Bisección de $f(x) = cos(\pi x)$
Informe Científico-Técnico

Carmen Laura Martín González y David Tómas Montesdeoca Flores

Grupo (1)

Técnicas Experimentales. 1^{er} *curso.* 2^{do} *semestre*

Lenguajes y Sistemas Informáticos

Facultad de Matemáticas

Universidad de La Laguna

La Laguna, 11 de Mayo de 2014

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Motivación y objetivos

Este método , se utiliza para resolver ecuaciones de una variable, está basado en el "Teorema de los Valores Intermedios"(TVM), en el cual se establece que toda función continua 'f', en un intervalo cerrado $[a, b]$, toma todos los valores que se hallan entre $f(a)$ y $f(b)$, de tal forma que la ecuación $f(x) = 0$ tiene una sola raíz que verifica $f(a) * f(b) < 0$.

1.1. ¿Qué es el método de bisección?

El método de la bisección es un algoritmo que nos permite encontrar una raíz de la función que tengamos. Su funcionamiento es muy similar al de la búsqueda binaria, sólo que estamos utilizando valores continuos (∞) en lugar de discretos (finitos). Vamos a asumir que las funciones con las que trabajamos son continuas.

Sabemos que si la función $f(x)$ atraviesa el eje x, en la intersección existe una raíz, por lo que antes de la raíz $f(x) > 0$ y después $f(x) < 0$. Entonces, teniendo dos puntos 'a' y 'b' tales que $f(a)$ y $f(b)$ tienen signos opuestos, sabemos que debe haber una raíz en el intervalo $[a, b]$.

1.2. ¿Cómo funciona su algoritmo?

El algoritmo funciona de la siguiente forma:

- Primero tomamos el punto medio entre 'a' y 'b', al cual llamaremos $c()$. Si el valor de $f(c) = 0$, o está suficientemente cerca.
- Segundo tomamos a 'c' como el valor de la raíz.

En caso contrario:

- Primero reemplazamos a 'a' o 'b' por 'c', de acuerdo al signo de $f(c)$, de tal forma que los signos de $f(a)$ y $f(b)$ sigan siendo diferentes.

Finalmente, repetimos el método con el nuevo intervalo.

1.3. Ventaja del método de bisección.

La principal ventaja de este método es que es muy eficaz aunque menos que el método de Newton, ya que, está garantizado que el método converge si los valores de $f(a)$ y $f(b)$ son de signos contrarios. La convergencia de este método es lineal, y su error absoluto después de n iteraciones es: $\frac{|b-a|}{2^n}$

Capítulo 2

Fundamentos teóricos

Este trabajo trata sobre la bisección de la función $f(x) = cos(\pi * x)$. Para realizarlo utilizamos programas como LaTeX para la realizacion del pdf, Beamer para realizar la presentación y Python para calcular las raices de la función nombrada anteriormente y crear su grafica correspondiente.

2.1. LaTeX

LaTeX es un paquete de macros que permite a los autores componer e imprimir su trabajo con la mayor calidad tipográfica posible, usando un formato profesional predefinido. LaTeX due escrito originalmente por Leslie Lamport. Emplea el formateador de LaTeX como motor de composición.

LaTeX es un programa de ordenador creado por Donald E. Knuth. Sirve para componer texto y fórmulas matemáticas.

2.2. Beamer

Beamer fue desarrollado por Till Tantau para la presentación de su tesis doctoral, como un conjunto de macros que facilitara el uso de otras clases, como seminar o prosper.

Beamer es una clase de LaTeXque permite crear diapositivas. Aunque también puede ser utilizada para crear presentaciones dinamicas que pueden ser proyectadas desde el ordenador.

2.3. Python

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses "Monty Python". Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible.

Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos

Capítulo 3

Procedimiento experimental

Este capítulo ha de contar con seccciones para la descripción de los experimentos y del material. También debe haber una sección para los resultados obtenidos y una última de análisis de los resultados.

3.1. Descripción de los experimentos

bla, bla, etc.

3.2. Descripción del material

('default', 'Feb 27 2014 20:00:17')
Linux-3.2.0-61-generic-i686-with-Ubuntu-12.04-precise
('Linux', 'FISICA-PC', '3.2.0-61-generic', '93-Ubuntu SMP Fri May 2 21:33:33 UTC 2014', 'i686', 'i686')
2.7.3
Genuine Intel(R) CPU 2160 @ 1.80GHz
GenuineIntel
1200.000 Hz
1024 KB

3.3. Resultados obtenidos

bla, bla, etc.

3.4. Análisis de los resultados

bla, bla, etc.

Figura 3.1: Ejemplo de figura

Figura 3.2: Ejemplo de figura con gráfico

Tiempo (± 0.001 s)	**Velocidad (± 0.1 m/s)**
1.234	67.8
2.345	78.9
3.456	89.1
4.567	91.2

Cuadro 3.1: Resultados experimentales de tiempo (s) y velocidad (m/s)

