INTERPOLACIÓN POLINOMIAL.

con OOoBasic Y OOoCalc

(VERSIÓN 1.0 - AGOSTO 16, 2011.)

Walter Mora F.,
Escuela de Matemática
Instituto Tecnológico de Costa Rica.

Textos Universitarios
Revista digital Matemática, Educación e Internet (www.cidse.itcr.ac.cr/revistamate/)

Contenido

PARTE I

INTERPOLACIÓN POLINOMIAL.

ASPECTOS PRÁCTICOS (con OOoBasic Y OOoCalc)

Introducción

La interpolación polinomial es la base de muchos tipos de integración numérica y tiene otras aplicaciones teóricas. En la práctica a menudo tenemos una tabla de datos $\{(x_i, y_i),\ i = 0,1,2,...,n\}$, obtenida por muestreo o experimentación. Suponemos que los datos corresponden a los valores de una función f desconocida (a veces es conocida, pero queremos cambiarla por una función más sencilla de calcular). El "ajuste de curvas" trata el problema de construir una función que aproxime muy bien estos datos (es decir, a f). Un caso particular de ajuste de curvas es la interpolación polinomial: En este caso se construye un polinomio $P(x)$ que pase por los puntos de la tabla.

La interpolación polinomial consiste en estimar $f(x^*)$ con $P(x^*)$ si x^* no está en la tabla pero se puede ubicar entre estos valores. Una situación típica se muestra en el siguiente ejemplo en el que tenemos datos que relacionan temperatura con el segundo coeficiente virial.[1]

Ejemplo 1

Considere los siguientes datos para el nitrógeno (N_2):

$T(K)$	100	200	300	400	450	500	600
$B(cm^3/mol)$	-160	-35	-4.2	9.0	?	16.9	21.3

donde T es la temperatura y B es el segundo coeficiente virial. ¿Cuál es el segundo coeficiente virial a $450K$?. Para responder la pregunta, usando interpolación polinomial, construimos un polinomio P que pase por los seis puntos de la tabla (ya veremos cómo), tal y como se muestra en la figura (I.1). Luego, el segundo coeficiente virial a $450K$ es aproximadamente $P(450) = 13.5cm^3/mol$.

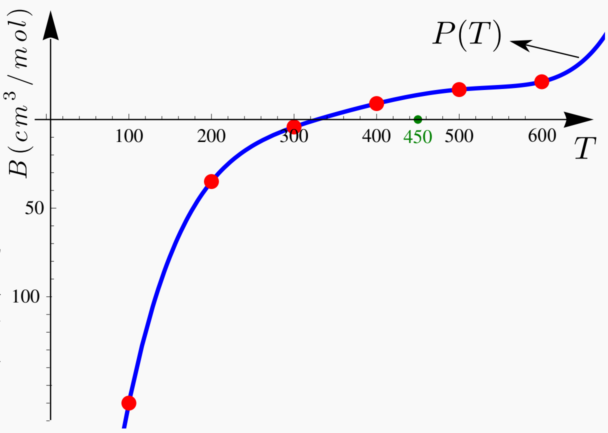


Figura I.1 Polinomio interpolante

1 El comportamiento de gases no ideales se describe a menudo con la *ecuación virial de estado*

$$\frac{PV}{RT} = 1 + \frac{B}{V} + \frac{C}{V^2} + ...,$$

donde P es la presión, V el volumen molar del gas, T es la temperatura Kelvin y R es la constante de gas ideal. Los coeficientes $B = B(T)$, $C = C(T)$,... son el segundo y tercer coeficiente virial, respectivamente. En la práctica se usa la serie truncada

$$\frac{PV}{RT} \approx 1 + \frac{B}{V}$$

Ejemplo 2

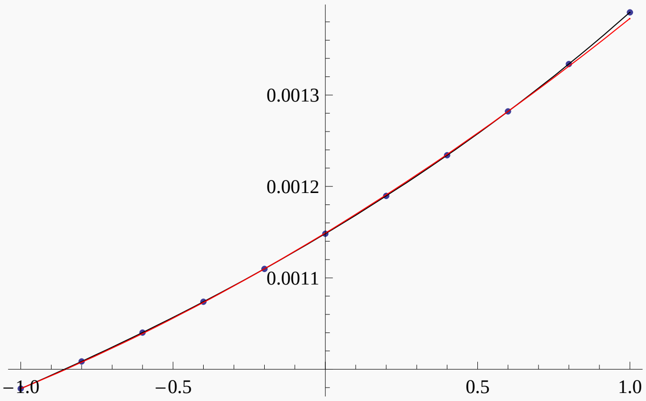
Consideremos la función f definida por

$$f(x) = \int_5^\infty \frac{e^{-t}}{t-x}\,dt, \quad -1 \leq x \leq 1$$

La integral que define a f es una integral no trivial (no se puede expresar en términos de funciones elementales). La tabla de la izquierda nos muestra algunos valores para f.

x	$f(x)$
-1	0.0009788055864607286
-0.6	0.0010401386051341144
-0.2	0.0011097929435687336
0	0.0011482955912753257
0.2	0.0011896108201581322
0.25	?
0.6	0.0012820294923443982
1.	0.0013903460525251596

Tabla I.1



Podemos usar un polinomio interpolante para interpolar $f(0.25)$.

En el mundillo del ajuste de curvas hay varias alternativas,

- Usar un polinomio interpolante. Es el método de propósito general más usado.
- Usar trazadores (splines). Estas son funciones polinomiales a trozos.
- Usar Polinomios trigonométricos en $[0, 2\pi]$. Son la elección natural cuando la función f es periódica de periodo 2π.
- Usar sumas exponenciales. Se usan si conocemos que f presenta decaimiento exponencial conforme $x \longrightarrow \infty$.
- Si los datos son aproximados ("datos experimentales"), lo conveniente sería usar *Mínimos Cuadrados*

Aquí solo vamos a tratar con interpolación polinomial y trazadores cúbicos.

I.1 Interpolación polinomial.

Un problema de interpolación polinomial se especifica como sigue: dados $n+1$ *pares* $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$, siendo todos los x_i's distintos, y $y_i = f(x_i)$ para alguna función f; encontrar un polinomio $P_n(x)$ de grado $\leq n$ tal que

$$P(x_i) = y_i, \quad i = 0, 1, 2, ..., n \tag{I.1}$$

Teorema 3 (Polinomio interpolante).

Dados $n+1$ puntos (x_0,y_0), (x_1,y_1), ... ,(x_n,y_n) con $x_i \neq x_j$ si $i \neq j$; existe un único polinomio $P_n(x)$ de grado $\leq n$ tal que $P(x_i) = y_i \ \forall\, i = 0,1,...,n$

A $P_n(x)$ se le llama *polinomio interpolante*, a cada x_i le decimos *nodo de interpolación* y a cada y_i *valor interpolado.*

- El problema tiene solución única, es decir hay un único polinomio que satisface (I.2).
- No se requiere que los datos estén igualmente espaciados ni en algún orden en particular.
- Si f es un polinomio de grado $k \leq n$, el polinomio interpolante de f en $n+1$ puntos coincide con f.

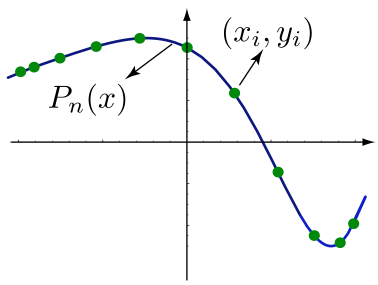
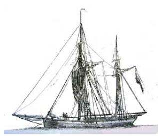


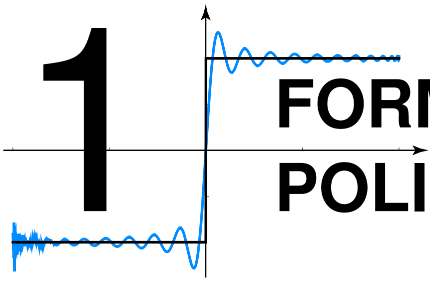
Figura I.2 Polinomio interpolante.

Definición 4

Si de una función f conocemos los puntos (x_0,y_0), (x_1,y_1), ... ,(x_n,y_n), con los x_i's todos distintos, y si $N = \{x_0, x_1, ..., x_n\}$ y $x^* \notin N$ pero $\text{mín}\, N < x^* < \text{máx}\, N$; entonces interpolar f en x^* con un subconjunto de $k+1$ nodos consiste en calcular $P_k(x^*)$ donde P_k es el polinomio interpolante obtenido con un subconjunto de $k+1$ nodos alrededor de x^*.

El polinomio interpolante es único, es decir, solo hay un polinomio que pasa por estos $n+1$ puntos. Aquí vamos a ver cuatro maneras de calcular este polinomio interpolante: La forma de Lagrange del polinomio interpolante, la fórmula baricéntrica de Lagrange, la modificada de Lagrange y la forma de Newton del polinomio interpolante (método de diferencias divididas de de Newton). Los cuatro métodos dan el mismo polinomio (aunque con diferente aspecto), y los cuatro métodos son importantes porque de ellos se hacen otras derivaciones teóricas.





1 FORMA DE LAGRANGE DEL POLINOMIO INTERPOLANTE.

Lagrange[2] calculó el único polinomio interpolante de manera explícita: El polinomio $P_n(x)$ de grado $\leq n$ que pasa por los $n+1$ puntos $(x_0,y_0),(x_1,y_1),...,(x_n,y_n)$ (con $x_i \neq x_j$ para todo i,j) es

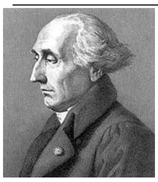
$$P_n(x) = y_0 L_{n,0}(x) + y_1 L_{n,1}(x) + ... + y_n L_{n,n}(x)$$

donde $L_{n,k}(x) = \displaystyle\prod_{\substack{i=0\\ i\neq k}}^{n} \frac{x-x_i}{x_k-x_i} = \frac{(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n).}$

Por ejemplo,

$$\begin{aligned}
L_{n,0}(x) &= \frac{(x-x_1)\cdot(x-x_2)\cdots(x-x_n)}{(x_0-x_1)\cdot(x_0-x_2)\cdots(x_0-x_n).}\\
L_{n,1}(x) &= \frac{(x-x_0)\cdot(x-x_2)\cdots(x-x_n)}{(x_1-x_0)\cdot(x_1-x_2)\cdots(x_1-x_n).}\\
L_{n,3}(x) &= \frac{(x-x_0)\cdot(x-x_2)\cdot(x-x_4)\cdots(x-x_n)}{(x_3-x_0)\cdot(x_3-x_2)\cdot(x_3-x_4)\cdots(x_3-x_n).}\\
\vdots \quad &\vdots \quad \vdots\\
L_{n,n}(x) &= \frac{(x-x_0)\cdot(x-x_1)\cdots(x-x_{n-1})}{(x_n-x_0)\cdot(x_n-x_1)\cdots(x_n-x_{n-1}).}
\end{aligned}$$

2



Joseph Louis Lagrange (1736-1813) fue uno de los más grandes matemáticos de su tiempo. Nació en Italia pero se nacionalizó Francés. Hizo grandes contribuciones en todos los campos de la matemática y también en mecánica. Su obra principal es la "Mécanique analytique"(1788). En esta obra de cuatro volúmenes, se ofrece el tratamiento más completo de la mecánica clásica desde Newton y sirvió de base para el desarrollo de la física matemática en el siglo XIX.

Ejemplo 5

Determine la forma de Lagrange polinomio interpolante, de grado ≤ 2, que pasa por $(0,1),(1,3),$ $(2,0)$.

Solución:

$$\begin{aligned} P_2(x) &= y_0 L_{2,0}(x) + y_1 L_{2,1}(x) + y_2 L_{2,2}(x) \\ &= 1 \cdot L_{2,0}(x) + 3 \cdot L_{2,1}(x) + 0 \cdot L_{2,2}(x) \\ &= 1 \cdot \frac{(x-1)(x-2)}{(0-1)(0-2)} + 3 \cdot \frac{(x-0)(x-2)}{(1-0)(1-2)} \end{aligned}$$

Ejemplo 6

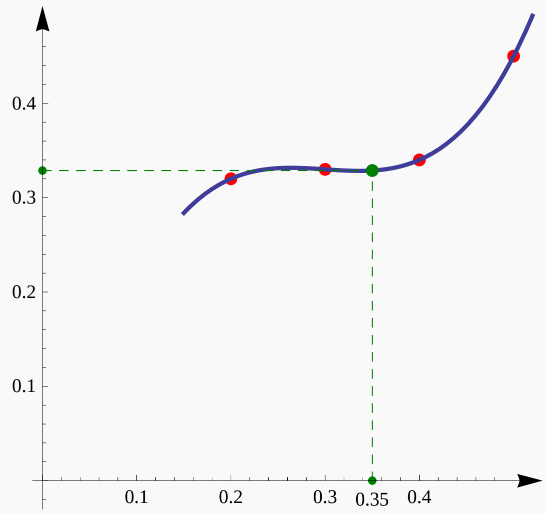
De una función f, conocemos la información de la tabla que sigue. Interpolar $f(0.35)$ usando un polinomio interpolante $P_3(x)$. (Indicar la subtabla de datos que va a usar.)

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$f(x)$	.3	.31	.32	.33	.34	.45	.46	.47

Solución: Como se requiere un polinomio interpolante $P_3(x)$, se necesita una subtabla de *cuatro* datos. Una opción es

x	0.2	0.3	0.4	0.5
$f(x)$	0.32	0.33	0.34	0.45

Si usamos la forma de Lagrange del polinomio interpolante, entonces



$$\begin{aligned} P_3(x) &= 0.32 \cdot \frac{(x-0.3)(x-0.4)(x-0.5)}{(0.2-0.3)(0.2-.4)(0.2-0.5)} \\ &+ 0.33 \cdot \frac{(x-0.2)(x-0.4)(x-0.5)}{(0.3-0.2)(0.3-0.4)(0.3-0.5)} \\ &+ 0.34 \cdot \frac{(x-0.2)(x-0.3)(x-0.5)}{(0.4-0.2)(0.4-0.3)(0.4-0.5)} \\ &+ 0.45 \cdot \frac{(x-0.2)(x-0.3)(x-0.4)}{(0.5-0.2)(0.5-0.3)(0.5-0.4)} \end{aligned}$$

y entonces $f(0.35) \approx P_3(0.35) = 0.32875.$

Ejemplo 7 (Interpolación lineal).

Verifique que el polinomio interpolante de grado ≤ 1 que pasa por $(x_0,y_0),(x_1,y_1)$

$$P_1(x) = \frac{(y_0 - y_1)}{(x_0 - x_1)}(x - x_1) + y_1.$$

Solución: Usando la fórmula de Lagrange,

$$\begin{aligned} P_1(x) &= y_0 L_{n,0}(x) + y_1 L_{n,1}(x) \\ &= y_0 \frac{(x - x_1)}{(x_0 - x_1)} + y_1 \frac{(x - x_0)}{(x_1 - x_0)}. \text{ Simplificando,} \\ &= \frac{(y_0 - y_1)}{(x_0 - x_1)}(x - x_1) + y_1 \end{aligned}$$

Ejemplo 8

En la tabla que sigue aparece las estadísticas de un curso con la cantidad de estudiantes en cada rango de notas.

Rango de Notas	30-40	40-50	50-60	60-70	70-80
N^o Estudiantes	35	48	70	40	22

Estime la cantidad de estudiantes con nota menor o igual a 55.

Solución: Para hacer la estimación necesitamos una tabla con las frecuencias acumuladas,

x$\leq$	40	50	60	70	80
y	35	83	153	193	215

Ahora calculamos el polinomio interpolante,

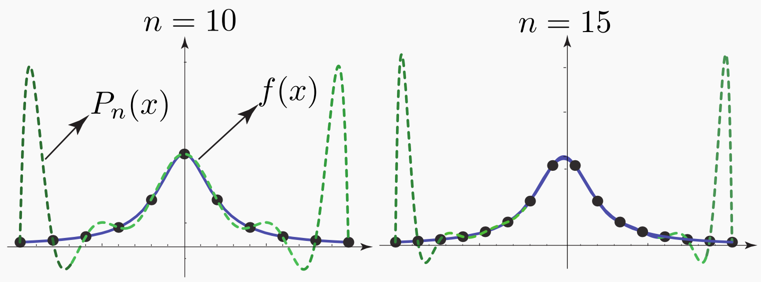
$$\begin{aligned} P_4(x) &= \frac{7(x-80)(x-70)(x-60)(x-50)}{48000} \\ &+ \frac{83(80-x)(x-70)(x-60)(x-40)}{60000} \\ &+ \frac{153(x-80)(x-70)(x-50)(x-40)}{40000} \\ &+ \frac{193(80-x)(x-60)(x-50)(x-40)}{60000} \\ &+ \frac{43(x-70)(x-60)(x-50)(x-40)}{48000} \end{aligned}$$

Así, la cantidad de estudiantes con nota menor o igual a 55 es aproximadamente $P_4(x) = 120$.

Ejemplo 9 (Nodos igualmente espaciados-fenómeno de Runge).

En general, el polinomio interpolante se podría ver afectado por el conjunto $\{x_0,...,x_n\}$ y por la función f.

Este ejemplo es algo extremo y es conocido como 'fenómeno de Runge'; si $f(x) = \dfrac{1}{1+25x^2}$, el polinomio interpolante presenta problemas de convergencia si tomamos los x_i's igualmente espaciados en $[-1,1]$, es decir si $x_i = -1 + i \cdot h$ con $h = 2/n$.



Observe que la interpolación se ve afectado hacia los extremos del intervalo no asi en el centro; esto parece ser una tendencia general.

Si se puede escoger los nodos, una buena opción de ajuste se obtiene con nodos de Tchebychev [3]

Ejemplo 10 (Nodos de Tchebychev).

Si hay posibilidad de escoger los puntos de interpolación, en el intervalo $[-1,1]$, la elección podría ser los nodos

$$x_i = \cos\left(\frac{2i+1}{2n+2}\pi\right),$$

conocidos como nodos de Tchebychev. A diferencia de lo que podría suceder con nodos igualmente espaciados, con estos nodos el polinomio interpolante ajusta bien si $f \in C^1[-1,1]$.

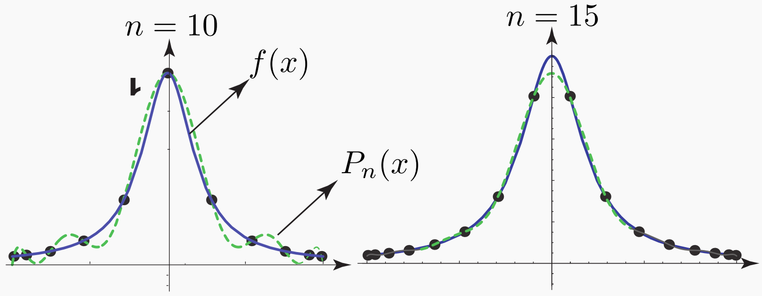
Para un intervalo $[a,b]$ es válido hacer el cambio de variable $u = \dfrac{(b-a)(x-1)}{2} + b$ que mapea el intervalo $[-1,1]$ en el intervalo $[a,b]$. En este caso, los nodos serían

$$u_i = \frac{(b-a)(x_i-1)}{2+b} \quad \text{con} \quad x_i = \cos\left(\frac{2i+1}{2n+2}\pi\right).$$

[3]

Pafnuti Lvóvich Tchebychev (1821 - 1894). El más prominente miembro de la escuela de matemáticas de St. Petersburg. Hizo investigaciones en Mecanismos, Teoría de la Aproximación de Funciones, Teoría de los Números, Teoría de Probabilidades y Teoría de Integración. Sin embargo escribió acerca de muchos otros temas: formas cuadráticas, construcción de mapas, cálculo geométrico de volúmenes, etc.

Continuación..



Como se prueba más adelante, en este caso, si $x^* \in [a,b]$,

$$|f(x^*) - P_n(x^*)| \leq \frac{M}{(n+1)!}\frac{1}{2^n} \quad \text{si} \quad |f^{(n+1)}(x)| \leq M \quad \text{para todo} \quad x \in [a,b].$$

1.1 Forma modificada y forma baricéntrica de Lagrange.

La forma de Lagrange del polinomio interpolante es atractiva para propósitos teóricos. Sin embargo se puede re-escribir en una forma que se vuelva eficiente para el cálculo computacional además de ser numéricamente mucho más estable (ver [2]). La forma modificada y la forma baricéntrica de Lagrange son útiles cuando queremos interpolar una función en todo un intervalo con un con un polinomio interpolante.

Supongamos que tenemos $n+1$ nodos distintos $x_0, x_1, ..., x_n$. Sea $\ell(x) = \prod_{i=0}^{n}(x - x_i)$ y definimos los *pesos baricéntricos* como

$$\omega_k = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{1}{x_k - x_i}, \quad k = 0,1,...,n.$$

Es decir, $\ell(x) = (x - x_0)(x - x_1)\cdots(x - x_n)$ y

$$\omega_k = \frac{1}{x_k - x_0} \cdot \frac{1}{x_k - x_1} \cdots \frac{1}{x_k - x_{k-1}} \cdot \frac{1}{x_k - x_{k+1}} \cdots \frac{1}{x_k - x_n}.$$

Ahora podemos definir la "forma modificada" y "forma baricéntrica" de Lagrange:

Definición 11

La *forma modificada* del polinomio de Lagrange se escribe como

$$P_n(x) = \ell(x) \sum_{j=0}^{n} \frac{\omega_j}{x - x_j} y_j \tag{1.1}$$

Definición 12

La *forma baricéntrica* del polinomio de Lagrange se escribe

$$P_n(x)\begin{cases} = & y_i & \text{si} \quad x = x_i, \\ = & \dfrac{\displaystyle\sum_{k=0}^{n} \frac{\omega_k}{x-x_k} y_k}{\displaystyle\sum_{k=0}^{n} \frac{\omega_k}{x-x_k}}, & \text{si} \quad x \neq x_i \end{cases} \tag{1.2}$$

Ejemplo 13

Consideremos la siguiente tabla de datos,

x	$f(x)$
0.2	3.2
0.3	3.3
0.4	3.4
0.5	4.5

Calcule la forma modificada y la forma baricéntrica de Lagrange e interpole con ambos polinomios, $f(0.35)$.

