SERIES NUMÉRICAS
Función trigonométrica: sin(x)

Jorge Antonio Herrera Alonso

Elizabeth Hernández Martín

Yessica Sabrina Gómez Buso

Grupo (2 | F)

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

Lenguajes y Sistemas Informáticos

Facultad de Matemáticas

Universidad de La Laguna

La Laguna, 10 de mayo de 2014

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Motivación y objetivos

A lo largo de este curso hemos aprendido a implementar diferentes códigos en *Python*, los cuales han logrado generar nuestra curiosidad por saber más. Esto nos permitió ir más allá y poder fusionar dicho lengutaje de programación con el procesador de texto LaTeX y una clase de este, el *Bearmer*, que utilizamos para realizar presentaciones. A partir de todos ellos, hemos conseguido llevar a cabo esta memoria.

1.1. Objetivo principal:

Profundizar nuestros conocimientos con el lenguaje de programación *Python*, el procesador de texto LaTeX y el creador de presentaciones *Bearmer* sobre el estudio de las series de Taylor.

1.2. Objetivo específico:

Hallar la aproximación de $f(x) = sin(x)$ mediante el método de Taylor, el error cometido y el estudio del tiempo de programación.

Capítulo 2

Fundamentos teóricos

2.1. Historia

Brook Taylor nació el 18 de agosto de 1685 en Edmonton. Hijo de John Taylor, del Parlamento de Bifrons, Kent, y de Olivia Tempest (hija de Sir Nicholas Tempest).

En "Los métodos de incrementación directa e inversa"de Taylor (1715) agregaba a las matemáticas una nueva rama llamada ahora «El cálculo de las diferencias finitas» , e inventó la integración por partes y descubrió la célebre fórmula conocida como la Serie de Taylor, la importancia de esta fórmula no fue reconocida hasta 1772, cuando Lagrange proclamó los principios básicos del Cálculo Diferencial. Taylor también desarrolló los principios fundamentales de la perspectiva en "Perspectivas Lineales"(1715). En su Methodus Incrementorum Directa et Inversa (Londres, 1715) desarrolló una nueva parte dentro de la investigación matemática, que hoy se llama cálculo de las diferencias finitas. Junto con "Los nuevos principios de la perspectiva lineal". Taylor da cuenta de un experimento para descubrir las leyes de la atracción magnética (1715) y un método no probado para aproximar las raíces de una ecuación dando un método nuevo para logaritmos computacionales (1717).

Brook Taylor murió en Somerset House el 29 de diciembre de 1731.

2.2. Cálculo de la serie de Taylor

Sea $f(x)$ una función definida en un intervalo que contiene al punto a, con derivadas en todos los órdenes.

El polinomio de primer grado

$$p_1(x) = f(a) + f'(a)(x - a)$$

tiene el mismo valor que $f(x)$ en el punto $x = a$ y también, como se comprueba fácilmente, la misma derivada que $f(x)$ en este punto. Su gráfica es una recta tangente a la gráfica de $f(x)$ en el punto a.

Es posible elegir un polinomio de segundo grado,

$$p_2(x) = f(a) + f'(a)(x - a) + \tfrac{1}{2}f''(a)(x - a)^2$$

tal que en el punto $x = a$ tenga el mismo valor que $f(x)$ y también valores iguales para su primera y segunda derivada. Su gráfica en el punto a se acercará a la de $f(x)$ más que la anterior. Es natural esperar que si construimos un polinomio que en $x = a$ tenga las mismas n primeras derivadas que $f(x)$ en el mismo punto, este polinomio se aproximará más a $f(x)$ en los puntos x próximos a a. Así obtenemos la siguiente igualdad aproximada, que es la fórmula de Taylor:

$$f(x) = f(a) + f'(a)(x-a) + \frac{1}{2!}f''(a)(x-a)^2 + ... + \frac{1}{n!}f^n(a)(x-a)^n$$

El segundo miembro de esta fórmula es un polinomio de grado n en $(x-a)$. Para cada valor de x puede calcularse el valor de este polinomio si se conocen los valores de $f(a)$ y de sus n primeras derivadas.

Para el caso de la función $sin(x)$ el Polinomio de Taylor sería de la siguiente forma:

$$f(x) = sin(a) + cos(a)(x-a) - \frac{1}{2!}sin(a)(x-a)^2 - \frac{1}{3!}cos(a)(x-a)^3 +$$

$$\frac{1}{4!}sin(a)(x-a)^4 + ... + \frac{1}{n!}f^n(a)(x-a)^n$$

Capítulo 3

Procedimiento experimental

3.1. Descripción de los experimentos

El experimento llevado a cabo en esta memoria ha consistido en la realización de varios códigos en lenguaje *Python*. Los algoritmos implementados que solucionan dichos códigos estiman la aproximación $f(x) = sin(x)$ mediante el método de Taylor, solicitando el grado del polinomio de Taylor, el punto central y el punto x donde se evalua dicho polinomio.

