Capítulo 3

Polinomios

3.1 Polinomios de variable real

3.1.1 Evaluación de polinomios

Para el cálculo eficiente de los valores de un polinomio se utiliza el algoritmo de Horner, también conocido como *multiplicación anidada* y como *división sintética*. La regla de Horner para la evaluación de polinomios establece que una expresión polinómica puede reescribirse como una factorización anidada. Por ejemplo, el siguiente polinomio de grado 5:

$$p(x) = a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \tag{3.1}$$

puede reescribirse como:

$$p(x) = ((((a_5x + a_4)x + a_3)x + a_2)x + a_1)x + a_0 \tag{3.2}$$

Mediante la regla de Horner es posible construir un algoritmo eficiente para el cálculo del valor, en un punto x_0, de un polinomio $p(x)$ de grado n:

input n,$(a_i : 0 \leq i \leq n)$, x_0
$\alpha \leftarrow a_n$
for $k = n-1, n-2, \ldots, 0$ **do**
    $\alpha \leftarrow \alpha x_0 + a_k$
end
output α.

Nota para el Prof.: La regla de Horner funciona bien en los casos normales en los que el exponente más grande es relativamente pequeño. Sin embargo, para polinomios con exponente máximo grande, las multiplicaciones repetidas forzadas por los paréntesis anidados pueden dar lugar a una pérdida importante de precisión y a la degradación del rendimiento de las operaciones.

El algoritmo de Horner también se utiliza para realizar una *deflación*, que es el proceso mediante el cual se cancela un factor lineal del polinomio. Si x_0 es un cero del polinomio $p(x)$, entonces $(x - x_0)$ es un factor de $p(x)$ y los ceros restantes de $p(x)$ son los $n-1$ ceros del polinomio:

$$q(x) = p(x)/(x - x_0) \tag{3.3}$$

Si representamos los polinomios $p(x)$ y el desconocido $q(x)$ como:

$$p(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_{n-1}x^{n-1} + a_nx^n \tag{3.4}$$
$$q(x) = b_0 + b_1x + b_2x^2 + b_3x^3 + \cdots + b_{n-1}x^{n-1} \tag{3.5}$$

cuando las ecuaciones (3.4) y (3.5) se substituyen en la ecuación (3.3), los coeficientes de las potencias correspondientes de x de ambos lados de la ecuación pueden igualarse término a término. Las ecuaciones que se obtienen son:

$$
\begin{aligned}
b_{n-1} &= a_n \\
b_{n-2} &= a_{n-1} + x_0 b_{n-1} \\
b_{n-3} &= a_{n-2} + x_0 b_{n-2} \\
&\vdots \\
b_1 &= a_2 + x_0 b_2 \\
b_0 &= a_1 + x_0 b_1
\end{aligned}
\tag{3.6}
$$

A partir del conjunto de ecuaciones (3.6), se puede escribir de forma compacta el algoritmo de Horner para calcular $q(x)$ a partir de $p(x)$ y una de sus raíces x_0 como se indica a continuación:

> **input** n,$(a_i : 0 \leq i \leq n)$, x_0
> $b_{n-1} \leftarrow a_n$
> **for** $k = n-1, n-2, \ldots, 1$ **do**
> $b_{k-1} \leftarrow a_k + x_0 b_k$
> **end**
> **output** $(b_i : 0 \leq i \leq n-1)$.

Un tercer uso del algoritmo de Horner, que se deriva directamente del punto anterior, es el cálculo eficiente del valor de las derivadas de cualquier orden del polinomio $p(x)$ en un punto x_0. Presentamos a continuación el pseudocódigo que genera $\alpha = p(x_0)$, $\beta = p'(x_0)$ y $\gamma = p''(x_0)$ suponiendo que $p(x)$ es de la forma dada por la ecuación (3.4) y que se proporciona x_0:

> **input** n,$(a_i : 0 \leq i \leq n)$, x_0
> $\alpha \leftarrow a_n$
> $\beta \leftarrow 0$
> $\gamma \leftarrow 0$
> **for** $k = n-1, n-2, \ldots, 0$ **do**
> $\gamma \leftarrow \beta + x_0 \gamma$
> $\beta \leftarrow \alpha + x_0 \beta$
> $\alpha \leftarrow a_k + x_0 \alpha$
> **end**
> $\gamma \leftarrow 2\gamma$
> **output** α, β, γ.

Ejercicio: Desarrolle el algoritmo expresado mediante el pseudocódigo anterior para calcular el polinómio $p(x) = x^4 - 4x^3 + 7x^2 - 5x - 2$ en el punto $x_0 = 1$.

