Triseccion

Sebastian Angarita, Hector Rodriguez, Aldemar Ramirez

12/8/2019

## Problema

Hayar la raiz de una función en un rango [a,b] a través del método de trisección.

## Solución

**Lenguaje de programación:** R

## Función principal-método de bisección

Parametros:

-f <- funcion

-xa <- a en el rango [a,b] donde se busca la raiz

-xb <- b en el rango [a,b] donde se busca la raiz

* tol <- tolerancia minima que debe tener la funcion

Valores de retorno:

-a\_c <- valor de a durante las iteraciones

-b\_c <- valor de b durante las iteraciones

-m\_c1 <- valor del primer corte m1 durante las iteraciones

-m\_c2 <- valor del segundo corte m2 durante las iteraciones

-i <- iteraciones necesarias para llegar a la tolerancia

-dx <- Error estimado.

triseccion = function(f, xa, xb, tol){  
 if( sign(f(xa)) == sign(f(xb)) ){ stop("f(xa) y f(xb) tienen el mismo signo") }  
 a = min(xa,xb)  
 b = max(xa,xb)  
 k = 0  
   
 #k es el numero de iteraciones  
 a\_c <- c()  
 b\_c <- c()  
 m\_c1 <- c()  
 m\_c2 <- c()  
 dx <- c()  
 errorRelativo = c()  
 repeat{  
 m1 = a + (1/3)\*(b-a)  
 m2 = m1 - a  
 m2 = m2 + m1  
 m\_c1 = c(m\_c1, m1)  
 m\_c2 = c(m\_c2, m2)  
   
 a\_c = c(a\_c,a)  
 b\_c = c(b\_c,b)  
   
 dx = c(dx,(b-a)/3) #error estimado  
   
 if( f(m1)==0 ){   
 lista = list("a\_c" = a\_c, "b\_c" = b\_c, "m\_c1" = m\_c1, "m\_c2" = m\_c2 , "i" = k, "resultado" = m1,  
 "dx" = dx)  
 return(lista)   
 }  
 if( f(m2)==0 ){   
 lista = list("a\_c" = a\_c, "b\_c" = b\_c, "m\_c1" = m\_c1, "m\_c2" = m\_c2 , "i" = k, "resultado" = m2,  
 "dx" = dx)  
 return(lista)   
 }  
   
 if( sign(f(a)) != sign(f(m1)) ){  
 b = m1  
 }   
 if( sign(f(m1)) != sign(f(m2)) ){  
 a = m1  
 b = m2  
 }   
 if( sign(f(m2)) != sign(f(b)) ){  
 a = m2   
 }  
   
 k = k+1  
   
 #until  
 if( dx[k] < tol ){  
 lista = list("a\_c" = a\_c, "b\_c" = b\_c, "m\_c1" = m\_c1, "m\_c2" = m\_c2 , "i" = k, "resultado" = m2,  
 "dx" = dx)  
 return(lista)  
 break;  
 }  
 } #repeat  
}

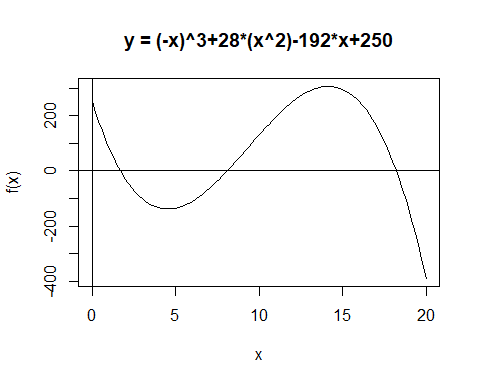
La función basicamente toma un intervalo [a,b] de la función, tal que f(a)\*f(b)<0, y basandose en el teorema de los valores intermedios se sabe que al ser la funcion continua hay al menos un raiz de la funcion en este intervalo.

En la anterior implementacion a diferencia a bisección se realizar tres cortes.

## Implementación

**Gráfica de la función**

f = function(x) (-x)^3+28\*x^2-192\*x+250  
curve(f, 0,20); abline(h=0, v=0) #gráfico para decidir un intervalo  
title(main="y = (-x)^3+28\*(x^2)-192\*x+250")



A través de la gráfica se puede ver que la función f tiene 3 raices.

