TEMA 4: INTERPOLACIÓN

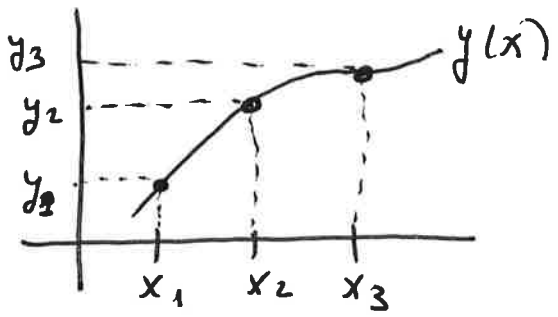
1. INTRODUCCIÓN

El problema de interpolación consiste en encontrar una función $y(x)$ que pasa por n-medidas $y_1, y_2, \dots, y_n$ en n-puntos $x_1, x_2, \dots, x_n$.

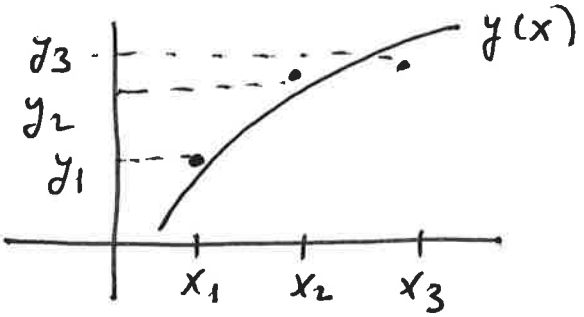
Es decir, se ha de cumplir que

$$y(x_i) = y_i, \quad 1 \le i \le n.$$

El problema de interpolación es diferente del problema de aproximación por mínimos cuadrados, donde no se requiere que la función de aproximación pase por los puntos seleccionados.



Interpolación



Aproximación por mínimos cuadrados.

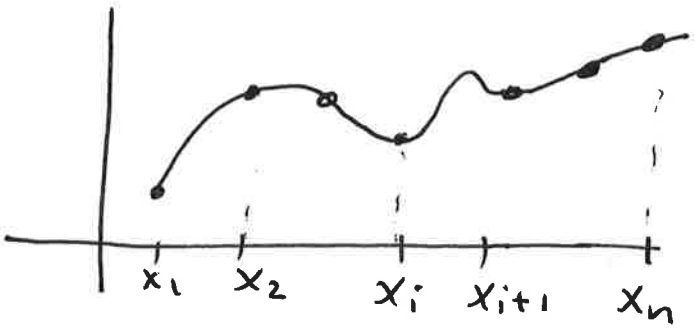
Típicamente, la función de interpolación es un polinomio de grado alto, o bien, un polinomio de grado bajo en cada intervalo $[x_i, x_{i+1}]$,

es decir,

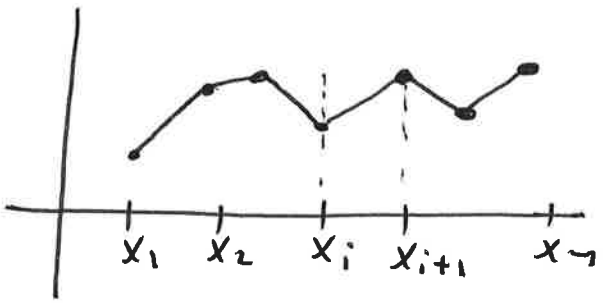
$$y(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$$

o

$$y(x) = y_i \frac{x - x_{i+1}}{x_i - x_{i+1}} + y_{i+1} \frac{x - x_i}{x_{i+1} - x_i}$$



polinomio de grado alto



polinomio de grado 1 en cada intervalo.

¡Los polinomios de grado alto no son estables!

Este tipo de interpolación con polinomios de grado alto es útil cuando los valores $y_1, y_2, \ldots, y_n$ provienen de funciones analíticas y los puntos $x_1, x_2, \ldots, x_n$ se eligen de forma adecuada.

Interpolación con polinomios de grado 1 en cada intervalo es completamente estable, pero la precisión puede dejar mucho que desear. Usaremos la interpolación con "splines", polinomios cúbicos (de grado 3) en cada intervalo para conseguir mejorar la precisión, en concreto continuidad de pendientes y momentos (derivadas segundas) en cada nodo (x_i, y_i).

¿Para qué necesitamos la interpolación?

1.) La gran mayoría de magnitudes de interés en Ingeniería se calculan por medio de integrales, las cuales, como bien sabemos, son difíciles de calcular analíticamente. La interpolación polinomial juega un papel también importante en la "integración numérica", que veremos en el próximo tema.

2) Cuando discretizamos una ecuación diferencial, calculamos el valor de la solución en una serie de nodos $(x_1, y_1), \ldots, (x_n, y_n)$.
La interpolación polinomial a trozos, en cada intervalo, es elemento esencial para estimar el valor de la solución en un punto distinto del de los nodos

$x_1, \ldots, x_n$;

2. Interpolación con polinomios de grado alto

Supongamos conocidos $n+1$ valores $y_0, y_1, \dots, y_n$ de una función $y(x)$ en los puntos $x_0, x_1, \dots, x_n$.