3.1.2 Interpolación

Nos centraremos ahora en el problema de obtener, a partir de una tabla de parejas $(x,f(x))$ definida en un cierto intervalo $[a,b]$, el valor de la función para cualquier x perteneciente a dicho intervalo.

Supongamos que disponemos de las siguientes parejas de datos:

x	x_0	x_1	x_2	$\cdots$	x_n
y	y_0	y_1	y_2	$\cdots$	y_n

El objetivo es encontrar una función continua lo más sencilla posible tal que

$$f(x_i) = y_i \qquad (0 \le i \le n) \tag{3.7}$$

Se dice entonces que la función $f(x)$ definida por la ecuación (3.7) es una **función de interpolación** de los datos representados en la tabla.

Existen muchas formas de definir las funciones de interpolación, lo que da origen a un gran número de métodos (polinomios de interpolación de Newton, interpolación de Lagrange, interpolación de Hermite, etc). Sin embargo, nos centraremos exclusivamente en dos funciones de interpolación:

1. Los polinomios de interpolación de Lagrange.
2. Las funciones de interpolación **splines**. Estas funciones son especialmente importantes debido a su idoneidad en los cálculos realizados con ordenador.

Polinomios de interpolación de Lagrange

Un polinomio de interpolación de Lagrange, p, se define en la forma:

$$p(x) = y_0\ell_0(x) + y_1\ell_1(x) + \cdots + y_n\ell_n(x) = \sum_{k=0}^{n} y_k\ell_k(x) \tag{3.8}$$

en donde $\ell_0, \ell_1, \ldots, \ell_n$ son polinomios que dependen sólo de los nodos tabulados $x_0, x_1, \ldots, x_n$, pero no de las ordenadas $y_0, y_1, \ldots, y_n$. La fórmula general del polinomio ℓ_i es:

$$\ell_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \tag{3.9}$$

Para el conjunto de nodos $x_0, x_1, \ldots, x_n$, estos polinomios son conocidos como **funciones cardinales**. Utilizando estos polinomios en la ecuación (3.8) obtenemos la forma exacta del polinomio de interpolación de **Lagrange**.

Ejemplo: Suponga la siguiente tabla de datos:

x	5	-7	-6	0
y	1	-23	-54	-954

Construya las funciones cardinales para el conjunto de nodos dado y el polinomio de interpolación de Lagrange correspondiente.

Las funciones cardinales, empleando la expresión (3.9), resultan ser:

$$\begin{array}{ll} \ell_0(x) = \frac{(x+7)(x+6)x}{(5+7)(5+6)5} & \ell_1(x) = \frac{(x-5)(x+6)x}{(-7-5)(-7+6)(-7)} \\ \ell_2(x) = \frac{(x-5)(x+7)x}{(-6-5)(-6+7)(-6)} & \ell_3(x) = \frac{(x-5)(x+7)(x+6)}{(0-5)(0+7)(0+6)} \end{array}$$

El polinomio de interpolación de Lagrange es:

$$p_3(x) = \ell_0(x) - 23\ell_1(x) - 54\ell_2(x) - 954\ell_3(x)$$

Interpolación de splines

Una *función spline* está formada por varios polinomios, cada uno definido sobre un subintervalo, que se unen entre sí obedeciendo a ciertas condiciones de continuidad.

Supongamos que disponemos de $n+1$ puntos, a los que denominaremos *nudos*, tales que $t_0 < t_1 < \cdots < t_n$. Supongamos además que se ha fijado un entero $k \geq 0$. Decimos entonces que una **función spline de grado** k con nudos en $t_0, t_1, \ldots, t_n$ es una función S que satisface las condiciones:

(i) en cada intervalo $[t_{i-1}, t_i)$, S es un polinomio de grado menor o igual a k.

(ii) S tiene una derivada de orden $(k-1)$ continua en $[t_0, t_n]$.

Los splines de grado 0 son funciones constantes por zonas. Una forma explícita de presentar un spline de grado 0 es la siguiente:

$$S(x) = \begin{cases} S_0(x) = c_0 & x \in [t_0, t_1) \\ S_1(x) = c_1 & x \in [t_1, t_2) \\ \vdots & \vdots \\ S_{n-1}(x) = c_{n-1} & x \in [t_{n-1}, t_n) \end{cases}$$

Los intervalos $[t_{i-1}, t_i)$ no se intersectan entre sí, por lo que no hay ambigüedad en la definición de la función en los nudos. Un spline de grado 1 se puede definir por:

$$S(x) = \begin{cases} S_0(x) = a_0 x + b_0 & x \in [t_0, t_1) \\ S_1(x) = a_1 x + b_1 & x \in [t_1, t_2) \\ \vdots & \vdots \\ S_{n-1}(x) = a_{n-1} x + b_{n-1} & x \in [t_{n-1}, t_n) \end{cases}$$

En las figuras (3.1) y (3.2) se muestran las gráficas correspondientes a los splines de grado cero y de grado 1 respectivamente.

