Universidad de La Laguna

Polinomio Interpolador de Newton

$f(x) = sin(x)$

Francisco Javier Reyes Sánchez
Zoilo González García

25 de abril de 2014

Facultad de Matemáticas
Universidad de La Laguna

Índice

Índice

Índice

Índice

Índice

Índice

Interpolador Polinómico

- En análisis numérico, la interpolación polinomica es una técnica de interpolación de un conjunto de datos o de una función por un polinomio.
- El objetivo de esta técnica es el de hallar un polinomio que permita hallar aproximaciones de valores desconocidos para la función.

Cálculo del Polinomio Interpolador de Newton

Definición: Sea f_n una variable discreta de n elementos y sea x_n otra variable discreta de n elementos los cuales corresponden a la imagen y la abcisa de los datos que se quieren interpolar:

$$f(x_k) = f_k, \quad k = 1, ..., n$$

El polinomio de grado n-1 resultante de aplicar este método, tendrá la forma:

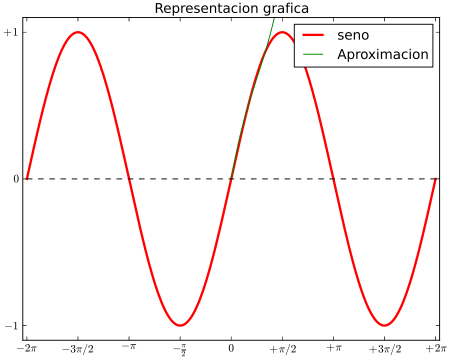
$$\sum_{i=0}^{n-1} a_j \left(\prod_{j=1}^{j-1} (x - x_i)\right)$$

Los coeficientes a_j son las llamadas diferencias divididas.

$$a_j = f[x_0, x_1, ..., x_{j-1}, x_j,]$$

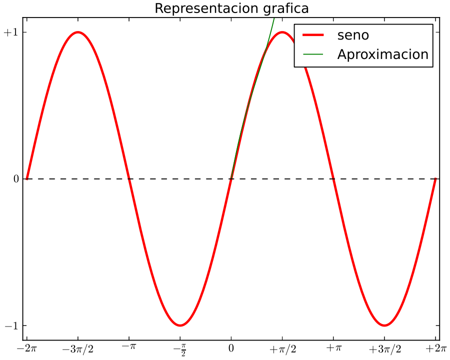
Descripción de los experimentos

- Seno



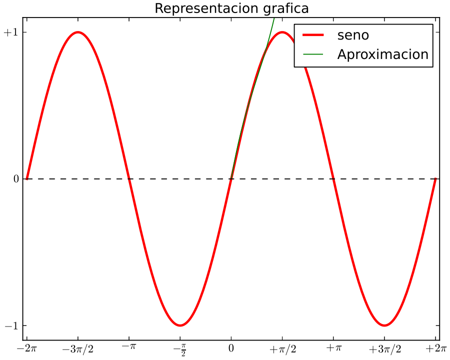
Descripción de los experimentos

- Seno
- Polinomio interpolador



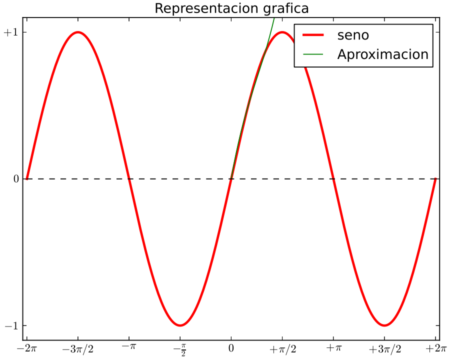
Descripción de los experimentos

- Seno
- Polinomio interpolador
- Diferencias divididas



Descripción de los experimentos

- Seno
- Polinomio interpolador
- Diferencias divididas
- Comprobación



Descripción del material

Análisis de los resultados obtenidos

```
El valor aproximado por el polinomio para x=0.785398
y=0.683013
El seno de 0.785398 es
sen(x)=0.707107


El valor aproximado por el polinomio para x=0.628319
y=0.575349
El seno de 0.628319 es
sen(x)=0.587785


El valor aproximado por el polinomio para x=0.523599
y=0.500000
El seno de 0.523599 es
sen(x)=0.500000
```

A.1. Algoritmo sin(x)

```
##########################################################################################
# seno.py
##########################################################################################
##!/usr/bin/python
#!encoding: UTF-8
from math import pi
from math import sin
#definimos una función, con la que obtendremos el valor de los diferentes nodos
empleados, en nuestro caso la funcion es el seno(x)def seno(x):
  f=sin(x)
  return f

if __name__=="__main__":#Prueba para la funcion seno(x)
  n=int(raw_input("Introduzca un punto de prueba: "))
  print "El seno de dicha función es: %.3f" %seno(n)
```

