



FACULDADE DE MATEMÁTICAS

Traballo Fin de Grao

Interpolación con splines cúbicos

Alejandro Barros González

2023/2024

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Interpolación con splines cúbicos

Alejandro Barros González

Xullo, 2024

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Traballo proposto

Área de Coñecemento: Matemática Aplicada
Título: Interpolación con splines cúbicos
Breve descripción do contido
Necesidade da interpolación por funcións splines. O espacio de splines polinómicos. B-splines. Interpolación con splines cúbicos: condicións nos extremos (naturais, encastramento, periodicidade), algoritmo de cálculo, análise do erro.
Recomendacións
Outras observacións

Índice

Resumo	VI
Introdución	IX
1. Estudo dos splines polinómicos e B-Splines	1
1.1. Necesidade das funcións tipo spline	1
1.2. O espazo $S_m(\Omega_n)$ de splines polinómicos	3
1.3. B-splines	9
2. Interpolación con splines cúbicos	19
2.1. Formulación do problema. Condicións nos extremos	19
2.2. Cálculo do spline cúbico de interpolación usando B-splines	26
2.3. Exemplos	28
2.4. Problema xeral de interpolación con splines	30
3. Interpolación con splines cúbicos: análise do erro	33
3.1. O erro en splines lineais	33
3.2. Relación integral	36
3.3. O erro en splines cúbicos	38
3.4. Consecuencias da relación integral	41
A. Códigos de MATLAB	45
Bibliografía	69

Resumo

Neste traballo introduciremos os splines e amosaremos como, en contraposición á interpolación polinómica e a cachos, resulta unha moi boa ferramenta á hora de interpolar de forma regular. Unha vez presentada a idea intuitiva, estudaremos o espazo de splines de grao m con respecto a unha partición de $n + 1$ nós, e moi especialmente a base do espazo constituída polos B-spline, de gran importancia nos cálculos prácticos. Posteriormente resolveremos o problema de interpolación con splines cúbicos (de grao 3), por ser estes os máis empregados na práctica. Trataremos a existencia, a unicidade e o algoritmo de cálculo, que implementamos nun código MATLAB co que amosamos algúns exemplos. Finalmente analizaremos o erro entre o spline interpolador e a función da que proveñen os nós interpolados cando se verifican certas condicións de regularidade nesta última.

Abstract

In this dissertation, we will introduce splines and prove how, in contrast to polynomial and piecewise interpolation, provide a superior tool for interpolation with regular functions. Once we present the intuitive idea, we will study the space of splines of degree m with respect to a partition of $n + 1$ nodes, paying special attention to the basis of this space constituted by B-splines, which are of great importance in practical computations. Thereafter, we will solve the interpolation problem using cubic splines, as these are the most commonly used in practice. We will discuss the existence, uniqueness and the calculation algorithm, which we will implement in a MATLAB code to illustrate some examples. Finally, we will analyze the error between the interpolating spline and the function from which the interpolation nodes are derived, assuming certain regularity conditions in the latter.

Palabras chave

Splines, splines cúbicos, B-splines, interpolación, aproximación.

Introdución

O problema da interpolación consiste en construír unha función simple (polinómica, trigonométrica...) que tome uns valores dados en puntos coñecidos chamados nós.

A interpolación polinómica é unha das máis estudadas, sendo Lagrange un dos primeiros en dar fórmulas para a obtención do polinomio interpolador. Posteriormente Newton deu fórmulas alternativas facendo uso das súas diferenzas divididas, que mediante a fórmula recursiva permitían aproveitar o polinomio calculado ao engadir novos nós. Non obstante, posto que o grao do polinomio interpolador está ligado ao número de nós, pronto se viu que a interpolación polinómica presentaba comportamentos oscilatorios cando o número de nós é relativamente grande (efecto Runge). A interpolación a cachos permite unha solución parcial deste problema. Ao dividir o conxunto de nós en grupos menores pódense empregar polinomios de menor grao. Deste xeito, sacrifícase a regularidade da función a cambio de evitar o efecto oscilatorio.

A interpolación con splines permite dar resposta a estas dificultades: evita o fenómeno Runge e ao mesmo tempo permite regularidade global da función interpolante. Se ben no pasado xa outras persoas traballaban de forma implícita con eles, non foi ata 1946 que I. J. Schoenberg deu unha definición formal de spline. Os splines son funcións polinómicas (habitualmente dun grao pequeno) a cachos ás que se lles impón condicións de regularidade nos nós de unión. Deste xeito, podemos aumentar o número de nós sen necesidade de incrementar o grao do polinomio.

