Método de la bisección aplicado a la función $f(x) = 5^x - 5$

Claudia Ballester Niebla, Cathaysa Pérez Quintero y Carlos Herrera Carballo

14 de mayo de 2014

Facultad de Matemáticas
Universidad de La Laguna

Índice

Índice

Índice

Índice

Índice

Índice

Objetivos

- **Objetivo principal**: Implementación con Python del método de bisección.

Objetivos

- **Objetivo principal**: Implementación con Python del método de bisección.
- **Objetivo específico**: Cómo se aproximan las raíces de una función, mediante el método de bisección.

Fundamentos teóricos. Método de bisección.

El método de la bisección se basa en dos teoremas, el de Bolzano y el del Valor Intermedio.

Fundamentos teóricos. Método de bisección.

El método de la bisección se basa en dos teoremas, el de Bolzano y el del Valor Intermedio.

Definición

- *Teorema de Bolzano: Sea $f(x)$ una función continua en un intervalo [a,b] tal que $f(a) * f(b) < 0$, entonces existe un punto c perteneciente al intervalo (a,b) tal que $f(c) = 0$.*

Fundamentos teóricos. Método de bisección.

El método de la bisección se basa en dos teoremas, el de Bolzano y el del Valor Intermedio.

Definición

- *Teorema de Bolzano: Sea $f(x)$ una función continua en un intervalo [a,b] tal que $f(a) * f(b) < 0$, entonces existe un punto c perteneciente al intervalo (a,b) tal que $f(c) = 0$.*
- *Teorema del Valor Intermedio: Sea $f(x)$ una función continua en un intervalo [a,b], tal que $f(a) < f(b)$ entonces, para todo k tal que $f(a) < k < f(b)$ existe x_0 que pertenece al intervalo (a,b) tal que $f(x_0) = k$.*

Fundamentos teóricos. Método de bisección.

Definición

Dados dos puntos a y b, tal que $f(a)$ *y* $f(b)$ *tengan signos distintos debe tener, al menos, una raíz en el intervalo [a,b]. Este método divide el intervalo en dos utilizando un tercer punto c=* $\frac{a+b}{2}$*. De esta forma, se darán dos posibilidades:* $f(a)$ *y* $f(c)$*, ó* $f(c)$ *y* $f(b)$ *tienen distinto signo. Se aplica este método al subintervalo donde ocurre el cambio de signo. Así se realizará tantas veces como sea necesario para conseguir la máxima precisión.*

Descripción de los experimentos.

- **Experimento 1:** Con el fin de verificar la precisión del algoritmo propuesto, se ejecuta el programa con los mismos valores y la misma función tres veces. De esta forma, si los resultados coinciden, significará que es exacto.

Descripción de los experimentos.

- **Experimento 1:** Con el fin de verificar la precisión del algoritmo propuesto, se ejecuta el programa con los mismos valores y la misma función tres veces. De esta forma, si los resultados coinciden, significará que es exacto.
- **Experimento 2:** Se ejecutará el algoritmo con distintas funciones para así observar que es válido para cualquier $f(x)$.

Descripción de los experimentos.

- **Experimento 1:** Con el fin de verificar la precisión del algoritmo propuesto, se ejecuta el programa con los mismos valores y la misma función tres veces. De esta forma, si los resultados coinciden, significará que es exacto.
- **Experimento 2:** Se ejecutará el algoritmo con distintas funciones para así observar que es válido para cualquier $f(x)$.
- **Experimento 3:** En este tercer experimento, se ejecutará el programa con distintos intervalos para verificar la existencia de raíces en los mismos.

Descripción de los experimentos.

- **Experimento 1:** Con el fin de verificar la precisión del algoritmo propuesto, se ejecuta el programa con los mismos valores y la misma función tres veces. De esta forma, si los resultados coinciden, significará que es exacto.
- **Experimento 2:** Se ejecutará el algoritmo con distintas funciones para así observar que es válido para cualquier $f(x)$.
- **Experimento 3:** En este tercer experimento, se ejecutará el programa con distintos intervalos para verificar la existencia de raíces en los mismos.
- **Experimento 4:** Se observa cómo varían las soluciones obtenidas con distintos valores de tolerancia de error.

Descripción de los experimentos.

- **Experimento 1:** Con el fin de verificar la precisión del algoritmo propuesto, se ejecuta el programa con los mismos valores y la misma función tres veces. De esta forma, si los resultados coinciden, significará que es exacto.
- **Experimento 2:** Se ejecutará el algoritmo con distintas funciones para así observar que es válido para cualquier $f(x)$.
- **Experimento 3:** En este tercer experimento, se ejecutará el programa con distintos intervalos para verificar la existencia de raíces en los mismos.
- **Experimento 4:** Se observa cómo varían las soluciones obtenidas con distintos valores de tolerancia de error.
- **Experimento 5:** Para determinar el rendimiento del algoritmo, se introduce un cronómetro en el mismo, de forma que imprima el tiempo que tarda la CPU de la computadora en ejecutar el programa.