Solución: Primero calculamos $\ell(x) = (x-0.2)(x-0.3)(x-0.4)(x-0.5)$. Ahora, los pesos baricéntricos,

$$\begin{aligned} \omega_0 &= \frac{1}{0.2-0.3} \cdot \frac{1}{0.2-0.4} \cdot \frac{1}{0.2-0.5} &= -166.667, \\ \omega_1 &= \frac{1}{0.3-0.2} \cdot \frac{1}{0.3-0.4} \cdot \frac{1}{0.3-0.5} &= 500, \\ \omega_2 &= \frac{1}{0.4-0.2} \cdot \frac{1}{0.4-0.3} \cdot \frac{1}{0.4-0.5} &= -500, \\ \omega_3 &= \frac{1}{0.5-0.2} \cdot \frac{1}{0.5-0.3} \cdot \frac{1}{0.5-0.4} &= 166.667 \end{aligned}$$

Entonces, la forma modificada de Lagrange es,

$$P_3(x) = (x-0.2)(x-0.3)(x-0.4)(x-0.5)\left(-\frac{533.333}{x-0.2} + \frac{1650.}{x-0.3} - \frac{1700.}{x-0.4} + \frac{750.}{x-0.5}\right),$$

y la forma baricéntrica es,

$$P_3(x) = \frac{-\dfrac{533.333}{x-0.2} + \dfrac{1650.}{x-0.3} - \dfrac{1700.}{x-0.4} + \dfrac{750.}{x-0.5}}{-\dfrac{166.667}{x-0.2} + \dfrac{500.}{x-0.3} - \dfrac{500.}{x-0.4} + \dfrac{166.667}{x-0.5}}$$

En ambos casos, $f(0.35) \approx P_3(0.35) = 3.2875$.

1.2 Forma baricéntrica con nodos igualmente espaciados.

La forma baricéntrica toma una forma especialmente simple cuando los nodos son igualmente espaciados.

Sea $h > 0$ y $x_k = x_0 + k \cdot h;\ k = 0,1,2...,n,$ entonces

$$\omega_m^{-1} = \prod_{\substack{k=0\\k\neq m}}^{n} (x_m - x_k) = \prod_{\substack{k=0\\k\neq m}}^{n} (x_0 + m \cdot h - x_0 - k \cdot h) = (-1)^{n-m} h^n m!(n-m)!$$

Ahora, como la fórmula (1.2) no cambia si cambiamos ω_m por $\omega_m^* = c\omega_m$ con $c \neq 0$, entonces tomando $c = (-1)^n h^n n!$, los pesos modificados se convierten en coeficientes binomiales con signo alternado,

$$\omega_m^* = (-1)^m \binom{n}{m},\ m = 0,1,...,n$$

Finalmente, la forma baricéntrica para nodos igualmente espaciados no depende del peso h y sus coeficientes son enteros,

$$P_n(x) \begin{cases} = \quad y_i & \text{si} \quad x = x_i, \\ = \quad \dfrac{\sum_{m=0}^{n} (-1)^m \binom{n}{m} \dfrac{y_k}{x - x_m}}{\sum_{m=0}^{n} (-1)^m \binom{n}{m} \dfrac{1}{x - x_m}}, & \text{si} \quad x \neq x_i \end{cases} \tag{1.3}$$

EJERCICIOS

1.1 Considere los cuatro puntos $(0,1),(1,2),(3,0),(4,4)$.

a) Calcule el polinomio interpolante $P_3(x)$, en la forma de Lagrange.

b) Verifique que efectivamente $P_4(x_i) = y_i$, es decir, $P_3(0) = 1, etc.$

c) Interpolar $f(3.5)$.

1.2 Considere los cuatro puntos $(0,1),(1,2),(3,0),(4,4)$. en la forma de modificada y la forma baricéntrica de Lagrange.

a) Calcule el polinomio interpolante $P_3(x)$, en la forma de modificada.

b) Calcule el polinomio interpolante $P_3(x)$, en la forma de Baricéntrica.

c) Verifique que efectivamente $P_3(x_i) = y_i$, es decir, $P(0) = 1, etc.$

d) Interpolar $f(3.5)$.

1.3 Consideremos la siguiente tabla de datos,

x	$f(x)$
0.2	1.2
0.3	5.3
0.4	9.4
0.5	10.5

Calcule la forma modificada y la forma baricéntrica de Lagrange e interpole $f(0.35)$. **Ayuda:** Estas fórmulas permiten reutilizar los cálculos!

1.4 Usando la forma de Lagrange del polinomio interpolante verifique que si $P(x)$ pasa por $(x_0,y_0),(x_1,y_1)$ entonces $P(x) = \dfrac{(y_0 - y_1)}{(x_0 - x_1)}(x - x_1) + y_1$. **Ayuda:** En algún momento de la simplificación debe sumar y restar $y_1 x_1$.

1.5 Considere la función de Bessel $J_0(x) = \dfrac{1}{\pi}\displaystyle\int_0^{\pi} \cos(x\,\text{sen}\,\theta)\,d\theta$. Tenemos la siguiente información,

x	$\pi J_0(x)$
0	3.59
0.2	3.11
0.4	3.08

a) Obtener la forma de Lagrange del polinomio interpolante.
b) Interpolar $J_0(0.25)$

1.6 Considere la siguiente tabla de salarios,

Salarios ($)	0-1000	1000-2000	2000-3000	3000-4000
Frecuencia	9	30	35	42

Estimar la cantidad de personas con salario entre $1000 y $1500.

1.7 Interpolar $\cos(1.75)$ usando la tabla

x_i	$\cos(1+3x_i)$
0	0.540302
1/6	0.070737
1/3	-0.416147

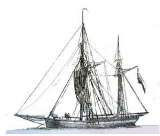
Ayuda: La estimación que se obtiene con el polinomio interpolante es -0.17054.

1.8 Considere la siguiente tabla de vapor para H_2O calentada a $200\,$MPa.

$v\ (m^3/kg)$	0.10377	0.11144	0.1254
$s\ (kJ/Kg \cdot K)$	6.4147	6.5453	6.7664

a) Use interpolación lineal para encontrar la entropía s para un volumen especifico v de $0.108 m^3/kg$.
b) Use interpolación cuadrática para encontrar la entropía s para un volumen especifico v de $0.108 m^3/kg$.

1.9 Usando la tabla del ejemplo (2), interpolar $f(0.25)$.



2 FORMA DE NEWTON PARA EL POLINOMIO INTERPOLANTE.

La representación

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)\cdots(x - x_{n-1}),$$

para el polinomio interpolante que pasa por los $n+1$ puntos $(x_0,y_0),...,(x_n,y_n)$, es conocida como *la representación de Newton* del polinomio interpolante.

2.1 Diferencias Divididas de Newton.

La manera más conocida para calcular la representación de Newton del polinomio interpolante, está basada en el método de *diferencias divididas.* Una gran ventaja sobre la forma clásica del método de Lagrange es que podemos agregar más nodos a la tabla de datos y obtener el polinomio interpolante sin tener que recalcular todo. Comparado con la forma modificada de Lagrange, no hay ganancia y más bien esta última forma es más estable. Aún así, el método de diferencias divididas tiene aplicaciones adicionales en otros contextos.

Podemos calcular los a_i's usando el hecho de que $P(x_i) = y_i$,

$$\begin{cases} P(x_0) = & y_0 & = & a_0 & \Longrightarrow & a_0 = y_0, \\ P(x_1) = & y_1 & = & a_0 + a_1(x_1 - x_0) & \Longrightarrow & a_1 = \dfrac{y_1 - y_0}{x_1 - x_0} \\ P(x_2) = & y_2 & = & a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) & \Longrightarrow & a_2 = \dfrac{y_2 - a_0 - a_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \\ & & & \cdots & & \end{cases}$$

Si $y_k = f(x_k)$, la fórmula anterior nos muestra que cada a_k depende de $x_0,x_1,...,x_k$. Desde muchos años atrás se usa la notación $a_k = f[x_0,x_1,...x_k]$ para significar esta dependencia.

Al símbolo $f[x_0,x_1,...x_n]$ se le llama *diferencia divida* de f. Usando esta nueva notación tendríamos que la forma de Newton del polinomio interpolante es

$$\begin{aligned} P(x) &= f[x_0] + f[x_0,x_1](x - x_0) + f[x_0,x_1,x_2](x - x_0)(x - x_1) \\ &\quad + \cdots + f[x_0,...,x_n](x - x_0)\cdots(x - x_{n-1}), \end{aligned}$$

donde $f[x_0] = y_0$ y $f[x_0,...,x_i]$ es el coeficiente principal de la forma de Newton del polinomio que interpola la función f en los nodos $x_0,x_1,...,x_i$.

Ejemplo 14 (Interpolación lineal).

El polinomio interpolante de grado ≤ 1 que pasa por $(x_0,y_0),(x_1,y_1)$ es

$$P_1(x) = f[x_0] + f[x_0,x_1](x-x_0) \text{ donde } f[x_0,x_1] = \frac{(y_0-y_1)}{(x_0-x_1)} \text{ y } f[x_0] = y_0$$

Si consideramos al coeficiente $f[x_0,x_1,...x_n]$ como una función de $n+1$ variables, entonces esta función es *simétrica*, es decir, permutar las variables de cualquier manera no afecta el valor de la función. Esto es así porque el polinomio que interpola los puntos $\{(x_i,y_i)\}_{i=0,...,n}$ es único, por lo tanto sin importar el orden en que vengan los puntos, el coeficiente principal siempre es $a_n = f[x_0,x_1,...x_n]$.

¿Qué es $f[x_k,x_{k+1},...,x_{k+j}]$?. Es el coeficiente principal de la forma de Newton del polinomio que interpola una función f en los nodos $x_k,x_{k+1},...,x_{k+j}$. Por ejemplo, si tenemos $n+1$ datos $(x_0,y_0),(x_1,y_1),...,(x_n,y_n)$, el polinomio que interpola $(x_3,y_3),(x_4,y_4)$ sería

$$P_1(x) = y_3 + f[x_3,x_4](x-x_3).$$

El nombre "*diferencia divida*" viene del hecho de que cada $f[x_k,x_{k+1},...,x_{k+j}]$ se puede expresar como un cociente de diferencias.

Teorema 15

La diferencia dividida $f[x_k,x_{k+1},...,x_{k+j}]$ satisface la ecuación

$$f[x_k,x_{k+1},...,x_{k+j}] = \frac{f[x_{k+1},x_{k+2},...x_{k+j}] - f[x_k,x_{k+1},...x_{k+j-1}]}{x_{k+j}-x_k} \tag{2.1}$$

Ejemplo 16

El teorema (15) indica que cada diferencia dividida se puede calcular en términos de otras "diferencias" previamente calculadas. Los ejemplos que siguen son casos particulares para mostrar cómo se aplica el teorema.

$$\begin{aligned} f[x_i,x_j] &= \frac{y_i-y_j}{x_i-x_j} \\ f[x_0,x_1,x_2] &= \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0} \\ f[x_1,x_2,x_3] &= \frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1} \end{aligned}$$

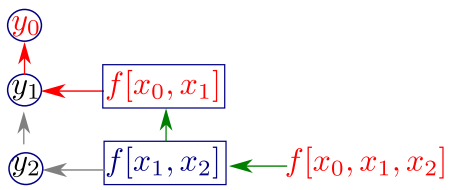
Continuación..

$$
\begin{aligned}
&\vdots\\
f[x_0,x_1,x_2,x_3] &= \frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0}\\
f[x_1,x_2,x_3,x_4] &= \frac{f[x_2,x_3,x_4]-f[x_1,x_2,x_3]}{x_4-x_1}\\
&\vdots\\
f[x_0,x_1,\ldots,x_k] &= \frac{f[x_1,x_2,\ldots,x_k]-f[x_0,x_1,\ldots,x_{k-1}]}{x_k-x_0}
\end{aligned}
$$

Este esquema recursivo se puede arreglar en forma matricial como sigue,

$$
\begin{array}{lllll}
x_0 & y_0 & & & \\
x_1 & y_1 & f[x_0,x_1] & & \\
x_2 & y_2 & f[x_1,x_2] & f[x_0,x_1,x_2] & \\
x_3 & y_3 & f[x_2,x_3] & f[x_1,x_2,x_3] & f[x_0,x_1,x_2,x_3]\\
\vdots & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

En general, para calcular $f[x_0]$, $f[x_0,x_1]$, $f[x_0,x_1,x_2],\cdots,f[x_0,\ldots,x_n]$, debemos calcular una matriz en la que las nuevas columnas se construyen con los datos de la columna anterior.



$$f[x_0,x_1,x_2] = \frac{f[x_0,x_1]-f[x_1,x_2]}{x_2-x_0}$$

La misma matriz se puede usar para calcular la forma de Newton para subconjuntos de datos: En el arreglo que sigue, la diagonal principal (en rojo) corresponde a los coeficientes del polinomio que interpola los datos $(x_0,y_{)},\ldots,(x_n,y_n)$. La diagonal en azul corresponde a los coeficientes del polinomio que interpola los datos $(x_1,y_1),\ldots,(x_n,y_n)$.

$$
\begin{array}{llllll}
y_0 & & & & & \\
y_1 & f[x_0,x_1] & & & & \\
y_2 & f[x_1,x_2] & f[x_0,x_1,x_2] & & & \\
y_3 & f[x_2,x_3] & f[x_1,x_2,x_3] & & & \\
\vdots & \vdots & \vdots & & \ddots & \\
y_n & f[x_{n-1},x_n] & f[x_{n-2},x_{n-1},x_n] & \cdots & f[x_1,\ldots,x_n] & f[x_0,x_1,\ldots,x_n]
\end{array}
$$

Por ejemplo, para calcular el polinomio que interpola los datos $(x_3,y_3),\ldots,(x_6,y_6)$ se usa la (sub)matriz,

$$
\begin{array}{llll}
y_3 & & & \\
y_4 & f[x_3,x_4] & & \\
y_5 & f[x_4,x_5] & f[x_3,x_4,x_5] & \\
y_6 & f[x_5,x_6] & f[x_1,x_2,x_3] & f[x_3,x_4,x_5,x_6]
\end{array}
$$

La diagonal principal (en rojo) corresponde a los coeficientes del polinomio que interpola estos cuatro datos.
(● Clic para ir al programa en Internet: Liga 1 | Liga 2)

Ejemplo 17

Usando diferencias divididas, calcular el polinomio interpolante para los datos $(-1,2),(1,1),(2,2),(3,-2)$ y el polinomio interpolante para los datos $(1,1),(2,2),(3,-2)$.

Solución: Primero construimos la matriz de diferencias divididas usando todos los datos. En rojo están los coeficientes del polinomio que interpola todos los datos y en azul los coeficientes del polinomio que interpola los datos $(1,1),(2,2),(3,-2)$.

$$\begin{array}{llll} x_0 & y_0 & & \\ x_1 & y_1 & f[x_0,x_1] & \\ x_2 & y_2 & f[x_1,x_2] & f[x_0,x_1,x_2] \\ x_3 & y_3 & f[x_2,x_3] & f[x_1,x_2,x_3] & f[x_0,x_1,x_2,x_3] \end{array} \longmapsto \begin{array}{rrrr} 2 & & & \\ 1 & -1/2 & & \\ 2 & 1 & 1/2 & \\ -2 & -4 & -5/2 & -3/4 \end{array}$$

El polinomio interpolante, en la forma de Newton, para todos los datos es

$$P(x) = 2 - 1/2(x+1) + 1/2(x+1)(x-1) - 3/4(x+1)(x-1)(x-2)$$

El polinomio interpolante, en la forma de Newton, para los datos $(1,1),(2,2),(3,-2)$ es

$$P(x) = 1 + 1\cdot(x-1) + -5/2(x-1)(x-2)$$

Ejemplo 18

De una función f, conocemos la información de la tabla (2.1). Interpolar $f(0.35)$ usando un polinomio interpolante $P_3(x)$. Primero que todo, escriba la tabla de datos que va a usar.

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
$f(x)$	3	3.1	3.2	3.3	3.4	4.5	4.6	4.7

Tabla 2.1

Solución: Como se requiere un polinomio interpolante $P_3(x)$, se necesita una tabla de cuatro datos. Una opción es

x	0.2	0.3	0.4	0.5
$f(x)$	3.2	3.3	3.4	4.5

Si usamos la forma de Newton del polinomio interpolante, entonces

3.2			
3.3	1		
3.4	1	0	
4.5	11	50	$166.\overline{6}$

$$\begin{aligned} P_3(x) &= 3.2 + 1\cdot(x-0.2) \\ &+ 0\cdot(x-0.2)(x-0.3) \\ &+ 166.\overline{6}\cdot(x-0.2)(x-0.3)(x-0.4) \end{aligned}$$

Por tanto $f(0.35) \approx P_3(0.35) = 3.2875$

2.2 Forma de Newton en el caso de nodos igualmente espaciados.

Si tenemos nodos igualmente espaciados con $x_k = x_0 + k \cdot h, k = 0,1,,...,n,$ entonces la *diferencia hacia adelante* de orden 1 en y_k es $\Delta^1 y_k = y_{k+1} - y_k$. La diferencia hacia adelante de orden m se define recursivamente como: $\Delta^m y_k = \Delta(\Delta^{m-1} y_k)$. Así,

$$
\begin{array}{lcl}
\Delta^0 y_k & := & y_k, \\
\Delta^1 y_k & = & y_{k+1} - y_k, \\
\Delta^2 y_k & = & \Delta(y_{k+1} - y_k) \; = \; y_{k+2} - y_{k+1} - y_{k+1} + y_k \; = \; y_{k+2} - 2y_{k+1} + y_k, \\
 & \cdots & \\
\Delta^n y_k & = & \displaystyle\sum_{j=0}^{n} (-1)^j \binom{n}{j} y_{k+n-j}
\end{array}
$$

En particular

$$\Delta^n y_0 = \sum_{j=0}^{n} (-1)^j \binom{n}{j} y_{n-j}.$$

Recordemos que si $s \in \mathbb{R}$,

$$
\begin{array}{lcl}
\binom{s}{0} & = & 1, \\
\binom{s}{1} & = & s, \\
\binom{s}{2} & = & \dfrac{s(s-1)}{2}, \\
\binom{s}{3} & = & \dfrac{s(s-1)(s-2)}{6}, \\
\binom{s}{4} & = & \dfrac{s(s-1)(s-2)(s-3)}{24}, \ldots
\end{array}
$$

La relación entre estas diferencias hacia adelante y los coeficientes de la forma de Newton del polinomio interpolante (en el caso de nodos igualmente espaciados) se expresa mediante la fórmula,

$$k!\,h^k f[x_0, x_1, ..., x_k] = \Delta^k y_0.$$

De esta manera, la forma de Newton del polinomio interpolante, para nodos igualmente espaciados, es

$$P_n(x) = y_0 + (x - x_0)\frac{\Delta f(x_0)}{1!h} + (x - x_0)(x - x_1)\frac{\Delta^2 f(x_0)}{2!h^2} + \cdots + (x - x_0)(x - x_1)\cdots(x - x_{n-1})\frac{\Delta^n f(x_0)}{n!h^n}.$$

Se puede hacer una simplificación más; si $x = x_0 + s \cdot h$ entonces

$$
\begin{array}{lcl}
\dfrac{x - x_0}{h} & = & s \\
\dfrac{x - x_i}{h} & = & \dfrac{x - (x_0 + i \cdot h)}{h} = s - i
\end{array}
$$

De este modo,

$$\begin{aligned}
(x-x_0)\frac{\Delta f(x_0)}{1!h} &= \frac{(x-x_0)}{h}\Delta^1 f(x_0) = \binom{s}{1}\Delta^1 f(x_0),\\
(x-x_0)(x-x_1)\frac{\Delta^2 f(x_0)}{2!h^2} &= \frac{(x-x_0)(x-x_1)}{2!h^2}\Delta^2 f(x_0) = \frac{s(s-1)}{2!}\Delta^2 f(x_0) = \binom{s}{2}\Delta^2 f(x_0),\\
&\dots\\
(x-x_0)(x-x_1)\cdots(x-x_{n-1})\frac{\Delta^n f(x_0)}{n!h^n} &= \frac{(x-x_0)(x-x_1)\cdots(x-x_{n-1})}{n!h^n}\Delta^n f(x_0)\\
&= \frac{s(s-1)(s-2)\dots(s-n+1)}{n!}\Delta^n f(x_0)\\
&= \binom{s}{n}\Delta^n f(x_0).
\end{aligned}$$

Es decir, si los nodos son igualmente espaciados (de paso h) y $x = x_0 + s\cdot h$,

$$P_n(x) = P_n(x_0 + s\cdot h) = \sum_{k=0}^{n}\binom{s}{k}\Delta^k y_0 \;=\; \sum_{k=0}^{n}\binom{s}{k} k!\,h^k f[x_0,x_1,...x_k].$$

Ejemplo 19

Usando la tabla de datos, interpolar $f(0.35)$.

x	$f(x)$
0.2	3.2
0.3	3.3
0.4	3.4
0.5	4.5

Solución: Los nodos son igualmente espaciados con $h = 0.1$. La matriz de diferencias divididas es,

$$\begin{array}{llll}
3.2 & & & \\
3.3 & 1 & & \\
3.4 & 1 & 0 & \\
4.5 & 11 & 50 & 166.6\overline{6}
\end{array}$$

Como $0.35 = 0.2 + 1.5\cdot 0.1, \Longrightarrow s = 1.5$,

$$\begin{aligned}
f(0.35)\approx P_3(0.35) &= \binom{1.5}{0}0!\,(0.1)^0\cdot 3.2 + \binom{1.5}{1}1!\,(0.1)^1\cdot 1 + \binom{1.5}{3}3!\,(0.1)^3\cdot 166.6\overline{6}\\
&= 1\cdot 3.2 + 1.5\cdot 0.1\cdot 1 - 0.0625\cdot 6\cdot (0.1)^3\cdot 166.6\overline{6} = 3.2875.
\end{aligned}$$

EJERCICIOS

2.1 Sea $P(x) = a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0$, si se conoce que este polinomio pasa por $(-1,3),(0,0),(1,4),(2,0),(3,1),(4,0)$, determine los coeficientes "a_i" del polinomio.

