

Ampliación de interpolación con Splines

Miguel Anguita Ruiz Pablo Baeyens Fernández Pablo David Medina Sánchez
Rubén Morales Pérez Francisco Javier Morales Piqueras

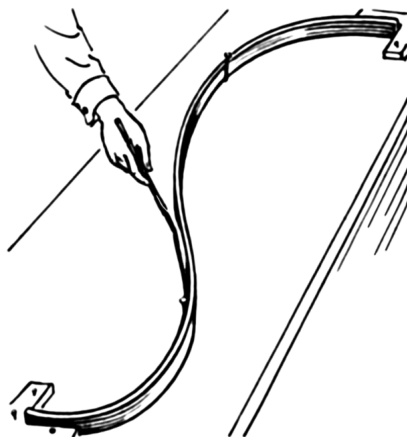
Índice

1. Splines cuadráticos	2
1.1. Introducción a los splines	2
1.2. Descripción del espacio de splines cuadráticos	2
1.3. Interpolación con splines cuadráticos	3
1.3.1. Método local: cálculo trozo a trozo	3
1.3.2. Método global: cálculo con una base de potencias truncadas	4
1.4. Error en los splines cuadráticos	5
1.5. Ejemplos	5
2. Splines cúbicos	8
2.1. Construcción a partir de los valores de s'' en los nodos $\{x_i\}$	9
2.2. Propiedades de minimización	11
2.3. Error en los splines cúbicos	13
2.4. Ejemplos	13
3. Implementación en ordenador: Octave	15
3.1. Spline Lineal	15
3.2. Splines cuadráticos	15
3.3. Splines cúbicos	16
3.3.1. Spline sujeto	16
3.3.2. Spline natural	17
3.3.3. Spline periódico	17
A. Definiciones y notación	19
B. Bibliografía	19

1. Splines cuadráticos

1.1. Introducción a los splines

La palabra **spline** con el tiempo se usó para referirse a una larga banda flexible generalmente de metal, que podía usarse para dibujar curvas continuas suaves, forzando a la banda a pasar por puntos específicos y trazados a lo largo de dicha curva.



La formalización del concepto de función spline, es decir, una curva continua que pasa por ciertos puntos se resume en la siguiente definición:

Definición. Sea $[a, b]$ un intervalo, $P = \{x_i\}_{i=0 \dots n} \in \mathcal{P}([a, b])$, $k, r \in \mathbb{N}$, $r < k$. Se dice que $s : [a, b] \rightarrow \mathbb{R}$ es un spline si $s \in C^r([a, b])$ y para todo $1 \leq i \leq n$, $s|_{[x_{i-1}, x_i]} \in \mathbb{P}_k$. $S_k^r(P)$ es el espacio de dichas funciones.

1.2. Descripción del espacio de splines cuadráticos

Partimos de $[a, b]$ un intervalo y $P \in \mathcal{P}([a, b])$. En esta primera sección nos centramos en los splines cuadráticos: los pertenecientes a $S_2^1(P)$.

Sus trozos son polinomios de grado menor o igual que 2 de la forma $ax^2 + bx + c$. Además son funciones de clase 1 (derivables en $[a, b]$ con derivada continua), lo que proporciona unas condiciones interesantes para resolver problemas de interpolantes.

Veamos algunas propiedades:

Proposición. Sea $[a, b]$ intervalo, $P = \{x_i\}_{i=0 \dots n} \in \mathcal{P}([a, b])$, entonces $\dim(S_2(P)) = n + 2$.

Demostración. Sea $s \in S_2(P)$.

- Para cada intervalo $[x_{i-1}, x_i]$ $s|_{[x_{i-1}, x_i]}(x) = ax^2 + bx + c$ para ciertos $a, b, c \in \mathbb{R}$. Por lo tanto cada trozo está determinado por 3 parámetros. Con n trozos tenemos $3n$ parámetros en total.
- Si imponemos la continuidad y derivabilidad en los extremos tenemos que

$$s_i(x_i) = s_{i+1}(x_i) \quad s'_i(x_i) = s'_{i+1}(x_i)$$

para todo $i = 1 \dots n - 1$. De cada condición se obtienen $n - 1$ ecuaciones, por lo tanto obtendremos: $n - 1 + n - 1 = 2n - 2$ ecuaciones linealmente independientes.

Por lo tanto, $\dim(S_2(P)) = 3n - (2n - 2) = n + 2$. □

Con el conocimiento de la dimensión del espacio podemos describir una base del espacio de splines cuadráticos con el uso de potencias truncadas. Una **base del espacio** es:

$$\{1, x, x^2, (x - x_1)_+^2, \dots, (x - x_{n-1})_+^2\}$$

1.3. Interpolación con splines cuadráticos

1.3.1. Método local: cálculo trozo a trozo

El problema que debemos resolver es el siguiente:

Problema. Sea $[a, b]$ intervalo, $P \in \mathcal{P}([a, b])$ partición. Hallar $s \in S_2(P)$ tal que:

$$s(x_i) = y_i \quad i = 0, 1, \dots, n$$

$$s'(x_k) = d_k$$

Es decir, sabemos los valores de la función en todos los nodos y el valor de la derivada en el nodo k .

