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

Carmen Laura Martín González y David Tómas Montesdeoca Flores

*Grupo (*1*)*

Técnicas Experimentales. 1er *curso.* 2do *semestre*

Lenguajes y Sistemas Informáticos

Facultad de Matemáticas

Universidad de La Laguna

La Laguna, 11 de Mayo de 2014

Índice general

1. Motivación y objetivos **1**
- 1.1. ¿Qué es el método de bisección? 1
- 1.2. ¿Cómo funciona su algoritmo? 1
- 1.3. Ventaja del método de bisección. 2

2. Fundamentos teóricos **3**
- 2.1. LaTeX 3
- 2.2. Beamer 3
- 2.3. Python 3

3. Procedimiento experimental **4**
- 3.1. Descripción de los experimentos 4
- 3.2. Descripción del material 4
- 3.3. Resultados obtenidos 4
- 3.4. Análisis de los resultados 5

4. Conclusiones **7**

A. Apéndice A: Código para la bisección y Hardware **9**
- A.1. Codigo Python 9
- A.2. Codigo Python para el Hardware 10

B. Apéndice B: Código para la representación de funciones **13**
- B.1. Explicacion del codigo Python 13

Bibliografía **15**

Índice de figuras

3.1. Ejemplo de figura 5
3.2. Ejemplo de figura con gráfico 5

Índice de cuadros

3.1. Datos de f(x) 5
3.2. Algunas raices de f(x) 6

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 Fue 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

3.1. Descripción de los experimentos

En los experimentos, el tiempo de cálculo depende del ordenador, los intervalos, y el error que utilicemos, por ello explicaremos eso junto con el código.

Ver Apéndice A.1. (Codigo Python)

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

Ver Apéndice A.2.(Codigo Python para el Hardware)

3.3. Resultados obtenidos

Al realizar la bisección de la función, obtenemos una serie de datos. Un claro ejemplo es el cálculo del tiempo de error. Otros de estos datos son las raíces calculadas, y otros son simples puntos calculados al sustituir un valor de x en f(x). Con estos puntos podemos realizar las siguientes tabla y gráfica.

Figura 3.1: Ejemplo de figura

Figura 3.2: Ejemplo de figura con gráfico

X	Y
- 2	1
-1.5	0
-1	-1
-0.5	0
0	1
0.5	0
1	-1
1.5	0
2	1

Cuadro 3.1: Datos de f(x)

3.4. Análisis de los resultados

Como sabemos, lo que conseguimos con la bisección es calcular las raíces de la función tratada. Dependiendo del intervalo dado y el margen de error, este valor se aproximará de mejor o peor modo, pero después de utilizar este programa realizado en Python podemos dar una tabla de raíces de nuestra función, que al ser periodica tiene infinitas.

X	Y
- 2.5	0
-1.5	0
-0.5	0
0.5	0
1.5	0
2.5	0

Cuadro 3.2: Algunas raices de f(x)

Capítulo 4

Conclusiones

Para concluir tenemos que mencionar varios puntos del trabajo:

- Python esta en movimiento y en pleno desarrollo, pera ya es una realidad y una interesante opcion para realizar todo tipo de programas que se ejecuten en cualquier maquina. El equipo de desarrollo esta trabajando de manera cada vez mas organizada y cuentan con el apoyo de una comunidad que esta creciendo rapidamente.
- Algunas de las ventajas que tiene LaTeX son que, crea con facilidad estructuras complejas como pies de pagina, indices, tablas, etc, y para la rama de ciencias tiene principalmente una ventaja primordial que es que se pueden crear con facilidad formulas matematicas.
- Beamer tiene todas las ventajas de LaTeX. Su presentacion en PDF es estandar y portable. Ademas tiene unos estilos predefinidos con botones de navegacion, tablas de contenidos, pies de pagina informativos, etc.
- El metodo de la Biseccion converge lentamente, lo que genera la propagacion de error por la cantidad de operaciones e iteraciones necesarias para que el metodo converja.

Finalmente, al complementar los elementos mencionados anteriormente y centrandonos en el calculo de las raices mediante el metodo de biseccion, hemos aprendido a crear un fichero de texto y una presentacion a modo de diapositivas, con un cierto grado de exigencia. Tambien hemos aprendido a ser independientes buscando la informacion correcta sobre el tema propuesto y a la vez aprender a trabajar en equipo. Bajo nuestro punto de vista este es el objetivo de la programacion de este trabajo, ya que, esto nos serviria para los trabajos futuros que tengamos que realizar.