2.2 Considere los datos $(x_0,1),(x_2,2),(x_3,3),(x_4,4),(x_5,5)$, donde $x_0 = 0.1, x_1 = 0.2, x_3 = 0.3, x_4 = 0.4$ y $x_5 = 0.5$. Calcule $f[x_2,x_3,x_4]$.

2.3 Verifique que $f[x_0,x_1,x_2] = \dfrac{y_0}{(x_0-x_1)(x_0-x_2)} + \dfrac{y_1}{(x_1-x_0)(x_1-x_2)} + \dfrac{y_2}{(x_2-x_0)(x_2-x_1)}$.

2.4 Considere los 4 datos $(0,1),(1,2),(3,0),(4,4)$.

a) Determine la matriz de diferencias divididas y la forma de Newton del polinomio interpolante $P_3(x)$.

b) Verifique que efectivamente $P_3(x_i) = y_i$, es decir, $P(0) = 1, etc.$

c) Interpolar $f(3.5)$.

2.5 Considere la siguiente tabla de datos para el nitrógeno,

$T(K)$	100	200	300	400	500	600
$B(cm^3/mol)$	-160	-35	-4.2	9.0	16.9	21.3

Tabla 2.2 Segundos Coeficientes viriales $B(cm^3/mol)$ para el nitrógeno

donde T es la temperatura y B es el segundo coeficiente virial. Interpolar el segundo coeficiente virial a $450K$.

2.6 Usar la forma de Newton del polinomio interpolante para completar la siguiente tabla de datos para el agua,

$T(C)$	50	60	65	68	75	80
$\rho(kg/m^3)$	988	985.7	980.5	?	974.8	971.6

Tabla 2.3

donde T es temperatura y ρ es la densidad.

2.7 Verifique que $f[x_i,x_j] = f[x_j,x_i]$.

2.8 Usando la forma de Newton del polinomio interpolante, obtenga el polinomio $P_1(x)$ que pasa por $(x_6,y_6),(x_7,y_7)$.

2.9 Considere la función de Bessel $J_0(x) = \dfrac{1}{\pi}\displaystyle\int_0^{\pi} \cos(x \operatorname{sen}\theta)\, d\theta$. Tenemos la siguiente información,

x	$\pi J_0(x)$
0	3.59
0.2	3.11
0.4	3.08

a) Obtener la forma de Newton del polinomio interpolante.

b) Interpolar $J_0(0.25)$

2.10 En la tabla que sigue aparece las estadísticas de un curso con la cantidad de estudiantes en cada rango de notas.

Rango de Notas	30-40	40-50	50-60	60-70	70-80
N^o Estudiantes	35	48	70	40	22

a) Estime la cantidad de estudiantes con nota mayor o igual a 65.

b) Estime la cantidad de estudiantes en el rango $55 - 65$

2.11 La siguiente tabla muestra los pesos normales de bebés durante los primeros 12 meses de vida,

Edad en meses	0	2	5	8	10	12
Peso en libras	7.5	10.25	15	16	18	21

Determine el peso de los bebés entre los 5 y 5.6 meses de vida.

2.12 Interpolar $\cos(1.75)$ usando la tabla

x_i	$\cos(1+3x_i)$
0	0.540302
1/6	0.070737
1/3	-0.416147

Ayuda: la estimación que se obtiene con el polinomio interpolante es -0.17054.

2.13 Considere la siguiente tabla de vapor para H_2O calentada a 200 MPa.

$v\ (m^3/kg)$	0.10377	0.11144	0.1254
$s\ (kJ/Kg \cdot K)$	6.4147	6.5453	6.7664

a) Use interpolación lineal para encontrar la entropía s para un volumen especifico v de $0.108m^3/kg$.
b) Use interpolación cuadrática para encontrar la entropía s para un volumen especifico v de $0.108m^3/kg$.

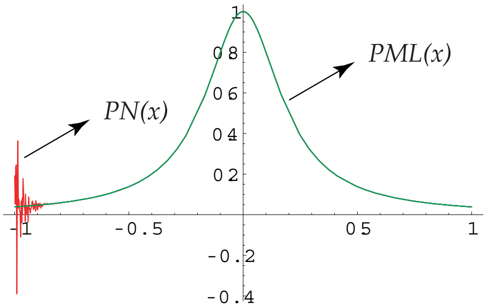
2.14 En la siguiente tabla de diferencias divididas, complete los datos que faltan.

x_i	y_i				
0	2				
1	3	$\square$			
$\square$	2	-1	-1		
3	1	$\square$	0	$\square$	
4	3	2	1.5	0.5	$\square$

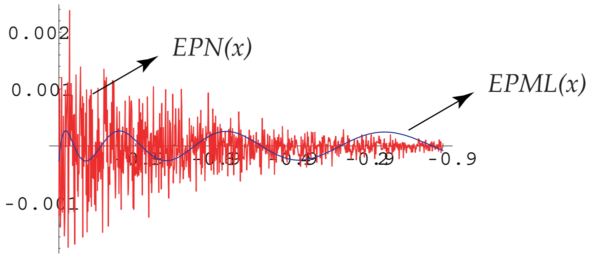
2.3 Forma de Lagrange vs Forma de Newton.

Usualmente se reserva la forma de Lagrange del polinomio interpolante para trabajo teórico y diferencias divididas de Newton para cálculos. La realidad es que la *forma modificada de Lagrange* es tan eficiente como diferencias divididas de Newton en cuanto a costo computacional y además es numéricamente mucho más estable. Hay varias ventajas que hacen de esta forma modificada de Lagrange, el método a escoger cuando de interpolación polinomial se trata ([9], [10]).

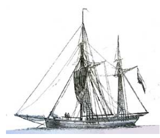
Para mostrar la inestabilidad del polinomio interpolante obtenido con diferencias divididas versus el obtenido con la forma modificada de Lagrange, consideramos la función de Runge $f(x) = 1/(1+25x^2)$ en $[-1,1]$. Para un buen ajuste, usamos 52 nodos de TChebyshev. En la figura (2.3,(a)) se muestra la gráfica de f junto con la gráfica del polinomio interpolante obtenido con diferencias divididas ($PN(x)$) y del polinomio interpolantes obtenido con la forma modificada de Lagrange ($PML(x)$). Usando la aritmética usual de la máquina, se nota inestabilidad de $PN(x)$ en las cercanías de $x=-1$. En la figura (2.3,(b)) se muestra el error relativo de la aproximación a f con cada polinomio en $[-1,-0.9]$. $EPN(x)$ corresponde al error relativo entre f y la forma de Newton del polinomio interpolante y $EPML(x)$ corresponde al error relativo entre f y la forma modificada de Lagrange.

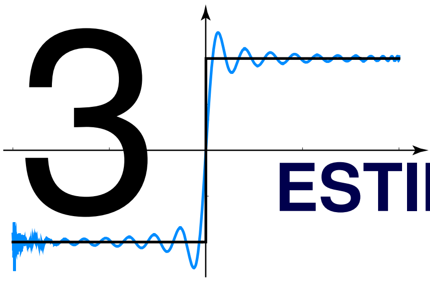


(a) Interpolación. Diferencias divididas vs forma modificada de Lagrange



(b) Error relativo.





3 ESTIMACIÓN DEL ERROR.

3.1 Introducción

La estimación del error, cuando interpolamos con un polinomio interpolante, es de interés práctico en varias áreas, por ejemplo en el desarrollo de métodos de aproximación en ecuaciones diferenciales ordinarias y en ecuaciones diferenciales en derivadas parciales.

Una estimación del error se puede obtener si conocemos alguna información acerca de la función f y sus derivadas. Sea $f \in C^{n+1}[a,b]$ y $P_n(x)$ el polinomio de interpolación de f en (x_0,y_0), (x_1,y_1), ..., (x_n,y_n), con $x_i \in [a,b]$. Entonces, usando polinomios de Taylor podemos establecer la siguiente fórmula para el error

$$f(x) - P_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n)$$

donde $a < \xi(x) < b$ y $x \in [a,b]$. Aquí, la expresión "$\xi(x)$" significa que ξ no es una constante fija, sino que varía según el valor que tome x.

Para efectos prácticos, a y b son el mínimo y el máximo del conjunto $\{x_0, x_1, ..., x_n\}$. Si M_n es el el máximo absoluto de la función $|f^{(n+1)}|$ en $[a,b]$, es decir, $|f^{(n+1)}(x)| \leq M_n$ para todo $x \in [a,b]$, entonces podemos obtener una estimación del error $f(x) - P_n(x)$ con la desigualdad,

$$|f(x) - P_n(x)| \leq \frac{M_n}{(n+1)!}|(x-x_0)(x-x_1)\cdots(x-x_n)|;\; x \in [a,b]. \tag{3.1}$$

Observe que un polinomio interpolante de grado alto no garantiza una mejora en el error: Si usamos más puntos (posiblemente más cercanos entre ellos) se puede esperar que el producto $\prod_i (x-x_i)$ se haga más pequeño con n, pero todavía debería pasar que la derivada de orden $n+1$ no crezca más rápido que $(n+1)!$ y esto parece no ser la regla[4].

Si los nodos son igualmente espaciados, y suponiendo que tenemos n y M_n fijos, la estimación del error depende de la función $\ell(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$. La forma general de esta función se muestra en la figura que sigue,

[4] Georg Faber (1912) demostró que para cada juego de nodos, existe un función continua para la cual los polinomios interpolantes no convergen uniformemente a f y también, para cada función continua existe un juego de nodos donde los polinomios interpolantes si convergen de manera uniforme. Aún en este último caso, los nodos no siempre fáciles de obtener.

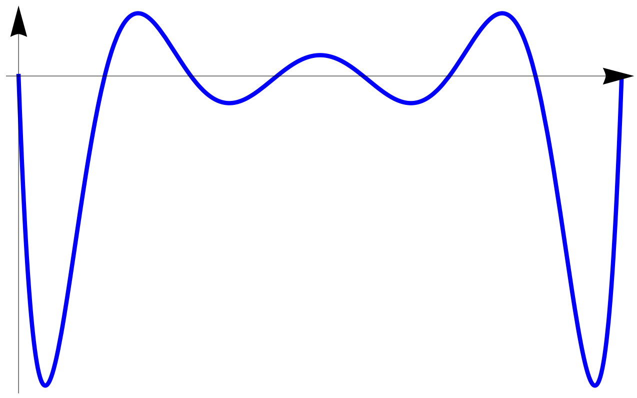


Figura 3.1 $\ell(x) = (x - x_0) \cdots (x - x_7)$ con 7 nodos igualmente espaciados.

Esto sugiere que en el caso de nodos igualmente espaciados (excepto $n = 1$), el error es más pequeño si x está hacia el centro y empeora en los extremos.

La desigualdad (3.1) sería suficiente para estimar el error al interpolar en un valor x, pero nos interesa también una estimación que nos sirva para todo $x \in [x_0, x_n]$.

3.2 Error en interpolación lineal.

Si tenemos dos puntos $(x_0, y_0), (x_1, y_1)$ con $x_0 < x_1$, el error es $f(x) - P_1(x) = \dfrac{(x - x_0)(x - x_1)}{2} f''(\xi(x))$. ¿Cuál es el error máximo si x está entre x_0 y x_1 y si f'' permanece acotada en $[x_0, x_1]$?.

Si $|f''(x)| \leq M_2$ en $[x_0, x_1]$, entonces

$$|f(x) - P_1(x)| \leq \frac{M_2}{2!} \left|(x - x_0)(x - x_1)\right|.$$

El error máximo depende del máximo valor de la función $\left|\dfrac{(x - x_0)(x - x_1)}{2}\right|$ en el intervalo $[x_0, x_1]$.
Como $\ell(x) = \dfrac{(x - x_0)(x - x_1)}{2}$ es una parábola cóncava hacia arriba (figura 3.2), es negativa si $x \in [x_0, x_1]$, por lo tanto el máximo en valor absoluto lo alcanza en $x = \dfrac{x_0 + x_1}{2}$, y es

$$\frac{(x_1 - x_0)^2}{8}.$$

$\therefore$ Si se usa interpolación lineal, el error general esta acotado por

$$|f(x) - P_1(x)| \leq M_2 \frac{(x_1 - x_0)^2}{8}.$$

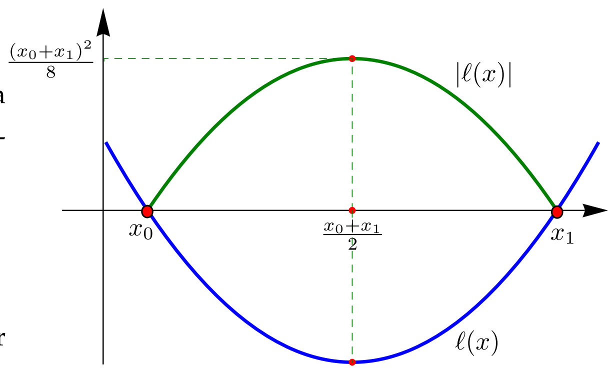


Figura 3.2 $\ell(x) = (x - x_0)(x - x_1)$ y $|\ell(x)|$

Ejemplo 20

Si tabulamos la función $\operatorname{sen} x$ para $x_0 = 0$, $x_1 = 0.002$, $x_2 = 0.004, \ldots$ entonces el error general al *interpolar linealmente* es

$$|\operatorname{sen} x - P_1(x)| \leq 1 \cdot \left| \frac{(0.002)^2}{8} \right| = 0.5 \times 10^{-6},$$

pues $|\operatorname{sen} x| \leq 1\ \forall\, x$ (Aquí suponemos que el polinomio se evalúa de manera exacta). Esto nos dice que la función $\operatorname{sen} x$ es apropiada para interpolación. lineal.

Si deseamos más precisión en un caso particular, podemos usar la fórmula (3.1). Si por ejemplo $x = 0.003$, entonces $|\operatorname{sen}(0.003) - P_1(0.003)| \leq \frac{\operatorname{sen}(0.004)}{2} |(0.003 - 0.002)(0.003 - 0.004)| \approx -1.99 \times 10^{-9}$ pues el máximo absoluto de $|\operatorname{sen} x|$ en el intervalo $[0.002,\, 0.004]$ es $\operatorname{sen}(0.004)$.

3.3 Error en interpolación cuadrática

Si interpolamos con tres puntos ($n = 2$) igualmente espaciados x_0, $x_1 = x + h$ y $x_2 = x_0 + 2h$; entonces si $x \in [x_0, x_2]$ y si $|f'''(x)| \leq M_3$ en $[a,b]$, la estimación general del error es,

$$\begin{aligned} |f(x) - P_2(x)| &\leq \frac{M_3}{3!} |(x - x_0)(x - x_1)(x - x_2)| \\ &\leq \frac{M_3}{6} |(x - x_0)(x - x_0 - h)(x - x_0 - 2h)| \end{aligned}$$

Para obtener el máximo absoluto de la función $\ell(x) = (x - x_0)(x - x_1)(x - x_2)$ calculamos sus puntos críticos: $\ell'(x) = 3x^2 \;+\; x(-6h - 6x_0) \;+ 6hx_0 + 3x_0^2 + 2h^2$, los ceros de esta cuadrática son $r_1 = \frac{1}{3}(3h + \sqrt{3}h + 3x_0)$ y $r_2 = \frac{1}{3}(3h - \sqrt{3}h + 3x_0)$.

Como $\ell(x)$ se anula en x_0 y x_2, el máximo absoluto de $|\ell(x)|$ es $\max\{|\ell(r_1)|, |\ell(r_2)|\} = \dfrac{2h^3}{3\sqrt{3}}$.

$\therefore$ El error general al interpolar con tres puntos igualmente espaciados es $|f(x) - P_2(x)| \leq \dfrac{M_3 h^3}{9\sqrt{3}}$, $x \in [x_0, x_2]$.

Ejemplo 21

Si tabulamos la función $\operatorname{sen} x$ para $x_0 = 0$, $x_1 = 0.01$, $x_2 = 0.02, \ldots$ entonces el error general al *interpolar con un polinomio de grado dos* es

$$|\operatorname{sen} x - P_2(x)| \leq \frac{1 \cdot (0.01)^3}{9\sqrt{3}} \approx 6.415 \times 10^{-8},$$

pues $|\operatorname{sen} x| \leq 1\ \forall\, x$.

3.4 Error en interpolación cúbica

Si tenemos cuatro puntos *igualmente espaciados* $(x_0,y_0),(x_1,y_1),(x_2,y_2),(x_3,y_3)$ con $x_0 < x_1 < x_2 < x_3$, una estimación del error es

$$|f(x) - P_3(x)| \leq \frac{M_4}{4!}|(x-x_0)(x-x_1)(x-x_2)(x-x_3)|, \quad \text{con} \quad |f^{(4)}(x)| \leq M_4 \text{ en } [x_0, x_3].$$

De nuevo, dados n y M_4 fijos, la estimación del error general depende del máximo absoluto del polinomio $|\ell(x)| = |(x-x_0)(x-x_1)(x-x_2)(x-x_3)|$. Como $x_i = x_0 + i \cdot h, i = 1,2,3;$

$$\begin{aligned} \ell(x) &= (x-x_0)(x-x_1)(x-x_2)(x-x_3) \\ &= (x-x_0)(x-x_0-h)(x-x_0-2h)(x-x_0-3h) \end{aligned}$$

$$\ell'(x) = 2(2x-3h-2x_0)(x^2+x(-3h-2x_0)+h^2+3hx_0+x_0^2)$$

Los puntos críticos son $r_1 = 0.5(3h+2x_0)$, $r_2 = 0.5(3h-\sqrt{5}h+2x_0)$ y $r_3 = 0.5(3h+\sqrt{5}h+2x_0)$. Como $\ell(x)$ se anula en x_0 y x_3, entonces el máximo absoluto de $|\ell(x)|$ es $\text{máx}\{|\ell(r_1)|,|\ell(r_2)|,|\ell(r_3)|\} = \left\{\frac{9h^4}{16}, h^4\right\} = h^4$. Finalmente,

$\therefore$ el error general al interpolar con cuatro puntos igualmente espaciados es $|f(x) - P_3(x)| \leq \dfrac{M_4h^4}{24}, \; x \in [x_0,x_3].$

$\therefore$ Si solo interpolamos valores $x \in [x_1,x_2]$, el máximo absoluto de $|\ell(x)|$ en este intervalo se alcanza en el punto medio $x = (x_1+x_3)/2 = 0.5(3h+2x_0)$ y es $\dfrac{9h^4}{16}$. En este caso la estimación del error general es

$$|f(x) - P_3(x)| \leq \frac{3M_4h^4}{128}, \; x \in [x_1,x_2].$$

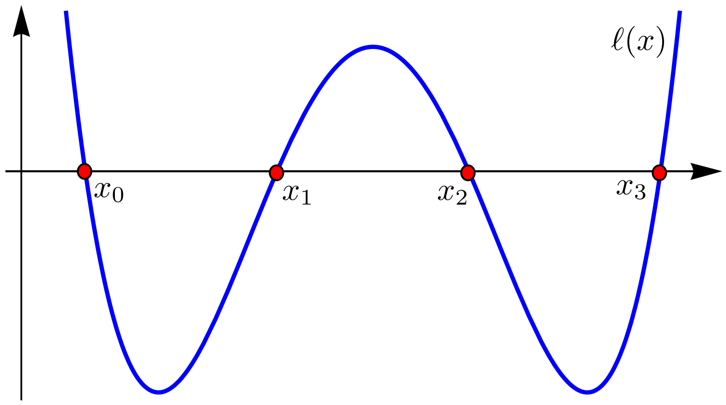


Figura 3.3 $\ell(x) = (x-x_0)(x-x_1)(x-x_2)(x-x_3)$

Ejemplo 22

Si tabulamos la función $\operatorname{sen} x$ para $x_0 = 0$, $x_1 = 0.05$, $x_2 = 0.10, x_3 = 0.15, \ldots$ entonces el error al interpolar con P_3 entre x_1 y x_2 es

$$|\operatorname{sen} x - P_3(x)| \leq 1 \cdot \frac{3}{128}(0.05)^4 \approx 1.46 \times 10^{-7},$$

pues $|\operatorname{sen} x| \leq 1 \ \forall\, x$ (Aquí suponemos que el polinomio se evalúa de manera exacta).