Solución. Si $k > 0$, para calcular s_k podemos calcular la tabla de diferencias divididas:

x	y	DD1	DD2
x_{k-1}	y_{k-1}		
x_k	y_k	p_k	
x_k	y_k	d_k	$\frac{d_k - p_k}{h_k}$

De esta forma, s_k queda, para $x \in [x_{k-1}, x_k]$

$$s_k(x) = y_{k-1} + p_k(x - x_{k-1}) + \frac{d_k - p_k}{h_k}(x - x_{k-1})(x - x_k) \quad (1)$$

Conocida la expresión de s_k podemos calcular $d_{k-1} = s'_k(x_{k-1})$, y repetir este proceso para calcular s_{k-1} , hasta llegar a $k = 0$.

Si $k < n$, debemos calcular s_{k+1} . Como sabemos la derivada d_k , calculamos la tabla de diferencias divididas:

x	y	DD1	DD2
x_k	y_k		
x_k	y_k	d_k	
x_{k+1}	y_{k+1}	p_{k+1}	$\frac{p_{k+1}-d_k}{h_k}$

De esta forma, s_{k+1} queda para $x \in [x_k, x_{k+1}]$ de la siguiente forma:

$$s_{k+1}(x) = y_k + d_k(x - x_k) + \frac{p_{k+1} - d_k}{h_{k+1}}(x - x_k)(x - x_k) \quad (2)$$

El método queda entonces de la siguiente forma:

1. Para i desde k hasta 0:
 - a) Calculamos d_i , (conocida en el primer caso) haciendo $d_i = s'_{i+1}(x_i)$.
 - b) Aplicamos la fórmula (1) para calcular s_i .
2. Para i desde $k + 1$ hasta n :
 - a) Calculamos d_i haciendo $d_i = s'_{i-1}(x_i)$
 - b) Aplicamos la fórmula (2) para calcular s_i .

1.3.2. Método global: cálculo con una base de potencias truncadas

Para este método usaremos esta base del espacio vectorial $S_2(P)$:

$$\{1, x, x^2, (x - x_1)_+^2, \dots, (x - x_{n-1})_+^2\}$$

Tenemos los siguientes matrices y vectores:

- G : matriz de Gram. Evaluamos los elementos de la base en todos los nodos.
- X : vector de coeficientes
- b : vector con los valores que queremos interpolar.

De esta forma, deberíamos resolver el sistema $GX = b$.

Si notamos por x_k el nodo en el que conocemos la derivada y d_k la derivada en el nodo, el sistema queda:

$$\begin{pmatrix} 1 & x_0 & x_0^2 & 0 & \cdots & 0 \\ 1 & x_1 & x_1^2 & (x_1 - x_1)_+^2 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \cdots & \vdots \\ \vdots & & \vdots & \vdots & \cdots & \vdots \\ 1 & x_n & x_n^2 & (x_n - x_1)_+^2 & \cdots & (x_n - x_{n-1})_+^2 \\ 0 & 1 & 2x_k & 2(x_k - x_1)_+ & \cdots & 2(x_k - x_{n-1})_+ \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ \alpha \\ \vdots \\ \omega \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_n \\ d_k \end{pmatrix}$$

Es una matriz escalonada ya que las potencias truncadas serán 0 antes del nodo que las define. Finalmente la solución sería, para $x \in [a, b]$:

$$s(x) = a + bx + cx^2 + \alpha(x - x_1)_+^2 + \cdots + \omega(x - x_{n-1})_+^2$$

1.4. Error en los splines cuadráticos

Teorema. Sean $f \in C^2([a, b])$, $\{x_i\}_{i=0 \dots n} \in \mathcal{P}([a, b])$, $s \in S_2^1(\{x_i\}_{i=0, \dots, n})$ spline para f , $h = \max\{x_i - x_{i-1}\}_{i=1 \dots n}$, $E = f - s$. Además, sea $M > 0$ tal que:

$$M \geq \sup\{|f''(x) - f''(y)| : |x - y| \leq h, x, y \in [a, b]\}$$

Entonces, se verifica, para todo $x \in [a, b]$:

$$E(x) \leq \frac{h^2 M}{2} \quad (3)$$

La demostración, así como cotas para las derivadas y cotas más precisas en función de la localización de x puede encontrarse en *Quadratic Interpolatory Splines*, W. Kammerer, G. Reddien y R.S. Varga, (1973).

