

# Polinomio interpolador de Newton

Iciar González Alonso, Serezade González Torres

Universidad de La Laguna

16 de mayo de 2014

Facultad de Matemáticas  
Universidad de La Laguna

## 1 Motivación y objetivos

# Índice

- 1 Motivación y objetivos
- 2 Fundamentos teóricos
  - Historia
  - Teorema

- 1 Motivación y objetivos
- 2 Fundamentos teóricos
  - Historia
  - Teorema
- 3 Procedimiento experimental
  - Descripción de los experimentos
  - Resultados obtenidos
  - Análisis de los resultados

- 1 Motivación y objetivos
- 2 Fundamentos teóricos
  - Historia
  - Teorema
- 3 Procedimiento experimental
  - Descripción de los experimentos
  - Resultados obtenidos
  - Análisis de los resultados
- 4 Conclusiones

- 1 Motivación y objetivos
- 2 Fundamentos teóricos
  - Historia
  - Teorema
- 3 Procedimiento experimental
  - Descripción de los experimentos
  - Resultados obtenidos
  - Análisis de los resultados
- 4 Conclusiones
- 5 Apéndice 1. Programa Python
  - Almacenamiento de  $x$
  - Almacenamiento de  $f(x)$
  - Coeficientes
  - Partes literales del polinomio
  - Valores

- 1 Motivación y objetivos
- 2 Fundamentos teóricos
  - Historia
  - Teorema
- 3 Procedimiento experimental
  - Descripción de los experimentos
  - Resultados obtenidos
  - Análisis de los resultados
- 4 Conclusiones
- 5 Apéndice 1. Programa Python
  - Almacenamiento de  $x$
  - Almacenamiento de  $f(x)$
  - Coeficientes
  - Partes literales del polinomio
  - Valores
- 6 Bibliografía

Se comentará la finalidad del trabajo. Distinguiremos entre:

- Objetivo principal: implementación con Python de las series de potencias de Newton. Concretamente hemos trabajado con dicho programa para aproximar a través de este método el valor de la función  $f(x) = e^x$ .



Se comentará la finalidad del trabajo. Distinguiremos entre:

- Objetivo principal: implementación con Python de las series de potencias de Newton. Concretamente hemos trabajado con dicho programa para aproximar a través de este método el valor de la función  $f(x) = e^x$ .
- Objetivos específicos: aprender a utilizar el método de Newton para aproximar funciones a través de series de potencias, centrándonos en la aproximación de la función nombrada anteriormente.

En este capítulo se pasará a desarrollar de forma teórica el tema a tratar.

- Historia.

En este capítulo se pasará a desarrollar de forma teórica el tema a tratar.

- Historia.
- Teorema.

## Definición

*La serie de potencias de Newton se trata de un teorema descubierto hacia 1664-1665 como se puede deducir, fue desarrollado por Isaac Newton. El descubrimiento de la generalización de la serie binómica fue un gran resultado en sí mismo, pero además este desencadenó nuevos estudios. Este teorema fue publicado por Wallis en 1685, pero este atribuyó el descubrimiento a Newton.*

## Definición

*La forma general del polinomio interpolante de Newton para  $n + 1$  datos  $(x_0, f(x_0)), (x_1, f(x_1)), \dots, (x_n, f(x_n))$  es:*

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2) + \dots \\ + a_n(x - x_0)(x - x_1)(x - x_2) \dots (x - x_{n-1})$$

*Los coeficientes  $a_i$  se obtienen calculando un conjunto de cantidades denominadas diferencias divididas. La notación para las diferencias divididas de una función  $f(x)$  están dadas por:*

$$f[x_i, x_{i+1}, x_{i+2}, \dots, x_{i+j-1}, x_{i+j}] = \frac{f[x_{i+1}, x_{i+2}, \dots, x_{i+j}] - f[x_i, x_{i+1}, \dots, x_{i+j-1}]}{x_{i+j} - x_i}$$

## Definición

*A continuación se enumerará y se desarrollarán los pasos que se han seguido en la elaboración del programa de las series de Newton en Python.*

- *Descripción de los experimentos.*

## Definición

*A continuación se enumerará y se desarrollarán los pasos que se han seguido en la elaboración del programa de las series de Newton en Python.*

- *Descripción de los experimentos.*
- *Resultados obtenidos.*

## Definición

*A continuación se enumerará y se desarrollarán los pasos que se han seguido en la elaboración del programa de las series de Newton en Python.*

- *Descripción de los experimentos.*
- *Resultados obtenidos.*
- *Análisis de los resultados*



- 1 Función del experimento:  $f(x) = e^x$

# Descripción de los experimentos

- 1 Función del experimento:  $f(x) = e^x$
- 2 Intervalo del experimento:  $[0,1]$

# Descripción de los experimentos

- 1 Función del experimento:  $f(x) = e^x$
- 2 Intervalo del experimento:  $[0,1]$
- 3 Nodos equidistantes: 0.00, 0.25, 0.50, 0.75, 1.00