No capítulo 1 expoñemos os problemas sinalados da interpolación polinómica e presentaremos o concepto de splines, estudamos o espazo de splines de grao m respecto a unha partición de $n+1$ nós, xunto con algunhas propiedades das súas raíces que resultan imprescindibles posteriormente. Falaremos tamén da base alternativa conformada polos B-splines, que son esenciais na parte computacional pois reducen os cálculos, e expresaremos de forma explícita os máis habituais.

Deseguido, no capítulo 2 trataremos o problema de interpolación con splines cúbicos. Probarémos que a solución existe e é única baixo condicións de contorno adicionais. Posteriormente, reformularemos o problema con B-splines cúbicos para implementalo nun algoritmo de MATLAB, co que calculamos o spline cúbico de interpolación nalgúns exemplos amosando as súas vantaxes con respecto á interpolación polinómica. Finalmente, estudaremos a existencia e solución ao

problema xeral de interpolación con splines.

No capítulo 3 poremos a nosa atención sobre o erro cometido na aproximación cunha función polo spline de interpolación baixo determinadas condicións de regularidade. Comezaremos co caso lineal, que nos servirá de base en splines de maior orde. Posteriormente introduciremos a relación integral, que nos permitirá afirmar que os splines cúbicos son, baixo certas condicións de contorno, a función interpolante con menor curvatura. Esta propiedade fai que os splines poidan ser unha boa ferramenta para o deseño do trazado de liñas ferroviarias.

Capítulo 1

Estudo dos splines polinómicos e B-Splines

O obxectivo deste capítulo é introducir a noción de splines e a súa necesidade fronte a outras familias de funcións no proceso da interpolación. Unha vez coñezamos a base máis intuitiva, presentaremos os B-splines, unha familia de splines que dan lugar a problemas de interpolación mellor condicionados. Tamén introducimos varios conceptos da interpolación clásica que serán fundamentais a posteriori. Con esa fin seguiremos as referencias [4], [7] e [12]

1.1. Necesidade das funcións tipo spline

En interpolación, a primeira familia que se nos ocorre como ferramenta para interpolar son os polinomios, xa que dados $(n + 1)$ nós distintos, existe un único polinomio de grao n que pasa por todos os puntos. Non obstante, conforme aumenta o número de nós, en polinomios de graos cada vez maior comeza a aparecer un comportamento oscilatorio que non é interesante cando se pretenden estudar fenómenos nos que é esperable que sigan unha curva máis suave. Ademais, tampouco é coherente cando a variación da magnitude dos valores dos nós é pequena.

A continuación presentamos un exemplo clásico onde este problema se fai evidente.

Exemplo 1.1. Consideremos no intervalo $[-10, 10]$ a partición uniforme: $x_i = x_0 + (20i)/(n + 1)$, $i = 0, \dots, n$, e supoñamos que os valores asociados son $y_i = f(x_i)$, $i = 0, \dots, n$, con $f = 1/(1 + x^2)$. Na figura 1.1 podemos apreciar que independentemente de aumentarmos o número de nós, nos intervalos iniciais e finais a interpolación con polinomios presenta unha desviación moi pronunciada que non se corresponden coa función á que pertencen os nós que se están interpolando, que ten un carácter máis suave.