1.5. Ejemplos

Problema. Dados los datos de la tabla, halla mediante el método global el spline cuadrático que interpole los nodos x_i con $i = 0, \dots, 3$ y cuya derivada en x_1 sea 4.

$$\begin{array}{c|cccc} x_i & 2 & 4 & 5 & 8 \\ y_i & 7 & 3 & 5 & 5 \\ d_i & & & 4 & \end{array}$$

Solución. Debemos hallar $s \in S_2(P)$ con $a, b, c, \alpha, \beta \in \mathbb{R}$ tales que, para $x \in [2, 8]$:

$$s(x) = a + bx + cx^2 + \alpha(x - 4)_+^2 + \beta(x - 5)_+^2$$

Planteamos el sistema de ecuaciones $GX = b$:

$$\begin{pmatrix} 1 & 2 & 4 & 0 & 0 \\ 1 & 4 & 16 & 0 & 0 \\ 1 & 5 & 25 & 0 & 0 \\ 1 & 8 & 64 & 16 & 9 \\ 0 & 1 & 8 & 0 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 7 \\ 3 \\ 5 \\ 5 \\ 4 \end{pmatrix}$$

Resolviendo el sistema, obtenemos la solución $a = 35, b = -20, c = 3, \alpha = -5, \beta = 2$. Por tanto, para $x \in [2, 8]$:

$$s(x) = 35 - 20x + 3x^2 - 5(x-4)_+^2 + 2(x-5)_+^2$$

Es decir:

$$s(x) = \begin{cases} 3x^2 - 20x + 35 & x \in [2, 4] \\ -2x^2 + 20x - 45 & x \in (4, 5] \\ 5 & x \in (5, 8] \end{cases}$$

Problema. Dados los siguientes datos, calcula el spline cuadrático que los interpola:

$$\begin{array}{c|ccccc} x_i & -1 & 1 & 3 & 6 & 7 \\ y_i & 1 & 4 & 8 & 2 & 9 \\ d_i & & & 5 & & \end{array}$$

Solución. Nos dan la derivada en el nodo 3, procedemos a calcular las diferencias divididas en los nodos 1 y 3 para hallar s_2 :

x	y	DD1	DD2
1	4		
3	8	2	
3	8	5	$3/2$

s_2 queda en su intervalo:

$$s_2(x) = 4 + 2(x-1) + \frac{3}{2}(x-1)(x-3)$$

Ahora estimamos la derivada en el nodo 1:

$$s_2'(x) = 2 + \frac{3}{2}((x-3) + (x-1)) = 3x - 4$$

$$s_2'(1) = 3 \cdot 1 - 4 = -1$$

Realizamos de nuevo la tabla de diferencias divididas:

s_1 queda en su intervalo:

x	y	DD1	DD2
-1	1		
1	4	$\frac{3}{2}$	
1	4	-1	$-\frac{5}{4}$

$$s_1(x) = 1 + \frac{3}{2}(x+1) + \frac{-5}{4}(x+1)(x-1)$$

Ahora que hemos calculado la expresión de s para todos los intervalos a la izquierda de la derivada, calculamos la función para todos los valores a la derecha de la derivada.

Calculamos las diferencias divididas para nodos 3 y 6:

x	y	DD1	DD2
3	8		
3	8	5	
6	2	-2	$-\frac{7}{3}$

s_3 y su derivada quedan en su intervalo:

$$s_3(x) = 8 + 5(x-3) - \frac{7}{3}(x-3)(x-3)$$

$$s'_3(x) = 5 - \frac{7}{3}2(x-3) = 5 - \frac{7}{3}(2x-9)$$

Estimamos la derivada del nodo 6:

$$s'_3(6) = 5 - \frac{7}{3}3 = -2$$

Finalmente, calculamos s_4 :

x	y	DD1	DD2
6	2		
6	2	-9	
7	9	7	16

De esta forma, la expresión de s_4 sería:

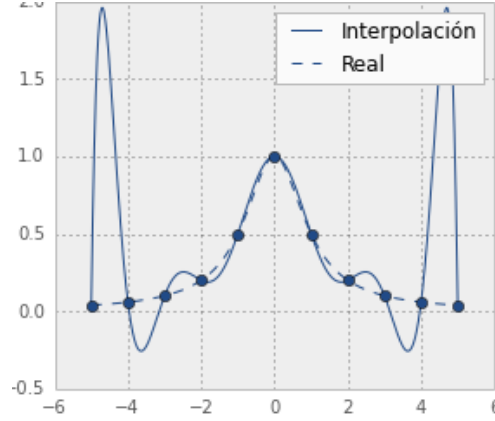
$$s_4(x) = 2 - 9(x-6) + 16(x-6)(x-6)$$

Por lo tanto, nuestra solución sería:

$$s(x) = \begin{cases} s_1(x) = 1 + \frac{3}{2}(x+1) + \frac{-5}{4}(x+1)(x-1) & \text{si } x \in [-1, 1) \\ s_2(x) = 4 + 2(x-1) + \frac{3}{2}(x-1)(x-3) & \text{si } x \in [1, 3) \\ s_3(x) = 8 + 5(x-3) - \frac{7}{3}(x-3)(x-3) & \text{si } x \in [3, 6) \\ s_4(x) = 2 - 9(x-6) + 16(x-6)(x-6) & \text{si } x \in [6, 7] \end{cases}$$

2. Splines cúbicos

Uno de los problemas de la interpolación polinomial es que, al ir aumentando el número de nodos el grado del polinomio requerido para interpolarlos aumenta. Esto conlleva fluctuaciones en los extremos de la interpolación.