3.5 Error con interpolación con polinomios de grado n.

Si interpolamos sobre puntos igualmente espaciados $x_i = x_0 + i \cdot h,\ \ i = 0,1,\ldots,n;$ y si h es pequeño entonces $f^{(n+1)}(\xi(x))$ en general no se espera que varíe gran cosa. El comportamiento del error es entonces principalmente determinado por $\ell(x) = (x - x_0)(x - x_1)\cdots(x - x_n)$.

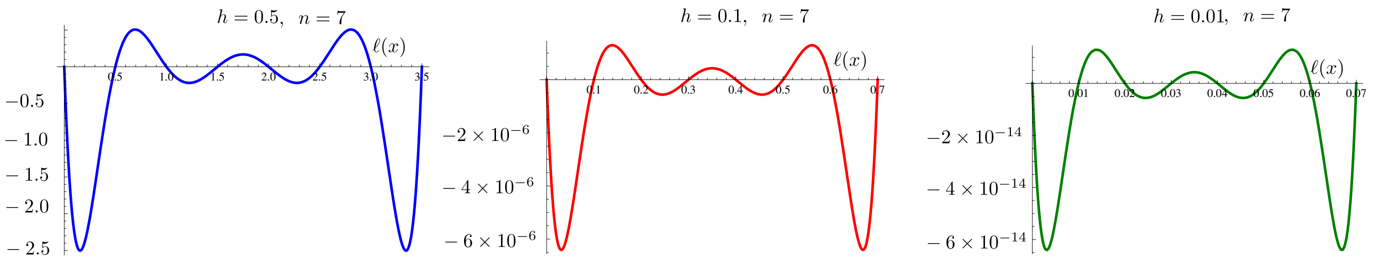


Figura 3.4 $\ell(x) = (x - x_0)(x - x_1)\cdots(x - x_n)$ con $n = 7$.

Pero las oscilaciones de $\ell(x)$ se hacen más violentas si n crece,

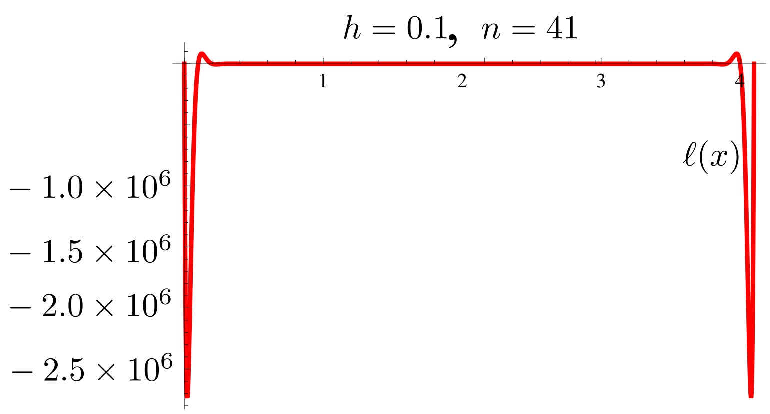


Figura 3.5 $\ell(x) = (x - x_0)(x - x_1)\cdots(x - x_n)$ con $n = 41$.

Sin embargo, la sucesión de polinomios interpolantes $\{P_n(x)\}$ podría converger a f (sin importar si los nodos son o no igualmente espaciados); esto depende del comportamiento de la derivada $k-$ésima de f: La sucesión $\{P_n(x)\}$

converge a f uniformente en $[a,b]$ (que contiene a los nodos) si

$$\lim_{k\to\infty} \frac{(b-a)^k}{k!} M_k = 0$$

y esto sucede si f es *analítica* en una región suficientemente grande, en el plano complejo, que contenga a $[a,b]$ ([1, pág 84]).

3.6 Interpolación Iterada de Neville

Si no tenemos información acerca de las derivadas de una función no podemos usar la fórmula para el cálculo del error. Entonces, ¿cuál es el grado del polinomio de interpolación más adecuado para interpolar un valor?. Para responder esta pregunta podemos usar el *algoritmo de Neville,* este método interpola un valor particular con polinomios de grado cada vez más alto (iniciando en grado cero) hasta que los valores sucesivos están suficientemente cercanos. Luego por inspección podemos decidirnos por un valor en particular.

Usemos la siguiente notación: $P_{0,1}$ es el polinomio interpolante que pasa por $(x_0,y_0),(x_1,y_1)$; $P_{0,1,2}$ es el polinomio interpolante que pasa por $(x_0,y_0),(x_1,y_1),(x_2,y_2)$; $P_{1,2,3,4}$ es el polinomio interpolante que pasa por $(x_1,y_1),(x_2,y_2)$, $(x_3,y_3),(x_4,y_4)$; etc. Como no tenemos información acerca de las derivadas de f, el criterio para estimar el error es empírica e implícita: Nos quedamos con la estimación que presente 'menos variación'.

Consideremos la siguiente tabla de datos,

x	1.2	1.3	1.4	1.5	1.6
$f(x)$	0.7651977	0.6200860	0.4554022	0.2818186	0.1103623

Para interpolar en $x=1.35$ tenemos varias opciones y combinaciones, con tres nodos, con cuatro nodos, etc. Usando nuestra notación, algunos resultados son $P_{0,1,2}(1.35)=0.5401905$; $P_{123}(1.35)=0.5388565$; $P_{0123}(1.35)=0.5395235$; $P_{1234}(1.35)=0.5395457$; $P_{01234}(1.35)=0.5395318$. La menor variación la encontramos con $P_{0123}(1.35)=0.5395235$; $P_{1234}(1.35)=0.5395457$ y $P_{01234}(1.35)=0.5395318$ y de estos tres, los más cercanos son $P_{0123}(1.35)=0.5395235$ y $P_{01234}(1.35)=0.5395318$. En este caso parece lo mejor quedarnos con la aproximación $P_{01234}(1.35)=0.5395318$ ya que toma en cuenta toda la tabla.

El problema en el anális anterior es la gran cantidad de polinomios que se deben evaluar, el algoritmo de Neville precisamente automatiza esta tarea usando cálculos anteriores para obtener el nuevo cálculo. El algoritmo de Neville no calcula $P(x)$ sino que *evalúa* varios polinomios interpolantes de Lagrange en un valor dado.

Sea $Q_{i,j}$ el polinomio interpolante que pasa por $(x_{i-j},y_{i-j})\ldots\ (x_i,y_i)$, es decir, $Q_{i,j}=P_{i-j,i-j+1,i-j+2,...,i-1,i}$ es el polinomio interpolante (en la forma de Lagrange) que pasa por los nodos $(x_{i-j},y_{i-j}),(x_{i-j+1},y_{i-j+1}),...,(x_i,y_i)$, $0\le j\le i$. Por ejemplo,

$Q_{0,0}=P_0$ pasa por (x_0,y_0), es decir, $P_0(x_0)=y_0$.

$Q_{4,0}=P_4$ pasa por (x_4,y_4), es decir, $P_4(x_4)=y_4$.

$Q_{5,2}=P_{3,2,1}$ pasa por $(x_3,y_3),(x_4,y_4),(x_5,y_5)$

$Q_{4,4}=P_{0,1,2,3,4}$ pasa por $(x_0,y_0),(x_1,y_1),...,(x_4,y_4)$

Con esta definición de $Q_{i,j}$ se tiene la siguiente relación recursiva,

$$Q_{i,j}(x) = \frac{(x - x_{i-j})Q_{i,j-1}(x) - (x - x_i)Q_{i-1,j-1}(x)}{x_i - x_{j-1}} \tag{3.2}$$

Aplicando esta relación para $i = 1,2,..,n;\ j = 1,2,...,i$ se logra calcular varios polinomios interpolantes de Lagrange en un valor x, como se muestra en la siguiente tabla (para el caso de 5 nodos)

x_0	$Q_{0,0} = y_0$				
x_1	$Q_{1,0} = y_1$	$Q_{1,1} = P_{0,1}$			
x_2	$Q_{2,0} = y_2$	$Q_{2,1} = P_{1,2}$	$Q_{2,2} = P_{0,1,2}$		
x_3	$Q_{3,0} = y_3$	$Q_{3,1} = P_{2,3}$	$Q_{3,2} = P_{1,2,3}$	$Q_{3,3} = P_{0,1,2,3}$	
x_4	$Q_{4,0} = y_4$	$Q_{4,1} = P_{3,4}$	$Q_{4,2} = P_{2,3,4}$	$Q_{4,3} = P_{1,2,3,4}$	$Q_{4,4} = P_{0,1,2,3,4}$

(Clic para ir al programa en Internet: Liga 1 | Liga 2)

Ejemplo 23

La distribución gamma se define como

$$F(x;\beta,\alpha) = \int_0^{x/\beta} \frac{u^{\alpha-1}e^{-u}}{\Gamma(\alpha)} \mathrm{du}$$

Supogamos que tenemos la siguiente tabla de datos, obtenida con $\beta = 1$ y $\alpha = 2$.

x	$F(x;1,2)$
$x_0 = 0$	0.0
$x_1 = 0.1$	0.00467884
$x_2 = 0.2$	0.01752309
$x_3 = 0.3$	0.03693631
$x_4 = 0.4$	0.06155193

Si queremos estimar F en 0.25 debemos usar polinomios que al menos pasen por x_2 y x_3. Por ejemplo $P_{0,1,2,3}$, $P_{1,2,3,4}$, etc.

Aplicando el algoritmo de Neville en $x = 0.25$, obtenemos la tabla (redondenado a 7 cifras decimales),

x_0	P_0				
0	0.0				
x_1	P_1	$P_{0,1}$			
0.1	0.0046679	0.0116697			
x_2	P_2	$P_{1,2}$	$P_{0,1,2}$		
0.2	0.0175231	0.0239507	0.0270209		
x_3	P_3	$P_{2,3}$	$P_{1,2,3}$	$P_{0,1,2,3}$	
0.3	0.0369363	0.0272297	0.0264099	0.0265118	
x_4	P_4	$P_{3,4}$	$P_{2,3,4}$	$P_{1,2,3,4}$	$P_{0,1,2,3,4}$
0.4	0.0615519	0.0246285	0.0265794	0.0264947	0.0265011

La menor variación la tenemos entre $P_{1,2,3,4}$ y $P_{0,1,2,3,4}(0.25)$. Como $F(0.25;1,2) = 0.026499021...$, la mejor aproximación en realidad es $P_{1,2,3,4}$, pero en la práctica, por suspuesto, tomamos decisiones sin esta información.

3.6.1 Algoritmo

El algoritmo es muy parecido al método de diferencias divididas de Newton, escribimos la primera columna de la matriz Q (las y_i's) y luego completamos la matriz con la relación recursiva (3.2).

Algoritmo 3.1: Algoritmo de Neville

Datos: Valor a interpolar x y los nodos $(x_0,y_0),(x_1,y_1),(x_2,y_2),...,(x_n,y_n)$
Salida: Matriz Q

for $i=0,...,n$ **do**
    $Q_{i,0}=y_i$
for $i=1,...,n$ **do**
    for $j=1,...,i$ **do**
        $Q_{i,j}(x)=\dfrac{(x-x_{i-j})Q_{i,j-1}(x)-(x-x_i)Q_{i-1,j-1}(x)}{x_i-x_{j-1}}$
return Matriz Q

3.7 Otros casos.

Si la función f y sus derivadas son conocidas, se puede hacer una estimación del error con el máximo absoluto.

Ejemplo 24

Sea $f(x)=\dfrac{1}{2}\,e^{(x-1)/2}$. Usando la fórmula de error, estime el error que se cometería al interpolar $f(1)$ con el polinomio interpolante obtenido de la tabla

x	$f(x)$
0.7	0.43
0.8	0.45
1.1	0.53
1.2	0.55

Solución: Son cuatro datos (no igualmente espaciados), $n+1=4$. Luego, la fórmula para estimar el error es

$$|f(1)-P_3(1)|=\left|\frac{f^{(4)}(\xi)}{4!}\,(1-0.7)(1-0.8)(1-1.1)(1-1.2)\right|\leq\left|\frac{M}{4!}\,(1-0.7)(1-0.8)(1-1.1)(1-1.2)\right|$$

donde M es el máximo absoluto de $|f^{(4)}(x)|=|\frac{1}{32}\,e^{\frac{x-1}{2}}|$, en $[0.7,\,1.2]$

Cálculo de M

Puntos críticos: La ecuación $f^{(5)}(x)=\frac{1}{64}\,e^{\frac{x-1}{2}}=0$ no tiene solución, así que no hay puntos críticos.

Comparación: $M=\text{máx}\{|f^{(4)}(0.7)|,\,|f^{(4)}(1.2)|\}=0.0345366...$

Finalmente, la estimación del error es $|f(1)-P_3(1)\leq\left|\dfrac{M}{4!}\,(1-0.7)(1-0.8)(1-1.1)(1-1.2)\right|=1.72683\times10^{-6}$

EJERCICIOS

3.1 Sea $f(x) = x^2 \ln x - x^2$. Supongamos que $P(x)$ es el polinomio interpolante de f obtenido con los datos $(1,-1),(2,-1.2),(3,0.88)$. Estime el error cometido al aproximar $f(2.71)$ con $P(2.71)$.

3.2 Sea $f(x) = \ln(4x^2+2)$ Usando la fórmula de error, estime el error que se cometería al interpolar $f(1.22)$ con el polinomio interpolante obtenido de la tabla

x	$f(x)$
0.5	1.09861
1.1	1.92279
1.2	2.04898
1.3	2.1702

3.3 Considere la tabla de datos

x	e^x
0	1
0.5	$e^{0.5}$
1	e

Estime el error cometido al aproximar $e^{0.6}$ con el polinomio de interpolación correspondiente, en el intervalo $[0,1]$.

3.4 Sea $f(x) = \cos(3x+1)$. Supongamos que $P(x)$ es el polinomio interpolante de f obtenido con los datos $(0.,0.54)$, $(0.5,-0.80)$, $(1.,-0.65)$. Estime el error cometido al estimar $f(0.71)$ con $P(0.71)$.

3.5 Considere la tabla de datos

x_i	$\cos(1+3x_i)$
0	0.540302
1/6	0.070737
1/3	-0.416147

Estime el error cometido al interpolar $\cos(1.75)$ con el polinomio de interpolación obtenido con la tabla anterior, en el intervalo $[0,1/3]$. **Ayuda:** Observe que en este caso, $x \neq 1.75$!

3.6 Sea $f(x) = \dfrac{1}{2}(\cos x + \operatorname{sen} x)$. Considere el conjunto de puntos $\{(x_i, f(x_i))\}_{i=0,1,2,3}$ con $x_i = i \cdot \pi/2$. Estime el error *general* cometido al aproximar $f(3\pi/4)$ con $P_3(3\pi/4)$.

3.7 Sea $f(x) = \dfrac{x^6}{84} - \dfrac{3\cos(2x)}{8}$. Considere el conjunto de puntos $\{(x_i, f(x_i))\}_{i=0,1,2,3}$ con $x_i = i \cdot 0.2$. Estime el error *general* cometido al aproximar $f(0.65)$ con $P_3(0.65)$.

3.8 Complete la fila 6 en la tabla

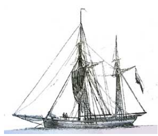
x_0	$Q_{0,0} = P_0$				
x_1	$Q_{1,0} = P_1$	$Q_{1,1} = P_{0,1}$			
x_2	$Q_{2,0} = P_2$	$Q_{2,1} = P_{1,2}$	$Q_{2,2} = P_{0,1,2}$		
x_3	$Q_{3,0} = P_3$	$Q_{3,1} = P_{2,3}$	$Q_{3,2} = P_{1,2,3}$	$Q_{3,3} = P_{0,1,2,3}$	
x_4	$Q_{4,0} = P_4$	$Q_{4,1} = P_{3,4}$	$Q_{4,2} = P_{2,3,4}$	$Q_{4,3} = P_{1,2,3,4}$	$Q_{4,4} = P_{0,1,2,3,4}$
x_5	$Q_{5,0} = P_5$	...	...	...	

3.9 Aproximar $F(0.25;1,2)$ (función gamma, ver ejemplo 23) usando interpolación lineal, cuadrática y cúbica.

3.10 Use el algoritmo de Neville para aproximar $F(0.25;1,2)$ usando nuestro criterio empírico para obtener una "mejor aproximación".

3.11 Supongamos que $x_0, x_1, ..., x_n$ son nodos distintos de un intervalo finito $[a,b]$. Sea $P_n(x)$ el polinomio interpolante obtenido con los datos $\{(x_i, f(x_i)\}_{i=0,1,...,n}$. Si $|f^{(n+1)}(x)| \leq M$ para x en $[a,b]$, muestre que si $x^* \in [a,b]$,

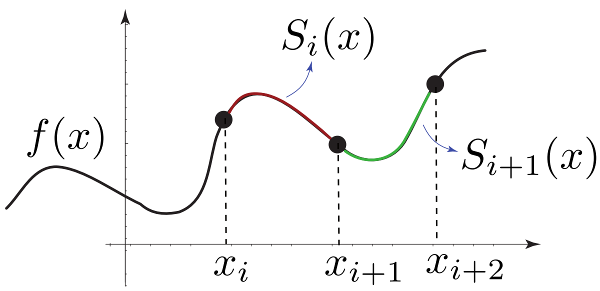
$$|f(x^*) - P_n(x^*)| \leq \frac{M}{(n+1)!}(b-a)^{n+1}$$



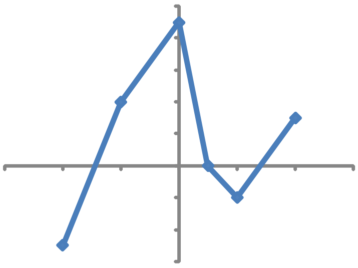
4 TRAZADORES CÚBICOS (CUBIC SPLINES).

Un trazador (spline) es una banda de hule delgada y flexible que se usa para dibujar curvas suaves a través de un conjunto de puntos. Los trazadores cúbicos (cubic splines) *naturales* se utilizan para crear una función que interpola un conjunto de puntos de datos. Esta función consiste en una unión de polinomios cúbicos, uno para cada intervalo, y está construido para ser una función con derivada primera y segunda continuas. El 'spline' cúbico natural también tiene su segunda derivada igual a cero en la coordenada x del primer punto y el último punto de la tabla de datos.

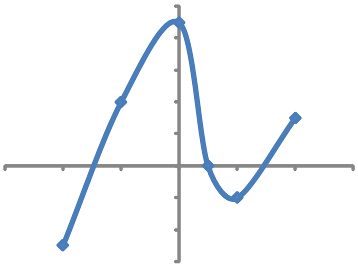
Supongamos que tenemos $n+1$ puntos $(x_0,y_0),...,(x_n,y_n)$ con $x_0 < x_1 < ... < x_n$. En vez de interpolar f con un solo polinomio que pase por todos estos puntos, interpolamos la función f en cada subintervalo $[x_k,x_{k+1}]$ con un polinomio cúbico (en realidad de grado ≤ 3) $S_k(x)$ de tal manera que el polinomio cúbico (o trazador cúbico) $S_i(x)$ en $[x_i,x_{i+1}]$ y el trazador cúbico $S_{i+1}(x)$ en $[x_{i+1},x_{i+2}]$, coincidan en x_{i+1} y que también sus derivadas primera y segunda coincidan en este punto. Cada trazador cúbico coincide con f en los extremos de cada intervalo.



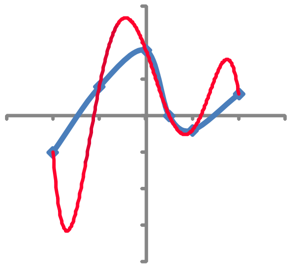
Una aplicación directa de los trazadores cúbicos es la de "suavizar curvas". Tanto en MSExcel como en *Calc* de OpenOffice o LibreOffice, en las gráficas de dispersión, los pares ordenados (x_i,y_i) se pueden unir con segmentos, con trazadores cúbicos o con el polinomio interpolante (también hay otras opciones, según el modelo o tendencia que se esté aplicando). En la gráfica de la figura (4) se muestra un conjunto de datos unidos por segmentos, unidos por trazadores cúbicos y unidos por el polinomio interpolante.



(a) Gráfico de líneas



(b) Gráfico ajustado con trazadores cúbicos



(c) Gráfico ajustado con trazadores y ajustado con el polinomio interpolante

Definición 25 (Trazador Cúbico).