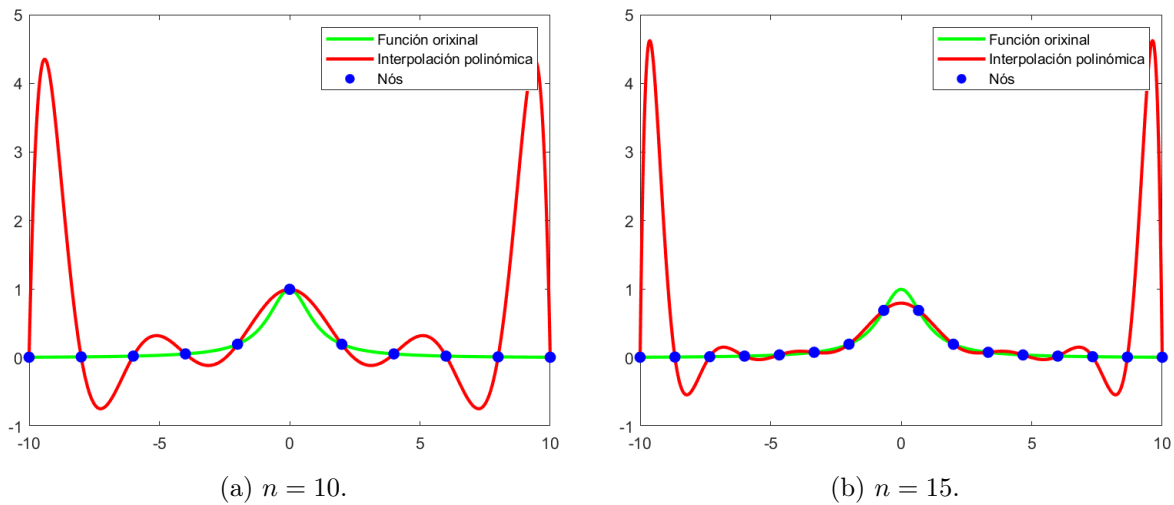


Figura 1.1: Exemplo dun comportamento oscilatorio na interpolación polinómica

Isto vai en contra da noción da interpolación, onde un esperaría que a maior número de nós mellor é a aproximación que se fai. A maiores, computacionalmente resulta moi custoso resolver o problema de interpolación polinómica cando aumenta moito o número de nós.

Como podemos ver en [7, p.44], o seguinte paso lóxico é dividir os nós en grupos de k nós consecutivos e axustar un polinomio de orde k para eses nós. Deste xeito evítase elevar de forma desmesurada o grao do polinomio, o que mitiga o comportamento oscilatorio. Deixando de lado as molestias derivadas de escoller de forma adecuada n e k para que a solución sexa única (se n non é múltiplo de k nos últimos nós habería máis dun polinomio de grado k que os interpole) a priori esta parece unha boa solución.

Exemplo 1.2. Seguindo co exemplo anterior, na figura 1.2 podemos ver que tanto na interpolación lineal a cachos como a cadrática a cachos desaparece o efecto oscilatorio e a función interpoladora axústase mellor á orixinal. Porén, non podemos pasar por alto que nos nós onde se xuntan dous polinomios presenta picos.

Como quedou de forma manifesta no exemplo, este procedemento ten a desvantaxe de que nos nós onde se pasa dun polinomio ao seguinte pérdese a regularidade, o que non é desexable se sabemos que os nós proceden dunha función cunha certa regularidade. Para evitalo, introducimos os splines, unha familia de funcións polinómicas a cachos que se pegan entre si de forma regular. O concepto de spline foi introducido por Schoenberg en 1946 (ver [8]) e dende entón non parou de desenvolverse.

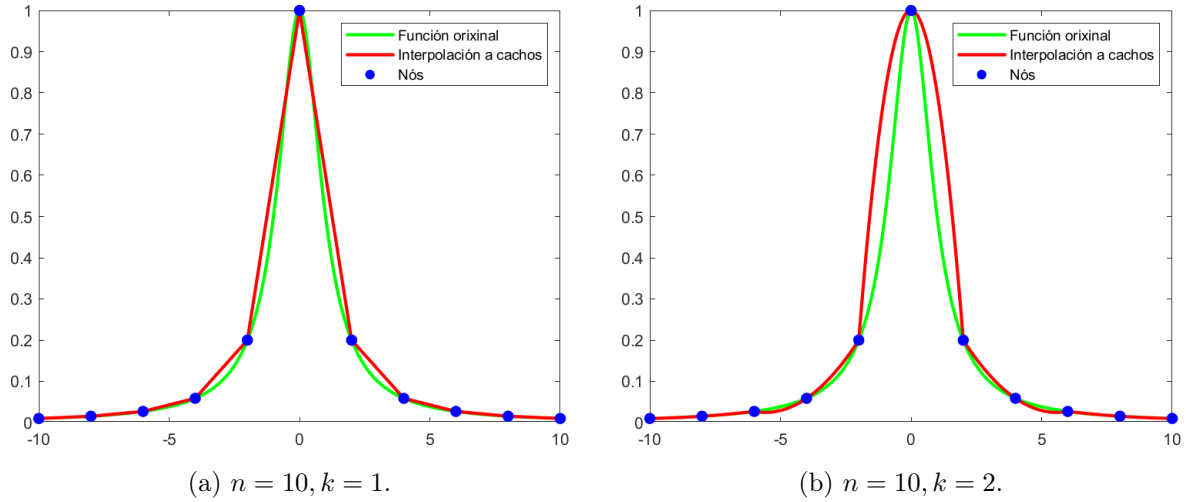


Figura 1.2: Interpolación lineal e parabólica a cachos

1.2. O espazo $S_m(\Omega_n)$ de splines polinómicos

Un spline é unha función definida a cachos onde cada cacho é un polinomio e onde, a diferenza do exemplo anterior, cada polinomio está unido entre si de forma suave. Consideremos un intervalo $[a, b] \subset \mathbb{R}$.