Si dividimos el intervalo en una partición podemos interpolar utilizando un polinomio $S_i(x)$ de grado 3 en cada intervalo, es decir, utilizando **splines cúbicos**. Como veremos después este método minimiza la cota de error.

$$S(x) = \begin{cases} S_0(x) & \text{si } x \in [x_0, x_1) \\ S_1(x) & \text{si } x \in [x_1, x_2) \\ \vdots & \vdots \\ S_i(x) & \text{si } x \in [x_i, x_{i+1}) \\ \vdots & \vdots \\ S_{n-1}(x) & \text{si } x \in [x_{n-1}, x_n] \end{cases} \quad (4)$$

Esta interpolación lineal fragmentaria pasa por los puntos: $\{(x_0, f(x_0)), (x_1, f(x_1)), \dots, (x_n, f(x_n))\}$

Dentro de los cúbicos encontramos los de clase 1 y 2, denotados por S_3^1 y S_3^2 (ó S_3).

1. Los splines cúbicos de **clase 1** son continuos y derivables con derivada continua. Conforman un espacio vectorial de dimensión $2(n+1)$. Una base es:

$$\{1, x, x^2, x^3, (x - x_1)_+^2, (x - x_1)_+^3, \dots, (x - x_{n-1})_+^2, (x - x_{n-1})_+^3\}$$

Estos splines no aseguran derivabilidad en los extremos. En un contexto geométrico esto significa que la función no es *suave* en los puntos de unión. Generalmente las condiciones físicas necesitan esa suavidad, y es aquí donde intervienen los splines cúbicos de clase 2.

2. Los splines cúbicos de **clase 2** son continuos y 2 veces derivables. A partir de la fórmula general, la dimensión de este espacio para una partición $\{x_i\}_{i=0, \dots, n}$ es $\dim(S_3^2(P)) = (3-2)n + 2 + 1 = n + 3$. Como tenemos $n + 1$ variables, tenemos 2 libertades en la resolución.

2.1. Construcción a partir de los valores de s'' en los nodos $\{x_i\}$

Vamos a plantear un método de resolución utilizando las segundas derivadas, denotamos, para $i = 1, \dots, n-1$: $M_i = S''(x_i)$, que son desconocidos a priori salvo en un spline natural.

Como el spline es de clase 2, tenemos para $i = 1, \dots, n-1$:

$$S''(x_i) = S'_i(x_i) = S'_{i+1}(x_i)$$

La restricción a cada intervalo de S es un polinomio S_i de grado 3, por ende, S'_i es lineal, con expresión para $x \in [x_{i-1}, x_i]$:

$$S'_i(x) = M_{i-1} \frac{x_i - x}{h_i} + M_i \frac{x - x_{i-1}}{h_i}$$

Integramos dos veces usando que $S_i(x_{i-1}) = y_{i-1}$ y $S_i(x_i) = y_i$ para las constantes de integración, obteniendo, para $x \in [x_{i-1}, x_i]$:

$$S_i(x) = M_{i-1} \frac{(x_i - x)^3}{6h_i} + M_i \frac{x - x_{i-1}}{6h_i} + (y_{i-1} - \frac{M_{i-1}h_i^2}{6}) \cdot \frac{x_i - x}{h_i} + (y_i - \frac{M_i h_i^2}{6}) \cdot \frac{x - x_{i-1}}{h_i}$$

Esta ecuación nos permite calcular S conocidas M_i con $i = 0, 1, \dots, n$. Las condiciones de suavidad en las ligaduras nos permiten igualar $S'_{i+1}(x_i) = S'_i(x_i)$. Derivando una vez, si $x \in [x_{i-1}, x_i]$:

$$S'_i(x) = -M_{i-1} \frac{(x_i - x)^2}{2h_i} + M_i \frac{(x - x_{i-1})^2}{2h_i} + \frac{y_i - y_{i-1}}{h_i} - (M_i - M_{i-1}) \frac{h_i}{6}$$

Si $x \in [x_i, x_{i+1}]$:

$$S'_{i+1}(x) = -M_i \frac{(x_{i+1} - x)^2}{2h_i} + M_{i+1} \frac{(x - x_i)^2}{2h_{i+1}} + \frac{y_{i+1} - y_i}{h_{i+1}} - (M_{i+1} - M_i) \frac{h_{i+1}}{6}$$

Recordando que $h_i = x_i - x_{i-1}$ e igualando $S'_{i+1}(x_i) = S'_i(x_i)$:

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

Agrupamos los M_i :

