Capítulo 2

Interpolación y aproximación de funciones

Es una situación frecuente en el ejercicio de la Física el encontrarse ante tablas de valores de algún suceso físico obtenidos experimentalmente y necesitar conocer los valores en puntos distintos de los medidos, tanto para responder a preguntas concretas necesarias en la práctica como para tratar de deducir leyes empíricas o constantes fundamentales.
La interpolación resuelve precisamente ese problema: dada una tabla de datos, que podemos entender como una colección de puntos en el plano, qué función describe el proceso de una forma sencilla? Por ejemplo, ¿existe un polinomio cuya gráfica pase por todos esos puntos? ¿Cuál es "el mejor" de ellos, si es que hay un concepto de "mejor"? ¿Cómo calcularlo? Si conseguimos un polinomio, que es una función muy sencilla, que pase por un conjunto de datos, es muy probable que podamos aproximar bien los valores de la función en otros puntos fuera de nuestra tabla de datos: ¡simplemente por el valor que toma el polinomio en esos puntos! Ese es el objetivo de este capítulo: entender este problema y su solución a fondo. Ponemos un par de ejemplos sencillos de la utilidad de este método.

- Pongamos un ejemplo con los datos expuestos en la figura 2.1. Nos preguntamos: cuál será la presión

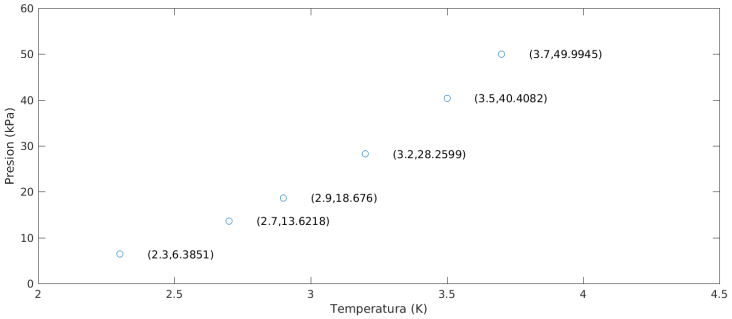

Figura 2.1: Temperatura en función de la presión. Datos obtenidos de Introductory Computational Physics (Andi Klein, Alexander Godunov)

del vapor a 3 K? Por la ley de Gay–lussac, la presión debería ser aproximadamente proporcional a la temperatura pero las mediciones no son exactamente así, lo que nos dice que la ley de Gay–Lussac es falsa para temperaturas muy bajas. Sin una "ley" a la que agarrarnos, necesitamos utilizar algún método numérico para responder a la pregunta que nos interesa. Veremos que la interpolación polinomial resuelve perfectamente este problema.

- Investigamos el *periodo de semidesintegración* o *semivida* o *half–life* de una partícula radioactiva, esto es, el tiempo que tarda en llegar a la mitad de su emisión radioactiva. Esto viene a decir, que buscamos el tiempo que tardarán la mitad de los átomos en pasar a ser no–radioactivos, una propiedad que es estadística por su naturaleza. Suponiendo que tenemos decaimiento exponencial y denotando por $M(t)$ el peso de los átomos radioactivos tenemos:

  $$M(t) = M_0 e^{-t/\tau},$$

  con M_0 la masa inicial, la semivida de este cuerpo es precisamente $\tau \log(2)$. Experimentalmente, si medimos la cantidad de radiación en tiempos $t_0, t_1, \ldots, t_k$, obtendremos medidas $M_0, \ldots, M_k$ que no obedecerán exáctamente a la fórmula de arriba. Pongamos que tenemos una fuente de radioactividad proveniente de una cantidad de Americio, isótopo 241, que emite ondas α, cuya semivida es de unos 430 años. Obviamente nuestro experimento para calcular esa semivida no puede durar tanto: pongamos que medimos la radiación una vez a la semana (los lunes) durante 5 meses. Luego nos preguntaremos cuál será la radiación el miércoles de la sexta semana, o dentro de 8 meses. Necesitamos una función sencilla $M_*(t)$ que cumpla $M_*(t_k) = M_k$ y que "sea razonable" para el problema. Además deberemos haber elegido correctamente los puntos t_j. Nótese que en ese caso un polinomio no es la mejor manera de aproximarnos a la verdad, necesitamos otras funciones.

Además de estos ejemplos de aplicación, la interpolación (muy en particular la polinomial) es una piedra básica que nos permitirá enfrentarnos a muchos otros problemas en los capítulos posteriores. Su utilidad ha de mirarse pues tanto desde el punto de vista de su uso directo en problemas aplicados como desde el punto de vista de la construcción de otros métodos numéricos cuyo conocimiento resulta imprescindible para un graduado en Física.

2.1. Interpolación polinomial

Primero tratamos el problema de encontrar un polinomio que alcance una serie de valores prescritos.

2.1.1. Un comentario sobre el Teorema de aproximación de Weierstrass y los polinomios de Taylor

Comenzamos por señalar un importante resultado cuya demostración no es sencilla:

Teorema 2.1.1 (Teorema de aproximación de Weierstrass) *Dada una función continua definida en $[a, b]$, y dado $\epsilon > 0$, existe un polinomio $p(x)$ con la propiedad de que $|f(x) - p(x)| < \epsilon$ para todo $x \in [a, b]$.*

Ilustramos este teorema en la Figura 2.2 El Teorema 2.1.1 nos dice, *grosso modo*, que los polinomios son funciones razonables para aproximar funciones continuas. Sin embargo, no nos dice qué polinomio es el que aproxima a la función, solo nos asegura la existencia de uno (para cada $\epsilon > 0$), pero bien podría ser que para $\epsilon = 0{,}1$ el polinomio en cuestión tuviera grado 10,000, eliminando toda posibilidad de uso práctico.
Como hemos visto en el Teorema 1.3.1, toda función suficientemente derivable puede ser aproximada por su polinomio de Taylor, luego podríamos pensar que esta es la respuesta a nuestros problemas. Sin embargo, observamos dos cosas:

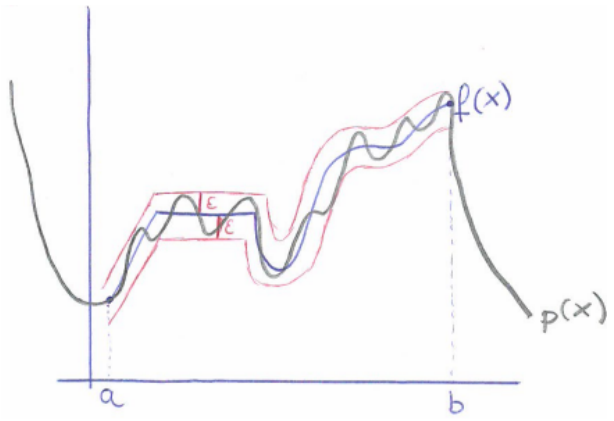

Figura 2.2: Ilustración del Teorema de Weierstrass: tomamos la función $f(x)$ y un $\epsilon > 0$. Existe un polinomio (en principio, de grado desconocido que podría ser altísimo) cuya gráfica está contenida en el tubo de radio ϵ mostrado en el dibujo en torno a la gráfica de la función.

I) Calcular el polinomio de Taylor exige conocer la función y sus derivadas en un único punto, lo que se encuentra muchas veces fuera de nuestro alcance (pensemos en el ejemplo de la presión de vapor del Helio 4 descrito arriba).

II) El polinomio de Taylor aproxima extremadamente bien la función en un intervalo, en ocasiones muy pequeño, del punto en torno al cual lo desarrollamos, pero nosotros necesitamos frecuentemente información no demasiado cerca de ese punto, más concretamente deberíamos aspirar a describir bien el valor de la función en puntos "no demasiado lejos de" los lugares en los que conocemos el valor de la función.

Es por ello que necesitamos otra herramienta más allá de los polinomios de Taylor.

2.1.2. Polinomios interpolantes de Lagrange

En esta sección tratamos el siguiente problema: dada una tabla de valores como la Tabla 2.1, ¿cómo encontramos, si es que existe, un polinomio de grado lo menor posible que alcance todos esos valores, esto es un polinomio $p(x)$ que cumpla $p(x_k) = y_k$ para $k = 0, \ldots, n$? El problema queda (aparentemente) totalmente resuelto con el

x	x_0	x_1	$\cdots$	x_n
y	y_0	y_1	$\cdots$	y_n

Cuadro 2.1: Tabla de valores para ajustar mediante un polinomio.

siguiente resultado.

Teorema 2.1.2 *Sean $x_0, \ldots, x_n$ todos ellos distintos, y sean $y_0, \ldots, y_n$ números reales. Entonces, existe un único polinomio $p(x)$ de grado menor o igual que n que cumple $p(x_k) = y_k$ para $k = 0, \ldots, n$. Es más, podemos*

escribir

$$p(x) = y_0 L_0(x) + \cdots + y_n L_n(x) = \sum_{k=0}^{n} y_k L_k(x),$$

donde

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

El polinomio $p(x)$ se llama polinomio interpolante de la tabla de datos, y escrito de la manera que acabamos de mostrar se dice que es el polinomio interpolante de Lagrange.