Buscamos un polinomio de grado n, ~~$p(x) = a_0 + a_1 x + \dots + a_i$~~

$$p(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

que interpole dichos valores, es decir,

$$p(x_j) = y_j, \quad j = 0, 1, \dots, n.$$

Por tanto, para encontrar los coeficientes $a_0, a_1, \dots, a_n$:

$$\left.\begin{array}{l} p(x_0) = y_0 \leftrightarrow a_0 + a_1 x_0 + \dots + a_n x_0^n = y_0 \\ p(x_1) = y_1 \leftrightarrow a_0 + a_1 x_1 + \dots + a_n x_1^n = y_1 \\ \quad\cdots\cdots \qquad\qquad \cdots \qquad\qquad \cdots \\ p(x_n) = y_n \leftrightarrow a_0 + a_1 x_n + \dots + a_n x_n^n = y_n \end{array}\right\},$$

sistema de ecuaciones lineales que, en forma matricial, se expresa como

$$\begin{bmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \qquad (*)$$

La matriz de este sistema

$$V = \begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \cdots & \cdots & \cdots & \cdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix}$$

se llama matriz de Vandermonde y cumple que su determinante es el producto de las diferencias $x_j - x_i$ entre los puntos de interpolación. Por tanto, si los puntos de interpolación son distintos, obtenemos una única solución $a_0, a_1, \ldots, a_n$ del sistema anterior, es decir, un único polinomio de interpolación

$$p(x) = a_0 + a_1 x + \cdots + a_n x^n.$$

Sin embargo, la obtención del polinomio $p(x)$ a través de la resolución del sistema de Vandermonde (*) no es la forma adecuada de proceder cuando el número de puntos de interpolación es grande. El motivo es que la matriz de Vandermonde V está muy mal condicionada.

Para una ilustración véase el código InterpolaVandermonde.m.

Buscamos una forma estable y rápida de calcular $p(x)$.

Supongamos en primer lugar que los puntos de interpolación son $y_i = 0$ excepto $y_j = p(x_j) = 1$, es decir,

$$p(x_i) = 0 \quad i \neq j$$
$$p(x_j) = 1 \quad i = j$$

$$p(x) = l_j(x) = \frac{(x-x_0)\cdots(x-x_j)\cdots(x-x_n)}{(x_j-x_0)\cdots(x_j-x_j)\cdots(x_j-x_n)}$$

Nótese que como

$$l_j(x) = \frac{(x-x_0)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)}{(x_j-x_0)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_n)}$$

se tiene que $l_j(x_i) = 0$ si $i \neq j$, $l_j(x_j) = 1$, con lo que $l_j(x)$ es el polinomio de interpolación buscado.

Por linealidad, el polinomio de interpolación para los valores $y_0, y_1, \ldots, y_n$ en los puntos $x_0, x_1, \ldots, x_n$ es

$$\boxed{p(x) = y_0 l_0(x) + y_1 l_1(x) + \cdots + y_n l_n(x),}$$

que es la llamada <u>fórmula de Lagrange</u> del polinomio de interpolación.

Nótese que para evaluar el polinomio de interpolación $p(x)$, mediante la fórmula de Lagrange, en un punto x fijo hemos de realizar n-multiplicaciones para cada $l_j(x)$. Como tenemos n términos $l_j(x)$, en total, el coste computacional es del orden de n^2, siendo n el nº de puntos de interpolación.

Otro inconveniente de la fórmula de Lagrange es que si se incorpora un nuevo punto de interpolación (x_{n+1}, y_{n+1}), entonces es preciso rehacer completamente todos los cálculos.

Veamos ahora que, manipulando adecuadamente la fórmula de Lagrange, podemos llegar a una nueva forma de calcular el polinomio de interpolación, que resulta mucho más eficiente desde un punto de vista computacional.

Para ello, consideremos el polinomio

$$L(x) = (x-x_0)(x-x_1)\cdots(x-x_n).$$

Nótese que $l_j(x)$ puede expresarse como

$$l_j(x) = \frac{\frac{L(x)}{x-x_j}}{L'(x_j)} = L(x)\frac{q_j}{x-x_j},$$

donde $q_j = L'(x_j)$.