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

Multiplicamos a ambos lados por 6, sacamos factor común y recordamos que $p_{i+1} = \frac{y_{i+1} - y_i}{h_{i+1}}$:

$$6M_i \frac{-3h_{i+1}}{6} + \frac{h_{i+1}}{6} - 3\frac{h_i}{6} + \frac{h_i}{6} + 6(p_{i+1} - p_i) = M_{i+1}h_{i+1} + M_{i-1}h_i$$

Agrupando y multiplicando M_i arriba y abajo por -2 :

$$-2M_i \frac{-2h_{i+1} - 3h_i + h_i}{-2} + 6(p_{i+1} - p_i) = M_{i+1}h_{i+1} + M_{i-1}h_i$$

Pasamos el M_i a la derecha y dividimos por $(h_{i+1} + h_i)$ en ambos lados:

$$6 \frac{p_{i+1} - p_i}{h_{i+1} + h_i} = M_{i+1} \frac{h_{i+1}}{h_{i+1} + h_i} + M_{i-1} \frac{h_i}{h_{i+1} + h_i} + 2M_i$$

Denotando por $\mu_i = \frac{h_i}{h_i + h_{i+1}}$, $\lambda_i = 1 - \mu_i = \frac{h_{i+1}}{h_i + h_{i+1}}$ y $\gamma_i = 6 \frac{p_{i+1} - p_i}{h_{i+1} + h_i}$:

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = \gamma_i (*)$$

Con los M_i en las ligaduras tendremos $4(n-1)$ variables, para que el sistema sea determinado nos faltan dos condiciones. Hay diferentes condiciones que se nos pueden presentar:

■ Spline sujeto

$S'_1(x_0) = f'_0$ y $S'_n(x_n) = f'_n$. De acuerdo con la fórmula de $S'(x)$ obtenemos:

$$f'_0 = -\frac{M_0 h_0}{2} + f[x_0, x_1] - \frac{(M_1 - M_0)h_0}{6}$$

$$\implies 2M_0 + M_1 = \frac{6(f[x_0, x_1] - f'_0)}{h_1} = 6f[x_0, x_0, x_1](*)$$

Equivalentemente para x_n :

$$S'_n(x_n) = -\frac{M_{n-1}(x_n - x_{n-1})^2}{2h_n} + \frac{M_n(x_n - x_{n-1})^2}{2h_n} + \frac{(y_n - y_{n-1})}{h_n} - \frac{(M_n - M_{n-1})h_n}{6}$$

$$\implies M_{n-1} + 2M_n = 6f[x_{n-1}, x_n, x_n](*)$$

Usando (*), la matriz del sistema es:

$$\begin{pmatrix} 2 & \lambda_0 & 0 & \cdots & 0 \\ \mu_1 & 2 & \lambda_1 & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & 0 & \mu_{n-1} & 2 & \lambda_{n-1} \\ 0 & \cdots & 0 & \mu_n & 2 \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_{n-1} \\ M_n \end{pmatrix} = \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{n-1} \\ \gamma_n \end{pmatrix}$$

■ Spline natural

En este caso $M_0 = 0$ y $M_n = 0$, $\lambda_0 = \mu_n = 1$ por lo que el sistema queda:

$$\begin{pmatrix} 2 & \lambda_1 & 0 & \cdots & 0 \\ \mu_2 & 2 & \lambda_2 & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & 0 & \mu_{n-2} & 2 & \lambda_{n-2} \\ 0 & \cdots & 0 & \mu_{n-1} & 2 \end{pmatrix} \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{pmatrix} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{n-2} \\ \gamma_{n-1} \end{pmatrix}$$

■ Spline periódico

En este caso $S'_1(x_0) = S'_n(x_n)$ y $S''_1(x_0) = S''_n(x_n)$. El sistema queda:

$$\begin{pmatrix} 2 & \lambda_0 & 0 & \cdots & 0 \\ \mu_1 & 2 & \lambda_1 & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & 0 & \mu_{n-1} & 2 & \lambda_{n-1} \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{pmatrix} = \begin{pmatrix} \gamma_0 = h_1 - S'_1(x_0) \\ \gamma_1 \\ \vdots \\ \gamma_{n-2} \\ \gamma_{n-1} \end{pmatrix}$$

En este caso añadimos:

$$S'_1(x_0) = -M_0 \cdot \frac{h_1}{2} + f[x_0, x_1] - \frac{M_1 - M_0}{6} \cdot h_1$$

2.2. Propiedades de minimización

Comenzamos planteando un problema de minimización sobre el espacio euclídeo $(C^2([a, b]), < \cdot, \cdot >)$, con la métrica y norma definida de la forma usual:

$$< f, g > = \int_a^b f g, \quad \|f\| = \sqrt{\int_a^b f^2}$$

Planteamos el problema:

Problema. Sea $f \in C^2([a, b])$, $P \in \mathcal{P}([a, b])$. Sea $H \subset C^2([a, b])$ definido por:

$$H = \{g \in C^2([a, b]) : \forall p \in P \ g(p) = f(p) \text{ y } g'(a) = f'(a), \ g'(b) = f'(b)\}$$

Hallar $u \in H$ tal que $\|u''\|$ sea mínima.