Definición 1.3. Unha partición do intervalo $[a, b]$ é un conxunto $\Omega_n := \{x_i\}_{i=0}^n$ de forma que $a = x_0 < x_1 < \dots < x_n = b$. Os puntos x_1, x_2, \dots, x_{n-1} son nós interiores e x_0 e x_{n+1} nós extremos.

Definición 1.4. Sexa $m \geq 1$ un enteiro. Dicimos que unha función $s : [a, b] \rightarrow \mathbb{R}$ é un spline polinómico de grao m relativo á partición Ω_n se verifica:

1. $s \in \mathcal{C}^{m-1}([a, b])$,
2. $s_i = s|_{[x_i, x_{i+1}]} \in \mathbb{P}_m$, $0 \leq i \leq n - 1$; onde \mathbb{P}_m é o espazo de polinomios de grao $\leq m$:
 $\mathbb{P}_m = \langle 1, x, x^2, \dots, x^m \rangle$

Denotaremos por $S_m(\Omega_n)$ ao conxunto de splines de grao m asociados á partición Ω_n .

Observación 1.5. Se ben nalgunhas referencias a regularidade que se lle impón aos nós interiores pode variar (véxase [5]), no noso caso ímonos limitar ao caso onde todos teñen regularidade $(m - 1)$.

Nótese que calquera polinomio $p \in \mathbb{P}_m$ é un spline de forma trivial (xa que en particular $p \in \mathcal{C}^\infty([a, b])$).

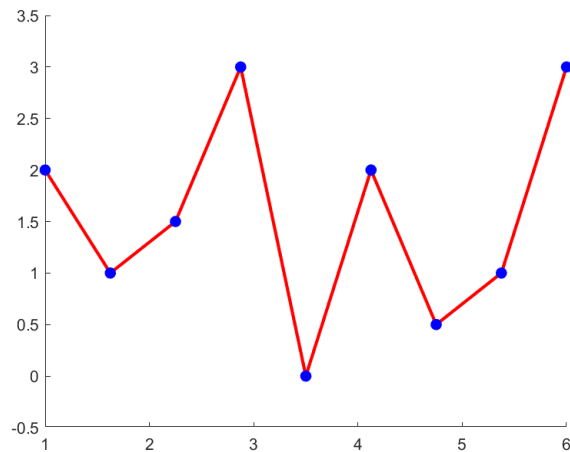


Figura 1.3: Spline de grau 1.

Un spline de grau 1 é unha función continua onde en cada subintervalo é un polinomio de grau ≤ 1 , o cal coincide coa interpolación lineal a cachos do apartado anterior. Podémolo escribir da seguinte maneira;

$$s(x) = \begin{cases} s_0(x) = a_0x + b_0, & \text{se } x \in [x_0, x_1] \\ s_1(x) = a_1x + b_1, & \text{se } x \in [x_1, x_2] \\ \vdots \\ s_{n-1}(x) = a_{n-1}x + b_{n-1}, & \text{se } x \in [x_{n-1}, x_n] \end{cases}$$

onde, para que $s \in C^0([a, b])$, tense que verificar:

$$s_{i-1}(x_i) = s_i(x_i), \quad i = 1, \dots, n - 1.$$

Deste xeito temos $2n$ coeficientes aos que se lle imponen $n - 1$ condicións.

Os splines cúbicos ($m = 3$) son os máis empregados, pois son función globalmente de clase 2 e en cada subintervalo polinomios de grau ≤ 3 . De forma explícita escríbense como segue:

$$s(x) = s_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i, \quad \text{se } x \in [x_i, x_{i+1}], \quad i = 0, \dots, n - 1;$$

de forma que s, s' e s'' sexan continuas, o cal se pode expresar da seguinte maneira:

$$\left. \begin{aligned} s_{i-1}(x_i) &= s_i(x_i) \\ s'_{i-1}(x_i) &= s'_i(x_i) \\ s''_{i-1}(x_i) &= s''_i(x_i) \end{aligned} \right\} i = 1, \dots, n - 1.$$

Logo neste caso hai $4n$ coeficientes aos que se lle fan $3(n - 1)$ restricións.