Un *trazador cúbico* S es una función a trozos que interpola a f en los $n+1$ puntos (x_0,y_0), (x_1,y_1), (x_2,y_2), ..., (x_n,y_n) (con $a=x_0<x_1<...<x_n=b$). S es definida de la siguiente manera,

$$S(x)=\begin{cases} S_0(x) & \text{si} \quad x\in[x_0,x_1],\\ S_1(x) & \text{si} \quad x\in[x_1,x_2],\\ \vdots & \quad\quad \vdots \\ S_{n-1}(x) & \text{si} \quad x\in[x_{n-1},x_n],\end{cases}$$

Donde,

1. $S_i(x)=a_i+b_i(x-x_i)+c_i(x-x_i)^2+d_i(x-x_i)^3$ para $i=0,1,\ldots n-1$
2. $S(x_i)=y_i,\ i=0,1,...,n$. Para efectos prácticos, $S_j(x_j)=y_j$, $j=0,1,...,n-1$ y $S_{n-1}(x_{n-1})=y_{n-1}$ y $S_{n-1}(x_n)=y_n$. El siguiente item asegura que $S_j(x_{j+1})=y_{j+1}$.
3. $S_i(x_{i+1})=S_{i+1}(x_{i+1})$ para $i=0,1,...,n-2$
4. $S_i'(x_{i+1})=S_{i+1}'(x_{i+1})$ para $i=0,1,...,n-2$
5. $S_i''(x_{i+1})=S_{i+1}''(x_{i+1})$ para $i=0,1,...,n-2$
6. Se satisface una de las dos condiciones que siguen,
   (a) $S''(x_0)=S''(x_n)=0$ **(frontera libre o natural)**
   (b) $S'(x_0)=f'(x_0)$ y $S'(x_n)=f'(x_n)$ **(frontera sujeta)**

Algunas curvas presentan "picos" así que se construye un trazador para cada curva entre cada dos picos. El tratamiento de picos requiere usualmente un trazador con frontera sujeta.

El proceso de construcción del trazador cúbico consiste en determinar cada polinomio cúbico $S_j(x)$, es decir, buscar sus coeficientes a_i, b_i, c_i y d_i. La definición nos da las condiciones que se deben cumplir. De estas condiciones podemos obtener un sistema de ecuaciones $4n\times 4n$, donde las incógnitas son todos los coeficientes a_i, b_i, c_i y d_i, $i=0,1,...,n-1$. Lo que obtenemos es un trazador cúbico único.

Ejemplo 26

Determinar el trazador cúbico (frontera libre) para la siguiente tabla,

x_i	$y_i = \cos(3x_i^2)\ln(x_i^3+1)$
$x_0 = 0$	0
$x_1 = 0.75$	-0.0409838
$x_2 = 1.5$	1.31799

Solución: El trazador es,

$$S(x) = \begin{cases} S_0(x) = a_0 + b_0(x-x_0) + c_0(x-x_0)^2 + d_0(x-x_0)^3 & \text{si} \quad x \in [x_0, x_1], \\ S_1(x) = a_1 + b_1(x-x_1) + c_1(x-x_1)^2 + d_1(x-x_1)^3 & \text{si} \quad x \in [x_1, x_2]. \end{cases}$$

Hay que determinar los coeficientes de S_0 y S_1 resolviendo el sistema 8×8,

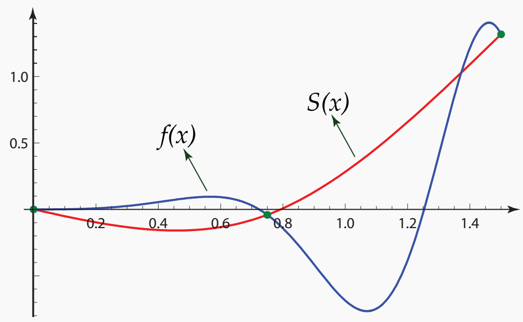
$$\begin{cases} S_0(x_0) &=& y_0 \\ S_1(x_1) &=& y_1 \\ S_1(x_2) &=& y_2 \\ S_0(x_1) &=& S_1(x_1) \\ S_0'(x_1) &=& S_1'(x_1) \\ S_0''(x_1) &=& S_1''(x_1) \\ S_0''(x_0) &=& 0 \\ S_1''(x_2) &=& 0 \end{cases} \iff \begin{cases} a_0 &=& 0 \\ a_1 &=& -0.0409838 \\ a_1 + 0.75b_1 + 0.5625c_1 + 0.421875d_1 &=& 1.31799 \\ a_0 + 0.75b_0 + 0.5625c_0 + 0.421875d_0 &=& a_1 \\ b_0 + 1.5c_0 + 1.6875d_0 &=& b_1 \\ 2c_0 + 4.5d_0 &=& 2c_1 \\ 2c_0 &=& 0 \\ 2c_1 + 4.5d_1 &=& 0 \end{cases}$$

La solución de este sistema es

$a_0 = 0$, $b_0 = -0.521299$, $c_0 = 0$, $d_0 = 0.829607$, $a_1 = -0.0409838$, $b_1 = 0.878663$, $c_1 = 1.86662$, y $d_1 = -0.829607$. Es decir,

$$S(x) = \begin{cases} S_0(x) &=& -0.521299x + 0.829607x^3 \quad \text{si} \quad x \in [0,\ 0.75] \\ S_1(x) &=& -0.0409838 + 0.878663(x-0.75) + 1.86662(x-0.75)^2 - 0.829607(x-0.75)^3 \quad \text{si} \quad x \in [0.75,\ 1.5]. \end{cases}$$

La representación gráfica para de S y f es



En las figuras (4.1) se muestra el trazador correspondiente a los nodos $x_0 = 0$, $x_1 = 0.5$, $x_2 = 1$, $x_3 = 1.5$ y $x_0 = 0$, $x_1 = 0.375$, $x_2 = 0.75$, $x_3 = 1.125$, $x_4 = 1.5$.

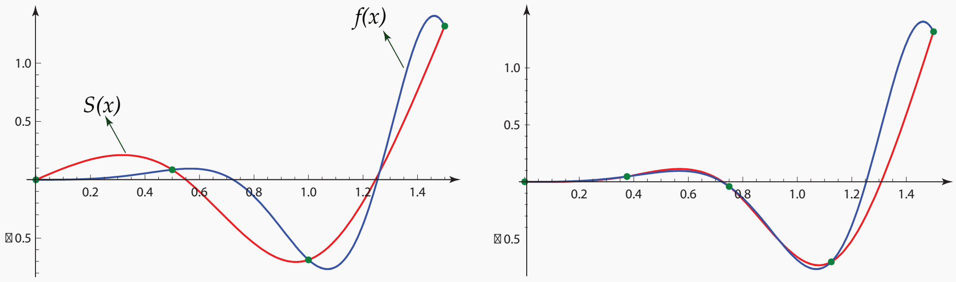


Figura 4.1 Trazador S y f con 3 y 4 puntos

Ejemplo 27

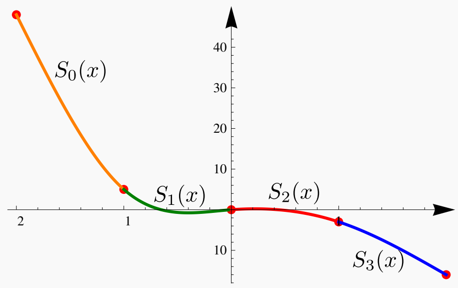
Determinar el trazador cúbico (frontera libre) para la siguiente tabla,

x_i	$y_i = x_i^4 - 4x_i^3$
$x_0 = -2$	48
$x_1 = -1$	5
$x_2 = 0$	0
$x_0 = -2$	-3
$x_1 = -1$	-16

Observe que la función $x^4 - 4x^3$ tiene un punto de inflexión en $x = 0$.

Solución: El trazador es,

$$S(x) = \begin{cases} S_0(x) = 9.85714(x+2)^3 - 52.8571(x+2) + 48 & \text{si} \quad x \in [-2,-1], \\ S_1(x) = -11.2857(x+1)^3 + 29.5714(x+1)^2 - 23.2857(x+1) + 5 & \text{si} \quad x \in [-1,0]. \\ S_2(x) = -0.714286x^3 - 4.28571x^2 + 2x & \text{si} \quad x \in [0,1], \\ S_3(x) = 2.14286(x-1)^3 - 6.42857(x-1)^2 - 8.71429(x-1) - 3 & \text{si} \quad x \in [1,22]. \end{cases}$$



Algoritmo para obtener el trazador cúbico (frontera natural). El proceso general sería como sigue. Sea $h_i = x_{i+1} - x_i$,

- De acuerdo al item (1.) de la definición (25), $S_i(x_i) = y_i \implies a_i = y_i$.

- Haciendo algunas manipulaciones algebraicas en el sistema, se obtiene

$$d_i = \frac{c_{i+1} - c_i}{3h_i} \quad \wedge \quad b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1}). \tag{4.1}$$

  La condición de frontera natural hace que $c_0 = c_n = 0$.

- Ahora todo depende del cálculo de los c_i's. Éstos se calculan resolviendo el sistema $(n+1) \times (n+1)$

$$\begin{pmatrix} 1 & 0 & 0 & & \cdots 0 \\ h_0 & 2(h_0+h_1) & h_1 & & \cdots 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \cdots 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \cdots & h_{n-3} & 2(h_{n-3}+h_{n-2}) & h_{n-2} \\ 0 & 0 & & & \cdots 1 \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix} = \begin{pmatrix} 0 \\ 3(f[x_2,x_1] - f[x_1,x_0]) \\ 3(f[x_3,x_2] - f[x_2,x_1]) \\ \vdots \\ 3(f[x_n,x_{n-1}] - f[x_{n-1},x_{n-2}]) \\ 0 \end{pmatrix}.$$

  Como antes, $f[x_i,x_j] = (y_i - y_j)/(x_i - x_j)$.

Ejemplo 28

Determinar el trazador cúbico (frontera natural) para la siguiente tabla,

x_i	$y_i = \cos(3x_i^2)\ln(x_i^3+1)$
$x_0 = 0$	0
$x_1 = 0.5$	0.0861805
$x_2 = 1$	−0.686211
$x_3 = 1.5$	1.31799

Solución: El trazador es,

$$S(x) = \begin{cases} S_0(x) = a_0 + b_0(x-x_0) + c_0(x-x_0)^2 + d_0(x-x_0)^3 & \text{si} \quad x \in [x_0, x_1], \\ S_1(x) = a_1 + b_1(x-x_1) + c_1(x-x_1)^2 + d_1(x-x_1)^3 & \text{si} \quad x \in [x_1, x_2]. \\ S_2(x) = a_2 + b_2(x-x_1) + c_2(x-x_1)^2 + d_2(x-x_1)^3 & \text{si} \quad x \in [x_2, x_3]. \end{cases}$$

Hay que determinar los coeficientes de S_0, S_1 y S_2. Iniciamos calculando los c_i's. Resolvemos el sistema 4×4. Recordemos que $h_i = x_{i+1} - x_i$,

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 3\left(\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}\right) \\ 3\left(\frac{y_3-y_2}{x_3-x_2} - \frac{y_2-y_1}{x_2-x_1}\right) \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 2. & 0.5 & 0 \\ 0 & 0.5 & 2. & 0.5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -5.15143 \\ 16.6596 \\ 0 \end{pmatrix}$$

Obtenemos $c_0 = 0$, $c_1 = -4.96871$, $c_2 = 9.57196$ y, por convenio, el comodín $c_3 = 0$.

Ahora podemos obtener el resto de coeficientes: $a_i = y_i$, los b_i's y los d_i's usando la ecuación (4.1).

$b_0 = 1.00048$, $b_1 = -1.48387$, $b_2 = 0.8177508$
$d_0 = -3.31247$, $d_1 = 9.69378$, $d_2 = -6.38131$.

Finalmente, el trazador cúbico es

$$\begin{cases} S_0(x) = -3.31247x^3 + 0.x^2 + 1.00048x & \text{si} \quad x \in [0, 0.5], \\ S_1(x) = 9.69378(x-0.5)^3 - 4.96871(x-0.5)^2 - 1.48387(x-0.5) + 0.0861805 & \text{si} \quad x \in [0.5, 1], \\ S_2(x) = -6.38131(x-1)^3 + 9.57196(x-1)^2 + 0.8177508(x-1.) - 0.686211 & \text{si} \quad x \in [1, 1.5]. \end{cases}$$

(● Clic para ir al programa en Internet: Liga 1 | Liga 2)

EJERCICIOS

4.1 Calcule el trazador cúbico (natural) para el conjunto de datos $(0,0), (1,1), (2,8)$.

4.2 Considere la tabla de datos,

$T(K)$	100	200	300	400	500	600
$B(cm^3/mol)$	-160	-35	-4.2	9.0	16.9	21.3

Tabla 4.1 Segundos Coeficientes viriales $B(cm^3/mol)$ para el nitrógeno

donde T es la temperatura y B es el segundo coeficiente virial.

a) Calcule el trazador cúbico (natural) para el conjunto de datos de la tabla.

b) ¿Cuál es el segundo coeficiente virial (interpolado) a $450K$?

c) Hacer la representación gráfica del trazador cúbico y del polinomio interpolante $P_5(x)$.

4.3 Considere la siguiente tabla de datos para el agua,

$T(C)$	50	60	65	75	80
$\rho(kg/m^3)$	988	985.7	980.5	974.8	971.6

Tabla 4.2

donde T es temperatura y ρ es la densidad. Hacer la representación gráfica del trazador cúbico y del polinomio interpolante $P_4(x)$.

5 ALGORITMOS E IMPLEMENTACIÓN CON OOOBASIC Y CALC.

5.1 Forma de Lagrange del polinomio interpolante

En esta primera implementación calculamos $P_n(x^*)$, es decir, no calculamos el polinomio interpolante; más bien calculamos este polinomio evaluado en un número x^*. Obtener el polinomio es sencillo. Al final de esta subsección se indica cómo hacerlo.

Recordemos que la forma de Lagrange del polinomio interpolante es

$$P_n(x) = y_0 L_{n,0}(x) + y_1 L_{n,1}(x) + ... + y_n L_{n,n}(x)$$

Para calcular de manera eficiente[5] los números $L_{n,k}(x^*)$, separamos el producto en dos factores

$$L_{n,k}(x^*) = \frac{(x^*-x_0)}{(x_k-x_0)} \cdot \frac{(x^*-x_1)}{(x_k-x_1)} \cdots \frac{(x^*-x_{k-1})}{(x_k-x_{k-1})} \cdot \frac{(x^*-x_{k+1})}{(x_k-x_{k+1})} \cdots \frac{(x^*-x_n)}{(x_k-x_n)}$$

El primer factor y el segundo factor se calcula con el ciclo

```
For k = 0 To n
      Ln(k) = 1
    Next k
For j = 1 To n+1
For k = 0 To j-1
          Ln(k)=(x*-X(j))/(X(k)-X(j))*Ln(k)
          Next k
For k = 0 To j-1
          Ln(j)=(x*-X(k))/(X(j)-X(k))*Ln(j)
          Next k
   Next j
```

En el ciclo principal, el segundo `For` produce el segundo factor de cada uno de los L_{nk}'s,

[5]Para n datos, el algoritmo que presentamos requiere $n^2 - n$ operaciones para calcular los L_{nk}. El algoritmo "directo"

```
For j = 0 To n-1
    For k=0 to n-1
        If k<>j Then
           Lnk=...
```

require $2n^2$ operaciones para hacer lo mismo.

$j=1$	`Ln(0) =`	$(x^*-x_1)/((x_0-x_1))$
$j=2$	`Ln(0) =` `Ln(1) =`	$(x^*-x_1)(x^*-x_2)/((x_0-x_1)(x_0-x_2))$ $(x^*-x_2)/((x_1-x_2))$

$j=3$	`Ln(0) =`
	`Ln(1) =`
	`Ln(2) =`
		$(x^*-x_1)(x^*-x_2)(x^*-x_3)/((x_0-x_1)(x_0-x_2)(x_0-x_3))$
		$(x^*-x_2)(x^*-x_3)/((x_1-x_2)(x_1-x_3))$
		$(x^*-x_3)/((x_2-x_3))$

$j=4$	`Ln(0) =`
	`Ln(1) =`
	`Ln(2) =`
	`Ln(3) =`
	...
		$(x^*-x_1)(x^*-x_2)(x^*-x_3)(x^*-x_4)/((x_0-x_1)(x_0-x_2)(x_0-x_3)(x_0-x_4))$
		$(x^*-x_2)(x^*-x_3)(x^*-x_4)/((x_1-x_2)(x_1-x_3)(x_1-x_4))$
		$(x^*-x_3)(x^*-x_4)/((x_2-x_3)(x_2-x_4))$
		$(x^*-x_4)/((x_3-x_4))$

Tabla 5.1

y el tercer `For` produce el primer factor de cada uno de los L_{nk}'s (excepto en el caso de `Ln(0)`).

Algoritmo 5.1: Forma de Lagrange del polinomio interpolante

Datos: $n+1$ datos $\{(x_i,y_i)\}_{i=0,1,...,n}$ con los x_i's distintos; y x^*
Salida: Polinomio interpolante evaluado en x^*, i.e. $P_n(x^*)$

```
suma= 0;
for k = 0 to n do
    Ln(k)= 1;
for j = 1 to n do
    for k = 0 to j − 1 do
        Ln(k)=(x* − x_j)/(x_k − x_j)*Ln(k)
    for k = 0 to j − 1 do
        Ln(j)=(x* − x_k)/(x_j − x_k)*Ln(j)
for k = 0 to n do
    suma= suma + Y(k)*Ln(k)
return suma
```

Implementación en OOo Basic y Calc. Implementamos una función `Lagrange(X,Y,x*)` que recibe los vectores $X=(x_0,x_1,...,x_n)$, $Y=(y_0,y_1,...,y_n)$, y el valor a interpolar `x*`. Esta función devuelve $P_n(x^*)$.

La función `Lagrange` la llamamos desde la subrutina `Main`. En esta subrutina usamos una variable `rango` para la selección de datos que hace el usuario. En OOoCalc los rangos inician en 0, por lo que si seleccionamos $n+1$ datos, `n = rango.Rows.getCount()-1`. El vector `X` y el vector `Y` los inicializamos con

```
For i=0 To n
    X(i)=rango.getCellByPosition(0, i).Value
    Y(i)=rango.getCellByPosition(1, i).Value
Next i
```

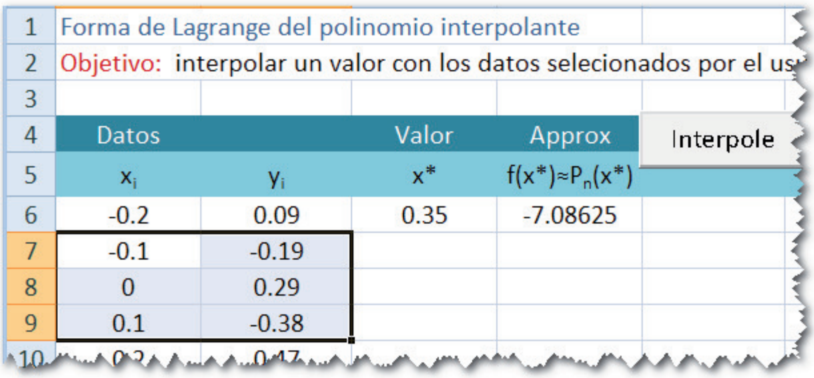
Una vez hecho esto, ya podemos llamar la función `Lagrange(X(),Y(),`x^*`)`. En esta función, como el último subíndice de `X` es n, ponemos `n = UBound(X)`.

```
 Function Lagrange(X(), Y(), xx)
 Dim suma,j, k, n
 Dim Ln()
   n = UBound(X)
   ReDim Ln(0 to n)

  For k = 0  To n
       Ln(k) = 1
  Next k
  For j = 1  To n
     For k = 0  To j-1
       Ln(k)=(xx-X(j))/(X(k)-X(j))*Ln(k)
    Next k
    For k = 0 { To} j-1
         Ln(j)=(xx-X(k))/(X(j)-X(k))*Ln(j)
    Next k
 Next j
  For k = 0 { To} n
       suma= suma+Y(k)*Ln(k)
  Next k
Lagrange= suma
End Function
```

Par usar esta función, vamos a usar el cuaderno de la figura (5.1). El usuario debe seleccionar una subtabla y hacer clic en el botón. Como ya indicamos, en OOoBasic este evento lo manejamos así: la selección se recibe en una variable `range`. Luego pasamos la información de este rango a los vectores `X()` e `Y()`. Luego llamamos a la función `Lagrange(X,Y,`x^*`)`. Observe que no es necesario pasar los valores del rango a los vectores X e Y, solo lo hacemos porque de esta manera la implementación va a la par de la teoría.

La lectura del rango y la lectura del valor x^* la hacemos desde la subrutina `Main`. Una vez leídos estos datos, se llama a la la función `Lagrange`. El botón tendrá asociada esta subrutina.