DEMOSTRACIÓN. Es un ejercicio verificar que $p(x)$ cumple la tabla de datos. Basta ver para ello que los $L_k(x)$ son polinomios de grado n tales que $L_k(x_i) = \delta_{ik}$ (delta de Kronecker, esto es $L_k(x_i) = 0$ si $i \neq k$, $L_k(x_k) = 1$). Para ver la unicidad: supongamos que $p_1(x)$ y $p_2(x)$ tienen grado menor o igual que n y cumplen ambos la tabla de datos. Entonces el polinomio $h(x) = p_1(x) - p_2(x)$ es de grado n y tiene $n+1$ ceros $x_0, \ldots, x_n$, luego ha de ser $h \equiv 0$, esto es, $p_1 \equiv p_2$ y por lo tanto el polinomio interpolante es único. □

Nótese que en el Teorema 2.1.2, aunque cada L_k tiene grado exáctamente n, $p(x)$ puede tener grado estrictamente menor.

Ejemplo 2.1.3 *En el ejemplo mostrado en la Figura 2.1 tenemos la tabla de valores*

T [K]	2,3	2,7	2,9	3,2	3,5	3,7
P [kPa]	6,38512	13,6218	18,676	28,2599	40,4082	49,9945

El polinomio interpolador tiene entonces la fórmula:

$$\begin{aligned} p(x) = 6{,}38512\frac{(x-2{,}7)(x-2{,}9)(x-3{,}2)(x-3{,}5)(x-3{,}7)}{-0{,}36288} + \\ 13{,}6218\frac{(x-2{,}3)(x-2{,}9)(x-3{,}2)(x-3{,}5)(x-3{,}7)}{0{,}0312} + \\ 18{,}676\frac{(x-2{,}3)(x-2{,}7)(x-3{,}2)(x-3{,}5)(x-3{,}7)}{-0{,}0864} + \\ 28{,}2599\frac{(x-2{,}7)(x-2{,}9)(x-3{,}2)(x-3{,}5)(x-3{,}7)}{0{,}135} + \\ 40{,}4082\frac{(x-2{,}3)(x-2{,}7)(x-2{,}9)(x-3{,}2)(x-3{,}7)}{-0{,}24} + \\ 49{,}9945\frac{(x-2{,}3)(x-2{,}7)(x-2{,}9)(x-3{,}2)(x-3{,}5)}{0{,}56} \end{aligned}$$

Si dibujamos ese polinomio junto con los datos de la tabla obtenemos la Figura 2.3.

Otra alternativa para demostrar que existe un único polinomio resulta muy ilustrativa: suponiendo que buscamos los coeficientes del polinomio $p(x) = a_0 + a_1 x + \cdots a_n x^n$ y que ha de cumplir la tabla de arriba, podemos escribir las ecuaciones como

$$\begin{aligned} a_0 + a_1 x_0 + \cdots a_n x_0^n &= y_0 \\ &\cdots \\ a_0 + a_1 x_n + \cdots a_n x_n^n &= y_n \end{aligned}$$

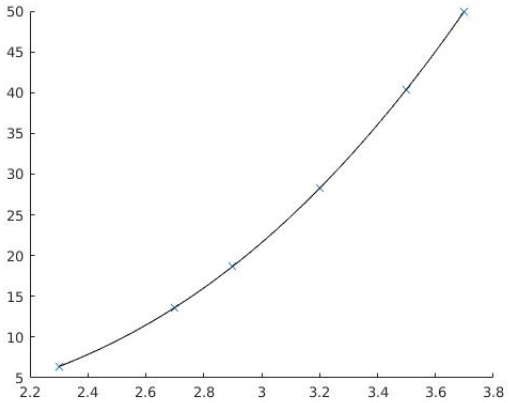

Figura 2.3: Datos del problema del Helio 4 interpolados usando interpolación de Lagrange. Se ve que la curva se ajusta a los datos y proporciona una manera razonable de aproximar los datos desconocidos.

esto es un sistema de $n+1$ ecuaciones lineales con $n+1$ incógnitas. En forma matricial,

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}.$$

La matriz de este sistema, llamada matriz de Vandermonde, tiene determiante igual a $\prod_{i<j}(x_j - x_i)$, que es distinto de cero por hipótesis, luego existe una única solución. Esta manera de enfocar el problema resultará de gran utilidad más adelante. El cálculo de la fórmula que damos para el determinante se puede hacer por inducción en n. Además, este enfoque nos da otra manera de calcular el polinomio: resolver el sistema de ecuaciones para encontrar los coeficientes del mismo.

Ejemplo 2.1.4 *De nuevo usamos ejemplo mostrado en la Figura 2.1, ahora con el método matricial. El sistema de ecuaciones que tenemos que resolver es:*

$$\begin{pmatrix} 1{,}0000 & 2{,}3000 & 5{,}2900 & 12{,}1670 & 27{,}9841 & 64{,}3634 \\ 1{,}0000 & 2{,}7000 & 7{,}2900 & 19{,}6830 & 53{,}1441 & 143{,}4891 \\ 1{,}0000 & 2{,}9000 & 8{,}4100 & 24{,}3890 & 70{,}7281 & 205{,}1115 \\ 1{,}0000 & 3{,}2000 & 10{,}2400 & 32{,}7680 & 104{,}8576 & 335{,}5443 \\ 1{,}0000 & 3{,}5000 & 12{,}2500 & 42{,}8750 & 150{,}0625 & 525{,}2188 \\ 1{,}0000 & 3{,}7000 & 13{,}6900 & 50{,}6530 & 187{,}4161 & 693{,}4396 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} 6{,}38512 \\ 13{,}6218 \\ 18{,}676 \\ 28{,}2599 \\ 40{,}4082 \\ 49{,}9945 \end{pmatrix}$$

Resolviendo este sistema con Matlab obtenemos

$$p(x) = 0{,}00986552 - 0{,}493265x + 6{,}2525x^2 - 18{,}6736784603x^3 + 23{,}0504462x^4 - 10{,}752716666x^5$$

que es obviamente solo una forma distinta de escribir el mismo polinomio que hemos calculado previamente con la fórmula de Lagrange.

Si bien en esta sección hemos visto dos métodos para calcular el polinomio interpolante, en los cálculos prácticos ninguna de las dos maneras resulta eficaz, por lo que se utiliza el algoritmo de diferencias divididas de Newton que se explicará en la Sección 2.1.4

2.1.3. Polinomios interpolantes y funciones derivables

Supongamos ahora que una tabla de la forma de la Tabla 2.1 corresponde a una función (conocida o no), por ejemplo que corresponde a una medición como la de la Figura 2.1, de modo que sabemos que *existe* en algún sentido físico la función, aunque no podamos representarla o no la conozcamos o no la podamos modelizar, pero al menos la podemos medir (la medimos, luego existe) con la suficiente precisión para hacer una tabla. ¿Qué podemos decir de la aproximación de la función desconocida con el polinomio de Lagrange de la tabla? Tenemos el siguiente importantísimo resultado (al que precede una definición).

Definición 2.1.5 *Dada una función definida en un intervalo $[a,b]$, y dados $x_0,\dots,x_n$ puntos distintos en el intervalo, definimos el polinomio interpolante de f con nodos $x_0,\dots,x_n$ al polinomio $p(x)$ (que existe y es único por el Teorema 2.1.2) de grado a lo más n que cumple $p(x_k)=f(x_k)$ para $k=0,\dots,n$.*

Teorema 2.1.6 *Sea $f\in C^{n+1}[a,b]$ y sean $x_0,\dots,x_n$ puntos distintos en el intervalo $[a,b]$. Sea $p(x)$ el polinomio interpolante de f con nodos $x_0,\dots,x_n$. Entonces, para cada $x\in[a,b]$ existe un número $\zeta_x\in(a,b)$ tal que*

$$f(x)=p(x)+\frac{f^{(n+1)}(\zeta_x)}{(n+1)!}(x-x_0)\cdots(x-x_n).$$

Demostración. Primero, si $x=x_k$ para algún k entonces la igualdad se cumple por definición. En otro caso, tomemos la función

$$g(t)=f(t)-p(t)-(f(x)-p(x))\prod_{j=0}^{n}\frac{t-x_j}{x-x_j},$$

que es de tipo $C^{n+1}[a,b]$ si completamos la definición con $g(x_k)=0$ para $0\le k\le n$. Se tiene $g(x_k)=0$ para $k=0,\dots,n$, y adicionalmente se tiene $g(x)=0$. Esto es, g tiene al menos $n+2$ ceros, lo que por el Corolario 1.1.11 implica que existe $\zeta_x\in(a,b)$ tal que $g^{(n+1)}(\zeta_x)=0$. Esto es, notando que $p(x)$ tiene grado a lo más n (con lo que su derivada $(n+1)$–ésima se anula) y con un poco de manipulación algebraica elemental),

$$0=g^{(n+1)}(\zeta_x)=f^{(n+1)}(\zeta_x)-(f(x)-p(x))\frac{(n+1)!}{\prod_{j=0}^{n}(x-x_j)}.$$

El resultado se sigue.