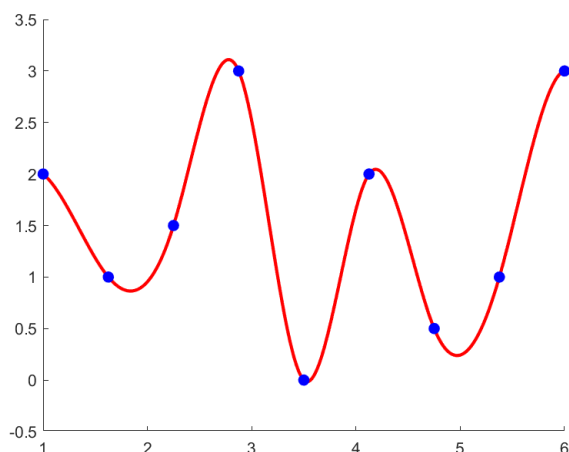


Figura 1.4: Spline de grao 3.

Exemplo 1.6. Outro exemplo de familias de splines son os splines unilaterais, que teñen o seguinte aspecto:

$$q_{m,i}(x) = (x - x_i)_+^m := \begin{cases} (x - x_i)^m, & \text{se } x \geq x_i, \\ 0, & \text{se } x < x_i. \end{cases}$$

Vexamos que son splines. En $[x_0, x_i]$ e en $[x_i, x_n]$ son polinomios de orde $\leq m$, logo só precisamos que sexa de clase $m - 1$. Isto tamén se verifica xa que

$$\lim_{x \rightarrow x_i^+} q_{m,i}^{(k)}(x) = \lim_{x \rightarrow x_i^+} q_{m,i}^{(k)}(x) = 0, \quad k = 0, \dots, m - 1.$$

Resultarannos de gran utilidade á hora de probar o seguinte resultado:

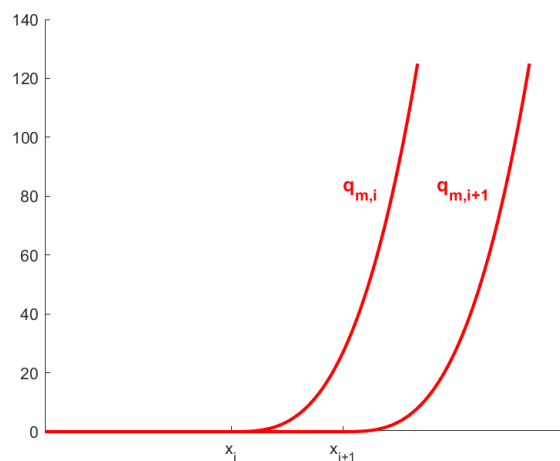


Figura 1.5: Splines unilaterais.

Teorema 1.7. *O conxunto $S_m(\Omega_n)$ é un espazo vectorial de dimensión $n + m$, e unha base do mesmo vén dada polo conxunto $\{p_0, p_1, \dots, p_m, q_{m,1}, \dots, q_{m,n-1}\}$, onde $p_j(x) := x^j$, $0 \leq j \leq m$.*

Demostración. Veremos en primeiro lugar que todo $s \in S_m(\Omega_n)$ se pode escribir do seguinte xeito:

$$s(x) = \sum_{j=0}^m a_j x^j + \sum_{i=1}^{n-1} b_i (x - x_i)_+^m, \quad (1.1)$$

o que proba que $\{p_0, p_1, \dots, p_m, q_{m,1}, \dots, q_{m,n-1}\}$ é un sistema de xeradores. En efecto, por ser $s \in \mathbb{P}_m$ no intervalo $[x_0, x_1]$, s é da forma:

$$s(x)|_{[x_0, x_1]} = \sum_{j=0}^m a_j x^j.$$

Facendo agora o desenvolvemento de Taylor desta función entorno ao punto x_1 , tense que:

$$s(x)|_{[x_0, x_1]} = \sum_{j=0}^m \frac{s^{(j)}(x_1)}{j!} (x - x_1)^j = \sum_{j=0}^m a_j^* (x - x_1)^j.$$