Por tanto,

$$P(x) = y_0 l_0(x) + y_1 l_1(x) + \cdots + y_n l_n(x)$$

$$= y_0 L(x)\frac{q_0}{x-x_0} + y_1 L(x)\frac{q_1}{x-x_1} + \cdots + y_n L(x)\frac{q_n}{x-x_n}$$

$$= L(x)\left(\frac{q_0 y_0}{x-x_0} + \frac{q_1 y_1}{x-x_1} + \cdots + \frac{q_n y_1}{x-x_n}\right).$$

Por otra parte, si $y_j = 1 \ \forall j$, entonces el polinomio de interpolación asociado es constante e igual a 1, es decir,

$$1 = l_0(x) + l_1(x) + \cdots + l_n(x) = L(x)\left(\frac{q_0}{x-x_0} + \cdots + \frac{q_n}{x-x_n}\right).$$

Se tiene con ello que

$$\boxed{P(x) = \frac{P(x)}{1} = \frac{L(x)\left(\frac{q_0 y_0}{x-x_0} + \frac{q_1 y_1}{x-x_1} + \cdots + \frac{q_n y_n}{x-x_n}\right)}{L(x)\left(\frac{q_0}{x-x_0} + \frac{q_1}{x-x_1} + \cdots + \frac{q_n}{x-x_n}\right)}}$$

$$\boxed{P(x) = \frac{\frac{q_0 y_0}{x-x_0} + \frac{q_1 y_1}{x-x_1} + \cdots + \frac{q_n y_n}{x-x_n}}{\frac{q_0}{x-x_0} + \frac{q_1}{x-x_1} + \cdots + \frac{q_n}{x-x_n}}}$$

que es la llamada fórmula del baricentro.

Veamos ahora de qué orden es el coste computacional de la fórmula del baricentro.

Para evaluar $p(x)$ en un punto dado, usando la fórmula del baricentro, necesitamos:

$2n$ sumas
$n+1$ restas
$n+1$ ~~divisiones~~ multiplicaciones
$n+2$ divisiones

$5n+4$ = total de operaciones.

En resumen:

* Lagrange $\simeq n^2$
* Baricentro $= 5n+4$

Para hacernos una idea, si $n=100$,

* Lagrange $\simeq 10^4 = 10.000$
* Baricentro $= 504$

Además, si en la fórmula del ~~lagrange~~ baricentro incorporamos un nuevo punto de interpolación (x_{n+1}, y_{n+1}), entonces los pesos q_j tienen un nuevo factor y q_{n+1} tiene n factores. El coste computacional se incrementa en $2n$, que es bien poquito.

Aún siendo muy eficiente computacionalmente, la fórmula del baricentro, sin embargo, ésta puede ser completamente inútil, en el sentido de que los errores de interpolación sean grandes, si los puntos $x_0, x_1, \ldots, x_n$ no se eligen de forma adecuada. Este fenómeno fue ilustrado por Runge para el caso de la función

$$f(x) = \frac{1}{1+16x^2}$$

donde usando nodos equiespaciados en el intervalo $[-1, 1]$ se generan unas oscilaciones espúreas, en los extremos del intervalo, del polinomio de interpolación, da igual la forma en que éste sea calculado. Resulta que prácticamente en todos los problemas de Ingeniería para los cuales necesitamos hacer uso de la interpolación polinomial, tenemos libertad de elección para los nodos $x_0, x_1, \ldots, x_n$, y aunque, en principio elegirlos equiespaciados puede resultar muy atractivo, sin embargo, el fenómeno de Runge anterior nos advierte de que esta elección puede ser muy mala.

Hemos de concentrar nodos en los extremos del intervalo donde estemos interpolando. Supongamos que el intervalo es $[-1,1]$. Los mejores nodos de interpolación que podemos elegir son los llamados nodos de Chebyshev:

$$x_j = \cos\frac{j\pi}{n}, \quad j=0,1,\dots,n,$$

$n = n^{\underline{o}}$ de puntos.

Además, de regalo, los nodos de Chebyshev simplifican enormemente los pesos q_j en la fórmula del baricentro. En efecto, haciendo uso de las identidades trigonométricas que transforman sumas y en productos se puede probar que

$$q_0 = \frac{1}{2}, \quad q_j = (-1)^j, \quad j=1,\dots,n-1, \quad q_n = \frac{1}{2}.$$

Nota.- ¿Cómo se transforman los nodos de Chebyshev a cualquier otro intervalo $[a,b]$?

Por medio de una transformación lineal que pase $[-1,1]$ a $[a,b]$, en concreto, la transformación $T: [-1,1] \longrightarrow [a,b]$

$$x \longmapsto \frac{b-a}{2}x + \frac{a+b}{2}$$

$$[-1,1] \xrightarrow{T} [a,b] \xrightarrow{f} \mathbb{R}$$

$$x \longmapsto T(x) = y \longmapsto z = f(y)$$

$$z_j = f(y_j)$$

$$y_j = T(x_j) = \frac{b-a}{2} x_j + \frac{a+b}{2}$$

x_j = nodos de Chebyshev en $[-1,1]$

$P_n(x)$ polinomio que interpola los datos

$$(x_j,\ z_j = f(T(x_j))),\quad 0 \le j \le n.$$

$P_n(y)$ polinomio de interpolación en $[a,b]$

$$\boxed{P_n(y) = P_n(T(x))}$$

de modo que los nodos de Chebyshev en $[a,b]$ son

$$\bar{x}_j = \frac{b-a}{2} x_j + \frac{a+b}{2}$$

con $x_j = \cos \frac{j\pi}{n}$, los nodos en $[-1,1]$.