□

Ejemplo 2.1.7 *Deseamos calcular el polinomio $p(x)$ de grado 3 que interpola $f(x)=-e^{-x}$ en los nodos $x=0,1,2,3$ y aproximar el valor de la función por el del polinomio en $x=0{,}5$.*
La tabla de valores de $f(x)$ en los nodos es::

x	0	1	2	3
y	-1	$-0{,}367879$	$-0{,}135335$	$-0{,}049787$

Utilizando cualquiera de los métodos descritos anteriormente se obtiene entonces el polinomio:

$$p(x)=0{,}0420970x^3+-0{,}326078x^2+0{,}916102x-1$$

El error $f(x)-p(x)$ *que cometemos al aproximar* $f(0{,}5)$ *por* $p(0{,}5) = -0{,}618206375$ *es de acuerdo con el Teorema 2.1.6 de la forma*

$$\frac{f^{(4)}(\zeta)}{4!}(,5)(-,5)(-1{,}5)(-2{,}5) = -0{,}0390625 f^{(4)}(\zeta),$$

para algún punto $\zeta \in (0,3)$*. Podemos dar algo más de información: dado que*

$$f^{(4)}(x) = -e^{-x},$$

tenemos que $f^{(4)}(x) \in [-1,0]$ *para* $x \in [0,2]$ *con lo que podemos decir:*

$$0 \leq f(0{,}5) - p(0{,}5) \leq 0{,}0390625.$$

De esta manera hemos acotado el error. De hecho, el error exacto (hasta la precisión de la calculadora) es

$$f(0{,}5) - p(0{,}5) = 0{,}0116757,$$

lo que en efecto concuerda con la cota que hemos caculado. Podemos dibujar juntos $f(x)$ *y* $p(x)$*, como hacemos en la Figura 2.4.*

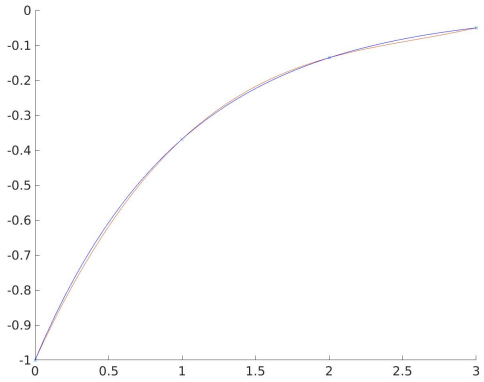

Figura 2.4: Comparación de $f(x) = -e^{-x}$ con el polinomio que la interpola en nodos $x = 0, 1, 2, 3$.

2.1.4. Diferencias divididas de Newton

Aunque la fórmula del Teorema 2.1.2 es perfectamente válida para calcular el polinomio interpolante de una tabla dada, resulta un poco molesto tener que recalcular todo de nuevo por el simple hecho de añadir otro nodo (lo que nos sucede inevitablemente con esa forma de resolver el problema). Para ayudarnos a combatir esta molestia, definimos las diferencias divididas de Newton, que nos permiten expresar el polinomio en otra *base* distinta. Recordemos que una base de un espacio vectorial es un conjunto de vectores linealmente independientes que lo generan. El espacio de los polinomios de grado menor o igual que n es un espacio vectorial sobre $\mathbb{R}$ (el lector puede comprobar que se cumplen todos los axiomas de la definición de un espacio vectorial), y una base de dicho espacio son los "vectores" (elementos del espacio, esto es, polinomios) $1, x, \ldots, x^n$. Pero hay otras bases,

infinitas de hecho, que podemos elegir. El método de las diferencias divididas de Newton elige una base asociada al conjunto de nodos $x_0, \ldots, x_n$, esto es la base:

$$1, (x - x_0), (x - x_0)(x - x_1), \ldots, (x - x_0) \cdots (x - x_{n-1}),$$

de forma que si añadimos otro nodo más x_{n+1} solo tenemos que añadir un último elemento a la base (el de la forma $(x - x_0) \cdots (x - x_n)$). Ya solo nos falta decir cómo se calculan los coeficientes en esa base del polinomio interpolante de Lagrange de una tabla de la forma 2.1. Se hace mediante las llamadas *diferencias divididas* de Newton.

Definición 2.1.8 *Definimos la n–ésima diferencia dividida de f respecto a $x_0, \ldots, x_n$ como el único número tal que el polinomio interpolante de f en esos nodos es de la forma:*

$$p(x) = f[x_0, \ldots, x_n]x^n + \textit{términos de orden inferior.}$$

Tenemos entonces el siguiente resultado:

Lema 2.1.9 *Las diferencias divididas no dependen del orden de los nodos $x_0, \ldots, x_n$, y adicionalmente satisfacen:*

$$p(x) = f[x_0] + f[x_0, x_1](x - x_0) + \cdots + f[x_0, \ldots, x_n](x - x_0) \cdots (x - x_{n-1}),$$

donde $p(x) = p_n(x)$ es el polinomio interpolante de f en $x_0, \ldots, x_n$.

Demostración. En efecto, la no dependencia se sigue de inmediato de la definición (el polinomio interpolante es único e independiente del orden en que tomamos los nodos). La igualdad se demuestra fácilmente por inducción: denotemos por $q_n(x)$ el polinomio en la parte de la derecha de la igualdad. Para $n = 0$ es obviamente cierto que $p_0(x) = q_0(x)$, y asumiendo que es cierta para $n - 1$ tenemos que $p_{n-1} = q_{n-1}$, luego q_n es un polinomio de grado n que satisface:

$$q_n(x_k) = q_{n-1}(x_k) = f(x_k), \quad k = 0, \ldots, n - 1,$$

y además el coeficiente principal de q_n es por definición igual al de p_n. Por lo tanto el polinomio $h(x) = q_n(x) - p_n(x)$ tiene grado a lo sumo $n - 1$ y tiene n raíces, con lo que $h \equiv 0$ y queda demostrado el lema. □

El siguiente resultado proporciona un sistema sencillo para calcular las diferencias divididas:

Teorema 2.1.10 *Las diferencias divididas de Newton satisfacen la siguiente relación recursiva:*

$$f[x_0, \ldots, x_n] = \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0}.$$

Demostración. Demostramos el resultado por inducción en n. El caso $n = 0$ es trivial. Supongamos que es cierto para el caso de n nodos, y veamos qué pasa con $n + 1$ nodos. Sea p_k el polinomio que interpola a f en $x_0, \ldots, x_k$, y sea $q(x)$ el que interpola a f en $x_1, \ldots, x_n$. Se comprueba fácilmente que

$$p_n(x) = q(x) + \frac{x - x_n}{x_n - x_0}(q(x) - p_{n-1}(x)),$$

basta ver para ello que ambos lados de la igualdad son polinomios de grado menor o igual que n que coinciden en $x_0, \ldots, x_n$. Ahora, el que ambos lados de la ecuación sean iguales como polinomios implica que sus coeficientes principales también coinciden, esto es:

$$f[x_0, \ldots, x_n] = \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0},$$

donde hemos usado la hipótesis de inducción para ver que los coeficientes principales de q y p_{n-1} son respectivamente $f[x_1, \ldots, x_n]$ y $f[x_0, \ldots, x_{n-1}]$.

□

Otros dos resultados importantísimos siguen.

Teorema 2.1.11 *Sea p el polinomio interpolante de una función f en $x_0, \ldots, x_n$, y sea t un punto en el dominio de la función. Entonces,*

$$f(t) = p(t) + f[x_0, \ldots, x_n, t](t - x_0) \cdots (t - x_n)$$

DEMOSTRACIÓN. En efecto, si tomamos el polinomio q que interpola a f en $x_0, \ldots, x_n, t$ tenemos por un lado $q(t) = f(t)$ y por el Lemma 2.1.9

$$q(t) = p(t) + f[x_0, \ldots, x_n, t](t - x_0) \cdots (t - x_n).$$

□

Teorema 2.1.12 *Si $f \in C^n[a, b]$ y $x_0, \ldots, x_n$ son puntos distintos en $[a, b]$ entonces hay un punto $\zeta \in (a, b)$ tal que*

$$f[x_0, \ldots, x_n] = \frac{f^{(n)}(\zeta)}{n!}$$

DEMOSTRACIÓN. Sea $p(x)$ el polinomio interpolante de f en $x_0, \ldots, x_{n-1}$. Por el Teorema 2.1.6 existe un $\zeta \in (a, b)$ tal que

$$f(x_n) = p(x_n) + \frac{f^{(n)}(\zeta_x)}{n!}(x_n - x_0) \cdots (x_n - x_{n-1}).$$

Por otro lado, el Teorema 2.1.11 implica que

$$f(x_n) = p(x_n) + f[x_0, \ldots, x_n](x_n - x_0) \cdots (x_n - x_{n-1}).$$

El resultado se sigue comparando ambas igualdades.

