

---

# Interpolación de Taylor

Mérari Afonso  
Ignacio Fragoso  
Lidia García

*Grupo 1F*

*Técnicas Experimentales. 1<sup>er</sup> curso. 2<sup>do</sup> semestre*

Facultad de Matemáticas

Universidad de La Laguna

---

La Laguna, 17 de mayo de 2013



# Índice general

<b>1. Motivacion y objetivos</b>	<b>5</b>
<b>2. Fundamentos teóricos</b>	<b>6</b>
2.1. Tipos de interpolación polinómica . . . . .	6
2.1.1. Interpolación Lineal . . . . .	7
2.1.2. Interpolación Cuadrática . . . . .	7
2.2. Interpolación por Splines . . . . .	8
2.3. Interpolación de Taylor . . . . .	9
<b>3. Procedimiento experimental</b>	<b>10</b>
<b>4. Conclusiones</b>	<b>13</b>
<b>A. Ápendice</b>	<b>14</b>
A.1. Programa Python . . . . .	14
A.2. Programa python de las gráficas . . . . .	15
<b>Bibliografía</b>	<b>17</b>



# Índice de figuras

2.1. Ejemplo 1. . . . .	7
2.2. Ejemplo 2. . . . .	8
2.3. Ejemplo 3. . . . .	9
3.1. Gráfica 1. . . . .	12
3.2. Gráfica 2. . . . .	12

# Índice de cuadros

3.1. Tabla 1. . . . .	10
3.2. Tabla 2. . . . .	11
3.3. Tabla 3. . . . .	11

# Capítulo 1

## Motivacion y objetivos

El objetivo de este trabajo principalmente ha sido desarrollar un experimento para evaluar la interpolación de Taylor de una función concreta. Para ello hemos desarrollado un programa en Python que calcula el polinomio de Taylor en un punto dado, la función real en el punto y el error que se comete al hallar el polinomio frente al valor real.

## Capítulo 2

# Fundamentos teóricos

Para comprender de forma correcta el tema a exponer es necesario conocer primero la definición de interpolar. Según la Real Academia Española (RAE) disponemos de estas cuatro opciones:

- \* Poner algo entre cosas.
- \* Intercalar palabras o frases en el texto de un manuscrito antiguo, o en obras y escritos ajenos.
- \* Interrumpir o hacer una breve intermisión en la continuación de algo, y volver luego a proseguirlo.
- \* Calcular el valor aproximado de una magnitud en un intervalo cuando se conocen algunos de los valores que toma a uno y otro lado de dicho intervalo.

En nuestro caso, la opción que define mejor el tema a tratar es la última, ya que se trata de la definición matemática de interpolación.

La interpolación surge porque muchas veces en una función sólo conocemos un conjunto de valores, y si queremos calcular el valor de la función para una abscisa diferente de las conocidas, debemos utilizar otra función que la aproxime, por tanto el valor que obtengamos será una aproximación del valor real. Lo que nos lleva a estar cometiendo un error, ya que se trata de una aproximación.

Pese a que existen varias formas de realizar este cálculo, se utiliza la interpolación porque es el método más sencillo.

En este caso se utilizan polinomios como funciones de aproximación. Este tipo de interpolación se denomina interpolación polinómica.

### 2.1. Tipos de interpolación polinómica

La función de interpolación a encontrar dependerá, entre otras cosas, de la cantidad de datos reales de los que partimos, y de cómo estos puntos se distribuyen por el plano



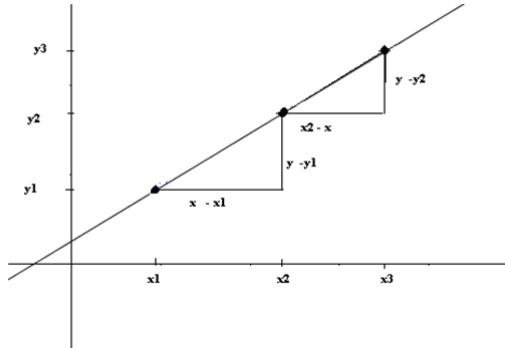


Figura 2.1: Ejemplo 1.

cartesiano. Esto nos dará una idea del tipo de función de interpolación que debemos buscar.