Nos ocupamos a continuación de la cuestión sobre la precisión - convergencia de la interpolación polinomial de Chebyshev.
Se tienen los dos resultados siguientes sobre la estimación de error:

- Convergencia polinomial: Sea $f:[a,b] \to \mathbb{R}$ de clase C^m y $P_n(x)$ su polinomio de interpolación asociado a $n+1$ nodos de Chebyshev. Entonces

$$|f(x) - P_n(x)| \leq \frac{C}{n^m} \quad \forall x \in [a,b].$$

es decir, el error de interpolación decrece como un polinomio de grado m respecto al nº de puntos de interpolación.

Convergencia exponencial. Sea $f:[-1,1]\to\mathbb{R}$ y supongamos que f admite una extensión analítica a la elipse de ecuación $\left(\frac{x}{a}\right)^2+\left(\frac{y}{b}\right)^2=1$. Sea $r=a+b$ y supongamos que $r>1$. Entonces

$$|f(x)-p_n(x)|\leq C\, e^{-n\log r} \qquad \forall\, x\in[-1,1].$$

con $n+1$ - puntos de interpolación de Chebyshev.

RESUMEN Y RECOMENDACIÓN PRÁCTICA

La interpolación polinomial p con polinomios de grado alto se ha de hacer usando la fórmula del baricentro y los nodos de Chebyshev.

Si la función f a interpolar es analítica, entonces pocos nodos de interpolación son suficientes porque la convergencia es exponencial. Conforme disminuye la regularidad de f, en términos de cuántas veces es derivable, se ha de usar un número mayor de nodos de Chebyshev porque la convergencia es polinomial.

EL MISTERIO DE LOS NODOS DE CHEBYSHEV

Resulta un tanto misterioso el hecho de que los nodos de Chebyshev tengan un comportamiento tan excepcional en la interpolación polinomial. Más sorprendente es, en principio, el hecho de que los nodos de Chebyshev se distribuyen en el intervalo $[-1,1]$ de igual

forma a como lo hacen las cargas eléctricas en un cable cuando se repelen con fuerza $\frac{1}{\text{distancia}}$,

lo que muestra una clara conexión de la interpolación Chebyshev con la teoría de potenciales de campos eléctricos. ¿De dónde procede esta conexión? No es fácil explicarlo, pero daremos alguna idea intuitiva. Tras unos cálculos matemáticos, más o menos sencillos, se prueba que el error de interpolación se expresa como

$$|f(x) - P_n(x)| = \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \right|.$$

Típicamente $|f^{(n+1)}(\xi)| \leq C$ con lo que el error está dominado por $\prod_{i=0}^{n} (x - x_i) := q(x)$.

Así, $|q(x)| = \prod_{i=0}^{n} |x - x_i|$, con lo que tomando logaritmos, $\log |q(x)| = \sum_{i=0}^{n} \log |x - x_i|$

Si pasamos al plano complejo, cambiando x por z y x_i por z_i, y denotamos por

~~$\Phi_N(z) = \frac{1}{N}$~~

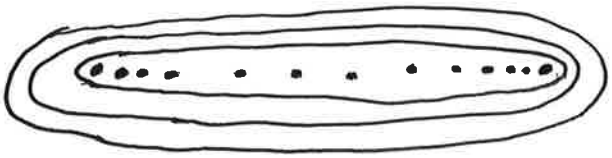
$$\Phi_n(z) = \frac{1}{n} \sum_{i=0}^{n} \log |z - z_i|$$

entonces esta función es armónica, es decir, solución de la ecuación de Laplace (o ecuación del potencial) lo que indica que $\phi_n(z)$ es el potencial eléctrico en z debido a las cargas z_i con potencial $\frac{1}{n} \log|z - z_i|$.

Por tanto,

$$|q(z)| = e^{n\,\phi_n(z)}.$$

Esta fórmula, que recordemos mide el error de interpolación, nos dice que si $\phi_n(z)$ varía mucho, entonces $q(z)$ también. Es lo que sucede cuando las cargas están uniformemente distribuidas (= nodos de interpolación aquiespaciados). Sin embargo, cuando las cargas se distribuyen con densidad $\frac{1}{\text{distancia}}$, caso Chebyshev, entonces se puede probar que $\phi_n(z) \approx \frac{1}{n} \log\left|\frac{z - \sqrt{z^2-1}}{2}\right|$, lo que da elipses para las líneas equipotenciales. Entonces, $|q(x)| \tilde{\approx} e^{n\,\phi(x)} = 2^{-n} = e^{-n \log 2}$, es decir, convergencia exponencial



Nodos Chebyshev versus líneas equipotenciales de $\phi(z)$.