Nombre	Edad	Nota
Pepe	24	10
Juan	19	8
Luis	21	9

Cuadro 3.2: Mi primer cuadro de datos

Capítulo 4

Conclusiones

Apéndice A

Título del Apéndice 1

A.1. Algoritmo XXX

```
################################################################################
# solucion .py
################################################################################
#
# David Tomas Montesdeoca Flores y Carmen Laura Martin Gonzalez
#
# 11-05-2014
#
# Es un programa que calcula la biseccià³n de la funcion takata en un intervalo que
# que debe introducir el usuario junto con un margen de error. Ademà¡s de calcular
# las raices de la funcià³n, calcula el tiempo de ejecucià³n y dibuja la grà!‘fica en
# otro intervalo distinto pedido al usuario.
#
################################################################################
#! /usr/bin/python
#!encoding: UTF-8
import time
import math
import matplotlib.pyplot as pl
import numpy as np
import numpy as np
import sys

def f(x):
  return (math.cos(x*math.pi))

def biseccion(a,b,e):
  c=(a+b)/2.0
  while((f(c)!=0.00000001) and (abs(b-a)>e)):
    if f(a)*f(c)<0.00000001:
      b=c
    else:
      a=c
    c=(a+b)/2.0
  return c
```

```
A=float(raw_input("Introduzca el extremo inferior del intervalo (a) en el que se desea buscar la raiz: "))
B=float(raw_input("Introduzca el extremo superior del intervalo (b) en el que se desea buscar la raiz: "))
E=float(raw_input("Introduzca el margen de error a partir del cual no afecte demasiado a sus calculos: "))
if f(A)*f(B)<0.00000001:
  start=time.time()
  r=biseccion(A,B,E)
  finish=time.time()-start
  print "La raÃz que se ha calculado en ese intervalo de forma aproximada es:%4.3f" %r
  print "El tiempo que ha tardado en ejecutarse el cÃ¡lculo en segundos es:"
  print finish

else:
  print "En el intervalo introducido no existe raÃz, lo sentimos."

x=int(raw_input("Introduzca el maximo valor de x para la representacion: "))
lista=[]
for i in range(x):
  y=math.cos(math.pi*i)
  lista.append(y)


pl.figure(figsize=(8,6), dpi=80)

X = np.linspace(-x, x, 256, endpoint=True)
C = np.cos(X*np.pi)
S = 0*(X)

pl.plot(X,C, color="cyan", linewidth=2.5, linestyle="-", label="Coseno")
pl.plot(X,S, color="black", linewidth=1.5, linestyle="-", label="Eje X")
pl.legend(loc='upper left')
pl.xlim(X.min()*1.1,X.max()*1.1)
pl.ylim(C.min()*1.1,C.max()*1.1)
pl.yticks([-1, 0, +1])
pl.title("Representacion grafica")
pl.savefig("cos.eps", dpi=72)
pl.show()
##############################################################################################
```

A.2. Algoritmo YYY

```
/##############################################################################################
 # prct12_1 .py
 ##############################################################################################
 #
 # David Tomas Montesdeoca Flores y Carmen Laura Martin Gonzalez
 #
 # 11-05-2014
 #
 # Este programa nos muestra la versiÃ³n de Linux que estamos utilizando y lo guarda
 # en un fichero de texto.
```

```
 #
 ###############################################################################
 #!encoding: UTF-8
#!usr/bin/python

import os
import platform

def SOFTinfo():
    softinfo={}
    softinfo={'Several':platform.uname(), 'S.O':platform.platform(), 'Pythons Version':platform.python_versi
    return softinfo

def CPUinfo():
# infofile on Linux machines:
  infofile = '/proc/cpuinfo'
  cpuinfo = {}
  if os.path.isfile(infofile):
    f = open(infofile, 'r')
    for line in f:
      try:
        name, value = [w.strip() for w in line.split(':')]
      except:
        continue
      if name == 'model name':
        cpuinfo['CPU type'] = value
      elif name == 'cache size':
        cpuinfo['cache size'] = value
      elif name == 'cpu MHz':
        cpuinfo['CPU speed'] = value + ' Hz'
      elif name == 'vendor_id':
        cpuinfo['vendor ID'] = value
    f.close()
  return cpuinfo

if __name__ == '__main__':
  softinfo = SOFTinfo()
  for keys in softinfo.keys():
    print softinfo[keys]
  cpuinfo = CPUinfo()
  for keys in cpuinfo.keys():
    print cpuinfo[keys]

  print "Introduzca el  ##########################################################################################no
  nombre_fichero = raw_input();
  f = open(nombre_fichero, "w")
  for keys in softinfo.keys():
    if type (softinfo[keys]) is list:
      f.write('\n'.join(softinfo[keys]))
    else:
      f.write(str(softinfo[keys]))
      f.write('\n')
  for keys in cpuinfo.keys():
    if type(cpuinfo[keys]) is list:
```

```
      f.write('\n'.join(cpuinfo[keys]))
    else:
      f.write(str(cpuinfo[keys]))
      f.write('\n')
  f.close()


platform.uname()
platform.platform()
platform.python_version()
platform.python_build()
 ################################################################################
```

Apéndice B

Título del Apéndice 2

B.1. Otro apendice: Seccion 1

Texto

B.2. Otro apendice: Seccion 2

Texto

Bibliografía

[1] Anita de Waard. A pragmatic structure for research articles. In *Proceedings of the 2nd international conference on Pragmatic web*, ICPW '07, pages 83–89, New York, NY, USA, 2007. ACM.

[2] J. Gibaldi and Modern Language Association of America. *MLA handbook for writers of research papers*. Writing guides. Reference. Modern Language Association of America, 2009.

[3] G.D. Gopen and J.A. Swan. The Science of Scientific Writing. *American Scientist*, 78(6):550–558, 1990.

[4] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison–Wesley Pub. Co., Reading, MA, 1986.

[5] Coromoto León. *Diseño e implementación de lenguajes orientados al modelo PRAM*. PhD thesis, 1996.

[6] Guido Rossum. Python library reference. Technical report, Amsterdam, The Netherlands, The Netherlands, 1995.

[7] Guido Rossum. Python reference manual. Technical report, Amsterdam, The Netherlands, The Netherlands, 1995.

[8] Guido Rossum. Python tutorial. Technical report, Amsterdam, The Netherlands, The Netherlands, 1995.

[9] ACM LaTeX Style. http://www.acm.org/publications/latex_style/.