# Descripción de los experimentos

- 1 Función del experimento:  $f(x) = e^x$
- 2 Intervalo del experimento:  $[0,1]$
- 3 Nodos equidistantes: 0.00, 0.25, 0.50, 0.75, 1.00
- 4 Valor  $x = 0.43$

# Resultados obtenidos

| x    | $f(x)$             | Coeficientes         | Parte literal |
|------|--------------------|----------------------|---------------|
| 0.0  | 1.0                | 1.0                  | 1.0           |
| 0.25 | 1.2840254166877414 | 1.1361016667509656   | 0.43          |
| 0.5  | 1.6487212707001282 | 0.6453634985971632   | 0.0774        |
| 0.75 | 2.117000016612675  | 0.12219975773607672  | -0.005418     |
| 1.0  | 2.718281828459045  | 0.026030877832598165 | 0.00173376    |

Table: Datos y resultados del programa

## Definición

*En el intervalo  $[0,1]$ , tomando los nodos  $x_0 = 0$ ,  $x_1 = 0.25$ ,  $x_2 = 0.5$ ,  $x_3 = 0.75$ ,  $x_4 = 1$ ,  $f(0.43) = 1.53785$  y el valor del polinomio en  $x = 0.43$  es  $1.53725$ . Con lo cual, se comete un error de  $6 \times 10^{-4}$ . El número de nodos ha de ser directamente proporcional al tamaño del intervalo.*

- 1 Obtención de la aproximación de la función  $f(x) = e^x$  mediante el polinomio de Newton implementado en Python.
- 2 Mejora del conocimiento en las estructuras iterativas de programación.
- 3 Cálculo del error cometido por el programa.
- 4 Aprender a aproximar funciones a través de las series de potencias.
- 5 Profundizar en los fundamentos teóricos del método.

## Definición

*En este capítulo se pasará a mostrar el código mediante el cual se implemento el polinomio de Newton para la funcion  $f(x) = e^x$ .*



# Almacenamiento de x

```
x = [ ]
for i in range (0,5):
    if i == 0:
        xi = 0.00
    else:
        xi = xi + 1/4.0
    x += [xi]

print 'Lista con los valores de x:'
print x # Imprime las x
```

# Almacenamiento de $f(x)$

```
fx = [ ]  
for i in range (0,5):  
    fxi = e**x[i]  
    fx += [fxi]  
  
print 'Lista con los valores de f(x):'  
print fx # Imprime las fx
```

# Coeficientes

```
d = [ ]
for i in range (0,5):
    if i == 0:
        di = fx[i]
    else:
        di = (fx[i] - fx[i-1])/(x[i] - x[i-1])
    d += [di]

a = [d[0], d[1], (d[2]-d[1])/(x[2]-x[0]), (d[3]-d[2]+d[1]-d[2])/(x[3]-x[0]), (d[4]-
d[3]+d[2]-d[3]+d[2]-d[3]+d[2]-d[1])/(x[4]-x[0])]

print 'Lista con los coeficientes:'
print a # Imprime los coeficientes del polinomio
```

# Partes literales

```
y = 0
l = [ ]
valorx = 0.43
for i in range (0,5):
    li = 1
    if i == 0:
        li = 1
    else:
        for j in range (0,i):
            li = li*(valorx-x[j])
            j+=1
        l = l + [li]

print 'Lista con la parte literal:'
print l #Imprime la parte literal del polinomio con el valor de x predeterminado
```

```
valorpol = 0

for i in range(0,5):
    valorpol += a[i] * l[i]

print 'Valor del polinomio en x = 0.43: %.35f' %valorpol # Imprime el valor del polinomio
con la x que le hemos dado
print ' '
real = e**valorx
print 'Valor de f(0.43) = e^(0.43) : %.35f' %real # Imprime el valor real de la función con
el valor de x
print ' '
print 'Error cometido en la aproximación del polinomio: %.35f' %(abs(valorpol-real))
# Imprime el valor absoluto del error cometido con el polinomio
```



"Solo Ciencia". <http://www.solociencia.com/cientificos/isaac-newton-teorema-binomio.htm>



"ULPGC Informática". <http://numat.net/tutor/newton.pdf>



"Universidad Politécnica de Madrid". [http://ocw.upm.es/matematica-aplicada/programacion-y-metodos-numericos/contenidos/TEMA\\_3/Apuntes/InterpolacionOCW.pdf](http://ocw.upm.es/matematica-aplicada/programacion-y-metodos-numericos/contenidos/TEMA_3/Apuntes/InterpolacionOCW.pdf)



"Instituto Tecnológico de Tuxtla Gutiérrez".  
<https://sites.google.com/site/danely7/unidad-5/polinomio-de-interpolacion-de-newton>



"Purificación González". ETSI  
Bilbao" <http://www.ehu.es/pegonzalez/I.Teleco/Apuntes/tema5.pdf>