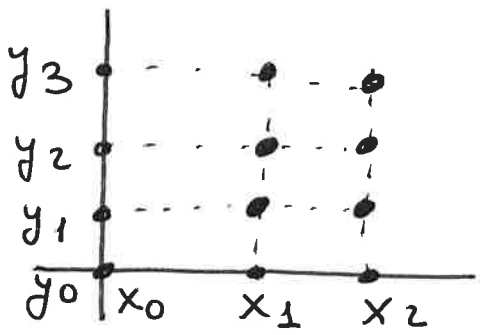
3. Interpolación con Polinomios de Varias Variables

Supongamos conocidos valores f_{ij}, de una determinada función $f(x,y)$ en el conjunto de nodos

$$(x_i, y_j), \quad 0 \le i \le N_x, \quad 0 \le j \le N_y.$$

Buscamos un polinomio $p(x,y)$ que pase por f_{ij} en (x_i, y_j), es decir,

$$p(x_i, y_j) = f_{ij}, \quad 0 \le i \le N_x, \quad 0 \le j \le N_y.$$



nodos (x_i, y_j)

Una forma sencilla, aunque costosa computacionalmente, de construir este polinomio es calculando el llamado producto tensorial de polinomios interpoladores en una dimensión. Concretamente, sean

$$l_i^x(x) = \frac{(x-x_0)\cdots(x-x_{i-1})\cdot(x-x_{i+1})\cdots(x-x_{N_x})}{(x_i-x_0)\cdots(x_i-x_{i-1})\cdot(x_i-x_{i+1})\cdots(x_i-x_{N_x})}$$

con $0 \le i \le N_x$. Nótese que

$$l_i^x(x_j) = \begin{cases} 1, & \text{si } i=j \\ 0, & \text{si } i \neq j. \end{cases}$$

y también

$$l_i^y(y) = \frac{(y-y_0)\cdots(y-y_{i-1})(y-y_{i+1})\cdots(y-y_{N_y})}{(y_i-y_0)\cdots(y_i-y_{i-1})(y_i-y_{i+1})\cdots(y_i-y_{N_y})}$$

$0 \le i \le N_y$, que cumplen

$$l_i^y(y_j) = \begin{cases} 1 & \text{si } i=j \\ 0 & \text{si } i \neq j \end{cases}$$

El polinomio de interpolación buscado es

$$p(x,y) = \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} f_{ij}\, l_i^x(x)\, l_j^y(y)$$

El mismo procedimiento es aplicable en dimensión 3 y superiores. Por ejemplo, en dimensión 3:

$$p(x,y,z) = \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} \sum_{k=0}^{N_z} f_{ijk}\, l_i^x(x)\, l_j^y(y)\, l_k^z(z)$$

4. INTERPOLACIÓN CON POLINOMIOS DE ORDEN BAJO.

EL CASO DE LOS SPLINES CÚBICOS

Supongamos dados $n+1$ valores o medidas

$y_0, y_1, \dots, y_n$ en $n+1$ puntos o nodos

$x_0, x_1, \dots, x_n$.

Buscamos una función de interpolación $y(x)$, que por tanto ha de cumplir $y(x_i) = y_i$, $0 \le i \le n$, y de modo que $y(x)$, en cada intervalo $[x_i, x_{i+1}]$, sea un polinomio de grado bajo. En concreto:

- Si suponemos que $y(x)$ es un polinomio de grado 1 en $[x_i, x_{i+1}]$, se dice que la interpolación es de Lagrange.
- Si $y(x)$ es un polinomio de grado 3 en cada intervalo $[x_i, x_{i+1}]$ y además exigimos continuidad de $y'(x)$ y de $y''(x)$ en cada nodo interior x_i, $1 \le i \le n-1$, entonces se dice que la interpolación es con splines cúbicos.

Nos centramos a continuación en la interpolación con splines, introducida en el campo de la Ingeniería Naval. Spline significa "lengüeta" por la forma

en que estos polinomios de interpolación se asemejan a una lengüeta.

Veamos cómo calcularlos.

Denotemos por $h_i = x_{i+1} - x_i$ y por $S_i(x)$ la restricción del spline interpolador $S(x)$ al intervalo $[x_i, x_{i+1}]$.

Como $S_i(x)$ es un polinomio de grado 3, su derivada segunda es un polinomio de grado 1, es decir, un segmento de recta. Por tanto, si denotamos por $z_i = S_i''(x_i)$, $z_{i+1} = S_i''(x_{i+1})$, se tiene

$$S_i''(x) = z_{i+1} \frac{x - x_i}{h_i} + z_i \frac{x_{i+1} - x}{h_i}.$$

Integrando se llega a

$$S_i'(x) = z_{i+1} \frac{(x-x_i)^2}{2h_i} + z_i \frac{(x_{i+1}-x)^2}{2h_i} + E_i$$

e integrando nuevamente

$$S_i(x) = z_{i+1} \frac{(x-x_i)^3}{6h_i} + z_i \frac{(x_{i+1}-x)^3}{6h_i} + E_i x + F_i.$$

El término lineal $E_i x + F_i$ se puede escribir en la forma

$$E_i x + F_i = C_i (x - x_i) + D_i (x_{i+1} - x).$$

En efecto:

$$E_i x + F_i = C_i x - C_i x_i + D_i x_{i+1} - D_i x$$

$$= (C_i - D_i) x - C_i x_i + D_i x_{i+1}$$

$$\left.\begin{aligned} E_i &= C_i - D_i \\ F_i &= -C_i x_i + D_i x_{i+1} \end{aligned}\right\} \Rightarrow D_i = \frac{E_i x_i + F_i}{h}$$

$$C_i = E_i + D_i$$

Por tanto:

$$S_i(x) = z_{i+1} \frac{(x-x_i)^3}{6h_i} + z_i \frac{(x_{i+1}-x)^3}{6h_i} + C_i(x-x_i) + D_i(x_{i+1}-x)$$