1	Forma de Lagrange del polinomio interpolante				
2	Objetivo: interpolar un valor con los datos selecionados por el us				
3					
4	Datos		Valor	Approx	Interpole
5	x_i	y_i	x*	$f(x^*) \approx P_n(x^*)$	
6	-0.2	0.09	0.35	-7.08625	
7	-0.1	-0.19			
8	0	0.29			
9	0.1	-0.38			

Figura 5.1 Cuaderno para la implementación de la forma de Lagrange del polinomio interpolante.

```
Option Explicit
Sub Main
  Dim xx     'no podemos poner x*
  Dim rango
  Dim X(), Y()
  Dim n,i

     xx  = ThisComponent.Sheets(0).getCellRangeByName("C6").Value
           'Rango seleccionado por el usuario
   rango = ThisComponent.getCurrentSelection()
       n = rango.Rows.getCount()-1 'n+1=número de datos

   If n<1 Then
     MsgBox "Por favor, seleccione los datos."
     Exit Sub
   End If

         'X() y Y()
   ReDim X(0 To n)
   ReDim Y(0 To n)
   For i=0 To n
     X(i)=rango.getCellByPosition(0, i).Value
     Y(i)=rango.getCellByPosition(1, i).Value
   Next i
         'Imprimimos en la celda "D6"
  ThisComponent.Sheets(0).getCellRangeByName("D6").Value=Lagrange(X,Y,xx)
End Sub
```

¿Cómo imprimir el polinomio?. Para imprimir el polinomio solo habría que hacer una pequeña modificación. La función `polyLagrange(X,Y)` devuelve un cadena de texto ('String'). Usamos la función `Str()` para convertir los números a cadena de texto y la concatenación del texto se hace con el operador '+'.

```
Function polyLagrange(X(), Y())
 Dim suma,j, k, n
 Dim Dn, Nn, Pn 'Variables para el texto
 Dim Ln()
 n = UBound(X)
 ReDim Ln(0 to n,1 to 2)
 For k = 0 To n
     Ln(k,1) =""
     Ln(k,2) =""
 Next k

 For j = 1 To n
    For k = 0 To j-1
       Ln(k,1)=Ln(k,1)+"(x-"+Str(X(j))+")"
       Ln(k,2)=Ln(k,2)+"("+Str(X(k))+"-"+Str(X(j))+")"
    Next k
```

```
    For k = 0 To j-1
       Ln(j,1)="(x-"+Str(X(k))+")"+Ln(j,1)
       Ln(j,2)="("+Str(X(j))+"-"+Str(X(k))+")"+Ln(j,2)
    Next k
Next j

For k = 0 To n
     Pn=Pn+"+"+Str(Y(k))+"*"+Ln(k,1)+"/("+Ln(k,2)+")"
Next k
                'Depuración -- = +,+- = -
 Pn=ReplaceString(Pn," + ","- -",)
 Pn=ReplaceString(Pn," - ","+ -")
 PolyLagrange = "P\_n(x)= " + Pn
End Function
```

En el caso de la tabla de la figura (5.1), la corrida entrega el polinomio

```
P_n(x)=  -0.19*(x - 0)(x - 0.1)/((-0.1 - 0)(-0.1 - 0.1))
          +0.29*(x + 0.1)(x-0.1)/((0 + 0.1)(0 - 0.1))
          -0.38*(x + 0.1)(x-0)/((0.1 + 0.1)(0.1 - 0))
```

Observe que usamos la función `ReplaceString`. En caso de que no esté disponible, puede agregar el código

```
Function ReplaceString(ByVal String, NewReplace, OldReplace as String) as String
 ReplaceString=join(split(String,OldReplace),NewReplace)
End Function
```

Implementación en *MATHEMATICA*. En *MATHEMATICA* podemos escribir el código de manera más directa. El código se puede escribir usando la paleta `"Basic Input"`. El módulo `Lagrange` calcula el polinomio interpolante,

$$
\begin{array}{l}
\texttt{Lagrange[XY\_] :=} \\
\quad \texttt{Module}\Big[\{j,\ k,\ n,\ X,\ Y\}, \\
\quad\quad X_{k\_} := \texttt{Transpose[XY]}_{[[1,k+1]]}; \\
\quad\quad Y_{k\_} := \texttt{Transpose[XY]}_{[[2,k+1]]}; \\
\quad\quad n = \texttt{Length[XY]} - 1; \\
\quad\quad \texttt{Return}\Big[\displaystyle\sum_{k=0}^{n} Y_k \left(\prod_{j=0}^{k-1} \frac{x - X_j}{X_k - X_j}\right)\left(\prod_{j=k+1}^{n} \frac{x - X_j}{X_k - X_j}\right)\Big]\Big];
\end{array}
$$

Para hacer una corrida con la misma selección de la figura (5.1), ejecutamos el código

```
XY = {-0.1, -0.19}, {0, 0.29}, {0.1, -0.38}};
P[x] = Lagrange[XY], P[0.35]
```

La salida es `{-9.5 (-0.1 + x) x - 29. (-0.1 + x) (0.1+ x) -19. x (0.1+ x), -7.08625}.`

EJERCICIOS

5.1 Vamos a usar la tabla (I.1, pág 3.) para interpolar $f(0.25)$ y comparar con el valor correcto $f(0.25) = 0.001200416039457...$

a) Interpolar $f(0.25)$ con tres datos
b) Interpolar $f(0.25)$ con cuatro datos
c) Interpolar $f(0.25)$ con toda la tabla

5.2 En este ejercicio vamos a hacer la representación gráfica de una función f conocida y su polinomio interpolante $P_3(x)$.

a) Calcule la forma de Lagrange del polinomio interpolante $P_3(x)$, usando la tabla de datos que está a la derecha. Hacer la representación gráfica de ambas funciones en $[0,2]$
b) Repita el ejercicio anterior ampliando la tabla a 15 datos.

x	$y = \cos(3x^2)\ln(x^3+1)$
0	0
0.5	0.0861805
1	-0.686211
1.5	1.31799

5.3 Considere la función de Runge, $f(x) = \dfrac{1}{1+25x^2}$.

a) Considere el conjunto de datos

$$x_i = -1 \,+\, i \cdot \frac{2}{n},\, y_i = f(x_i),\, i = 0,1,...,n \,\subseteq\, [-1,1].$$

Hacer la representación gráfica de los polinomios interpolantes para $n = 5,10,20$. Represente estos polinomios conjuntamente con f.

b) **(Nodos de TChebyshev).** Usando los datos $\{(x_i, f(x_i))\}_{i=0,1,...,n} \subseteq [-1,1]$ con $x_i = \cos\left(\dfrac{2i+1}{2n+2}\pi\right)$, calcule los polinomios interpolantes para $n = 3,8,20$ y represente estos polinomios conjuntamente con f.

5.2 Forma modificada y forma baricéntrica de Lagrange.

La implementación se centra en el cálculo de los ω_k. Una vez calculados estos números, armar cada polinomio interpolante es algo directo. Recordemos que

$$P_n(x) = \ell(x)\sum_{k=0}^{n} \frac{\omega_k}{x - x_k}\, y_k$$

donde $\omega_k = \dfrac{1}{(x_k - x_1)(x_k - x_2)...(x_k - x_{k-1})(x_k - x_{k+1})...(x_k - x_n)}$.

- En el algoritmo, para calcular cada ω_k, separamos el denominador en dos factores

$$(x_k - x_1)(x_k - x_2)...(x_k - x_{k-1})(x_k - x_{k+1})...(x_k - x_n).$$

El primer factor y el segundo factor en el denominador de cada uno de los ω_k's se calcula con el ciclo

$w_k = 1,$ `para cada` $k = 0, ..., n$
`For` $j = 1$ `To` n
  `For` $k = 0$ `To` $j - 1$
    $w_k \ = \ (x_k - x_j) w_k$
  `For` $k = 0$ `To` $j - 1$
    $w_j \ = \ (x_j - x_k) w_j$

En este ciclo, el primer `For` anidado produce el segundo factor en el denominador de cada uno de los ω_k's,

$w_0 = (x_0 - x_1)(x_0 - x_2) \cdots (x_0 - x_n),$
$w_1 = (x_1 - x_2)(x_1 - x_3) \cdots (x_1 - x_n),$
$w_2 = (x_2 - x_3)(x_2 - x_4) \cdots (x_2 - x_n),$

y el segundo `For` anidado completa el producto,

$w_1 = (x_1 - x_0)(x_1 - x_2)(x_1 - x_3) \cdots (x_1 - x_n),$
$w_2 = (x_2 - x_0)(x_2 - x_1)(x_2 - x_3)(x_2 - x_4) \cdots (x_2 - x_n),$ etc.

Finalmente, $\omega_k = 1/w_k$.

- Si usamos nodos de TChebyshev, el cálculo es directo: $\omega_k = (-1)^k \operatorname{sen} \dfrac{(2k+1)\pi}{2n+2}$. Este último resultado se obtiene del siguiente cálculo:

$$\begin{aligned} \omega_k^{-1} &= \lim_{x \to x_k} \frac{\ell(x)}{(x - x_k)} \\ &= \lim_{x \to x_k} \frac{\ell(x) - \ell(k)}{(x - x_k)}, \quad \text{pues} \quad \ell(x_k) = 0; \\ &= \ell'(x_k). \end{aligned}$$

Algoritmo 5.2: Cálculo de los Pesos Baricéntricos

Datos: $n+1$ nodos distintos $\{x_i\}_{i=0,1,\ldots,n}$.
Salida: Pesos baricéntricos ω_k, $k = 0, 1, ..., n$

if $\{x_i\}_{i=0,\ldots,n}$ *son nodos de TChebyshev* **then**
  $w_k = (-1)^k \operatorname{sen} \dfrac{(2k+1)\pi}{2n+2}$, $k = 0, ..., n$
else
  for $k = 0$ **to** n **do**
    $w_k = 1$
  for $j = 1$ **to** n **do**
    for $k = 0$ **to** $j - 1$ **do**
      $w_k = (x_k - x_j) w_k$
    for $k = 0$ **to** $j - 1$ **do**
      $w_j = (x_j - x_k) w_j$
  for $k = 0$ **to** n **do**
    $w_k = 1/w_k$
return $w_0, w_1, ..., w_n$

Implementación en OOo Basic y Calc. Aquí solo implementamos la función `ModificadaLagrange(X,Y,x*)`. Esta subrutina recibe los vectores $X = (x_0, x_1, ..., x_n)$, $Y = (y_0, y_1, ..., y_n)$, y el valor a interpolar x*. Esta función devuelve $P_n(x^*)$. El código de esta función sería,

```
Function  ModificadaLagrange(X(),Y(), xval)
Dim suma,j, k, n,L
Dim W()
 n = UBound(X)
 ReDim W(0 to n)
 For k = 0 To n
    W(k) = 1
 Next k
 For j = 1 To n
     For k = 0 To j-1
        W(k)=(X(k)-X(j))*W(k)
     Next k
     For k = 0 To j-1
        W(j)=(X(j)-X(k))*W(j)
     Next k
 Next j

 For k = 0 To n
    W(k)=1/W(k)
 Next k

 L=1
 For k = 0 To n
    L=(xval-X(k))*L
 Next k

 For k = 0 To n
    suma = suma+Y(k)*W(k)/(xval-X(k))
 Next k
ModificadaLagrange = L*suma
End Function
```

EJERCICIOS

5.4 Agregar la forma modificada de Lagrange al cuaderno que contiene la función `Lagrange`.

5.5 Implementar la forma baricéntrica de Lagrange en el caso general y el caso de nodos igualmente espaciados.

5.3 Forma de Newton del polinomio interpolante.

La forma de Newton del polinomio interpolante es

$$\begin{aligned} P(x) &= f[x_0] + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1) \\ &\quad + \cdots + f[x_0,\ldots,x_n](x-x_0)\cdots(x-x_{n-1}), \end{aligned}$$

donde los coeficientes están en la diagonal de la matriz de diferencias divididas,

$$\begin{array}{lllll} y_0 = f[x_0] & & & & \\ y_1 & f[x_0,x_1] & & & \\ y_2 & f[x_1,x_2] & f[x_0,x_1,x_2] & & \\ y_3 & f[x_2,x_3] & f[x_1,x_2,x_3] & & \\ \vdots & \vdots & \vdots & \ddots & \\ y_n & f[x_{n-1},x_n] & f[x_{n-2},x_{n-1},x_n] & \cdots & f[x_0,x_1,\ldots,x_n] \end{array}$$

Para la implementación de la fórmula de diferencias divididas de Newton es conveniente reescribir el polinomio como

$$\begin{aligned} P(x) &= C_{0,0} + C_{1,1}(x-x_0) + C_{2,2}(x-x_0)(x-x_1) \\ &\quad + \cdots + C_{n,n}(x-x_0)\cdots(x-x_{n-1}), \end{aligned}$$

y la matriz de diferencias divididas como

$$\begin{array}{lllll} y_0 = C_{0,0} & & & & \\ y_1 & C_{1,1} & & & \\ y_2 & C_{2,1} & C_{2,2} & & \\ y_3 & C_{3,1} & C_{3,2} & & \\ \vdots & \vdots & \vdots & \ddots & \\ y_n & C_{n,1} & C_{n,2} & \cdots & C_{n,n} \end{array}$$

Para el cálculo de los $C_{i,i}$'s usamos la fórmula recursiva:

$$\begin{aligned} C_{i,0} &= y_i, \;\; i = 0,2,\ldots,n \\ C_{i,j} &= \frac{C_{i,j-1} \; - \; C_{i-1,j-1}}{x_i - x_{i-j}}, \quad j = 1,2,\ldots n, \;\; i = j,2,\ldots,n \end{aligned} \tag{5.1}$$

Algoritmo 5.3: Diferencias Divididas de Newton

Datos: $\{(x_i, y_i)\}_{i=0,1,\ldots,n}$ con los x_i's distintos.
Salida: Coeficientes del polinomio interpolante: $C_{0,0}$, $C_{1,1}, \ldots, C_{n,n}$

for $i = 1$ **to** n **do**
  $C_{i,0} = y_i$
;
for $j = 1$ **to** n **do**
  for $i = j$ **to** n **do**
    $C_{i,j} = \dfrac{C_{i,j-1} \; - \; C_{i-1,j-1}}{x_i - x_{i-j}}$
return $C_{0,0}$, $C_{1,1}, \ldots, C_{n,n}$

Para evaluar el polinomio interpolante en x^* se requiere los coeficientes $C_{j,j}$ y los nodos $x_0, \ldots x_n$.

Algoritmo 5.4: Evaluar la forma de Newton del polinomio interpolante

Datos: Nodos $x_0, x_1, \ldots, x)n$, coeficientes $C_0, C_1, \ldots, C_n$ y x^*
Salida: $P_n(x^*)$

suma $= C_0$;
factor $= 1$;
for $j = 1$ **to** n **do**
  factor=factor$\cdot(x^* - x_{j-1})$;
  suma=suma+$C_j\cdot$factor;
return suma

Implementación en OOo Basic y Calc. Como en la implementación de la forma de Lagrange, aquí también tenemos los datos en dos vectores `X` e `Y`. La función `DDNewton(X,Y)` devuelve un vector con la diagonal de la matriz de diferencias divididas: $C_0, C_1, \ldots, C_n$. Para evaluar la forma de Newton implementamos la función `EvalDDNewton(C(),X(),x*)` usando el algoritmo (5.4). También se incluye la función `PolyDDNewton(C(),X())` que devuelve el polinomio $P_n(x)$ y la subrutina `MatrizDDNewton(X(), Y(),columna, fila)` para imprimir la matriz de diferencias divididas desde la celda `(columna,fila)`.

Para ejecutar estas macros, usamos como referencia el cuaderno de la figura (5.2).

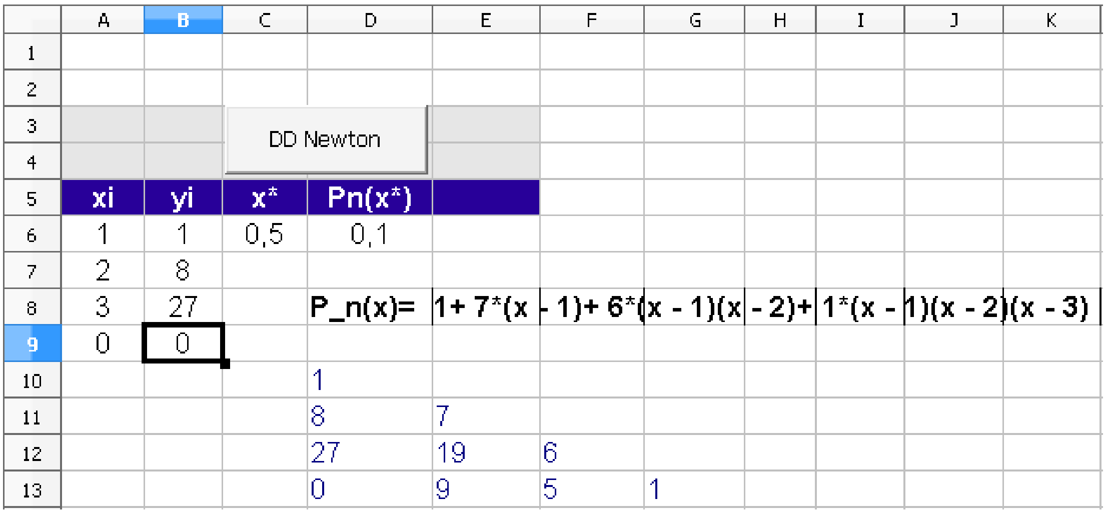


Figura 5.2 Cuaderno para la implementación de la forma de Newton del polinomio interpolante.

El botón ejecuta la subrutina `Main`.

```
Option Explicit
 Sub  Main
  Dim xx,i,Hoja
  Dim rango
  Dim X(), Y(),C()
  Hoja= ThisComponent.Sheets(0)
  xx  = Hoja.getCellRangeByName("C6").Value

           'Rango seleccionado por el usuario
  rango = ThisComponent.getCurrentSelection()
      n = rango.Rows.getCount()-1 'n+1=número de datos

  If n<1 Then
     MsgBox "Por favor, seleccione los datos."
     Exit Sub
  End If
  ReDim X(0 To n)
  ReDim Y(0 To n)
  ReDim C(0 To n)

  For i=0 To n
     X(i)=rango.getCellByPosition(0, i).Value
     Y(i)=rango.getCellByPosition(1, i).Value
  Next i
  C= DDNewton(X, Y)
         'imprimir Pn(x*)
  Hoja.getCellRangeByName("D6").Value= EvalDDNewton(C,X, xx)
         'imprimir Pn(x)
  Hoja.getCellRangeByName("D8").setString(PolyDDNewton(C,X))
         'imprimir la matriz
  Call MatrizDDNewton(X, Y,3,9)
 End Sub
```

Las funciones `DDNewton` usa dos vectores `C1` y `C2`. En `C1` se almacena la columna actual y en `C2` la nueva columna.Cada elemento de la diagonal `C2(j)` se almacena en el vector `Cij`. Para recalcular `C2` se actualiza `C1`. En vez de poner `C1=C2` se usa un `For` para hacer la copia componente a componente (en las nuevas versiones de OpenOffice y LibreOffice esto puede haber cambiado y se podría poner directamente `C1=C2` de manera segura).

```
Function  DDNewton(X(), Y()) As Variant
  Dim  i, j, n,k
  Dim C1() As double, C2() As double, Cij() as double

  n = UBound(X)
  ReDim C1(0 to n)
  ReDim C2(0 to n)
  ReDim Cij(0 to n)
  'continua en la pág siguiente
```

```
  For i = 0 To n
    C1(i) = Y(i)
  Next i
  Cij(0)=Y(0)
  For j = 1 To n
     For i = j To n
        'Calcula la columna j
         C2(i) = (C1(i) - C1(i - 1)) / (X(i) - X(i - j))
     Next i
     Cij(j)=C2(j)
        'Actualiza C1
     For k = 0 To n
         C1(k) = C2(k)
     Next k
  Next j
  DDNewton = Cij()
 End Function
```

```
Function  EvalDDNewton(Cij(),X(), xx) As Double
 Dim j, n, suma, factor
 suma=Cij(0)
 factor=1
 n = UBound(Cij)
  For j = 1 To n
     factor=factor*(xx-X(j-1))
     suma = suma+Cij(j)*factor
  Next j
  EvalDDNewton=suma
End Function
```

```
 Function PolyDDNewton(Cij(),X()) As String
 Dim j,n
 Dim Pn, factor
 Pn= Str(Cij(0))
 factor=""
 n = UBound(Cij)
 For j = 1 To n
     factor=factor+"(x -"+Str(X(j-1))+")"
     Pn = Pn+"+"+Str(Cij(j))+"*"+factor
  Next j
               'Depuración -- = +,+- = -
  Pn=ReplaceString(Pn," + "," - -",)
  Pn=ReplaceString(Pn," - ","+ -")
  PolyDDNewton="P\_n(x)= " + Pn
End Function
```

Si no esta disponible la subrutina `ReplaceString`, se agrega

```
Function ReplaceString(ByVal String, NewStr, OldStr as String) as String
    ReplaceString=join(split(String,OldStr),NewStr)
End Function
```

Para imprimir la matriz de diferencias divididas se modifica la función `DDNewton`.

```
Sub MatrizDDNewton(X(), Y(),columna, fila)
 Dim  i, j,n, k, Hoja
 Dim C1(), C2(), Cij()
 Hoja= ThisComponent.Sheets(0)
 n = UBound(X)
 ReDim C1(0 to n)
 ReDim C2(0 to n)
 ReDim Cij(0 to n)
 For i = 0 To n
  C1(i) = Y(i)   'Imprime primera columna
  Hoja.getCellByPosition(columna, fila+i).Value=C1(i)
 Next i
 Cij(0)=Y(0)
 For j = 1 To n
  For i = j To n 'Imprime la columna j
   C2(i) = (C1(i) - C1(i - 1)) / (X(i) - X(i - j))
   Hoja.getCellByPosition(columna+j, fila+i).Value=C2(i)
  Next i
   Cij(j)=C2(j)
   For k = 0 To n
    C1(k) = C2(k)
   Next k
 Next j
 End Sub
```

EJERCICIOS

5.6 Vamos a usar la tabla (I.1, pág 3) para interpolar $f(0.25)$ y comparar con el valor correcto $f(0.25) = 0.001200416039457...$

a) Interpolar $f(0.25)$ con tres datos
b) Interpolar $f(0.25)$ con cuatro datos
c) Interpolar $f(0.25)$ con toda la tabla

5.7 En este ejercicio vamos a hacer la representación gráfica de una función f conocida y su polinomio interpolante $P_3(x)$.

a) Calcule la forma de Newton del polinomio interpolante $P_3(x)$, usa-ndo la tabla de datos que está a la derecha. Hacer la representación gráfica de ambas funciones en $[0,2]$

b) Repita el ejercicio anterior ampliando la tabla a 15 datos.

x	$y = \cos(3x^2)\ln(x^3+1)$
0	0
0.5	0.0861805
1	−0.686211
1.5	1.31799

5.8 Considere la función de Runge, $f(x) = \dfrac{1}{1+25x^2}$.

a) Considere el conjunto de datos

$$x_i = -1 \, + \, i \cdot \frac{2}{n},\ y_i = f(x_i),\ i = 0,1,...,n \ \subseteq \ [-1,1].$$

Hacer la representación gráfica de los polinomios interpolantes para $n = 5,10,20$. Represente estos polinomios conjuntamente con f.

b) **(Nodos de Chebyshev).** Usando los datos $\{(x_i, f(x_i))\}_{i=0,1,...,n} \subseteq [-1,1]$ con $x_i = \cos\left(\dfrac{2i+1}{2n+2}\pi\right)$, calcule los polinomios interpolantes para $n = 3,8,20$ y represente estos polinomios conjuntamente con f.

