi= x inicial, err = error
    real::x,xi,err
    real::F,Fp
    print*,"Indique su x inicial para aplicar newton-rhaphso a la funcion  x^3-x-2"
    read*,xi
Do
    !xi - F(xi)/F'(xi)
    x=xi-(F(xi)/Fp(xi))
    err=abs((x-xi)/x)*100
    print*,"X= ",x,"err= ",err
    xi=x
    if(err<.00001)exit
end do
end program nr
```