- Imponemos ahora las condiciones de interpolación:

$$y_i = S_i(x_i) \Leftrightarrow y_i = z_i \frac{h_i^2}{6} + D_i h_i$$

$$\Rightarrow D_i = \frac{y_i}{h_i} - \frac{z_i h_i}{6}$$

$$y_{i+1} = S_i(x_{i+1}) \Leftrightarrow y_{i+1} = z_{i+1} \frac{h_i^2}{6} + C_i h_i$$

$$C_i = \frac{y_{i+1}}{h_i} - \frac{z_{i+1} h_i}{6}$$

quedando $S_i(x)$ escrito como:

$$S_i(x) = z_{i+1} \frac{(x-x_i)^3}{6h_i} + z_i \frac{(x_{i+1}-x)^3}{6h_i} + \left(\frac{y_{i+1}}{h_i} - \frac{z_{i+1} h_i}{6}\right)(x-x_i)$$

$$+ \left(\frac{y_i}{h_i} - \frac{z_i h_i}{6}\right)(x_{i+1}-x)$$

• Imponemos ahora las condiciones de continuidad de las derivadas primeras, es decir, $S_{i-1}'(x_i) = S_i'(x_i)$, $i = 1, 2, \dots, n-1$:

$$S_{i-1}'(x) = z_i \frac{(x-x_{i-1})^2}{2h_{i-1}} - z_{i-1}\frac{(x_i-x)^2}{2h_{i-1}} + \frac{y_i}{h_{i-1}} - \frac{z_i h_{i-1}}{6} - \frac{y_{i-1}}{h_{i-1}} + \frac{z_{i-1}h_{i-1}}{6}$$

$$S_i'(x) = z_{i+1}\frac{(x-x_i)^2}{2h_i} - z_i\frac{(x_{i+1}-x)^2}{2h_i} + \frac{y_{i+1}}{h_i} - \frac{z_{i+1}h_i}{6} - \frac{y_i}{h_i} + \frac{z_i h_i}{6}$$

$$S_{i-1}'(x_i) = S_i'(x_i)$$

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

$$\frac{z_{i-1}h_{i-1}}{6} + z_i\frac{h_{i-1}+h_i}{3} + \frac{z_{i+1}h_i}{6} = -\frac{y_i+y_{i-1}}{h_{i-1}} + \frac{y_{i+1}-y_i}{h_i} \;;$$

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

$i = 1, 2, \dots, n-1.$

Tenemos pues un sistema tridiagonal de $n-1$ ecuaciones con $n+1$ incógnitas $z_0, z_1, \dots, z_n$.

El cierre del sistema (mismo número de ecuaciones que de incógnitas) se suele hacer de dos formas:

1º) Fijar $z_0 = z_{n+1} = 0$. La función spline de Son los actual splines o splines naturales Matlab/Octave hace eso.

2º) Condición "not-a-knot": se impone que el spline $S(x)$ cumpla

$$y\left(\frac{x_0+x_1}{2}\right) = \frac{y_0+y_1}{2}, \quad y\left(\frac{x_{n-1}+x_n}{2}\right) = \frac{y_{n-1}+y_n}{2}.$$

Esta última posibilidad está implementada en el Toolbox Spline de MatLab.

Nota.- En el caso de nodos equiespaciados, el sistema lineal para calcular $z_1, z_2, \ldots, z_{n-1}$ es

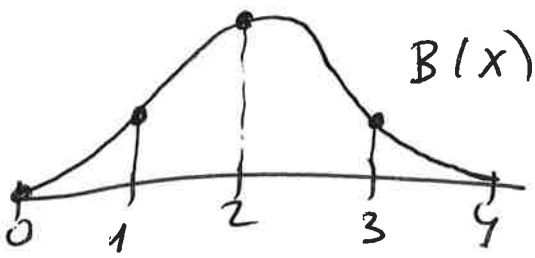
$$h(z_{i-1} + 4z_i + z_{i+1}) = 3y_{i+1} - 3y_{i-1}.$$

Nota.- Nótese que la condición de continuidad $S_{i-1}''(x_i) = S_i''(x_i)$ se cumple por construcción.

Otra forma de construir los polinomios splines de interpolación es a partir de una base. Son los llamados B-splines, que se construyen del siguiente modo:

1º) Se consideran los nodos $(0, 1, 2, 3, 4)$ a los que se asignan los valores $\frac{1}{6}(0, 1, 4, 1, 0)$ y las pendientes $\frac{1}{2}(0, 1, 0, -1, 0)$.