□

Cuando calculamos a mano un polinomio interpolante mediante el método de diferencias divididas de Newton resulta útil estructurar las cuentas en una matriz en la primera columna contiene $f[x_0]$, $f[x_1]$,..., la segunda columna contenga $f[x_0, x_1]$, $f[x_1, x_2]$... etcétera. Además, desplazamos ligeramente hacia abajo cada columna para que resulte más fácil ver de donde vienen los cálculos, puesto que por el Teorema 2.1.10 cada columna se calcula fácilmente a partir de la anterior y de la lista de nodos. A la hora de representarlo en un ordenador, resulta más fácil tratar esta tabla de datos como una matriz.

Ejemplo 2.1.13 *Resolvemos de nuevo el Ejemplo 2.1.7 usando diferencias divididas de Newton. El algoritmo de cálculo de diferencias divididas proporciona:*

−1,0000	0,6321	−0,1998	0,0421
−0,3679	0,2325	−0,0735	0
−0,1353	0,0855	0	0
−0,0498	0	0	0

Con ello, el polinomio obtenido es:

$$p(x) = -1 + 0{,}6321x - 0{,}1998x(x-1) + 0{,}0421x(x-1)(x-2) = 0{,}0421x^3 - 0{,}3261x^2 + 0{,}91601x - 1,$$

que es (si obviamos los errores de redondeo cometidos en ambos casos al escribir los números con un número de cifras decimales) el mismo que el obtenido en el Ejemplo 2.1.7.

Una implementación en Matlab para calcular las diferencias divididas a partir de una tabla de valores de f (y que también permite interpolar el polinomio en los valores deseados y dibujar el resultado) sería como sigue.

```
function [A,p,y]=difdiv(a,b,x,plotme)
% A partir de dos vectores columna a,b de igual tamanno
% calcula las diferencias divididas acumulandolas en una matriz
% la primera columna de la matriz son las dif. div de orden 0
% la segunda las de orden 1, etc.
% Devuelve tambien los coeficientes en la base de Newton del
% polinomio interpolador de la tabla, esto es, la primera fila de A
% Finalmente devuelve el valor del polinomio en los valores de la lista x
% Si "plotme" vale 1 entonces hace un dibujo de los datos y el polinomio
n=length(a);
A=zeros(n);
vector=b;
A(:,1)=vector;
for j=2:n
  vector=(vector(2:end)-vector(1:end-1))./(a(j:n)-a(1:n-j+1));
  A(1:n-j+1,j)=vector;
end
A
p=A(1,:);
y=zeros(size(x))+p(1);
producto=1;
for k=2:n
  producto=producto.*(x-a(k-1));
  y=y+p(k)*producto;
end

if plotme==1
  plot(a,b,'x',x,y);
end
```

2.1.5. La elección de los nodos y los polinomios de Chebyshev

Hasta ahora hemos trabajado con el problema de ajustar un polinomio a una tabla de datos dada, que suponemos corresponde a una función desconocida, amparados por el resultado teórico 2.1.6 que nos da un término de error sobre la función a aproximar. Sin embargo, en ocasiones tenemos la "suerte" de tener que generar nosotros la tabla de datos, y cabe preguntarse, ¿en qué nodos deberíamos calcular los valores de la función si queremos segurarnos de que el error de interpolación sea mínimo? Esto es, si los nodos los elegimos nosotros, ¿podemos sacar algún beneficio de una buena elección?
Esa es la pregunta a la que respondemos a continuación. Recordemos del Teorema 2.1.6 que el error en la interpolación viene dado por la fórmula

$$\frac{f^{(n+1)}(\zeta_x)}{(n+1)!}(x-x_0)\cdots(x-x_n).$$

No podemos actuar para cambiar el valor de $f^{(n+1)}$ (eso normalmente le corresponde a la naturaleza), pero si queremos elegir los nodos apropiadamente lo que debemos es tomar $x_0,\ldots,x_n$ que minimizan la cantidad

$$\max_x |(x-x_0)\cdots(x-x_n)|. \tag{2.1}$$

Supongamos inicialmente que nuestro intervalo de interpolación es $[-1,1]$, de forma que todos nuestros nodos debemos elegirlos ahí. Nos preguntamos entonces, ¿qué puntos en $[-1,1]$ minimizan la cantidad (2.1), con $x \in [-1,1]$? La respuesta viene dada por el siguiente resultado.

Teorema 2.1.14 *Para cada $n \geq 0$, la función $T_n(x) = \cos(n \arc\cos(x))$ es un polinomio de grado n, y además es, de entre todos los polinomios mónicos de grado n, aquél cuyo máximo valor absoluto en $[-1,1]$ es el menor. En otras palabras, para cualquier otro polinomio mónico $p(x)$ de grado n se tiene*

$$\max_{x\in[-1,1]} |p(x)| \geq \max_{x\in[-1,1]} |T_n(x)| = \frac{1}{2^{n-1}}.$$

Por lo tanto, los ceros $x_0, \ldots, x_n$ del polinomio de Chebyshev T_{n+1}, que están todos en $[-1,1]$, son los que minimizan la cantidad (2.1) cuando $x \in [-1,1]$, de entre todos los conjuntos de $n+1$ puntos en $[-1,1]$. Es más, dichos ceros vienen dados por la fórmula

$$\hat{x}_k = \cos\left(\frac{2k+1}{2n+2}\pi\right), \quad k = 0, \ldots, n. \tag{2.2}$$

Finalmente, los polinomios de Chebyshev pueden obtenerse mediante la fórmula recursiva:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geq 1,$$

con la puntualización de que la secuencia así obtenida ha de ser normalizada para obtener polinomios mónicos.

Demostración.
La demostración de este resultado es demasiado compleja para este curso. Recomendamos leerla en el libro *Análisis Numérico* de Burden y Faires.

□

Como consecuencia, si tenemos que elegir $n+1$ puntos $x_0, \ldots, x_n$ en el intervalo $[-1,1]$ para interpolar, los elegiremos de acuerdo con la fórmula (2.2), lo que nos garantiza que el error cometido en la interpolación es a lo sumo

$$\frac{\left|f^{(n+1)}(\zeta_x)\right|}{2^n(n+1)!}.$$

En el siguiente resultado explicamos la situación para un intervalo general $[a,b]$:

Corolario 2.1.15 *Dada $f \in C^{n+1}[a,b]$, eligiendo los nodos*

$$x_k = \hat{x}_k \frac{b-a}{2} + \frac{b+a}{2}, \quad k = 0, \ldots, n$$

el polinomio interpolante $p(x)$ de f en nodos $x_0, \ldots, x_n$ satisface (para cada $x \in [a,b]$):

$$|f(x) - p(x)| \leq \frac{(b-a)^{n+1}\left|f^{(n+1)}(\zeta)\right|}{2^{2n+1}(n+1)!},$$

para algún $\zeta \in (a,b)$, que depende de x.

Demostración.

Sea $g : [-1, 1] \to \mathbb{R}$ definido por

$$f(x) = g\left(\frac{2x - a - b}{b - a}\right), \quad \text{esto es,} \quad g(x) = f\left(x\frac{b-a}{2} + \frac{b+a}{2}\right)$$

que es una función de tipo $C^{n+1}[-1, 1]$. El polinomio $q(x)$ que interpola a g en los $n + 1$ nodos de Chebyshev $\hat{x}_0, \ldots, \hat{x}_n$ satisface

$$|g(x) - q(x)| \leq \frac{\left|g^{(n+1)}(\zeta_x)\right|}{2^n(n+1)!},$$

para algún $\zeta_x \in (-1, 1)$. Ahora, dado que $q(\hat{x}_k) = g(\hat{x}_k) = f(x_k)$, el polinomio $p(x)$ definido en $[a, b]$ por:

$$p(x) = q\left(\frac{2x - a - b}{b - a}\right)$$

tiene grado n y cumple $p(x_k) = q(\hat{x}_k) = f(x_k)$, luego es el polinomio interpolante de f en los nodos del enuciado. Además, para $x \in [a, b]$,

$$|f(x) - p(x)| = \left|g\left(\frac{2x - a - b}{b - a}\right) - q\left(\frac{2x - a - b}{b - a}\right)\right| \leq \frac{\left|g^{(n+1)}(\zeta_x)\right|}{2^n(n+1)!} = \frac{\left|f^{(n+1)}(\zeta)\right|}{2^n(n+1)!}\frac{(b-a)^{n+1}}{2^{n+1}},$$

para algunos $\zeta_x, \zeta \in (-1, 1)$. La última igualdad se obtiene a partir de la relación entre f y g en el principio de la demostración.