A.2. Algoritmo Diferencias Divididas

```
###############################################################################
# difdiv.py
###############################################################################

#!/usr/bin/python
#!encoding: UTF-8
from math import pi
from math import sin
import seno
def difdiv(x):
  c=[]#lista que guardará cada una de las diferencias divididas.
  c=c+[0]#ya que la primera diferencia dividida es cero (f(0))
  c1=(seno.seno(x[1])-seno.seno(x[0]))/(x[1]-x[0])
  c=c+[c1]
  c12=(seno.seno(x[2])-seno.seno(x[1]))/(x[2]-x[1])#diferencia dividida entre x1 y x2
  c2=(c12-c1)/(x[2]-x[0])
  c=c+[c2]
  c23=((seno.seno(x[3])-seno.seno(x[1]))/(x[3]-x[2]))
  c13=((c23-c12)/(x[3]-x[1]))
  c3=((c13-c2)/(x[3]-x[0]))
  c=c+[c3]
  return c
```

A.3. Algoritmo Interpolador de Newton

```
##################################################################################
# interpoladornewton.py
##################################################################################
#!/usr/bin/python
#!encoding: UTF-8
from math import pi
from math import sin
import difdiv
l=[]
s=[]
n=(0, pi/6, pi/3, pi/2)#tomamos cuatro nodos equidistantes en el intervalo (0,pi/2)
l=difdiv.difdiv(n)
print "El polinomio interpolador de Newton en el intervalo (0, pi/2), para los puntos
\nx0=0\nx1=pi/6\nx2=pi/3\nx3=pi/2\nes: "
print "P(x)= (%f) + (%f)*(x-%f) + (%f)*(x-%f)(x-%f) + (%f)*(x-%f)(x-%f)(x-%f)" %(l[0],
l[1], n[0], l[2], n[0], n[1], l[3], n[0], n[1], n[2])
```

A.4. Algoritmo para comprobacion

```
##################################################################################
# comprobacion.py
##################################################################################
#!/usr/bin/python
#!encoding: UTF-8
from math import pi
from math import sin
import seno
valores=(pi/4, pi/5, pi/6,2*pi/5)
for i in valores:
  print "El valor aproximado por el polinomio para x=%f es" %i
  y=(0.311104)*(i**3)+(-0.733021)*(i**2)+(1.253448)*i
  print "y=%f" %y
  print "El seno de %f es" %i
  f=seno.seno(i)
  print "sen(x)=%f" %f
  print "\n"
```

A.5. Algoritmo de representación gráfica

```
##########################################################
####representacionseno.py
##########################################################
#! encoding: utf8

import matplotlib.pylab as pl
import numpy as np

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

pl.subplot(1,1,1)

X = np.linspace(-np.pi*2, np.pi*2, 256, endpoint=True)
S = np.sin(X)
E = X*0
Y = np.linspace(0, np.pi/2, 150, endpoint=True)
A = ((0.311104)*(Y**3)+(-0.733021)*(Y**2)+(1.253448)*Y)

pl.plot(X,S, color="red", linewidth=2.5, linestyle="-", label="seno")
pl.plot(Y,A, color="green", label="Aproximacion")
pl.plot(X,E, color="black", linewidth=1, linestyle="--")

pl.legend(loc='0')

pl.xlim(X.min()*1.02,X.max()*1.02)
pl.xticks([-2*np.pi,-3*np.pi/2,-np.pi, -np.pi/2, 0, np.pi/2, np.pi, 3*np.pi/2, 2*np.pi],
          [r'$-2\pi$',r'$-3\pi/2$',r'$-\pi$',r'$-\frac{\pi}{2}$',r'$0$',r'$+\pi/2$',r'$+\pi$', r'$+3\pi/2$',r'$+2\pi$']
          )

pl.ylim(-1.1,1.1)
pl.yticks([-1, 0, +1],
          [r'$-1$', r'$0$', r'$+1$']
         )

pl.title("Representacion grafica")

pl.savefig("representacionseno.eps", dpi=72)

pl.show()
```

Conclusiones

- Aspecto Teórico La realización de este trabajo nos ha sido muy útil pues hemos obtenido nuevos conocimientos matemáticos de gran utilidad tales como el cáculo de diferencias divididas o el método de inerpolación polinómica de Newton para la aproximación de una función en un intervalo.
- Aspecto Computacional Desde el punto de vista informático esta tarea nos ha servido para adquirir destrezas que sin duda nos ayudarán en un futuro en la realización de informes y presentaciones asi como la elaboración de algorimos para resolver determinados problemas.

Bibliografía

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

ACM LaTeX Style.
http://es.wikipedia.org/wiki/N