El spline correspondiente $B(x)$, que cumple el sistema de ecuaciones lineal correspondiente, y las condiciones $B(0) = B(4) = 0$ tiene la forma



Este B-spline produce la base requerida simplemente trasladando su gráfica de modo que en $[0,4]$ la base resultante es

$$\{B(x-3),\ B(x-2),\ B(x-1),\ B(x),\ B(x+1),\ B(x+2),\ B(x+3)\}.$$

Este tipo de implementación es la que suele estar implementada en el software de Cálculo Científico ~~pesar~~ en Ingeniería.

De esta forma, cualquier spline cúbico en $[0,4]$ es combinación lineal de la base anterior, es decir,

$$s(x) = c_3 B(x-3) + c_2 B(x-2) + c_1 B(x-1) + c_0 B(x) + c_1 B(x+1) + c_2 B(x+2) + c_3 B(x-3).$$

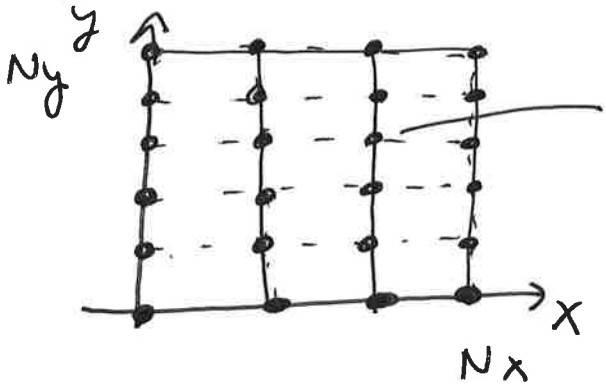
El mismo procedimiento se aplica para cualquier conjunto de n-puntos y también a dimensiones superiores.

Veamos con más detalle cómo proceder en el caso de la dimensión 2, donde el objetivo es generar una superficie de splines a partir de una nube de puntos. Pero antes, repasemos con un poco más de detalle los comentarios anteriores sobre B-splines.

EXTENSIÓN A LA DIMENSIÓN 2

Consideremos los datos de interpolación 2d:

$$\{x_i, y_j, f_{ij}\}_{i=1,j=1}^{N_x \; N_y}$$



$$f_{ij} = f(x_i, y_j).$$

Construimos ~~pas~~ bases de B-splines en las dos direcciones espaciales

$$\{B_i(x)\}_{i=1}^{N_x+2}, \quad \{\bar{B}_j(y)\}_{j=1}^{N_y}$$

y construimos el llamado "producto tensorial" de dichas bases, es decir, la base

$$\{B_i(x)\bar{B}_j(y)\}_{i=1,j=1}^{N_x \; N_y}$$

que tiene $N_x * N_y$ elementos. Por tanto, necesitamos $N_x \cdot N_y$ datos f_{ij} para poder calcular el polinomio interpolador tipo spline

$$s(x,y) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} s_{ij} B_i(x) \bar{B}_j(y).$$

Los coeficientes s_{ij} se determinan resolviendo el sistema

$$f_{ij} = s(x_i, y_j) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} s_{ij} B_i(x_i) \overline{B}_j(y_j).$$

La gráfica de la función $s(x,y)$, es decir, $z = s(x,y)$ proporciona una aproximación suave de superficies de interés en Ingeniería a partir de datos puntuales

$$(x_i, y_j, f_{ij}).$$

En resumen, los splines $B_i(x)$ juegan el mismo papel que los polinomios de Lagrange $l_i(x)$. Nos permiten interpolar funciones de varias variables "tensorizando" las bases (B_i ó l_i) 1d.

Repasemos un poco más en detalle, todo este asunto.

Resumiendo, a partir de un conjunto de datos (x_i, y_i), $0 \le i \le n$, podemos construir una función de interpolación $s(x)$ que sobre cada intervalo $[x_i, x_{i+1}]$ sea un polinomio de grado menor o igual a 3 y de forma que se garantice continuidad de $s(x)$, de $s'(x)$ y de $s''(x)$ en los nodos interiores x_j, $1 \le j \le n-1$. Para el caso de nodos equiespaciados, con $x_{i+1} - x_i = h$, hemos de resolver el sistema de $n-1$ ecuaciones con $n-1$ incógnitas

$$z_{i-1} + 4z_i + z_{i+1} = \frac{3y_{i+1} - 3y_{i-1}}{h}, \quad 1 \le i \le n-1,$$

donde $z_0 = z_n = 0$, en el caso de los splines naturales. Nótese que la matriz del sistema anterior es estrictamente diagonal dominante, lo que es una muy buena noticia desde el punto de vista computacional, ya que esa condición garantiza la convergencia de los métodos iterativos de Jacobi y Gauss-Seidel.

El sistema anterior tiene $n-1$ incógnitas z_i. Eso nos ~~dicho~~ da $n-1$ grados de libertad a la hora de calcular el spline $s(x)$. Recordemos que otros dos grados de libertad provienen del cálculo de las

constantes $D_i = \frac{y_i}{h} - \frac{z_i \cdot h}{6}$

$$C_i = \frac{y_{i+1}}{h} - \frac{z_{i+1}\, h}{6}$$

a través de las condiciones de interpolación. Tenemos pues

$n - 1 + 2 = n + 1$ grados de libertad, ~~es decir, el~~

que ~~con~~ coincide con el número de nodos x_i, $0 \le i \le n$.