**Resultados de las raices**

resultados <- triseccion(f, 0, 5, 1e-7)  
  
cat("Cero de f en [0,5] es approx: ",resultados$resultado, "con error <=",   
 resultados$dx[resultados$i],"\n")

## Cero de f en [0,5] es approx: 1.696277 con error <= 3.871762e-08

resultados <- triseccion(f, 5, 10, 1e-7)  
  
cat("Cero de f en [5,10] es approx: ",resultados$resultado, "con error <=",   
 resultados$dx[resultados$i],"\n")

## Cero de f en [5,10] es approx: 8.09322 con error <= 3.871762e-08

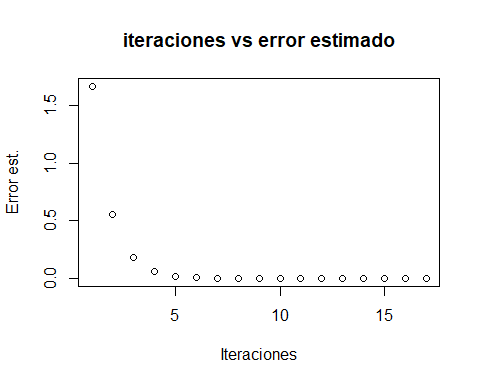
resultados <- triseccion(f, 15, 20, 1e-7)  
  
cat("Cero de f en [15,20] es approx: ",resultados$resultado, "con error <=",   
 resultados$dx[resultados$i],"\n")

## Cero de f en [15,20] es approx: 18.2105 con error <= 3.871762e-08

A comparacion del metodo de biseccion encontramos que el error es mucho mas cercano a la tolerancia.

**Grafica de iteraciones vs error estimado**

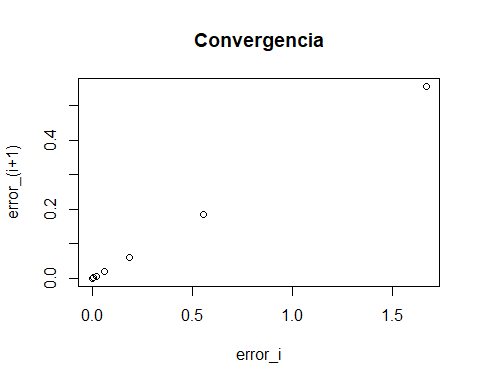
plot(x = 1:length(resultados$a\_c), y = resultados$dx,  
 xlab = "Iteraciones", ylab = "Error est.",   
 main = "iteraciones vs error estimado")



El error estimado siempre converge a cero. En comparación al metodo de biseccion encontramos que el metodo de trisección necesita de menor cantidad de iteraciones.

**Grafica de m(i) vs m(i+1) - Para el primer corte**

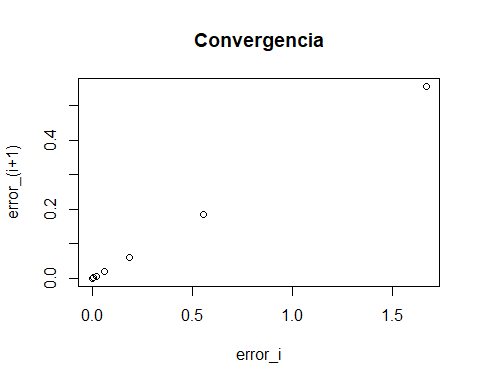
m\_i = resultados$dx[-length(resultados$m\_c1)]  
m\_i2 = resultados$dx  
m\_i2 = m\_i2[-1]  
  
plot(x =m\_i, y = m\_i2, xlab = "error\_i", ylab = "error\_(i+1)",  
 main = "Convergencia")



De acuerdo con la grafica el metodo tiene una convergencia lineal en el primer corte.

**Grafica de m(i) vs m(i+1) - Para el segundo corte**

m\_i = resultados$dx[-length(resultados$m\_c2)]  
m\_i2 = resultados$dx  
m\_i2 = m\_i2[-1]  
  
plot(x =m\_i, y = m\_i2, xlab = "error\_i", ylab = "error\_(i+1)",  
 main = "Convergencia")

 De acuerdo con la grafica el metodo tiene una convergencia lineal en el segundo corte exactamente parecida al primer corte y al metodo de bisección.

## Caso especial: dos raices en el intervalo

resultados <- triseccion(f, 1, 20, 1e-7)  
  
cat("Cero de f en [0,20] es approx: ",resultados$resultado, "con error <=",  
 resultados$dx[resultados$i],"\n")

## Cero de f en [0,20] es approx: 8.09322 con error <= 4.904232e-08

El metodo de triseccion se acerca solamente a una de las raices, sin embargo con este metodo tomo el mas cercano a la mitad del rango [a,b].