### 2.1.1. Interpolación Lineal

Este tipo de interpolación es empleado cuando suponemos que las variaciones son proporcionales.

Por ejemplo: Dados dos puntos  $(x_1, y_1)$  y  $(x_2, y_2)$ , entonces la interpolación lineal consiste en hallar una estimación del valor  $y$ , para un valor  $x$  tal que  $x_1 < x < x_2$ . Teniendo en cuenta estas suposiciones, podemos determinar:

$$y = y_1 + \frac{(y_2 - y_1)(x - x_1)}{x_2 - x_1}$$

### 2.1.2. Interpolación Cuadrática

Si en vez de utilizar rectas, es decir polinomios de primer grado, utilizamos polinomios de segundo grado para interpolar, estaremos realizando interpolación cuadrática. Para la interpolación lineal utilizábamos dos puntos, ya que dos puntos determinan una recta, ahora necesitaremos tres puntos para determinar la correspondiente parábola.

El forma de resolver el sistema para encontrar los valores que determinan a la función cuadrática  $(a, b, c)$  es:

$$y = a + b(x - x_0) + c(x - x_0)(x - x_1)$$

Lagrange [2] expuso una manera simplificada de calcular los polinomios interpoladores de grado  $n$  para el caso de un polinomio de 2º grado que pasa por los puntos  $(x_0, y_0), (x_1, y_1), (x_2, y_2)$ :

$$y = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

A parte de la interpolación de Lagrange, existen otros métodos muy conocidos como son el método de las diferencias divididas de Newton y la interpolación de Hermite. También existen métodos de Interpolación segmentaria que nos permiten aproximar funciones de un modo eficaz. Entre ellos cabe destacar la interpolación por splines y la interpolación de Taylor. Este último será el que estudiaremos nosotros.

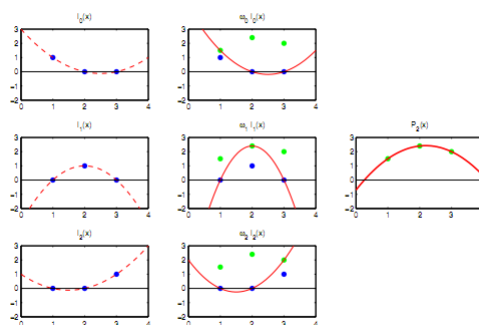


Figura 2.2: Ejemplo 2.

## 2.2. Interpolación por Splines

La Interpolación por Splines es un refinamiento de la interpolación polinómica que usa "pedazos" de varios polinomios en distintos intervalos de la función a interpolar para evitar problemas de oscilación como el llamado Fenómeno de Runge.

La idea es que agrupamos las abscisas  $x_0, x_1, \dots, x_m$  en distintos intervalos según el grado del spline que convenga emplear en cada uno. Así, un spline será un polinomio interpolador de grado  $n$  de  $f$  para cada intervalo. A la postre, los distintos splines quedarán unidos recubriendo todas las abscisas e interpolando a la función.

El principal problema que presenta la interpolación por splines reside en los puntos que son comunes a dos intervalos (extremos). Por esos puntos deben pasar los splines de ambos intervalos, pero para que la interpolación sea ajustada, conviene que el punto de unión entre dos splines sea lo más "suave" posible (ej. evitar puntos angulosos), por lo que se pedirá también que en esos puntos ambos splines tengan derivada común. Esto no será siempre posible y, a menudo, se empleará otro tipo de interpolación, quizás una interpolación no-polinómica.

## 2.3. Interpolación de Taylor

La Interpolación de Taylor [1] usa el Desarrollo de Taylor de una función en un punto para construir un polinomio de grado  $n$  que se aproxima a la función dada. Tiene dos ventajas esenciales sobre otras formas de interpolación:

1. Requiere sólo de un punto  $X_0$  conocido de la función para su cálculo.
2. El cálculo del Polinomio de Taylor es sumamente sencillo comparado con otras formas de interpolación polinómica:

$$P(x) = \sum_{i=1}^n \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i$$

Sin embargo, el error que se comete al resolver este polinomio puede alcanzar cotas demasiado elevadas.

Para hallar el error restamos el valor real de la función menos el valor de la interpolación de Taylor en valor absoluto.