Si liberamos las condiciones $z_0 = z_n = 0$ impuestas sobre los extremos, tenemos dos nuevos grados de libertad.

En total, $n+3$ grados de libertad.

Es decir, tenemos $n+3$ grados de libertad para construir un spline cúbico sobre $n+1$ nodos.

Desde el punto de vista del Álgebra Lineal diríamos que el espacio vectorial de splines cúbicos sobre $n+1$ nodos ~~es~~ tiene dimensión $n+3$.

Otra forma de ver esta dimensión es teniendo en cuenta que para construir un polinomio de grado 3 en un intervalo necesitamos 4 números (grados de libertad). Si tenemos $n+1$ nodos, eso nos da n intervalos. Por tanto, construimos $S(x)$ a partir de n polinomios cúbicos, lo que nos da

$4 \cdot n$ grados de libertad.

Dado que tenemos 3 condiciones (continuidad de $s(x)$, de $s'(x)$ y de $s''(x)$) en los nodos interiores

x_i, $1 \le i \le n-1$, a satisfacer, esto reduce en

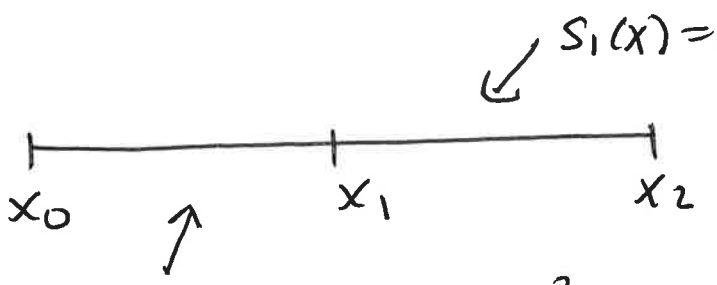
$3 \cdot (n-1)$ los grados de libertad,

resultando un número total de grados de libertad de

$$4n - 3(n-1) = n+3 = \underbrace{(n+1)}_{\text{nº de nodos}} + 2 \quad ,$$

es decir, número de nodos +2.

Veamos un ejemplo sencillo de 3 nodos.



Incógnitas (= grados de libertad) = 8, que son

$a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3$.

Condiciones de continuidad en el nodo x_1:

$$\left.\begin{array}{ll} S_0(x_1) = S_1(x_1) \Rightarrow & a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 = b_0 + b_1 x_1 + b_2 x_1^2 + b_3 x_1^3 \\ S_0'(x_1) = S_1'(x_1) \Rightarrow & a_1 x_1 + 2a_2 x_1 + 3a_3 x_1^2 = b_1 + 2b_2 x_1 + 3b_3 x_1^2 \\ S_0''(x) = S_1''(x_1) \Rightarrow & 2a_2 + 6a_3 x_1 = 2b_2 + 6b_3 x_1 \end{array}\right\}$$

Del sistema anterior podemos "despejar", por ejemplo, b_3, b_2, b_1 en función ~~de los anteriores~~ resto,

es decir, eliminamos 3 grados de libertad, con lo que nos quedan $8-3 = 5 = \underset{\text{nº de nodos}}{3} + 2$.

En resumen, la "base de splines" para $n+1$ nodos tiene $n+3$ elementos.

La construcción del spline $S(x)$, sobre $n+1$-nodos, usando una base de B-splines, resulta muy conveniente, sobre todo, cuando se quieren construir "superficies de splines" a partir de una malla de nodos bi-dimensional (x_i, y_j), $0 \le i \le N_x$, $0 \le j \le N_y$.

El B-spline, $B(x)$, construido anteriormente, sirve de pilar fundamental para construir una base de B splines, simplemente, trasladándolo a izquierda y derecha a lo largo de los nodos. No es difícil ver que estos $B(x-x_j)$ son linealmente independientes y por tanto, forman una base del correspondiente espacio de splines cúbicas sobre los nodos considerados.

Un último aspecto que no debemos olvidar es el de la "precisión" que proporciona la interpolación con splines cúbicas. Como es fácil de intuir, los splines cúbicos aproximan con

exactitud polinomios de grado 3. El error empieza con la potencia 4. De ello se puede deducir que para nodos equi-espaciados $x_{i+1} - x_i = h$, el error de interpolación es del orden de h^4, es decir,

$$|f(x) - s(x)| \leq C\, h^4 \quad , \quad \forall\, x \in [a,b]$$

con f la función a aproximar y $s(x)$ el spline cúbico usado en la interpolación. La constante C es una cota de $f^{(4)}(x)$ en $[a,b]$.

Se concluye que una función de clase C^4 se puede aproximar tanto como se quiera por un spline cúbico.

Como nota histórica, decir que los splines fueron introducidos por el matemático de origen rumano Isaac J. Schoenberg en la década de 1940.