



Universidad  
de La Laguna

# *Polinomio Interpolador de Newton*

$$f(x) = \sin(x)$$

Francisco Javier Reyes Sánchez

25 de abril de 2014

Facultad de Matemáticas  
Universidad de La Laguna

# Índice

## *Interpolador Polinómico*

# Índice

*Interpolador Polinómico*

*Cálculo del Polinomio Interpolador de Newton*

# Índice

*Interpolador Polinómico*

*Cálculo del Polinomio Interpolador de Newton*

*Ejercicios*

Una subsección

Creación de diapositivas

Otra subseccion

# Índice

*Interpolador Polinómico*

*Cálculo del Polinomio Interpolador de Newton*

*Ejercicios*

Una subsección

Creación de diapositivas

Otra subseccion

*Algoritmos*

# Índice

*Interpolador Polinómico*

*Cálculo del Polinomio Interpolador de Newton*

*Ejercicios*

Una subsección

Creación de diapositivas

Otra subseccion

*Algoritmos*

*Bibliografía*

## *Interpolador Polinómico*

- En análisis numérico, la interpolación polinómica 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 abscisa de los datos que se quieren interpolar:

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

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.

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



## *Título de la diapositiva*

Texto de la diapositiva

## *Diapositivas*

### *Definition*

Un ejemplo de definición

### *Example*

- Primero



## *Diapositivas*

### *Definition*

Un ejemplo de definición

### *Example*

- Primero
- Segundo



## *Diapositivas*

### *Definition*

Un ejemplo de definición

### *Example*

- Primero
- Segundo
- Tercero



## *Diapositivas*

### *Definition*

Un ejemplo de definición

### *Example*

- Primero
- Segundo
- Tercero
- Cuarto



## *Este es otro Título*

*Defnición*

*Otra definición*

*Ejemplo*

*1. Primero*



# *Este es otro Título*

*Defnición*

*Otra definición*

*Ejemplo*

- 1. Primero*
- 2. Segundo*



## 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"
```

## *Bibliografía*



Guido Rossum.

Python library reference.

Technical report, Amsterdam, The Netherlands, The Netherlands, 1995.



ACM LaTeX Style.

<http://es.wikipedia.org/wiki/N>