Sea  $p$  el valor real, y  $p^*$  el valor obtenido de realizar la interpolación, tenemos que el error es:  $|p - p^*|$

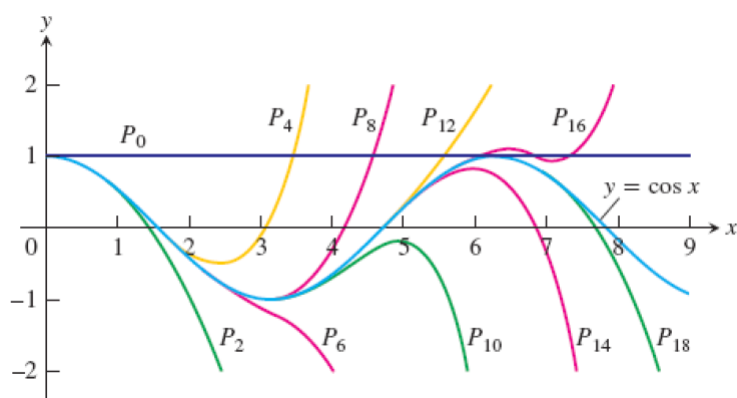


Figura 2.3: Ejemplo 3.

## Capítulo 3

# Procedimiento experimental

Para realizar los pertinentes experimentos hemos creado un programa en lenguaje Python que nos proporciona de forma rápida y eficaz el resultado real, el resultado de realizar la interpolación de Taylor y el error que este método nos proporciona.

Estos resultados varían dependiendo del valor de  $x$ , el valor del centro ( $x_0$  o  $c$ ) y el valor de  $n$  que tomemos. Ya que el error que se comete es lo que nos define si la interpolación es válida y ya que éste ha sido el incentivo para realizar el trabajo, hemos realizado una serie de tablas acompañadas de gráficas que nos permiten ver de una forma clara qué sucede al variar el centro, la  $x$  o la  $n$ .

Fijando el valor de  $x$  y la derivada  $n$ -ésima ( $x=1$ ,  $n=5$ )

Centro	Interpolación Taylor	Valor Real	Error
1.0	-1	-1	0
1.25	-1.00025363221339	-1	0.000253632213394583
1.5	-1.00452485553482	-1	0.00452485553481630
1.75	-0.900045035338955	-1	0.0999549646610447
2	0.123909925872083	-1	1.12390992587208

Cuadro 3.1: Tabla 1.

En este cuadro podemos observar como al ir incrementando el valor del centro (dentro del intervalo, por supuesto) el error incrementa notablemente.

Para verlo gráficamente:

En el siguiente caso hemos fijado el centro en 1.5 y la  $n$ -ésima = 1

x	Int. Taylor	Valor Real	Error
-1.5	-352.175700107991	-1.8369701e-16	352.1757
0.0	-3.40445153931272	1	4.4044515393127
1.0	-1.00025363221339	-1	0.00025363221339458
1.5	-0.000202346334410	-1.8369701e-16	0.00020234633441002
3	-3.03724338394450	-1	2.0372433839445

Cuadro 3.2: Tabla 2.

En esta ocasión podemos decir que no existe una pauta definida, es decir, no podemos decir que al aumentar  $x$  disminuya o aumente el error, o viceversa, pero sí podemos observar algo curioso: tomando valores de  $x$  que coinciden con el intervalo el error cometido es bastante menor que al tomar valores fuera de éste, pese a que estos valores no sean desmesurados, por ejemplo con  $x=-1.5$  el error que se produce es de 352,1757. Para apreciarlo mejor:

Por último, hemos realizado varios experimentos dejando fijo el centro y la  $x$  ( $c=1.5$  y  $x=1$ ):

n-ésima	Interpolación Taylor	Valor Real	Error
3	-0.924832229288648	-1	0.075167770711351
8	-0.99984 3101399498	-1	0.000156898600502386
10	-1.00000354258429	-1	3.54258428503229e-6
17	-1.000000000000004	-1	4.26325641456060e-14
20	-0.999999999999999	-1	1.1102302462516e-15

Cuadro 3.3: Tabla 3.

Esta vez se puede apreciar claramente como al aumentar la  $n$ -ésima el error se reduce hasta llegar a un valor de 1.1102302462516e-15, valor que curiosamente se mantiene constante a partir de  $n=20$ .