Para resolver el problema, demostramos el siguiente teorema:

Teorema (Minimización). Sea $f \in C^2([a, b])$, $P \in \mathcal{P}([a, b])$, s spline sujeto para f . Se verifica:

$$\forall u \in H : \|s''\| \leq \|u''\|$$

Demostración. Sea $u \in H$, $e = u - s$. Tenemos:

$$\|u''\|^2 = \|e'' + s''\|^2 = \|e''\|^2 + \|s''\|^2 + 2 \langle e'', s'' \rangle$$

Dividimos $\langle e'', s'' \rangle$ en intervalos:

$$\langle e'', s'' \rangle = \int_a^b e'' s'' = \sum_1^{n-1} \int_{x_i}^{x_{i+1}} e'' s''$$

En cada intervalo, integramos por partes:

$$\sum_1^{n-1} \int_{x_i}^{x_{i+1}} e'' s'' = \sum_1^{n-1} e'(x) s''(x) \Big|_{x_i}^{x_{i+1}} - \sum_1^{n-1} \int_{x_i}^{x_{i+1}} e' s'''$$

La primera sumatoria es una suma telescópica, por lo que conservamos el primer y último término:

$$\sum_1^{n-1} e'(x) s''(x) \Big|_{x_i}^{x_{i+1}} = e'(b) s''(b) - e'(a) s''(a) = (u'(b) - s'(b)) s''(b) - (u'(a) - s'(b)) s''(a) = 0$$

ya que $u, s \in H$.

En cuanto a la segunda, $s'''|_{[x_i, x_{i+1}]}$ es constante, por lo que podemos sacarlo de la integral:

$$\sum_1^{n-1} s_i \int_a^b e' = \sum_1^{n-1} s_i (e(b) - e(a)) = 0$$

Es decir, $\langle e'', s'' \rangle = 0$. Por tanto:

$$\|u''\|^2 = \|e''\|^2 + \|s''\|^2 + 2 \langle e'', s'' \rangle = \|e''\|^2 + \|s''\|^2 \geq \|s''\|^2$$

donde utilizamos que la norma siempre es positiva. □

Así, podemos observar que el **spline cúbico sujeto** asociado a una función f tiene la menor norma de su segunda derivada de entre las que interpolan a f en una partición dada, por lo que resuelve nuestro problema.

2.3. Error en los splines cúbicos

Teorema. Sea $f \in C^4([a, b])$, $n \in \mathbb{N}$, $P = \{x_i\}_{i=0 \dots n} \in \mathcal{P}([a, b])$ y $s \in S_3^1(P)$ spline para f . Además, sean $h = \max\{x_i - x_{i-1}\}_{i=1 \dots n}$, $M > 0$ cota superior de $|f^{(iv)}|$ en $[a, b]$ y $E = f - s$, $x \in [a, b]$.

Se verifica:

$$|E(x)| \leq \frac{5M}{384}h^4 \quad (5)$$

La demostración, así como cotas para las derivadas, puede consultarse en *Optimal Error Bounds for Cubic Spline Interpolation*, Charles Hall y Weston Meyer, (1976).

2.4. Ejemplos

Sujeto:

Problema. Hallar spline sujeto tal que:

1. Pasa por los puntos: $\{(0, 0), (1, 0,5), (2, 2), (3, 1,5)\}$
2. $S'(0) = 0,2$ y $S'(3) = -1$

Solución. Como los nodos están equiespaciados $h_i = 1 \ \forall i \in \{1..n\}$

$$\lambda_0 = \mu_3 = 1 \text{ y } \lambda_1 = \lambda_2 = \mu_1 = \mu_2 = \frac{1}{2}$$

Calculamos las diferencias divididas para obtener los γ_i

- $\frac{\gamma_0}{6} = f[x_0, x_0, x_1] = \frac{f[x_1, x_0] - f[x_0, x_0]}{x_1 - x_0} = \left(\frac{0,5 - 0}{1 - 0} - 0,2\right) / 1 = 0,3$
- $\frac{\gamma_1}{6} = f[x_0, x_1, x_2] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} = \left(\frac{2 - 0,5}{1 - 0} - \frac{0,5 - 0}{1 - 0}\right) / 2 = \frac{1}{2}$
- $\frac{\gamma_2}{6} = f[x_1, x_2, x_3] = \frac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_1} = \left(\frac{1,5 - 2}{1 - 0} - \frac{2 - 0,5}{1 - 0}\right) / 1 = -1$
- $\frac{\gamma_3}{6} = f[x_2, x_3, x_3] = \frac{f[x_3, x_3] - f[x_2, x_3]}{x_3 - x_2} = \left(-1 - \frac{1,5 - 2}{1 - 0}\right) / 1 = -\frac{1}{2}$