5.4 Trazadores cúbicos

Recordemos que un *trazador cúbico* S es una función a trozos que interpola a f en los $n+1$ puntos (x_0,y_0), (x_1,y_1), (x_2,y_2), $\ldots$, (x_n,y_n) (con $a = x_0 < x_1 < ... < x_n = b$). S es definida de la siguiente manera,

$$S(x) = \begin{cases} S_0(x) & \text{si} \quad x \in [x_0, x_1], \\ S_1(x) & \text{si} \quad x \in [x_1, x_2], \\ \quad \vdots & \quad\quad \vdots \\ S_{n-1}(x) & \text{si} \quad x \in [x_{n-1}, x_n], \end{cases}$$

Donde, $S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$ para $i = 0,1,\ldots n-1$.

Para el cálculo de los coeficientes a_i, b_i, c_i y d_i usamos el algoritmo

Algoritmo 5.5: Trazador cúbico con frontera natural

Datos: $(x_0, y_0), (x_1, y_1), \ldots (x_n, y_n)$ con $x_0 < x_1 < \cdots < x_n$.
Salida: Coeficientes a_j, b_j, c_j, d_j, para cada $j = 0, 1, \ldots n-1$.

for $j = 0, 1, \ldots n-1$ **do**
  $h_j = x_{j+1} - x_j$;
  $a_j = y_j$;

$a_n = y_n$;
for $j = 1, \ldots n-1$ **do**
  $\alpha_j = \dfrac{3}{h_j} \cdot (a_{j+1} - a_j) - \dfrac{3}{h_{j-1}} \cdot (a_j - a_{j-1})$.

$l_0 = 1$, $\mu_0 = 0$, $z_0 = 0$;
$l_n = 1$, $z_n = 0$, $c_n = 0$;
for $i = 1, 2, \ldots, n-1$ **do**
  $l_i = 2(x_{i+1} - x_{i-1})\mu_{i-1}$;
  $\mu_i = h_i / l_i$;
  $z_i = (\alpha_i - h_{i-1} z_{i-1}) / l_i$;

for $j = n-1, n-2, \ldots, 0$ **do**
  $c_j = z_j - \mu_j c_{j+1}$;
  $b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$;
  $d_j = (c_{j+1} - c_j)/(3h_j)$;

return a_j, b_j, c_j, d_j, $j = 0, 1, \ldots n-1$.

Implementación en OOo Basic y Calc. La función `TrazadorCubico(X(), Y())` recibe los datos en dos vectores `X,Y` y devuelve una matriz con los coeficientes de las polinomios $S_i(x)$. En esta implementación usamos la función `CeldaCF` de la biblioteca `BblMatematica`.

Para ejecutar estas macros, usamos como referencia el cuaderno de la figura (5.3).

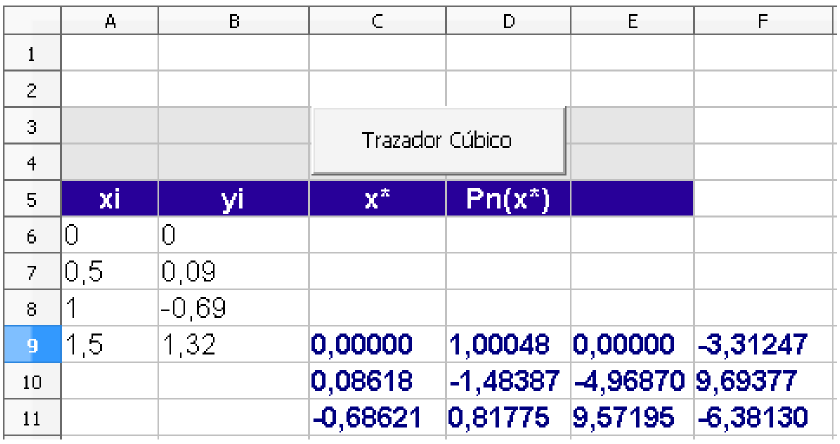


Figura 5.3 Cuaderno para la implementación de el trazador cúbico.

El botón ejecuta la subrutina `Main`.

```
Option Explicit
  Sub  Main
      'Cargar la biblioteca personal BblMatematica para usar CeldaCF
      'Si no usa la biblioteca, debe agregar la función (está más abajo).
  BasicLibraries.loadLibrary("BblMatematica")
  Dim xval

  Dim rango
  Dim X(), Y(),S()
  Dim n,i, j, fi, co

        'Rango selecionado por el usuario
  rango = ThisComponent.getCurrentSelection()
      n = rango.Rows.getCount()-1 'n+1=número de datos

 If n<1 Then
    MsgBox "Por favor, seleccione los datos."
    Exit Sub
 End If
 ReDim X(0 To n)
 ReDim Y(0 To n)
 Redim S(0 to n-1, 0 to 3)\icom{Coeficientes del trazador c□bico}

 For i=0 To n
    X(i)=rango.getCellByPosition(0, i).Value
    Y(i)=rango.getCellByPosition(1, i).Value
 Next i

 S = TrazadorCubico(X, Y)

         'Imprimir en columna 2, fila 8
 co=2 : fi=8
  For j = 0 To n - 1
        'CeldaCF' es una función especial, ver más abajo.
   CeldaCF(co,fi + j).Value    = S(j,0) 'a(j)
   CeldaCF(co+1,fi + j ).Value = S(j,1) 'b(j)
   CeldaCF(co+2,fi + j).Value  = S(j,2) 'c(j)
   CeldaCF(co+3,fi + j).Value  = S(j,3) 'd(j)
 Next j
End Sub
```

```
Function  TrazadorCubico(X(), Y()) As Variant
 Dim i, j, n
 Dim a(), b(), c(), d(), h(), l(), mu(), z(), alfa()
 Dim S()
 n = UBound(X)
 ReDim a(0 to n)
```

```
 ReDim b(0 to n)
 ReDim c(0 to n)
 ReDim d(0 to n)
 ReDim h(0 to n)
 ReDim l(0 to n)
 ReDim mu(0 to n)
 ReDim z(0 to n)
 ReDim alfa(0 to n)
 Redim S(0 to n-1, 0 to 3)

 For j = 0 To n - 1
    h(j) = X(j + 1) - X(j)
    a(j) = Y(j)
 Next j

 a(n) = Y(n)

 For j = 1 To n - 1
   alfa(j) = 3 / h(j) * (a(j + 1) - a(j)) - 3 / h(j - 1) * (a(j) - a(j - 1))
 Next j
 l(0) = 1
 mu(0) = 0
 z(0) = 0
 l(n) = 1
 z(n) = 0
 c(n) = 0

 For i = 1 To n - 1
    l(i) = 2 * (X(i + 1) - X(i - 1)) - h(i - 1) * mu(i - 1)
    mu(i) = h(i) / l(i)
    z(i) = (alfa(i) - h(i - 1) * z(i - 1)) / l(i)
 Next i
 For j = n - 1 To 0 Step -1
  c(j) = z(j) - mu(j) * c(j + 1)
  b(j) = (a(j + 1) - a(j)) / h(j) - h(j) * (c(j + 1) + 2 * c(j)) / 3
  d(j) = (c(j + 1) - c(j)) / (3 * h(j))
 Next j
     'Coeficientes
 For j = 0 To n - 1
    S(j,0)=a(j)
    S(j,1)=b(j)
    S(j,2)=c(j)
    S(j,3)=d(j)
 Next j
 TrazadorCubico=S()
End Function
```

Si la función `CeldaCF` no esta disponible, agregar

```
Function celdaCF(columna, fila)
celdaCF = thisComponent.Sheets(0).getCellByPosition(columna, fila)
End Function
```

EJERCICIOS

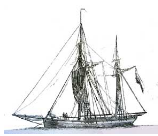
5.9 Implementar una función para imprimir cada uno de los polinomios $S_i(x)$.

5.10 Verifique su implementación con el ejemplo (28).

5.11 Implementar una función para evaluar $S(x^*)$.

5.12 Usar la implementación para interpolar $f(0.27)$ y $f(0.554)$; usando la tabla

x_i	y_i
$x_0 = 0$	0.656
$x_1 = 0.2$	-0.086
$x_2 = 0.4$	0.68
$x_3 = 0.6$	-1.799



PARTE II

INTERPOLACIÓN. ASPECTOS TEÓRICOS.

Un problema de interpolación polinomial se especifica como sigue: dados $n+1$ puntos $(x_0,y_0),(x_1,y_1),...,(x_n,y_n)$, siendo todos los x_i's distintos, encontrar un polinomio $P_n(x)$ de grado $\leq n$ tal que

$$P_n(x_i) = y_i, \;\; i = 0,1,2,...,n \qquad \text{(I.2)}$$

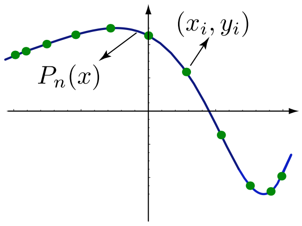


Figura I.4 $P_n(x)$

Teorema 29 (Existencia y unicidad).

Sean $x_0,x_1,...,x_n$ números reales distintos y $n \geq 0$. Si $y_0,y_1,...,y_n$ son números reales arbitrarios, existe un único polinomio P_n, de grado menor o igual a n tal que $P_n(x_i) = y_i$, $i = 0,1,...,n$.

Prueba: Unicidad: Sean P_n y Q_n son dos polinomios interpolantes de grado a lo sumo n tales que $P_n(x_i) = Q_n(x_i) = y_i$, $i = 0,1,2,...,n$. Entonces el polinomio $R(x) = P_n(x) - Q_n(x)$ es de grado a lo sumo n y tiene a lo sumo n ceros o es el polinomio nulo (teorema fundamental del algebra). Como $R(x)$ tiene $n+1$ ceros, debe ser el polinomio nulo, es decir, $P_n(x) \equiv Q_n(x)$.

Existencia: La prueba es por inducción.

Para $n = 0$ existe un polinomio P_0 de grado a lo sumo 0 tal que $P_0(x_0) = y_0$. Por supuesto, se trata del polinomio nulo.

Supongamos que el resultado es cierto para $n-1 \geq 0$, es decir, existe P_{n-1} de grado a lo sumo $n-1$ tal que $P_{n-1}(x_i) = y_i$, $i = 0,1,...,n-1$. Consideremos el polinomio

$$P_n(x) = P_{n-1}(x) + a_n\,(x-x_0)(x-x_1)...(x-x_{n-1}), \;\; a_n \;\; \text{una constante.}$$

Como $\prod_{j=0}^{n-1}(x-x_j)$ es de grado n, P_n tiene grado a lo sumo n (algún coeficiente podría ser nulo). Solo falta verificar que se puede elegir la constante a_n tal que $P_n(x_i) = y_i$, $i = 0,1,...,n$. En efecto, por hipótesis

$$P_n(x_i) = P_{n-1}(x_i) \; + \; 0 \; = \; y_i \;\; \text{si} \;\; i = 0,1,...,n-1.$$

Ahora, evaluando P_n en x_n se prueba que existe una constante a_n adecuada para que $P_n(x_n) = y_n$,

$$P_{n-1}(x_n) + a_n \prod_{j=0}^{n-1}(x_n - x_j) \;=\; y_n \;\Longrightarrow\; a_n = \frac{y_n - P_{n-1}(x_n)}{\prod_{j=0}^{n-1}(x_n - x_j).}$$

Observe que $\prod_{j=0}^{n-1}(x_n - x_j) \neq 0.$

Existen varias formas para el polinomio interpolante. $\mathbb{R}_{n+1}[x]$ es el conjunto de polinomios con coeficientes reales, de grado $\leq n$ y tiene dimensión $n+1$. Si $\Omega = \{B_0, B_2, ..., B_n\}$ es una base para $\mathbb{R}_{n+1}[x]$, entonces el polinomio interpolante P_n se puede escribir en la base Ω como

$$P_n(x) = a_1 B_0(x) + ... + a_n B_n(x)$$

Aunque el polinomio interpolante es único, la forma (antes de simplificar) puede cambiar según la base que se tome. Si tomamos la base $\{1, x, ..., x^n\}$ de $\mathbb{R}_{n+1}[x]$, entonces

$$P_n(x) = a_0 + a_1 x + ... + a_n x^n$$

Como $P_n(x_i) = y_i,\ i = 0, 1, ..., n,$ tenemos el sistema de ecuaciones lineales

$$\begin{pmatrix} 1 & x_0 & x_0^2 & ... & x_0^n \\ 1 & x_1 & x_1^2 & ... & x_1^n \\ \vdots & \vdots & & \cdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{I.3}$$

La matriz asociada V_n del sistema I.3 se llama matriz de Vandermonde. Las columnas de V_n conforman un conjunto de vectores linealmente independiente (ejercicio 5.1). Por lo tanto $\text{Det}(V_n) \neq 0.$ Además (ver 5.2),

$$\text{Det}(V_n) = \prod_{0 \leq i < j \leq n} (x_i - x_j)$$

Por ejemplo,

$$\text{Det}(V_3) = (x_0 - x_1)(x_0 - x_2)(x_1 - x_2)(x_0 - x_3)(x_1 - x_3)(x_2 - x_3)$$

Los coeficientes $a_0, a_1, ..., a_n$ se pueden obtener por regla de Cramer. El cálculo del polinomio interpolante con este método es muy costoso, además de que los coeficientes a_i no siempre se calculan tan exactos como uno quisiera (resolviendo el sistema).

Ejemplo 30

Para los datos $(x_0,y_0),(x_1,y_1)$ la matriz de Vandermonde es

$$\begin{pmatrix} 1 & x_0 \\ 1 & x_1 \end{pmatrix}$$

Usando la regla de Cramer, $P(x) = \dfrac{x_1y_0 - x_0y_1}{x_1 - x_0} + \dfrac{y_1 - y_0}{x_1 - x_0}x$.

I.5 Forma de Lagrange para el polinomio interpolante.

La forma de Lagrange del polinomio interpolante se obtiene usando la base $L_{n,0}(x),...,L_{n,n}(x)$ donde $L_{n,k}(x) = \prod_{\substack{i=0 \\ i\neq k}}^{n} \dfrac{x - x_i}{x_k - x_i}$. En este caso

$$P_n(x) = a_0L_{n,0}(x) + a_1L_{n,1}(x) + ... + a_nL_{n,n}(x)$$

Como $P_n(x_i) = y_i,\ i = 0,1,...,n$ y como $L_{n,j}(x_i) = \delta_{ij}$ (delta de Kronecker), tenemos el sistema de ecuaciones lineales

$$\begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & ... & 0 \\ \vdots & \vdots & & \cdots & \vdots \\ 0 & 0 & & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{I.4}$$

de donde $a_i = y_i,\ i = 0,1,...,n$.

I.6 Forma de Lagrange modificada y forma baricéntrica de Lagrange.

La forma de Lagrange del polinomio interpolante es atractiva para propósitos teóricos. Sin embargo se puede reescribir en una forma que se vuelva eficiente para el cálculo computacional. Una exposición más extensa se encuentra en ([2])

Forma Modificada.

$$P_n(x) = \ell(x)\sum_{j=0}^{n} \frac{\omega_j^{(n)}}{x - x_j}y_j \tag{I.5}$$

Forma Baricéntrica.

$$P_n(x)\begin{cases} = y_i & \text{si} \quad x = x_i, \\ = \dfrac{\displaystyle\sum_{k=0}^{n} \frac{\omega_k^{(n)}}{x-x_k} y_k}{\displaystyle\sum_{k=0}^{n} \frac{\omega_k^{(n)}}{x-x_k}}, & \text{si} \quad x \neq x_i \end{cases}$$

$\ell(x)$ y $\omega_k^{(n)}$ se definen así: Supongamos que tenemos $n+1$ nodos distintos $x_0, x_1, ..., x_n$. $\ell(x) = \prod_{i=0}^{n}(x - x_i)$ y definimos los *pesos baricéntricos* como

$$\omega_0^{(0)} = 1, \ \omega_k^{(n)} = \prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{1}{x_i - x_k}.$$

La forma modificada se obtiene notando que $L_{n,k}(x) = \dfrac{\omega_k^{(n)}}{x-x_k}\prod_{j=0}^{n}(x-x_j) = \dfrac{\omega_k^{(n)}}{x-x_k}\ell(x)$.

La forma baricéntrica se obtiene dividiendo el polinomio interpolante por $\sum_{k=0}^{n} L_{n,k}(x) \equiv 1$ (es decir, el polinomio interpolante de $f(x) \equiv 1$ es $P_n(x) \equiv 1$), en efecto

$$\begin{aligned} P_n(x) &= \sum_{k=0}^{n} y_k L_{n,k}(x) = \frac{\displaystyle\sum_{k=0}^{n} y_k L_{n,k}(x)}{\displaystyle\sum_{k=0}^{n} L_{n,k}(x)} \\ &= \frac{\displaystyle\sum_{k=0}^{n} y_k \frac{\omega_k^{(n)}}{x-x_k}\prod_{j=0}^{n}(x-x_j)}{\displaystyle\sum_{k=0}^{n} \frac{\omega_k^{(n)}}{x-x_k}\prod_{j=0}^{n}(x-x_j)} = \frac{\displaystyle\sum_{k=0}^{n} \frac{\omega_k^{(n)}}{x-x_k} y_k}{\displaystyle\sum_{k=0}^{n} \frac{\omega_k^{(n)}}{x-x_k}} \quad \text{si} \quad x \neq x_k. \end{aligned}$$

Esta forma se llama “forma baricéntrica” de Lagrange pues, aunque no todos los “pesos” son necesariamente positivos, expresa este polinomio como un promedio ponderado de los y_k's.

Nodos de TChebyshev.

En el caso de que podamos escoger los nodos, la elección son los nodos de TChebyshev. En este caso el cálculo de los pesos baricéntricos es muy sencillo ([2], p.249),

$$\omega_k^{(n)} = (-1)^k \operatorname{sen} \frac{(2k+1)\pi}{2n+2}$$

I.7 Forma de Newton para el polinomio interpolante.

La forma de Newton del polinomio interpolante se obtiene usando la base $B_0(x),...,B_n(x)$ donde

$$\begin{aligned}
B_0(x) &= 1\\
B_1(x) &= (x-x_0)\\
B_2(x) &= (x-x_0)(x-x_1)\\
&\vdots\\
B_n(x) &= (x-x_0)(x-x_1)\cdots(x-x_{n-1})
\end{aligned}$$

Tenemos entonces

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

Como $P_n(x_i) = y_i$, $i = 0,1,...,n$ tenemos

$$\begin{aligned}
P_n(x_0) &= a_0 && = y_0\\
P_n(x_1) &= a_0 + a_1(x_1-x_0) && = y_1\\
P_n(x_2) &= a_0 + a_1(x_2-x_0) + a_2(x_2-x_0)(x_2-x_1) && = y_2\\
&\vdots\\
P_n(x_n) &= a_0 + a_1(x_n-x_0) + ... + a_n(x_n-x_0)\cdots(x_n-x_{n-1}) && = y_n
\end{aligned}$$

Como el sistema es triangular, podemos despejar directamente cada a_i. Para obtener una fórmula recursiva un poco más limpia, definimos $Q_0 = a_0$ y

$$Q_j(x) = a_0 + a_1(x-x_0) + ... + a_j(x-x_0)\cdots(x-x_{j-1}), \ \ j = 1,...,n-1$$

De esta manera,

$$\begin{aligned}
a_0 &= y_0\\
a_1 &= \frac{y_1 - Q_0(x_1)}{x_1 - x_0}\\
a_2 &= \frac{y_2 - Q_1(x_2)}{(x_2-x_0)(x_2-x_1)}\\
&\vdots\\
a_n &= \frac{y_n - Q_{n-1}(x_n)}{(x_n-x_0)\cdots(x_n-x_{n-1})}
\end{aligned}$$

Obtenemos una fórmula recursiva: $a_0 = y_0$ y $a_k = \dfrac{y_k - Q_{k-1}(x_k)}{(x_k - x_0)\cdots(x_k - x_{k-1})}$, $k = 1, ..., n$

Diferencias divididas. Si $y_k = f(x_k)$, la fórmula anterior nos muestra que cada a_k depende de $x_0, x_1, ..., x_k$. Desde muchos años atrás se usa la notación $a_n = f[x_0, x_1, ... x_n]$ para significar esta dependencia.

Al símbolo $f[x_0, x_1, ... x_n]$ se le llama *diferencia divida* de f.