3.2. Descripción del material

Los materiales requeridos para la realización del trabajo han sido:

1. una nube
2. El sistema operativo empleado es Bardinux, adaptación de la distribución de Linux 'kubuntu' a las necesidades en cuanto a Software de los miembros de la comunidad universitaria de la ULL.
3. CPU speed 800.000Hz
4. cache size 3072 KB

3.3. Resultados obtenidos

3.4. Análisis de los resultados

bla, bla, etc.

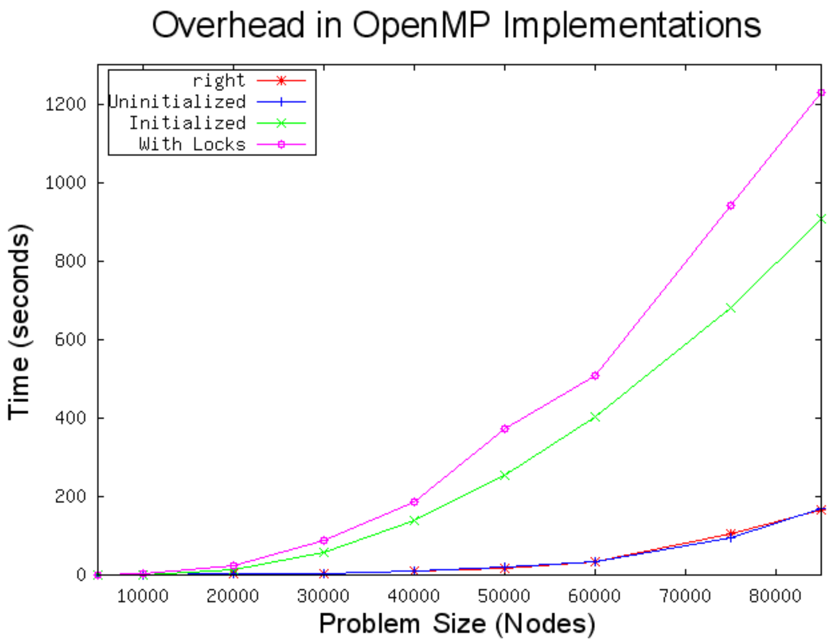

Figura 3.1: Ejemplo de figura

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)

Capítulo 4

Conclusiones

bla, bla, bla, etc.

Apéndice A

Título del Apéndice 1

A.1. Algoritmo XXX

```
#SE PONE EL CÓDIGO DE LOS PROGRAMAS.

import math
from sympy import *
import time

def factorial(numero):
  factorial = 1
  while(numero > 1):
    factorial = factorial * numero
    numero = numero - 1
  return factorial

def taylor(x, numero, centro):
  c = Symbol('c')
  f = sin(c)
  suma = f.evalf(subs={c: centro})
  for i in range (1,numero + 1):
    f_i = diff(f, c)
    v = f_i.evalf(subs={c: centro})
    s= (v / factorial(i)) * ((x - centro) ** i)
    suma = suma + s
    f = f_i
  return taylor

def taylor1(x, numero, centro):
  c = Symbol('c')
  f = sin(c)
  func = f.evalf(subs={c: centro})
  suma = func
  for i in range (1,numero + 1):
    f_i = diff(f, c)
    v = f_i.evalf(subs={c: centro})
    s= (v / factorial(i)) * ((x - centro) ** i)
```

```
    suma = suma + s
    f = f_i
  error = func - suma
  print 'El valor de la funcion original sin(%f) es igual a %f ' % (centro, func)
  print 'El Polinomio de Taylor de grado n=%d en el punto centro c=%f evaluada en el punto x=%f es igual a %
  print 'El Error de la funcion original con el Polinomio de Taylor es: error=%f' % error
  return taylor1

def lista():
  f=open("Taylor.tex", 'w')
  f.write('Grado del polinomio (n), Punto Central (c), Punto de evaluación (x), Aproximación, Error, Tiempo
  f.write('==================================================================================================
  f.close()
  for i in range (0,numero+1):
    p=[]
    inicio = time.time()
    p=p+[modulo.taylor(x, i, centro)]
    fin = time.time
    tiempo_total = fin - inicio
    p=p+[tiempo_total]
    f.write(str(p))
    f.write("\n")
  modulo.taylor1(x, i, centro)
  f.close()
```

A.2. Algoritmo YYY

```
/###################################################################################
 # Fichero .h
 ###################################################################################
 #
 # AUTORES
 #
 # FECHA
 #
 # DESCRIPCION
 #
 ##################################################################################
```

Apéndice B

Título del Apéndice 2

B.1. Apendice2

B.2. Otro apendice: Seccion 2

`Texto`

Bibliografía

[1] http://www.buscabiografias.com/bios/biografia/verDetalle/2155/Brook Taylor.

[2] 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.

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

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

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

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

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

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

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

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