□

Ejemplo 2.1.16 Calcular el polinomio $p(x)$ que interpola a la función $f(x) = \sin(\pi x)$ en 3 nodos de Chebyshev. *Respuesta: Primero hacemos la tabla de valores de $f(x)$ en los nodos:*

x	-$\sqrt{3}/2$	*0*	$\sqrt{3}/2$
y	*-0.4086*	*0*	*0.4086*

Calculamos entonces mediante las diferencias divididas de Newton el polinomio:

$$\begin{array}{cccc} -0{,}86603 & -0{,}4086 & 0{,}4695 & 0 \\ 0 & 0 & 0{,}4695 & \\ 0{,}86603 & 0{,}4086 & & \end{array}$$

Con ello el polinomio buscado es:

$$p(x) = -0{,}4086 + 0{,}4695(x + 0{,}86603) = 0{,}4695x - 0{,}0012,$$

donde podemos ver que el término independiente es debido a errores de redondeo. Podríamos hacer las cuentas de forma simbólica sin errores, con cualquiera de los métodos estudiados. Por ejemplo, usando la fórmula de Lagrange,

$$p(x) = -0{,}4086\frac{x(x - \sqrt{3}/2)}{3/2} + 0{,}4086\frac{x(x + \sqrt{3}/2)}{3/2} = 0{,}4086 \cdot \frac{2}{\sqrt{3}}x = 0{,}4718x.$$

Evitando errores de redondeo en la evaluación de la función, podemos dar una fórmula cerrada:

$$p(x) = \frac{2}{\sqrt{3}}\sin\left(\pi\sqrt{3}/2\right)x.$$

Gráficamente podemos ver la función original y el polinomio interpolador en la Figura 2.5

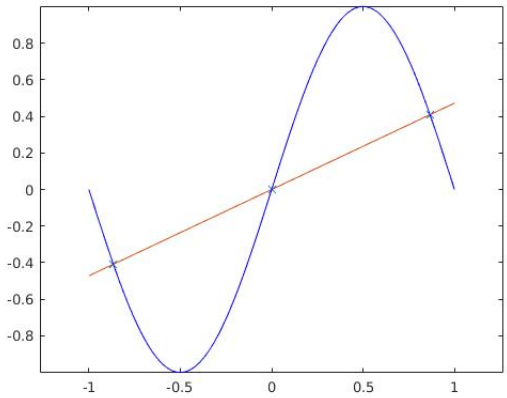

Figura 2.5: Comparación de $f(x) = \sin(\pi x)$ con el polinomio que la interpola en 3 nodos de Chebyshev. El polinomio interpolador resulta ser de grado 1, y claramente insuficiente para aproximar la función.

Ejemplo 2.1.17 Utilizando el método de diferencias divididas de Newton, calcula el polinomio interpolante de grado 3 de la función $f(x) = \cos x$ en el intervalo $[0, \pi]$ utilizando el número apropiado de nodos de Chebyshev. *Respuesta: para calcular el polinomio de grado* 3 *tenemos que interpolar en* 4 *nodos. Los* 4 *nodos de Chebyshev para el intervalo* $[-1, 1]$ *son:*

$$\hat{x}_0 = \cos(\pi/8) = 0{,}9239, \quad \hat{x}_1 = \cos(3\pi/8) = 0{,}3827,$$

$$\hat{x}_2 = \cos(5\pi/8) = -0{,}3827, \quad \hat{x}_3 = \cos(7\pi/8) = -0{,}9239.$$

Al pasarlos al intervalo $[0, \pi]$ *obtenemos:*

$$x_0 = \cos(\pi/8) = 3{,}0220, \quad x_1 = \cos(3\pi/8) = 2{,}1719,$$

$$x_2 = \cos(5\pi/8) = 0{,}9697, \quad x_3 = \cos(7\pi/8) = 0{,}1196.$$

Calculamos ahora el polinomio con el método de diferencias divididas, obteniendo:

3,0220	−0,9929	−0,5026	0,2135	0,1471
2,1719	−0,5656	−0,9409	−0,2135	0
0,9697	0,5656	−0,5026	0	0
0,1196	0,9929	0	0	0

Con ello, el polinomio es:

$$-0{,}9929 - 0{,}5026(x - 3{,}0220) + 0{,}2135(x - 3{,}0220)(x - 2{,}1719) +$$

$$0{,}1471(x - 3{,}0220)(x - 2{,}1719)(x - 0{,}9697)$$

Gráficamente podemos ver la función original y el polinomio interpolador en la Figura 2.6

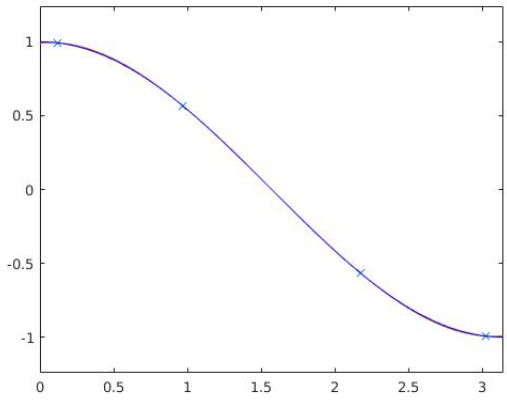

Figura 2.6: Comparación de $f(x) = \cos x$ con el polinomio que la interpola en 4 nodos de Chebyshev. La calidad de la aproximación es excelente.

Ejemplo 2.1.18 Sea $f : [0, 7] \to \mathbb{R}$ una función tal que todas las derivadas de cualquier orden de f están acotadas en valor absoluto por 4. Cuál es el mínimo número de nodos de interpolación que debemos usar si queremos asegurarnos de que el error de interpolación sea, en valor absoluto, a lo máximo de una centésima? Cuáles son entonces los nodos de interpolación? *Respuesta: Los nodos de interpolación que mejor garantizan que el error es pequeño son los de Chebyshev, que en el intervalo* $[0, 7]$ *son los de la forma*

$$\frac{7}{2}(x_k + 1),$$

donde $x_k = \cos\left(\frac{2k+1}{2n+2}\pi\right)$, $k = 0, \ldots, n$. *Con ellos, el error de interpolación en cualquier* $x \in [0, 7]$ *es a lo sumo:*

$$\frac{7^{n+1}|f^{(n+1)}(\zeta_x)|}{2^{2n+1}(n+1)!} \leq \frac{7^{n+1}4}{2^{2n+1}(n+1)!}.$$

Esta cantidad es menor que una centésima si y solamente si $n \geq 8$, *con lo que el mínimo número de nodos es* 9, *y dichos nodos son:*

$$0{,}0532, \quad 0{,}4689, \quad 1{,}2502, \quad 2{,}3029, \quad 3{,}5000, \quad 4{,}6971, \quad 5{,}7498, \quad 6{,}5311, \quad 6{,}9468$$

Terminamos esta sección indicando que una mala elección de los nodos produce el llamado "fenómeno Runge", que consiste en que polinomios interpolantes de grados cada vez mayores en lugar de aproximar la función producen tremendas oscilaciones, sobre todo en los extremos. Ello se puede comprobar tratando de interpolar la función $f(x) = (1 + x^2)^{-1}$ en $[-3, 3]$ en nodos equiespaciados. El uso de los nodos de Chebyshev evita la aparición del efecto Runge. Podemos ver este efecto gráficamente en la Figura 2.7.