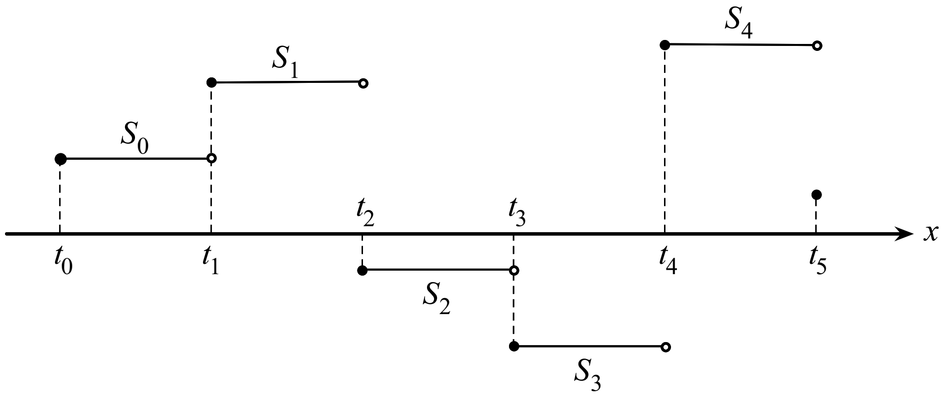

Figura 3.1: Spline de grado 0 con seis puntos.

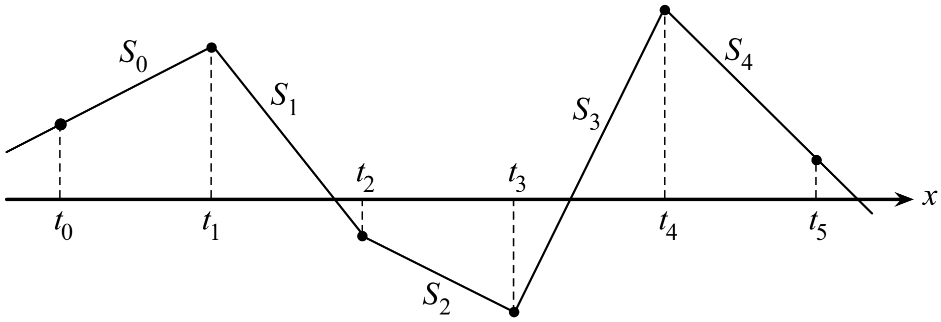

Figura 3.2: Spline de grado 1 con seis puntos.

Splines cúbicos

El spline cúbico ($k = 3$) es el spline más empleado, debido a que proporciona un excelente ajuste a los puntos tabulados y su cálculo no es excesivamente complejo.

Sobre cada intervalo $[t_0, t_1], [t_1, t_2], \ldots, [t_{n-1}, t_n]$, S está definido por un polinomio cúbico diferente. Sea S_i el polinomio cúbico que representa a S en el intervalo $[t_i, t_{i+1}]$, por tanto:

$$S(x) = \begin{cases} S_0(x) & x \in [t_0, t_1) \\ S_1(x) & x \in [t_1, t_2) \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [t_{n-1}, t_n) \end{cases}$$

Los polinomios S_{i-1} y S_i interpolan el mismo valor en el punto t_i, es decir, se cumple:

$$S_{i-1}(t_i) = y_i = S_i(t_i) \qquad (1 \leq i \leq n-1)$$

por lo que se garantiza que S es continuo en todo el intervalo. Además, se supone que S' y S'' son continuas, condición que se emplea en la deducción de una expresión para la función del spline cúbico.