El sistem queda:

$$\begin{pmatrix} 2 & 1 & 0 & 0 \\ 1/2 & 2 & 1/2 & 0 \\ 0 & 1/2 & 2 & 1/2 \\ 0 & 0 & 1 & 1/2 \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \end{pmatrix} = \begin{pmatrix} 6 \cdot \frac{3}{10} \\ 6 \cdot \frac{1}{2} \\ 6 \cdot (-1) \\ 6 \cdot (-\frac{1}{2}) \end{pmatrix}$$

Del que obtenemos la solución $M_0 = -0,36$, $M_1 = 2,52$, $M_2 = -3,72$ y $M_3 = 0,36$. Calculamos los trozos finalmente aplicando la fórmula:

$$S_1(x) = M_0 \frac{(x_1 - x)^3}{6} + M_1 \frac{(x - x_0)}{6} + (y_0 - \frac{M_0}{6}) \frac{x_1 - x}{1} + (y_1 - \frac{M_1}{6}) \frac{x - x_0}{1} = 0,48x^3 - 0,18x^2 + 0,2x$$

Equivalentemente para C_2 y C_3 , obtenemos la solución:

$$S(x) = \begin{cases} 0,48x^3 - 0,18x^2 + 0,2x & \text{si } 0 \leq x \leq 1 \\ -1,04x(x-1)^3 + 1,26(x-1)^2 + 1,28(x-1) - 0,5 & \text{si } 1 < x \leq 2 \\ 0,68(x-2)^3 - 1,86(x-2)^2 + 0,68(x-2) + 2 & \text{si } 2 < x \leq 3 \end{cases}$$

3. Implementación en ordenador: Octave

3.1. Spline Lineal

La implementación de la función que nos permite calcular un spline lineal es muy sencilla:

```
function s = SplineLineal(x,y)
    p = diff(y)./diff(x);
    A = [p' y(1:end-1)'];
    s = mkpp(x,A);
end
```

3.2. Splines cuadráticos

Utilizando el **método global**, podemos definir fácilmente una función que calcule un spline cuadrático de clase 1:

```
function s = SplineCuad(x, y, d_k, k)
    # Número de intervalos
    n = length(x) - 1;

    # 1, x, x^2
    A(:,1) = [ones(n+1,1); 0];
    A(:,2) = [x' ; 1];
    A(:,3) = [x'.^2 ; 2.*x(k+1)];

    # Potencias truncadas
    for j = 4 : n + 2
        t = @ (s) (s > x(j-2)) .* (s - x(j-2));
        A(:, j) = [t(x').^2; 2.*t(x(k+1))];
    end

    # Resolución del sistema
    sol = A \ [y' ; d_k];

    for k = 1:n
        p = sol(3:-1:1);

        for l = 2:k
            p += sol(l+2).*[1, -2.*x(l), x(l).^2];
        end

        B(k, :) = polyaffine(p,[-x(k) 1]);
    end

    s = mkpp(x,B);
end
```

Otra implementación posible es calcular el spline **a trozos**:

```
function z = SplineCuadLocal(x, y, d_k, k)
    s = zeros(length(x)-1, 3);
    d = d_k;
    #Recorremos todos los nodos de n+1 en adelante:

    for i = (k+1):(length(x)-1)
        p = (y(i+1)-y(i))/(x(i+1)-x(i));
        q = (p-d)/(x(i+1)-x(i));
        v = [x(i) x(i)];
        s(i,:) = [0, 0, y(i)]+[0, d, -d*x(i)]+q*poly(v);
        d = polyval(polyder(s(i,:)),x(i+1));
    end
    d = d_k;

    #Recorremos todos los nodos desde n hasta el 1:

    for j = k:-1:1
        p = (y(j+1)-y(j))/(x(j+1)-x(j))
        q = (d-p)/(x(j+1)-x(j))
        v = [x(j) x(j+1)]
        s(j,:) = [0 0 y(j)]+[0 p -p*x(j)]+q*poly(v);
        d = polyval(polyder(s(j,:)), x(j))
    end

    for i = 1:length(s)
        s(i,:) = polyaffine(s(i,:), [-x(i), 1]);
    end
    z = mkpp(x, s);
end
```

3.3. Splines cúbicos

Para el cálculo de splines cúbicos por medio de la segunda derivada nos hemos valido de la interpolación de splines lineales y de la función `ppint`, que realiza la integración de un spline.