2.2. Interpolación polinomial a trozos

En ocasiones sucede que tenemos una cantidad grande de datos en el plano y necesitamos una curva que pase exáctamente por todos ellos pero debido al fenómeno Runge o a otras causas no queremos que sea un polinomio,

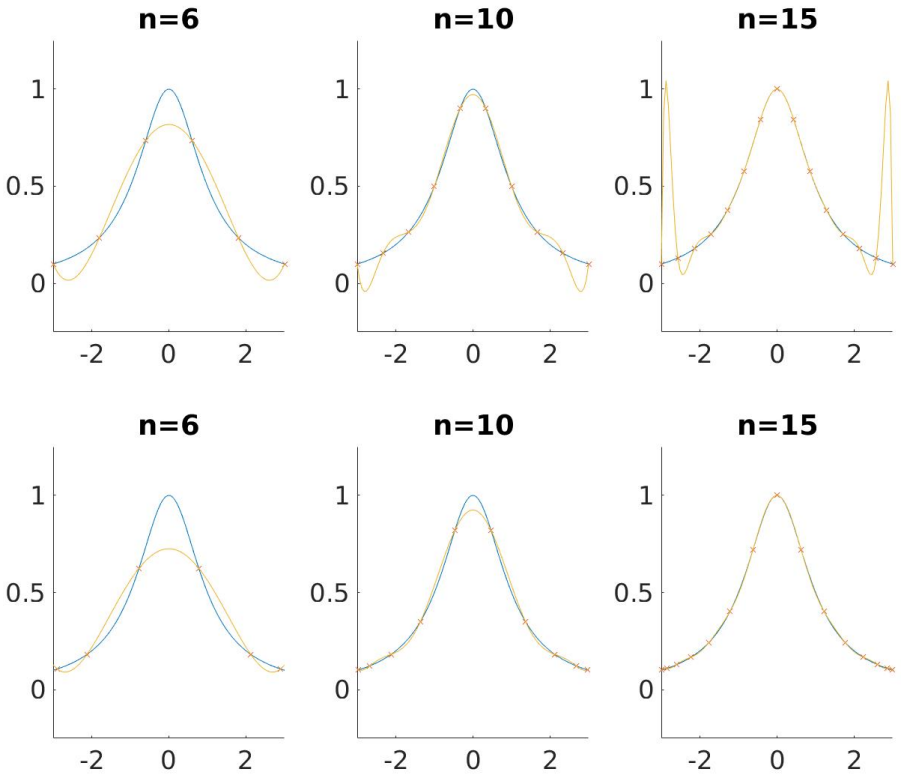

Figura 2.7: fenómeno Runge en acción: en la fila de arriba, tres polinomios interpolantes para $6, 10$ y 15 nodos equiespaciados de la función $f(x) = (1 + x^2)^{-1}$ en $[-3, 3]$. Obsérvese la oscilación del polinomio interpolante en los extremos, degradando la calidad de la interpolación. En la fila de abajo, mismo procedimiento utilizando los nodos de Chebyshev, lo que hace desaparecer el fenómeno Runge.

sino que preferimos definir la función a trozos.

2.2.1. Interpolación lineal a trozos

La opción más sencilla es la interpolación lineal: dados unos datos como los de la Tabla 2.1 que suponemos contiene algunos valores de la función f, y suponiendo que los nodos están ordenados de forma que $x_0 < \cdots < x_n$ podemos considerar la función interpolante:

$$g(x) = y_j \frac{x - x_{j+1}}{x_j - x_{j+1}} + y_{j+1} \frac{x - x_j}{x_{j+1} - x_j} = y_j + \frac{y_{j+1} - y_j}{x_{j+1} - x_j}(x - x_j) \quad \text{si} \quad x \in [x_j, x_{j+1}],$$

esto es $g(x)$ iguala al polinomio interpolante de grado 1 en cada subintervalo $[x_j, x_{j+1}]$. La principal ventaja de este método es que es tremendamente sencillo (no requiere gran capacidad de cálculo) y su principal desventaja es que la función $g(x)$ así obtenida ni siquiera es derivable. Este método es conocido como interpolación lineal. Una implementación en Matlab de este procedimiento sería como sigue:

```
function y=interpolacionlineal(a,b,x,plotme)
% Dada una tabla de datos a,b con a ordenado de menor a mayor
% devuelve un vector y con el resultado de interpolar en los
% puntos de x con interpolacion lineal a trozos.
% si "plotme" es 1 dibuja todo.
n=length(x);
for j=1:n
  xj=x(j);
  if (xj<a(1)) || (xj>a(end))
    sprintf('El valor %f que esta en el vector x no vale',xj)
    y(j)=NaN;
  elseif xj==a(1)
    y(j)=b(1);
  elseif xj==a(end)
    y(j)=b(end);
  else
    indice=find(a>xj,1); % Encuentra la primera posicion de a mayor que xj
    y(j)=b(indice-1)+(b(indice)-b(indice-1))/(a(indice)-a(indice-1))*(xj-a(indice-1));
  end
end

if plotme==1
  plot(a,b,'x',x,y);
end
```

Podemos ver un ejemplo gráfico de interpolación lineal en la Figura 2.8.

2.2.2. Trazadores o "splines" cúbicos

Más complicado, pero con un resultado mucho más satisfactorio en términos de derivabilidad, es el método de los llamados trazadores cúbicos.

Definición 2.2.1 *Dados unos datos como los de la Tabla 2.1 ordenados de forma que $x_0 < \cdots < x_n$, un trazador cúbico S para la tabla es una función con las siguientes propiedades:*

- *En cada subintervalo $[x_j, x_{j+1}]$ $j = 0, \ldots, n-1$, $S(x)$ es un polinomio cúbico distinto, que denotamos $S_j(x)$.*

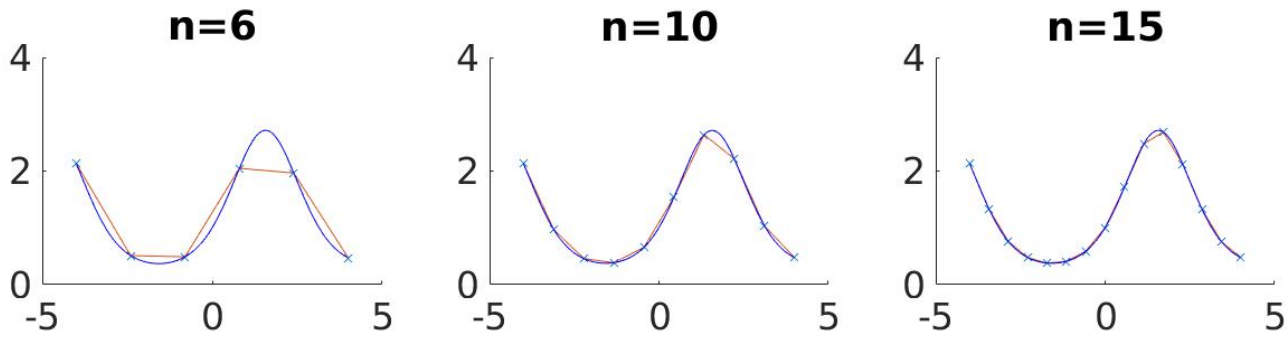

Figura 2.8: Interpolación lineal en distintos números de nodos equiespaciados para la función $f(x) = e^{\sin x}$ en el intervalo $[-4, 4]$.

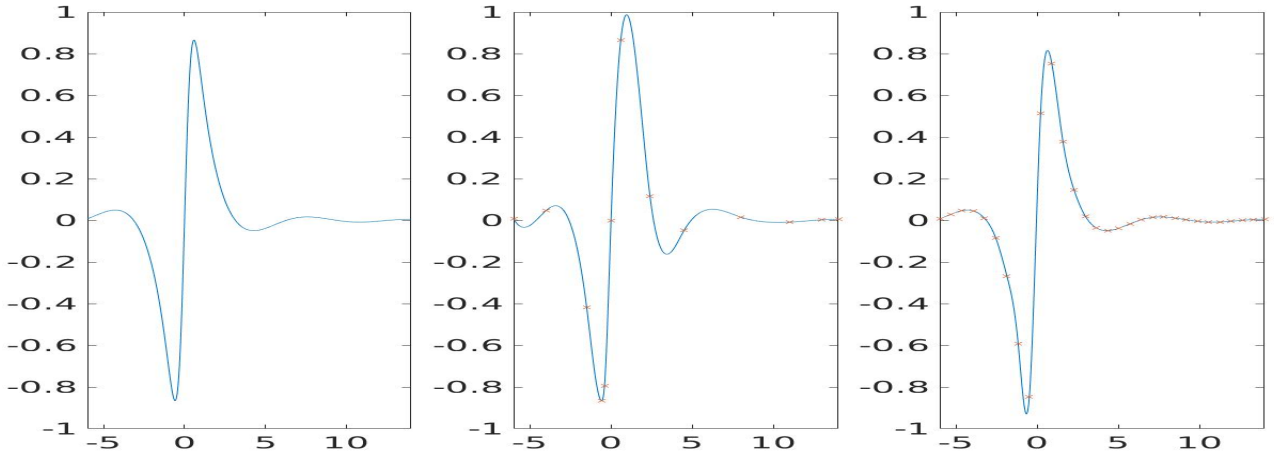

Figura 2.9: Interpolación mediante trazadores cúbicos para la función $f(x) = \sin(x)e^{\frac{1}{1+x^2}}/(1+x^2)$ en el intervalo $[-6, 15]$. En la primera imagen, la función original. En la segunda imagen, trazador cúbico con 12 nodos escogidos a mano. En la tercera, trazador cúbico con 30 nodos equiespaciados.