Aplicando las condiciones de continuidad del spline S y de las derivadas primera S' y segunda S'', es posible encontrar la expresión analítica del spline. No vamos a obtener esta expresión, ya que su demostración queda fuera del ámbito de estos apuntes. Simplemente diremos que la expresión resultante es:

$$\begin{aligned} S_i(x) &= \tfrac{z_i}{6h_i}(t_{i+1}-x)^3 + \tfrac{z_{i+1}}{6h_i}(x-t_i)^3 \\ &\quad + \left(\tfrac{y_{i+1}}{h_i} + \tfrac{z_{i+1}h_i}{6}\right)(x-t_i) + \left(\tfrac{y_i}{h_i} - \tfrac{z_i h_i}{6}\right)(t_{i+1}-x) \end{aligned}$$

En la expresión anterior, $h_i = t_{i+1} - t_i$ y $z_0, z_1, \ldots, z_n$ son incógnitas. Para determinar sus valores, utilizamos las condiciones de continuidad que deben cumplir estas funciones. El resultado (que tampoco vamos a demostrar) es:

$$h_{i-1}z_{i-1} + 2(h_i + h_{i-1})z_i + h_i z_{i+1} = \frac{6}{h_{i-1}}(y_{i+1} - y_i) - \frac{6}{h_{i-1}}(y_i - y_{i-1})$$

La ecuación anterior, con $i = 1, 2, \ldots, n-1$ genera un sistema de $n-1$ ecuaciones lineales con $n+1$ incógnitas $z_0, z_1, \ldots, z_n$. Podemos elegir z_0 y z_n de forma arbitraria y resolver el sistema de ecuaciones resultante para obtener los valores de $z_1, z_2, \ldots, z_{n-1}$. Una elección especialmente adecuada es hacer $z_0 = z_1 = 0$. La función spline resultante se denomina *spline cúbico natural* y el sistema de ecuaciones lineal expresado en forma matricial es:

$$\begin{pmatrix} u_1 & h_1 & & & & \\ h_1 & u_2 & h_2 & & & \\ & h_2 & u_3 & h_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & & h_{n-2} & u_{n-1} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{pmatrix} = \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \vdots \\ \nu_{n-2} \\ \nu_{n-1} \end{pmatrix} \tag{3.10}$$

en donde:

$$\begin{aligned} h_i &= t_{i+1} - t_i \\ u_i &= 2(h_i + h_{i-1}) - \frac{h_{i-1}^2}{u_{i-1}} \\ b_i &= \frac{6}{h_i}(y_{i+1} - y_i) \\ \nu_i &= b_i - b_{i-1} - \frac{h_{i-1}\nu_{i-1}}{u_{i-1}} \end{aligned} \tag{3.11}$$

Este sistema de ecuaciones, que es tridiagonal, se puede resolver mediante eliminación gaussiana sin pivoteo como se muestra en la figura (3.3). El código acepta como entrada un conjunto de nodos (t_i) y el conjunto de los valores de la función correspondiente (y_i) y produce un vector con los vectores z_i. Por último, el valor del spline S en un punto x cualquiera interpolado se puede calcular de forma eficiente empleando la siguiente expresión:

$$S_i(x) = y_i + (x - t_i)\left[C_i + (x - t_i)\left[B_i + (x - t_i)A_i\right]\right] \tag{3.12}$$

en donde

$$\begin{aligned} A_i &= \frac{1}{6h_i}(z_{i+1} - z_i) \\ B_i &= \frac{z_i}{2} \\ C_i &= -\frac{h_i}{6}z_{i+1} - \frac{h_i}{3}z_i + \frac{1}{h_i}(y_{i+1} - y_i) \end{aligned} \tag{3.13}$$

```
input n,(t_i),(y_i)
for i = 0,1,...,n-1 do
      h_i ← t_{i+1} - t_i
      b_i ← 6(y_{i+1} - y_i)/h_i
end

u_1 ← 2(h_0 + h_1)
ν_1 ← b_1 - b_0
for i = 2,3,...,n-1 do
      u_i ← 2(h_i + h_{i-1}) - h_{i-1}^2/u_{i-1}
      ν_i ← b_i - b_{i-1} - h_{i-1}ν_{i-1}/u_{i-1}
end

z_n ← 0
for i = n-1,n-2,...,1 do
      z_i ← (ν_i - h_i z_{i+1})/u_i
end
z_0 ← 0
output (z_i)
```

Figura 3.3: Algoritmo para encontrar los coeficientes z_i de un spline cúbico.

Veamos un ejemplo para ilustrar el empleo de los splines cúbicos para interpolar los valores de una tabla. En la tabla (3.1) se muestran algunos valores de una serie de valores tabulados a intervalos regulares de la función $f(x) = \sqrt{x}$ en el intervalo [0,2.25]. También se indican los valores interpolados empleando el correspondiente spline cúbico así como el error absoluto cometido. Obsérvese que el error es cero para los nudos. En la figura (3.4) se representan gráficamente los valores tabulados.