Resultados.

Experimento 1

Intento	Resultado
1	0.969
2	0.969
3	0.969

Table: Tabla experimento 1

Resultados.

Experimento 2

Funcion	**Resultado**
5^x	Error
$5x - 10$	1.969
$3x^2 - 1$	Error

Table: Tabla experimento 2

Resultados.

Experimento 3

Intervalo	Resultado
(-3,2)	1.023
(-2,3)	1.008
(2,5)	Error

Table: Tabla experimento 3

Resultados.

Experimento 4

Tolerancia	Resultado
0.01	0.996
0.02	0.992
0.001	1.000

Table: Tabla experimento 4

Resultados.

Experimento 5

Intento	Tiempo (segundos)
1	$2.8133 * 10^{-}5$
2	$2.4080 * 10^{-}5$
3	$3.1948 * 10^{-}5$

Table: Tabla experimento 5

Algoritmo empleado

```
import time
import timeit
def f(x):
  return (5**x)-5
def biseccion(a,b,tol):
  c=(a+b)/2.0
  while((f(c)!=0.000001) and (abs(b-a)>tol)):
    if f(a)*f(c)<0.000001:
      b=c
    else:
      a=c
    c=(a+b)/2.0
  return c

import sys
if (len(sys.argv)==4):
  A=float(sys.argv[1])
  B=float(sys.argv[2])
  TOL=float(sys.argv[3])
else:
  A=float(raw_input("Introduzca el extremo a del intervalo: "))
  B=float(raw_input("Introduzca el extremo b del intervalo: "))
  TOL=float(raw_input("Introduzca la tolerancia del error que desee: "))
if f(A)*f(B)<0.000001:
  start=time.time()
  raiz=biseccion(A,B,TOL)
  finish=time.time()-start
  print "El tiempo de ejecución es:"
  print finish
  print "La raíz aproximada de la función escogida es: %4.3f" %raiz
else:
  print "En ese intervalo no existe raíz, por favor vuelva a ejecutar el programa con otros valores"
```

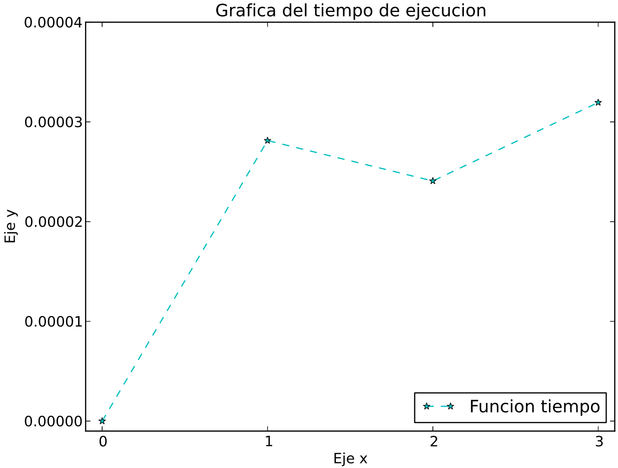

Figure: Gráfica del tiempo

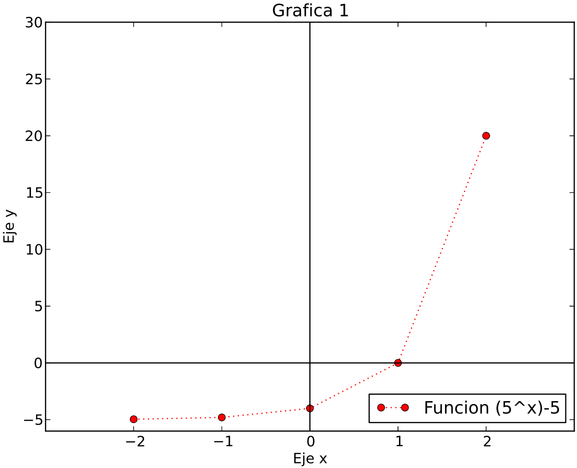

Figure: Gráfica de la funcion

Bibliografía

Guía docente de la asignatura: Técnicas Experimentales. (2013)
$http://eguia.ull.es/matematicas/query.php?codigo = 299341201$

Spivak, M. -Calculus, Ed. Cambridge, 2006; Ed. Reverté, 1981 [BULL]

Demostración del Método de Bisección. (2008)
$http://www.ma3.upc.edu/users/carmona/teaching/clases/08-09/trabajos/$

Python para todos. -Raúl González Duque.
$http://campusvirtual.ull.es/1314/pluginfile.php/197675/mod_resource/content/4/Primera_Parte/General2012/Python_para_tod$

The beamer class. User Guide for version 3.26.
$http://campusvirtual.ull.es/1314/pluginfile.php/197674/mod_resource/content/1/beameruserguide.pdf$