- $S(x_j) = y_j$ *para* $j = 0, \ldots, n-1$.
- $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$, $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$, $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ *para* $j = 1, \ldots, n-2$. *Dicho de otra forma,* $S(x)$ *es de tipo* C^2 *(aunque esté definida a trozos).*
- *Se cumple una de las dos siguientes condiciones:*
  - $S''(x_0) = S''(x_n) = 0$ *(condiciones de frontera libre o natural).*
  - $S'(x_0) = f'(x_0)$ *y* $S'(x_n) = f'(x_n)$ *(condiciones de frontera ligada). Esta condición solo tiene sentido si suponemos que la tabla corresponde a los valores de una función* f *derivable en* x_0 *y* x_n

Tenemos el siguiente resultado.

Teorema 2.2.2 *En las condiciones de la Definición 2.2.1, existe un único trazador cúbico con condiciones de frontera libre, y un único trazador cúbico con condiciones de frontera ligada (asumiendo que la tabla corresponde a los valores de una función* f *derivable en* x_0 *y* x_n*).*

Demostración. La demostración de este resultado es bastante larga, aunque no muy sofisticada. Recomendamos su lectura en el libro de Burden y Faires. □

No vamos a dar un código Matlab para resolver el problema de los trazadores cúbicos (en inglés, *splines*), limitándonos a proporcionar la función interna que realiza el trabajo: `spline`. Aprender a utilizar esta función resulta útil a la hora de proporcionar dibujos suaves y sencillos que pasen por una serie de puntos dados. La mayor desventaja es que no nos proporciona una fórmula, sino un conjunto de fórmulas cada una válida en un trozo. El uso que tiene es por lo tanto más gráfico que analítico.

Podemos ver un ejemplo gráfico de interpolación mediante interpolantes cúbicos en la Figura 2.9.

Mencionamos un teorema relativo al error cometido por la interpolación por trazadores cúbicos con frontera natural.

Teorema 2.2.3 *Sea $f \in C^4[a,b]$ tal que su cuarta derivada está acotada por M en valor absoluto. Si S es el trazador cúbico de frontera libre obtenido para una tabla de valores de f con nodos $x_0 < \cdots < x_n$, entonces*

$$\max_{a\le x\le b} |f(x) - S(x)| \le \frac{5M}{384} \max_{0\le j\le n-1} (x_{j+1} - x_j)^4.$$

Por ejemplo, si la cuarta derivada de f está acotada por 1 y tomamos nodos equiespaciados cada uno a distancia $1/2$ del anterior, podemos garantizar un error máximo de menos de una milésima.
Acabamos esta sección indicando que si tomáramos un alambre flexible y lo obligáramos a pasar por una colección de puntos del plano, su forma final sería parecida a la del trazador cúbico de frontera libre.

2.3. El método de mínimos cuadrados

Es frecuente que deseemos obtener una función dada por una fórmula matemática que se ajuste "suficientemente bien" a un conjunto de datos dado, y que la fórmula en cuestión no sea un polinomio, o sea un polinomio de grado muy pequeño (pese a tener una gran cantidad de valores para ajustarlo). Entre los muchos ejemplos de aplicación se encuentra el encontrar la *recta de regresión* de un conjunto de datos, esto es, la recta que que "mejor aproxima" los datos en su conjunto. En este caso se trata de encontrar una combinación lineal de dos funciones $f_1(x) = 1$ y $f_2(x) = x$ que aproxima de la mejor manera posible una colección de datos determinada. El método de mínimos cuadrados, una de las más exitosas creaciones de Karl Fiedrich Gauss, fue diseñado por éste para encontrar el asteroide Ceres, que se le había "perdido" a la humanidad a las pocas semanas de ser descubierto.

Definición 2.3.1 *Dada una colección de datos como los de la Tabla 2.1, y dadas unas funciones $f_1, \ldots, f_m$, la* solución de mínimos cuadrados *para aproximar los datos con las funciones dadas es una función de la forma*

$$f(x) = a_1 f(x) + \ldots + a_m f_m(x) \tag{2.3}$$

que minimiza el error cuadrático

$$\sum_{k=0}^{n} (f(x_k) - y_k)^2.$$

En esta definición, por lo general, consideramos $m << n$, aunque la definición es válida igualmente en el contexto general.

La elección de minimizar el error cuadrático (la elección de la potencia 2 en lugar de otra) obedece fundamentalmente a la facilidad de la resolución del problema en ese caso. Comenzamos por observar que, de existir una función de la forma (2.3) que hiciese cero el error tendríamos

$$a_1 f_1(x_k) + \ldots + a_m f_m(x_k) = y_k, \quad k = 1, \ldots, n,$$

o en forma matricial

$$\underbrace{\begin{pmatrix} f_1(x_0) & f_2(x_0) & \cdots & f_m(x_0) \\ \vdots & \vdots & & \vdots \\ f_1(x_n) & f_2(x_n) & \cdots & f_m(x_n) \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}}_{b}.$$

Notemos que (si $m < n+1$) este es un sistema sobredeterminado y por lo tanto no debemos esperar que tenga solución, salvo en casos extremadamente afortunados. La idea de Gauss consiste en sustituir la búsqueda de un x que resuelva el problema $Ax = b$, que no podemos esperar que tenga solución, por la búsqueda de un x que minimize $\|Ax-b\|^2$, esto es, por la búsqueda de un x que minimice el error cuadrático. Gauss entonces notó lo siquiente:

Lema 2.3.2 *Dadas A, b de forma que A es $n \times m$, b es $n \times 1$ y $m \leq n$, entonces el x de norma mínima que minimiza $Ax - b$ es precisamente el único x que cumple la* ecuación normal

$$A^T Ax = A^T b,$$

que es cuadrado y tiene una única solución, siempre que el rango de A sea máximo.

DEMOSTRACIÓN. De entre todos los $x \in \mathbb{R}^m$ que pueden minimizar la cantidad escrita, el de norma mínima pertenecerá al complemento ortogonal del núcleo de A. Sea entonces $g : Ker(A)^\perp \to \mathbb{R}$, $g(x) = \|Ax-b\|^2$, satisface $\lim_{\|x\|\to\infty} \|g(x)\| = \infty$. El problema

$$\min_{x \in Ker(A)^\perp} g(x)$$

se resuelve por lo tanto en un punto en el que el gradiente v_x de g se anule. Calcular este gradiente es sencillo, pues ha de cumplir $\langle \dot{x}, v_x \rangle = Dg(x)\dot{x}$ para todo $\dot{x} \in Ker(A)^\perp$, pero

$$Dg(x)\dot{x} = \frac{d}{dt} \mid_{t=0} g(x + t\dot{x}) = 2\langle Ax - b, A\dot{x} \rangle = \langle 2A^T(Ax-b), \dot{x} \rangle,$$

luego $v_x = \pi_{Ker(A)^\perp} 2A^T(Ax-b) = 2A^T(Ax-b)$. El único x en el que se anula dicho gradiente es precisamente la solución de la ecuación normal.

□

Aunque el Lema 2.3.2 proporciona en teoría un método para calcular la aproximación de mínimos cuadrados, en realidad es poco efectivo por motivos de condicionamiento matricial. Es mucho mejor calcular la factorización QR reducida $A = \hat{Q}R$ de A y resolver el sistema cuadrado $Rx = \hat{Q}^T b$, que (en el caso de que A tenga rango máximo) admite una única solución, la buscada.
Una implementación en Matlab de este procedimiento sería como sigue:

```
function [coefs,y]=aproximacionminimoscuadrados(a,b,F,x,plotme)
% Dado un vector fila de funciones F proporciona la combinacion lineal
% de las funciones que contiene que mejor aproxima por minimos
% cuadrados los datos en la tabla a,b, que son vectores de igual tamanno.
% el programa devuelve el vector de coeficientes y los valores de
% la aproximacion en el vector x.
% Si "plotme" vale 1 dibuja los resultados.
% Ejemplo de uso:
% a=[-1;0;1;2;3];
% b=[-2;0;0;-2;-6];
% F=@(x) [x x.^4]
% [coefs,y]=aproximacionminimoscuadrados(a,b,F,x,1);
n=length(a);
m=length(F(a(1)));
A=zeros(n,m);
for i=1:n
  A(i,:)=F(a(i));
```