3.3.1. Spline sujeto

La función para el cálculo del **spline sujeto** queda:

```
function s = SplineSuj (x, y, d_1, d_n)
    n      = length(x) - 1;
    twoes = 2*ones(1,n+1);
    h      = diff(x);
```



```

mu      = [(h(1:end-1)./(h(1:end-1) + h(2:end))) 1];
lambda = ones(1,n) - [0 mu(1:end-1)];

A = diag(mu,-1) + diag(twoes,0) + diag(lambda,1);

dd1 = diff(y)./h;
for i=1:(n-1)
    dd2(i)=(dd1(i+1)-dd1(i))/(x(i+2)-x(i));
end

gamma(1) = 6*(dd2(1)-d_1)/(x(2)-x(1));
gamma(2:n) = 6*dd2(1:n-1);
gamma(n+1) = 6*(d_n-dd2(n-1))/(x(n+1)-x(n));

m = A\gamma';
s = ppint(ppint(SplineLineal(x, m')));
end

```

3.3.2. Spline natural

Para el **spline natural** calculamos la matriz, resolvemos el sistema e integramos añadiendo los datos de las derivadas segundas en los extremos:

```

function s = SplineNat(x, y)
    n = length(x) - 1;

    h = diff(x);      #h_i
    p = diff(y)./h;   #p_i

    twoes = 2*ones(1,n-1);
    mu     = (h(1:end-1)./(h(1:end-1) + h(2:end)));
    lambda = ones(1,n-1) - mu;
    gamma  = diff(p)./(h(1:end-1) + h(2:end));

    if(n < 3)
        A = 2;
    else
        A = diag(twoes) + diag(lambda(1:end-1),1) + diag(mu(2:end),-1);
    end

    m = A \ gamma';
    s = ppint(ppint(SplineLineal(x, [0 m' 0])));
end

```

3.3.3. Spline periódico

La función del **spline periódico** queda:

```

function z = SplinePer (x, y)
    n=length(x);
    twoes=2*ones(1,n);
    h=diff(x);
    lambda=[1];
    for i=1:(n-2)
        lambda(i+1)=h(i+1)/(h(i+1)+h(i));
    endfor
    mu=[];
    for i=1:(n-2)
        mu(i)=h(i)/(h(i+1)+h(i));
    endfor
    mu(n-1)=1;
    A=diag(mu,-1)+diag(twoes,0)+diag(lambda,1);
    dd1=zeros(n-1,1);
    for i=1:(n-1)
        dd1(i)=(y(i+1)-y(i))/(x(i+1)-x(i));
    endfor
    dd2=zeros(n-2,1);
    for i=1:(n-2)
        dd2(i)=(dd1(i+1)-dd1(i))/(x(i+2)-x(i));
    endfor
    A(1,:)=h(1)*A(1,:);
    A(1,1)=A(1,1)-h(1)/3;
    A(1,2)=A(1,2)+h(1)/6;
    A(n,:)=h(n-1)*A(1,:);

    A(n,1)=A(1,1)+h(1)/3;
    A(n,2)=A(1,2)-h(1)/6;
    gamma=zeros(n,1);
    gamma(1)=h(1)-dd1(1);
    gamma(n)=-h(n-1)+dd1(1);
    for i=1:(n-2)
        gamma(i+1)=6*dd2(i);
    endfor
    m=A\gamma;
    s = ppint(ppint(SplineLineal(x, m)));
end

```

A. Definiciones y notación

Definición. Sea $I \subset \mathbb{R}$ un intervalo cerrado y acotado con extremos a, b :

- Una **partición** P de I es un subconjunto finito de I con $a, b \in P$.
- $\mathcal{P}(I)$ es el conjunto de todas las particiones de I .

Definición. Sea $a \in \mathbb{R}$, $n \in \mathbb{N}$. La **potencia truncada** en a de grado n , $(x - a)_+^n$ viene dada por:

$$(x - a)_+^n = \begin{cases} 0 & \text{si } x \leq a \\ (x - a)^n & \text{si } x > a \end{cases}$$

Cualquier potencia truncada de grado n es de clase $n - 1$, y su derivada de orden n presenta una discontinuidad en a . La derivada de $(x - a)_+^n$ en x es $n(x - a)_+^{n-1}$.

Su implementación en Octave es bastante sencilla: dados **a** y **n**, podemos definir la potencia truncada como función anónima de la siguiente forma:

```
pot = @(x) (x > a) * (x - a)^n
```

Como Octave tiene tipos dinámicos convertirá $(x > a)$ a 1 si $x > a$ y a 0 en otro caso.

Definición. Dado un intervalo $[a, b]$ y una partición $\{x_i\}_{i=0 \dots n} \in \mathcal{P}([a, b])$, se definen, para $1 \leq i \leq n$:

- $h_i = x_i - x_{i-1}$
- $p_i = \frac{y_i - y_{i-1}}{h_i}$

B. Bibliografía

- [Quadratic Interpolatory Splines](#) W.J.Kammerer, G. W. Reddien, and R.S. Varga
- [Cubic Spline Interpolation](#)
- *Análisis numérico* (Novena Edición) Richard L. Burden y J. Douglas Faires
- *Analysis of Numerical Methods* - Eugene Isaacson, Herbert Bishop Keller
- *Spline Approximation of Functions and Data* - Universidad de Oslo
- *Optimal Error Bounds for Cubic Spline Interpolation* - Charles A. Hall