Para reflejarlo mejor hemos decidido hacer dos gráficas, la primera nos muestra la variación entre los valores que toma la función real y los valores que toma la interpolación de Taylor para  $n=2$ . En la segunda gráfica hemos aumentado esta  $n$  y podemos observar como prácticamente no existe variación entre las dos funciones

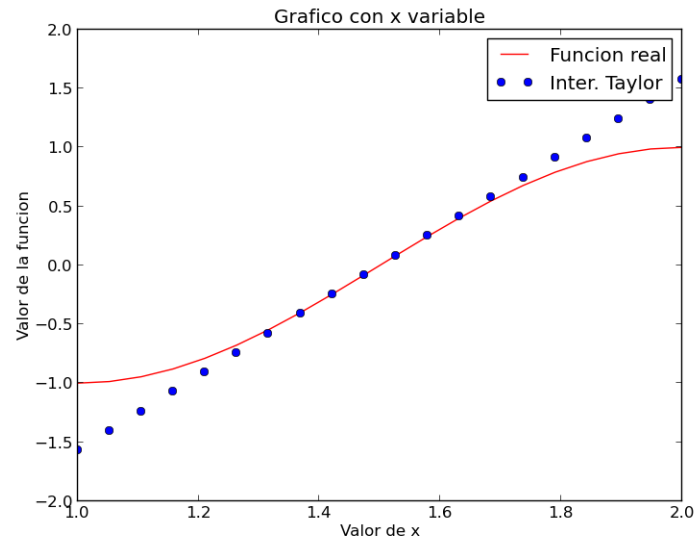


Figura 3.1: Gráfica 1.

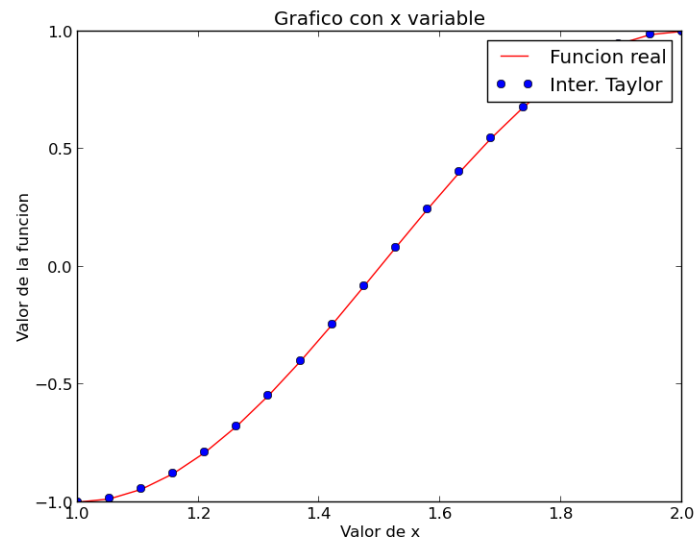


Figura 3.2: Gráfica 2.

## Capítulo 4

# Conclusiones

Tiene dos ventajas esenciales sobre otras formas de interpolación:

Requiere sólo de un punto  $x_0$  conocido de la función para su cálculo, si bien se pide que la función sea suficientemente diferenciable en un entorno de ese punto.

El cálculo del polinomio de Taylor es sumamente sencillo comparado con otras formas de interpolación polinómica.

Sin embargo, en ocasiones no será deseable su uso, dado que el error de interpolación puede alcanzar cotas demasiado elevadas. En nuestro caso particular, el método funciona bien ya que se trata de una función continua y cuyas derivadas son fáciles de calcular.

tomamos un polinomio de orden alto el error que cometemos al realizar la interpolación es prácticamente nulo, mientras que al tomar polinomios con un orden pequeño solo escasos valores tendrán semejanza con el valor real de la función.