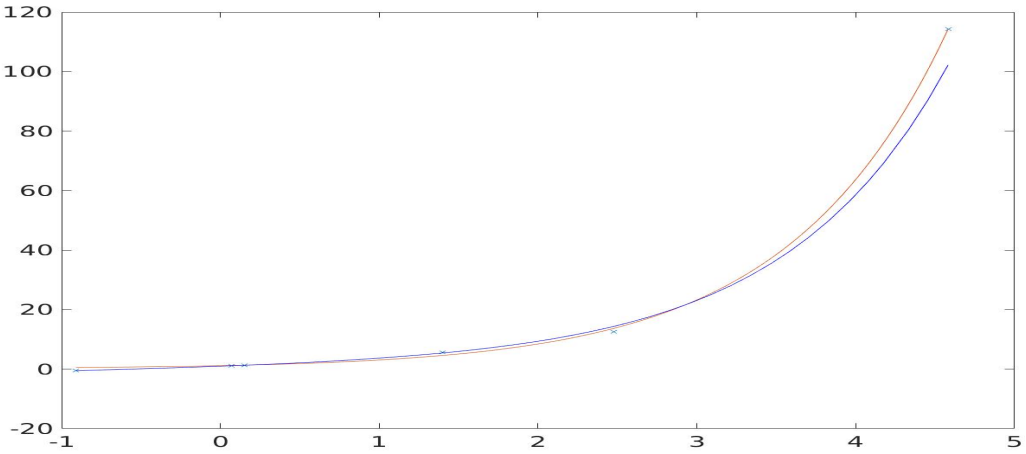

Figura 2.10: Partiendo de la función $f(x) = x + e^x$ (dibujada en azul), hacemos una serie de medidas a las que añadimos un error relativo de aproximadamente el 10 % del valor de la función en cada punto (marcadas con una "x"). Entonces, calculamos la función de la forma $a_1 x + a_x e^x$ que mejor aproxima a las medidas tomadas. Nótese que ni la función original ni la de mínimos cuadrados pasa por las medidas tomadas, y aún así ambas se parecen bastante.

```
end
coefs=A\b;  % Resuelve el problema con los metodos de Matlab
y=x;
for k=1:length(x)
  y(k)=F(x(k))*coefs;
end
if plotme==1
  plot(a,b,'x',x,y);
end
```

Podemos ver un ejemplo gráfico de interpolación mediante interpolantes cúbicos en la Figura 2.10.

2.4. Exercises. Interpolation

Exercise 2.1

Compute the polynomial $p(x)$ interpolating $f(x) = \sin(\pi x)$ in 3 Chebyshev nodes.
Answer: We first write a table of values for $f(x)$ in the Chebyshev nodes:

x	$-\sqrt{3}/2$	0	$\sqrt{3}/2$
y	-0.4086	0	0.4086

The demanded polynomial is then:

$$p(x) = -0{,}4086\frac{x(x-\sqrt{3}/2)}{3/2} + 0{,}4086\frac{x(x+\sqrt{3}/2)}{3/2} = 0{,}4086 \cdot \frac{2}{\sqrt{3}}x =$$

$$\frac{2}{\sqrt{3}}\sin\left(\pi\sqrt{3}/2\right)x = 0{,}4718x$$

Exercise 2.2

The following table describes an unknown function $y = f(x)$:

x	-2	-1	1	2
y	-9	-3	-1	6

We need to find the constants $a, b \in \mathbb{R}$ such that $f(x)$ is approximated by a function of the form $ax^3 + b$.
Answer: We construct the corresponding linear system

$$\begin{pmatrix} -8 & 1 \\ -1 & 1 \\ 1 & 1 \\ 8 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -9 \\ -3 \\ -1 \\ 6 \end{pmatrix},$$

The normal equations are then:

$$\begin{pmatrix} -8 & -1 & 1 & 8 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} -8 & 1 \\ -1 & 1 \\ 1 & 1 \\ 8 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -8 & -1 & 1 & 8 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} -9 \\ -3 \\ -1 \\ 6 \end{pmatrix},$$

that is

$$\begin{pmatrix} 130 & 0 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 122 \\ -7 \end{pmatrix},$$

which yields the solution:

$$a = \frac{122}{130} = 0{,}9384\ldots, \quad b = \frac{-7}{4} = -1{,}75.$$

Exercise 2.3

Compute the cubic polynomial $p(x)$ which interpolates:

x	-1	0	1	2
y	0	2	1	2

Answer: Using Lagrange's formula:

$$p(x) = 0\frac{x(x-1)(x-2)}{-6} + 2\frac{(x+1)(x-1)(x-2)}{2} +$$
$$1\frac{(x+1)x(x-2)}{-2} + 2\frac{(x+1)x(x-1)}{6} = \frac{5}{6}x^3 - \frac{3}{2}x^2 - \frac{1}{3}x + 2.$$

Exercise 2.4

For each of the following tables of data, functions and points, compute the function that approximates the data, compute the value of the function at the points and, if you have a computer, plot the values of the data and the function in the plane.

I) Function: Lagrange polinomial of degree 3 (using the original formula by Lagrange and Newton's divided differences). Points: $\{-2, 0\}$.

x	-3	-1	1	3
y	14	4	2	7

II) Function: Lagrange polinomial of degree 3 (using the original formula by Lagrange and Newton's divided differences). Points: $\{-2, 0\}$.

x	-3	-1	1	3
y	14	4	2	8

III) Function: Lagrange polinomial of degree 4 (using the original formula by Lagrange and Newton's divided differences). Points: $\{-1/2, 1, 3\}$.

x	-3	-2	-1	0	2
y	25	7	1	1	6

IV) Function: piecewise linear interpolation and cubic spline. Points: $\{3/2, 21/20, 7/2\}$.

x	1	2	3	4
y	100	80	70	65

Exercise 2.5

Function: piecewise linear interpolation cubic spline. Points: $\{-1, 0, 1\}$.

x	-3	-1/2	3	4
y	10	11	40	6

Exercise 2.6

Same as in the previous exercise, but for this one you need a computer.

- Function: $T(t) = c_1 t + c_2 e^{t/5}$ by minimal square error and by cubic spline. Points: $\{3, 10, 50\}$.

t=time	0	5	10	15	20	25	30	35	40
T=temperature	0	40	80	130	190	265	410	730	1600

- Function: degree 3 polynomial by minimal square error and by cubic spline. Points: $\{-4, 1/2, 4\}$.

x	-3	-2	-1	0	1	2	3
y	-1.05	1.57	2.60	2.23	1.63	3.41	8.32

- Function: $f(x) = ae^{-x}\cos(2\pi x) + be^{-x}\sin(2\pi x)$ by minimal square error and by cubic spline. Points: $\{1/10, 11/10\}$.

x	0	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
y	2.34	1.80	-0.28	-1.12	-0.09	0.87	0.74	-0.1	-0.23	-0.11	0.33

Exercise 2.7

Jon Doe made a series of experiments with the time needed for a baloon to get to the floor when thrown from different heigths: 0,1m, 0,5m, 1m, 1,5m. He then used polynomial interpolation to deduce the approximate value of the time for a height of 1,3 m, getting that the time was 1,8737s. Unfortunately, Jon lost some of the data he had originally collected so we only have the information of 3 experiments:

heigth (m)	0.1	0.5	1	1.5
time (s)	0.1	0.9	?	2

Can you tell which was the time measured by Jon for heigth 1m?

Exercise 2.8

The height of the mushroom of a nuclear bomb grows from approx. $10km$ for 1 kiloton bombs to $20km$ for 10 kiloton bombs and $35km$ for 30 megaton bombs. You are asked to provide an approximate value for the heigth of the mushroom for 1 megaton and 10 megaton bombs. *Answer: there are many ways to solve the problem. Reasonable answers for* 1 *and* 10 *megaton bombs are* 31,38 *and* 34,32 *kilometers respectively*

Exercise 2.9

Learn how to use the programs which have been sent to you by the proffessor. Use them to solve the interpolation problems you have done by hand, and check that you obtain the same answers.

Exercise 2.10

Study the quality of Lagrange interpolation for functions $f(x) = \sin(x^2)$ and $f(x) = e^{\sin(x)}$ calculated in 10 uniform points in the interval $[0, 5]$. Compare the results to using 10 Chebyshev's nodes in that interval.

Exercise 2.11

Learn how to use Matlab's function `interp1` to generate spline interpolation and other interpolations too.

Exercise 2.12

Compare the results in the previous to last exrercise with those of cubic spline interpolation.

Exercise 2.13

Same as above (polynomial and spline interpolation) for Runge's function $f(x) = \frac{1}{1+25x^2}$, this time using the points $-1, -8/10, -6/10, \ldots, 6/10, 8/10, 1$. Try also with other collections of points, for example $-1, -1/2, 0, 1/2, 1$. Try also with Chebyshev's nodes.

Exercise 2.14

The speed of a person in free fall has been measured each second after jump. The results are (in meters per second): $10, 19, 27, 32, 39, 43, 46, 48, 50, 51, 52, 53$. After 12 seconds the speed remaind approximately constant.

You are asked to try different forms of interpolation, and to ultimately provide a simple formula that approximates the movement. Use your results to deduce how much distance the human person has fallen after 12 seconds. Data has been extracted from http://www.johnnyutah.com/

Exercise 2.15

Using the web http://impact.ese.ic.ac.uk/ImpactEffects/ you can obtain simulation data for the effects of a meteor impact on the surface of Earth. Fix some (reasonable) numbers for the projectile speed, density, angle of impact. Then, check the maximum air velocity given by the simulation for 5 different diameters. Interpolate the table that you get with a degree 4 polinomial and then compare the results of your interpolation to those of the original simulation for different values of the diameters. You should decide carefully which diameter values you choose for the interpolation.