En la figura (3.5) se muestra otro ejemplo. Se representan gráficamente los puntos interpolados mediante una función spline cúbica para la función $y = \mathrm{sen}(x)$.

x	$\sqrt{x}$	$S_i(x)$	$\|\varepsilon_a(x)\|$
0.0000	0.0000	0.0000	0.0000E+00
0.0625		0.1426	1.0732E-01
0.1250		0.2782	7.5266E-02
0.1875		0.3997	3.3261E-02
0.2500	0.5000	0.5000	0.0000E+00
0.3125		0.5744	1.5440E-02
0.3750		0.6285	1.6155E-02
0.4375		0.6701	8.6732E-03
0.5000	0.7071	0.7071	0.0000E+00
⋮	⋮	⋮	⋮
1.7500	1.3228	1.3228	0.0000E+00
1.8125		1.3462	6.8994E-07
1.8750		1.3693	5.9953E-06
1.9375		1.3919	8.7004E-06
2.0000	1.4142	1.4142	0.0000E+00
2.0625		1.4361	2.4522E-05
2.1250		1.4577	4.7329E-05
2.1875		1.4790	4.6215E-05
2.2500	1.5000	1.5000	0.0000E+00

Cuadro 3.1: Valores interpolados mediante un spline cúbico para la función $y = \sqrt{x}$ e indicación del error cometido (en valor absoluto).

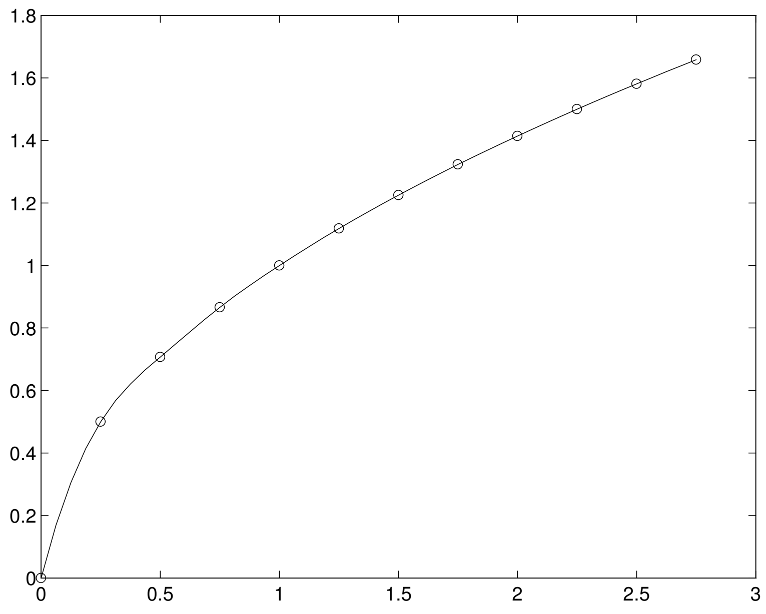

Figura 3.4: Representación de la función $y = \sqrt{x}$. Los círculos representan los valores tabulados de la función y la línea continua los puntos interpolados mediante una función spline cúbica.

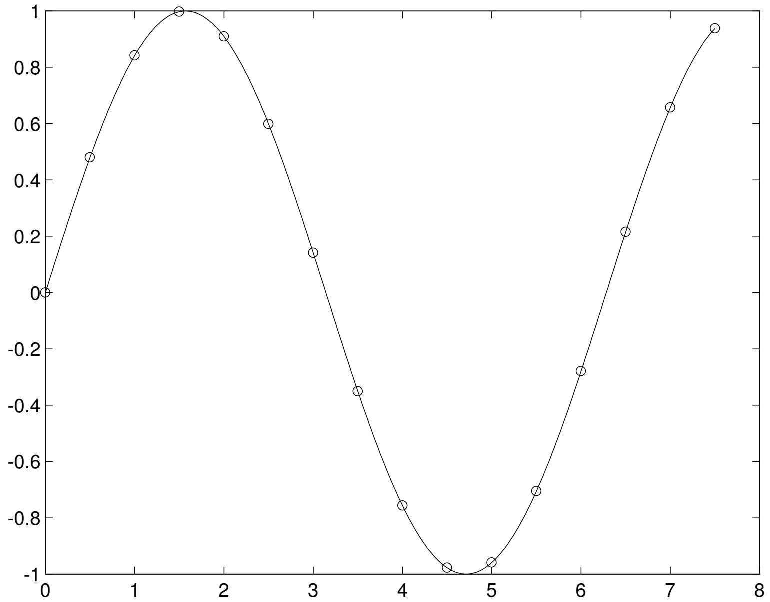

Figura 3.5: Representación de la función $y = \operatorname{sen}(x)$. Los círculos representan los valores tabulados de la función y la línea continua los puntos interpolados mediante una función spline cúbica.