# Apéndice A

## Ápendice

### A.1. Programa Python

```
#!/usr/bin/python
import random, sys
from sympy import *

valor_c = Symbol('c')
pi = 3.141592653589793238462643383279502884197169399375105820974944592307816406286
funcion = cos(pi*valor_c)

def fac(n):
    if n == 0:
        return 1
    else:
        return n * fac(n-1)

def Taylor(x,c,n):
    polinomio = 0
    for i in range(n + 1):
        derivada = eval(str(diff(funcion,valor_c,i)))
        polinomio += ((derivada/(fac(i)))*((x - c)**i))
    return polinomio

def real(x):
    resultado = eval(str(cos (pi*x)))
    return resultado

if __name__=='__main__':
    x=float(sys.argv[1])
    c=float(sys.argv[2])
    n=int(sys.argv[3])
    print "El resultado de evaluar la derivada {0}-esima en el punto {1}, cuyo centro es {2} es {3}".format(n,
    print "El valor real de la funcion en el punto {0} es: {1}".format(x, real(x))
    print "El error es ", abs(real(x)-Taylor(x,c,n))
```



## A.2. Programa python de las gráficas

- Gráfica para representar 20 puntos entre [1,2] fijando los valores de la  $n$ -ésima ( $n=2$ ) y el valor del centro ( $c=1.5$ )

```
from matplotlib.pyplot import *
from math import *
from sympy import *

x=linspace(1,2,20)
pi = 3.141592653589793238462643383279502884197169399375105820974944592307816406286
sym_c = Symbol('c')
c = 1.5
n = 2

def f1(x):
    funcion = []
    for i in x:
        funcion.append(cos(pi*i))
    return funcion

def fac(n):
    if n == 0:
        return 1
    else:
        return n * fac(n-1)

def f2(x):
    funcion = cos(pi*sym_c)
    polinomio = []
    for h in x:
        suma = 0
        for i in range(n + 1):
            derivada = eval(str(diff(funcion,sym_c,i)))
            suma += ((derivada/(fac(i)))*((h - c)**i))
        polinomio.append(suma)
    return polinomio

y1 = f1(x)
y2 = f2(x)

p1, = plot(x,y1,'r-')
p2, = plot(x,y2,'bo')
xlabel('Valor de x')
ylabel('Valor de la funcion')
legend([p1, p2], ['Funcion real', 'Inter. Taylor'])
title('Grafico con x variable')
xlim(1.0, 2.0)
show()
```

- Gráfica para representar 20 puntos entre [1,2] fijando los valores de la n-ésima (n=20) y el valor del centro (c=1.5)

```
from matplotlib.pyplot import *
from math import *
from sympy import *
```

```
x=linspace(1,2,20)
pi = 3.141592653589793238462643383279502884197169399375105820974944592307816406286
sym_c = Symbol('c')
c = 1.5
n = 20
```

```
def f1(x):
    funcion = []
    for i in x:
        funcion.append(cos(pi*i))
    return funcion
```

```
def fac(n):
    if n == 0:
        return 1
    else:
        return n * fac(n-1)
```

```
def f2(x):
    funcion = cos(pi*sym_c)
    polinomio = []
    for h in x:
        suma = 0
        for i in range(n + 1):
            derivada = eval(str(diff(funcion,sym_c,i)))
            suma += ((derivada/(fac(i)))*((h - c)**i))
        polinomio.append(suma)
    return polinomio
```

```
y1 = f1(x)
y2 = f2(x)
```

```
p1, = plot(x,y1,'r-')
p2, = plot(x,y2,'bo')
xlabel('Valor de x')
ylabel('Valor de la funcion')
legend([p1, p2], ['Funcion real', 'Inter. Taylor'])
title('Grafico con x variable')
xlim(1.0, 2.0)
show()
```

# Bibliografía

- [1] Brook Taylor, (1685 - 1731), Nacionalidad Inglesa, Campo: Matemáticas, Conocido por: Teorema de Taylor, Serie de Taylor
- [2] Joseph-Louis Lagrange, (1736 - 1813), Nacionalidad Italiano Francés, Campo: Matemáticas, Física, matemática. Instituciones: École polytechnique. Conocido por: Mecánica analítica, Mecánica celeste, Análisis matemático, Teoría de números.

*[http : //www.slideshare.net/taker85/interpolacion – de – taylor – y – hermite](http://www.slideshare.net/taker85/interpolacion-de-taylor-y-hermite)*  
*[http : //www.acm.org/publications/latex\\_style/](http://www.acm.org/publications/latex_style/)*  
*[http : //www.ctan.org/](http://www.ctan.org/)*