Apéndice A

Apéndice A: Código para la bisección y Hardware

A.1. Codigo **Python**

```
#####################################################################################
# solucion .py
#####################################################################################
#
# David Tomas Montesdeoca Flores y Carmen Laura Martin Gonzalez
#
# 11-05-2014
#
# Es un programa que calcula la biseccion de la funcion f(x)=cos(pix) en un intervalo que
# que debe introducir el usuario junto con un margen de error. Ademas de calcular
# las raices de la funcion, calcula el tiempo de ejecucion y dibuja la grafica 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 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 raiz que se ha calculado en ese intervalo de forma aproximada es:%4.3f" %r
  print "El tiempo que ha tardado en ejecutarse el calculo en segundos es:"
  print finish

else:
  print "No podemos asegurar que en el intervalo introducido exista raiz, 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. Codigo **Python** para el Hardware

```
/#################################################################################
 # prct12_1 .py
 #################################################################################
 #
 # David Tomas Montesdeoca Flores y Carmen Laura Martin Gonzalez
 #
 # 11-05-2014
 #
```

```
 # Este programa nos muestra la version 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 nombre del fichero para almacenar los resultados: "
  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

Apéndice B: Código para la representación de funciones

B.1. Explicacion del codigo **Python**

Como dijimos en el apendice anterior, se trata del codigo de un programa realizado en python para calcular la biseccion de f(x)=Cos(πx) en un intervalo introducido por el usuario juton con el margen de error. ademas calcula el tiempo de ejecucion y dibuja la grafica en otro intervalo que desee el usuario.

Primero importamos las librerias necesarias. En este caso las necesitamos para calcular el tiempo, para las funciones trigonometricas (Coseno), y para representar graficas.

```
#! /usr/bin/python
#!encoding: UTF-8
import time
import math
import matplotlib.pyplot as pl
import numpy as np
import sys
```

Definimos las dos funciones que vamos a usar principalmente, una para calcular los valores de f(x) al sustituir un x en ella, y otra para calcular la biseccion.

```
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 continuacion pedimos al usuario que ingrese los extremos superior e inferior del intervalo y la tolerancia o margen de error. Establecemos una condicion para poder realizar la biseccion, y es que exista un cambio de signo entre f(A) y f(B), asi que f(A)*f(B)¡0. Cuando esto se cumple el programa se ejecuta, empezando a contar el tiempo y parando este çronometro"despues del calculo.

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

else:
  print "No podemos asegurar que en el intervalo introducido exista raiz, lo sentimos."
```

Tras hallar la biseccion deseada, se pide al usuario un valor de x para representar la funcion en un intervalo de -x a x. Con la ayuda de un bucle for conseguimos dar valores desde el 0 hasta x y guardarlos en un vector en los valores de y. De este modo podemos tener los puntos a representar.

```
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)
```

Por ultimo representamos la funcion. Elejimos el color y el ancho de la linea. Como nos interesa ver el corte con el eje de abcisas, representamos una recta x multiplicada por 0, asi todos sus puntos valdran 0. Le ponemos un titulo a la grafica, la guardamos y la mostramos.

```
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)
```

Y aqui termina el codigo.

Bibliografía

[1] Gerald and Wheatley. Analisis Numerico con Aplicaciones.

[2] David kincaid y Ward Cheney. Las Matematicas del Calculo Cientifico.

[3] Wikipedia: es.wikipedia.org/wiki/Metodo_de_biseccion#Algoritmo

[4] Juegos de Logica: www.juegosdelogica.com/numero_$\pi.htm$

[5] Punto Q:

http://riunet.upv.es/handle/10251/5617

http://riunet.upv.es/handle/10251/2063

http://riunet.upv.es/handle/10251/5567

[6] Carmona Teaching: http://www.ma3.upc.edu/users/carmona/teaching/clases/08-09/trabajos/metodo%20biseccion.pdf

[7] GitHub

Puede acceder al Repositorio online en: https://github.com/alu0100833218/Equipo_1_A