Si consideramos $f[x_0, x_1, ... x_n]$ como una función de $n+1$ variables, entonces es una función *simétrica*, es decir, permutar las variables de cualquier manera no afecta el valor de la función. Esto es así porque el polinomio que interpola los puntos $\{(x_i, y_i)\}_{i=0,...,n}$ es único, por lo tanto sin importar el orden en que vengan los puntos, el coeficiente principal siempre es $a_n = f[x_0, x_1, ... x_n]$.

El nombre *diferencia divida* viene de la agradable propiedad

Teorema 31

La diferencia dividida $f[x_0, x_1, ... x_n]$ satisface la ecuación

$$f[x_0, x_1, ... x_n] = \frac{f[x_1, x_2, ... x_n] - f[x_0, x_1, ... x_{n-1}]}{x_n - x_0} \tag{I.6}$$

Prueba: Sea $P_k(x)$ el polinomio que interpola f en $x_0, x_1, ..., x_k$. Aquí solo necesitamos $P_n(x)$ y $P_{n-1}(x)$. Sea $R(x)$ el polinomio que interpola f en $x_1, x_2, ..., x_n$. Entonces (ejercicio I.2)

$$P_n(x) = R(x) + \frac{x - x_n}{x_n - x_0}[R(x) - P_{n-1}(x)] \tag{I.7}$$

Como en la ecuación (I.7) el polinomio de la izquierda y el de la derecha son idénticos, entonces su coeficiente principal debe ser el mismo, es decir,

$$\begin{aligned} f[x_0, x_1, ... x_n]x^n + ... &= f[x_1, x_2, ... x_n]x^{n-1} + ... + \frac{x\,R(x) - x\,P_{n-1}(x) + ...}{x_n - x_0} \\ &= f[x_1, x_2, ... x_n]x^{n-1} + \\ &\quad ... + \frac{f[x_1, x_2, ... x_n]x^n + ... - f[x_0, x_2, ... x_{n-1}]x^n + ...}{x_n - x_0} \\ &= \frac{(f[x_1, x_2, ... x_n] - f[x_0, x_2, ... x_{n-1}])x^n}{x_n - x_0} + ... \end{aligned}$$

de donde se obtiene (I.6).

Interpolando sobre $\{(x_i, y_i)\}_{i=k-j,...,k}$ tenemos

$$f[x_{k-j}, x_{k-j+1}, \cdots, x_k] = \frac{f[x_{k-j+1}, ..., x_k] - f[x_{k-j}, x_1, ..., x_{k-1}]}{x_k - x_{k-j}}$$

Resumiendo: En notación de diferencias divididas,

$$\begin{aligned}
a_0 &= y_0, \\
a_1 &= f[x_0, x_1], \\
a_2 &= f[x_0, x_1, x_2] \\
\vdots & \quad \vdots \\
a_n &= f[x_0, x_1, ..., x_n].
\end{aligned}$$

donde, $f[x_k] = y_k$ y, en general,

$$f[x_{k-j}, x_{k-j+1}, \cdots, x_k] = \frac{f[x_{k-j+1}, ..., x_k] - f[x_{k-j}, x_1, ..., x_{k-1}]}{x_k - x_{k-j}}$$

En particular,

$$\begin{aligned}
f[x_i, x_j] &= \frac{y_i - y_j}{x_i - x_j} \\
f[x_{k-2}, x_{k-1}, x_k] &= \frac{f[x_{k-1}, x_k] - f[x_{k-2}, x_{k-1}]}{x_k - x_{k-2}} \\
f[x_{k-3}, x_{k-2}, x_{k-1}, x_k] &= \frac{f[x_{k-2}, x_{k-1}, x_k] - f[x_{k-3}, x_{k-2}, x_{k-1}]}{x_k - x_{k-3}} \\
&\vdots \\
f[x_0, x_1, ..., x_k] &= \frac{f[x_1, x_2, ..., x_k] - f[x_0, x_1, ..., x_{k-1}]}{x_k - x_0}
\end{aligned}$$

Este esquema recursivo se puede arreglar en forma matricial como sigue,

$$\begin{array}{llll}
y_0 & = \boldsymbol{a_0} & & \\
y_1 & f[x_0, x_1] = \boldsymbol{a_1} & & \\
y_2 & f[x_1, x_2] & f[x_0, x_1, x_2] = \boldsymbol{a_2} & \\
 & f[x_2, x_3] & f[x_1, x_2, x_3] & \\
\vdots & \vdots & \vdots & \ddots \\
y_n & f[x_{n-1}, x_n] & f[x_{n-2}, x_{n-1}, x_n] & f[x_0, x_1, ..., x_n] = \boldsymbol{a_n}
\end{array}$$

I.8 Estimación del error.

Sea $P_n(x)$ el polinomio que interpola f en los puntos $x_0, x_1, ..., x_n$. La fórmula de error debe ser exacta para $x = x_i$, $i = 0, 1, ..., n$ y también debe ser exacta en el caso de que f sea un polinomio de grado inferior o igual a n. Esto sugiere que en la fórmula de error deben aparecer los factores $(x - x_0)(x - x_1)\cdots(x - x_n)$ y f^{n+1}. Probando con $f(x) = x^{n+1}$ se observa que debe aparecer el factor $1/(n+1)!$.

Teorema 32

Sea $f \in C^{n+1}[a,b]$. Sea $P_n(x)$ el polinomio de grado $\leq n$ que interpola f en los $n+1$ puntos (distintos) $x_0, x_1, ..., x_n$ en el intervalo $[a,b]$. Para cada valor fijo $x \in [a,b]$ existe $\xi(x) \in]a,b[$ tal que

$$f(x) - P_n(x) = \frac{f^{n+1}(\xi(x))}{(n+1)!}(x - x_0)(x - x_1)\cdots(x - x_n)$$

Prueba: Este es un razonamiento muy elegante, debido a Cauchy[6]. Si $x = x_i$, $i = 0, 1, ...n$, la fórmula de error es correcta. Consideremos un valor fijo $x \in [a,b]$, diferente de cada uno de los nodos x_i, $i = 0, 1, ..., n$. Definamos una función g, en la variable t, de la siguiente manera,

$$g(t) = f(t) - P_n(t) \; - \; \frac{f(x) \; - \; P_n(x)}{\displaystyle\prod_{i=0}^{n}(x - x_i)} \prod_{i=0}^{n}(t - x_i)$$

$g \in C^{n+1}[a,b]$ y $g(x) = 0$ y $g(x_i) = 0$, $i = 0, 1, ...n$.

Por tanto g tiene $n+2$ ceros distintos en $[a,b]$. El teorema de Rolle dice que si h es continua en $[a,b]$ y derivable en $]a,b[$, y si $h(a) = h(b) = 0$, entonces $h(\varepsilon) = 0$ para algún $\varepsilon \in]a,b[$. Aplicando repetidamente el teorema de Rolle a la función g en los intervalos $[x_0, x_1]$, $[x_1, x_2]$, ..., concluimos que g' tiene *al menos* $n+1$ ceros distintos en $]a,b[$. De manera similar concluimos que

g'' tiene *al menos* n ceros distintos en $]a,b[$

g''' tiene *al menos* $n-1$ ceros distintos en $]a,b[$

$\vdots$

g^{n+1} tiene *al menos* 1 cero distinto en $]a,b[$ (g^{n+1} es continua en [a,b]).

Ahora bien, sea $\xi(x)$ un cero de g^{n+1} en $]a,b[$. Como $\left.\dfrac{d^{n+1}g}{dt^{n+1}}\right|_{t=\xi(x)} = 0$, tenemos (ejercicio 5.4)

$$0 = f^{n+1}(\xi(x)) \; - \; \frac{f(x) \; - \; P_n(x)}{\displaystyle\prod_{i=0}^{n}(x - x_i)}(n+1)!$$

6

Agustín Louis Cauchy (1789-1857), padre del análisis moderno. Estableció las bases del análisis matemático basándolo en un concepto riguroso de límite. Fue el creador del análisis complejo. También trabajo en ecuaciones diferenciales, geometría, álgebra, teoría de números, probabilidad y física matemática de donde, $f(x) - P_n(x) = \dfrac{f^{n+1}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n).$

I.9 Polinomios de TChebyshev y convergencia.

En la fórmula de error,

$$f(x) - P_n(x) = \frac{f^{n+1}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n)$$

el factor $(x-x_0)(x-x_1)\cdots(x-x_n)$ puede ser "optimizado" escogiendo de manera adecuada los nodos $x_0, x_1, ..., x_n$. El proceso de optimización lleva, de manera natural, a un sistema de polinomios llamados polinomios de TChebyshev

El resultado principal es: Para cualquier polinomio mónico $P_n(x)$ de grado n,

$$||P_n(x)||_\infty \geq ||\overline{T}_n(x)||_\infty = \frac{1}{2^{n-1}}, \;\; n \geq 1$$

donde $\overline{T}_n(x)$ es el polinomio mónico de TChebyshev de grado n.

$\overline{T}_{n+1}(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$ con $x_i = \cos\left(\dfrac{2i+1}{2n+2}\pi\right), \;\; i = 0,1,....n$

Luego, si P_n interpola f, usando nodos de TChebyshev, tenemos

$$||f(x) - P_n(x)||_\infty \leq \frac{||f^{n+1}||_\infty}{(n+1)!}\,\frac{1}{2^n}$$

Por lo tanto, la convergencia se puede asegurar si la familia de derivadas $f^{(n)}$ es uniformemente acotada ($\exists M$ tal que $\forall n \in \mathbb{N}$, $||f^{(n+1)}(x)||_\infty \leq M$). En realidad se necesita menos para asegurar la convergencia!.

Los Polinomios de TChebyshev (de primera especie) se definen, de manera recursiva, de la siguiente manera:

$$\left\{\begin{array}{rcl} T_0(x) &=& 1, \;\; T_1(x) \;=\; x \\ \\ T_{n+1}(x) &=& 2xT_n(x) - T_{n-1}(x), \;\; n \geq 1. \end{array}\right. \tag{I.8}$$

Por ejemplo,

$$\left\{\begin{array}{rcl} T_2(x) &=& 2x^2 - 1 \\ T_3(x) &=& 4x^3 - 3x \\ T_4(x) &=& 8x^4 - 8x^2 + 1 \end{array}\right.$$

Teorema 33

Si $x \in [-1,1]$ entonces $T_n(x) = \cos\left(n\cos^{-1}x\right), \;\; n \geq 0$

Prueba: La idea de la prueba esta basada en el hecho de que $\cos n\theta$ es un polinomio en $\cos\theta$, es decir, $\cos n\theta = \sum_{k=0}^{n} a_k \cos^k\theta$. Luego $f_n(\cos\theta) = \cos n\theta$ satisface (I.8).

Recordemos que $\cos(a+b) = \cos a \cos b - \operatorname{sen} a \operatorname{sen} b$. Luego,

$$\cos(n+1)\theta = \cos\theta\cos n\theta - \operatorname{sen}\theta\operatorname{sen} n\theta.$$
$$\cos(n-1)\theta = \cos\theta\cos n\theta + \operatorname{sen}\theta\operatorname{sen} n\theta.$$

sumando miembro a miembro y reacomodando,

$$\cos(n+1)\theta = 2\cos\theta\cos n\theta - \cos(n-1)\theta \tag{I.9}$$

Ahora, sea $\theta = \cos^{-1} x$ y $x = \cos\theta$ y $f_n(x) = \cos(n\cos^{-1} x)$. Usando la ecuación (I.9) obtenemos que,

$$\begin{cases} f_0(x) = 1, \ f_1(x) = x \\ f_{n+1}(x) = 2xf_n(x) - f_{n-1}(x), \ n \geq 1. \end{cases}$$

Luego, $f_n = T_n$ para todo n.

- $\overline{T}_n(x) = 2^{n-1}T_n(x)$ **es mónico:** Si $n \geq 1$, de (I.8) se deduce que el coeficiente principal de $T_n(x)$ es 2^{n-1} (ejercicio 5.11). Luego $2^{1-n}T_n(x)$ es mónico.

$$\overline{T}_n(x) = 2^{1-n}T_n(x) \quad \text{y} \quad \overline{T}_0(x) = T_0(x)$$

- **Los ceros de** $\overline{T}_{n+1}(x)$ **:** Si $x \in [-1,1]$ podemos poner $x = \cos\theta$. Luego, $T_{n+1}(\cos\theta) = \cos(n+1)\theta$, así que los ceros $x_k^{(n+1)}$ de T_{n+1} en $[-1,1]$ se pueden obtener resolviendo $\cos(n+1)\theta = 0$.

  $\cos(n+1)\theta = 0 \implies (n+1)\theta = (2k-1)\pi/2, \ \ k \in \mathbb{Z}$. Como $x = \arccos\theta, \ 0 \leq \theta \leq \pi$, entonces

$$x_k^{(n+1)} = \cos\theta_k^{(n+1)}, \ \ \theta_k^{(n+1)} = \frac{(2k-1)\pi}{2n+2}, \ \ k = 1,2,...,n+1$$

  Por lo tanto, todos los ceros de T_{n+1} son reales, distintos y están contenidos en el intervalo $]-1,1[$.

  Ahora podemos escribir

$$\overline{T}_{n+1}(x) = \prod_{k=0}^{n}\left(x - x_k^{(n+1)}\right)$$

- **Extremos:** θ crece de 0 hasta π, por tanto $x = \cos\theta$ decrece de 1 hasta -1. Luego T_n oscila entre 1 y -1. En el ejercicio (5.12) se pide mostrar que

$$T_n \text{ alcanza sus valores extremos en } y_k^{(n)} = \cos\left(\frac{k\pi}{n}\right),\ k = 0,1,...n. \tag{I.10}$$

$$T_n\left(\cos\phi_k^{(n)}\right) = (-1)^k \quad \text{si} \quad \phi_k^{(n)} = \frac{k\pi}{n},\ k = 0,2,...n. \tag{I.11}$$

$$(-1)^k\overline{T}_n(y_k^{(n)}) = \frac{1}{2^{n-1}} \quad \text{y} \quad ||\overline{T}_n(x)||_\infty = \frac{1}{2^{n-1}} \tag{I.12}$$

Los polinomios de TChebyshev son útiles e importantes debido al siguiente teorema

Teorema 34

Para cualquier polinomio mónico P_n de grado n

$$||P_n(x)||_\infty \geq ||\overline{T}_n(x)||_\infty = \frac{1}{2^{n-1}}$$

Prueba: La prueba es por contradicción. Supongamos que hay un polinomio P_n mónico de grado n para el que

$$||P_n(x)||_\infty < \frac{1}{2^{n-1}} \tag{I.13}$$

Sea $x_k = \cos(k\pi/n)$. El polinomio $\overline{T}_n$ es mónico de grado n. Entonces de (I.12),

$$(-1)^k P_n(x_k) \leq |P_n(x_k)| < \frac{1}{2^{n-1}} = (-1)^k\overline{T}_n(x_k),\ k = 0,1,...,n.$$

Luego $0 < (-1)^k(\overline{T}_n(x_k) - P_n(x_k)),\ k = 0,1,...,n.$

Esto dice que el polinomio $\overline{T}_n - P_n$ cambia de signo al menos n veces en $[-1,1]$ y hay un cero por cada cambio de signo!

Pero como $\overline{T}_n$ y P_n son mónicos de grado n, $\overline{T}_n - P_n$ tiene grado a lo sumo $n-1$ y entonces no puede tener más de n ceros.

Luego, $\overline{T}_n = P_n$ pero esto contradice (I.13).

El teorema nos dice que si tomamos los $n+1$ ceros de $T_{n+1}(x)$ como los nodos x_i, entonces

$$||f(x) - P_n(x)||_\infty \quad \leq \quad \frac{||f^{n+1}||_\infty}{(n+1)!}||||\overline{T}_{n+1}||_\infty \; = \; \frac{||f^{n+1}||_\infty}{(n+1)!}||\frac{1}{2^n}$$

Se puede probar (ver [8]) que si usamos los nodos de TChebyshev, $P_n(x) \longrightarrow f(x)$ si $n \longrightarrow \infty$ uniformemente en $[-1,1]$ con solo que $f \in C^1[a,b]$.

EJERCICIOS

5.1 Use el teorema fundamental del álgebra para probar que las columnas de V_n conforman un conjunto de vectores linealmente independiente.

5.2 Pruebe, usando inducción y operaciones elementales de matrices, que

$$\text{Det}(V_n) = \prod_{0 \leq i < j \leq n} (x_i - x_j)$$

5.3 Probar (I.7). Primero pruebe el caso $i = 1, ..., k-1$ y luego los casos $i = 0$ e $i = k$.

5.4 Si $h(t) = \prod_{i=0}^{n}(t - x_i)$, muestre que $h^{(n+1)(t)} = (n+1)!$

5.5 Sea P un polinomio de grado a lo sumo n que interpola una función f en los $n+1$ nodos distintos $x_0, x_1, ...x_n$. Sea t un punto diferente de los nodos anteriores. Sea Q el polinomio de grado a lo sumo $n+1$ que interpola f en $x_0, x_1, ...x_n, t$. Mostrar que

a) $Q(x) = P(x) + f[x_0, x_1, ..., x_n, t]\prod_{j=0}^{n}(x - x_j)$

b) $f(t) - P(t) = f[x_0, x_1, ..., x_n, t]\prod_{j=0}^{n}(t - x_j)$

5.6 Sea $f \in C^n[a,b]$ y sean $x_0, x_1, ...x_n$ puntos distintos en $[a,b]$. Muestre, usando el ejercicio (5.5) y la fórmula de error, que existe un ξ en $]a,b[$ tal que

$$f[x_0, x_1, ..., x_n] = \frac{1}{n!}f^{(n)}(\xi)$$

Ayuda: Compare $f(t) - P(t)$ tal y como aparece en la fórmula del error y como aparece en el ejercicio (5.5).

5.7 Pruebe que si f es un polinomio de grado k, entonces para $n > k$ se tiene

$$f[x_0, x_1, ..., x_n] = 0$$

5.8 Muestre que si $f(x) = \text{sen}(x)$ es aproximada por un polinomio P de grado nueve que interpola f en diez puntos de $[0,1]$, entonces

$$|\operatorname{sen}(x) - P(x)| \leq \frac{1}{10!},\ x \in [0,1].$$

5.9 Sea $h = x_1 - x_0 > 0$, muestre que si P_1 interpola f en $\{x_0, x_1\}$, entonces

$$||f(x) - P_1(x)||_\infty \leq \frac{||f''||_\infty}{8}\, h^2,\ x \in [x_0, x_1]$$

5.10 Muestre que si P_2 interpola f en $\{x_0, x_0+h, x_0+2h\}, h > 0$, entonces

$$||f(x) - P_2(x)||_\infty \leq \frac{||f'''||_\infty}{9\sqrt{3}}\, h^3,\ x \in [x_0, x_0+2h]$$

Ayuda: Observe que $Q(x) = (x-x_0)(x-(x_0+h))(x-(x_0+2h))$ es un polinomio de grado tres. Sus puntos críticos son ceros de la cuadrática Q. Resulta sencillo determinar los extremos absolutos de Q.

5.11 Muestre que el coeficiente principal de $T_n(x)$ es 2^{n-1}.

5.12 Si $x \in [-1,1]$ y $x = \cos\theta$, $T_n(\cos\theta) = \cos n\theta$. Use este hecho para verificar que

a) T_n alcanza sus valores extremos en $y_k^{(n)} = \cos\left(\frac{k\pi}{n}\right)$, $k = 0,1,...n$.

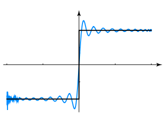
b) $T_n\left(\cos\phi_k^{(n)}\right) = (-1)^k$ si $\phi_k^{(n)} = \frac{k\pi}{n}$, $k = 0,2,...n$.

c) $(-1)^k\overline{T}_n(y_k^{(n)}) = \frac{1}{2^{n-1}}$ y $||\overline{T}_n(x)||_\infty = \frac{1}{2^{n-1}}$

5.13 Considere la función de Runge,

$$f(x) = \frac{1}{1+25x^2}$$

Haga la representación gráfica de $(x-x_0)(x-x_1)...(x-x_n)$ para los casos $n = 5,10,20$, en el caso de que los nodos sean igualmente espaciados y en el caso que sean nodos de TChebyshev. En cada caso usar un mismo sistema de coordenadas.



Solución de los Ejercicios

Bibliografía

[1] W. Gautschi. *Numerical Analysis. An Introduction.* Birkhäuser, 1997.

[2] P. Henrici.*Essentials of Numerical Analysis.* Wiley, New York, 1982.

[3] J. Stoer,. *Introduction to Numerical Analysis.* 3rd ed. Springer, 2002.

[4] Dahlquist, G. Björk, A. *Numerical Mathematics in Scientific Computation.*

[5] D. Kahaner, K. Moler, S. Nash. *Numerical Methods and Software.* Prentice Hall.1989.

[6] D. Kincaid, W. Cheney. *Numerical Analysis. Mathematics of Scientific Computing.* Brooks-Cole Publishing Co.1991.

[7] R. Burden, J. Faires. *Análisis Numérico.* 6ta ed. Thomson. 1998.

[8] E. Cheney, *Introduction to Approximation Theory.* Internat. Ser. Pure ans Applied Mathematics. McGraw-Hill. 1966.

[9] J.P. Berrut, L. N. Trefethen. "Barycentric Lagrange Interpolation" Siam Rewiew. Vol. 46, No. 3. 2004.

[10] J. Higham, "The numerical stability of barycentric Lagrange interpolation". IMA Journal of Numerical Analysis 24. 2004.