Analogamente, como $s \in \mathbb{P}_m$ no intervalo $[x_1, x_2]$, s pódese expresar do seguinte xeito:

$$s(x)|_{[x_1, x_2]} = \sum_{j=0}^m b_j x^j.$$

Facendo aquí tamén o desenvolvemento de Taylor desta función entorno ao punto x_1 , temos que:

$$s(x)|_{[x_1, x_2]} = \sum_{j=0}^m \frac{s^{(j)}(x_1)}{j!} (x - x_1)^j = \sum_{j=0}^m b_j^* (x - x_1)^j.$$

Dado que $s \in \mathcal{C}^{m-1}([a, b])$, da primeira expresión obtense que $s^{(j)}(x_1) = a_j^*$ e da segunda $s^{(j)}(x_1) = b_j^*$. Así, concluímos que $a_j^* = b_j^*$, $0 \leq j \leq m - 1$. Como na derivada m -ésima non se impón ningunha condición de continuidade, podemos expresar $b_m^* = a_m^* + b_1$. Polo tanto, a expresión de s no intervalo $[x_0, x_2]$ é:

$$s(x)|_{[x_0, x_2]} = \begin{cases} \sum_{j=0}^m a_j^* (x - x_1)^j, & \text{si } x \in [x_0, x_1], \\ \sum_{j=0}^m a_j^* (x - x_1)^j + b_1 (x - x_1)^m, & \text{si } x \in [x_1, x_2]. \end{cases}$$

ou o que é o mesmo:

$$s(x) = \sum_{j=0}^m a_j x^j + b_1 (x - x_1)_+^m.$$

Aplicando este mesmo razoamento para cada nó x_i , $1 \leq i \leq n - 1$, por recorrencia chegamos a expresión que buscábamos.

En segundo lugar precisamos garantir a independencia lineal de $\{p_0, \dots, p_m, q_{m,1}, \dots, l_{m,n-1}\}$; é dicir, se:

$$\sum_{j=0}^m a_j x^j + \sum_{i=1}^{n-1} b_i (x - x_i)_+^m = 0, \quad x \in [a, b];$$

entón $a_j = 0, j = 0, \dots, m$ e $b_i = 0, i = 1, \dots, n - 1$. No intervalo $[x_0, x_1]$ o spline pódese escribir como:

$$s(x)|_{[x_0, x_1]} = \sum_{j=0}^m a_j x^j = 0.$$

Así que é evidente que $a_0 = a_1 = \dots = a_m = 0$. Agora no intervalo $[x_1, x_2]$ o spline quedaranos unicamente como :

$$s(x)|_{[x_1, x_2]} = b_1 (x - x_1)^m = 0,$$

co cal podemos afirmar que $b_1 = 0$. Procedendo de esquerda a dereita, o que nos queda no intervalo $[x_i, x_{i+1}]$ é

$$s(x)|_{[x_i, x_{i+1}]} b_i (x - x_i)^m,$$

de onde se segue que $b_i = 0, i = 1, \dots, n - 1$. □

Non obstante, se ben probamos que é unha base, gustaríanos saber como expresar un spline coa mesma. Posto que en $[x_0, x_1]$ o spline é un polinomio de orde m , basta dar $m + 1$ puntos do spline nese subintervalo para coñecer os valores de a_0, \dots, a_m . Despois, para cada subintervalo consecutivo bastará dar un punto nese subintervalo para coñecer o valor de b_i . Polo tanto, dados $t_0, t_1, \dots, t_m \in (x_0, x_1), t_{m+1} \in (x_1, x_2), \dots, t_{m+n-1} \in (x_{n-1}, x_n)$, deberemos resolver o seguinte sistema lineal:

$$\sum_{j=0}^m a_j t_k^j + \sum_{i=1}^{n-1} b_i (t_k - x_i)_+^m = s(t_k), \quad k = 0, \dots, m + n - 1;$$

con matriz de coeficientes:

$$\begin{pmatrix} 1 & 1 & \dots & 1 & & & & & 1 \\ t_0 & t_1 & \dots & t_m & & & & & t_{m+n-1} \\ t_0^2 & t_1^2 & \dots & t_m^2 & & & & & t_{m+n-1}^2 \\ \vdots & \vdots & \ddots & \vdots & & & & & \vdots \\ t_0^m & t_1^m & \dots & t_m^m & & & & & t_{m+n-1}^m \\ 0 & 0 & \dots & 0 & (t_{m+1} - x_1)^m & (t_{m+2} - x_1)^m & \dots & & (t_{m+n-1} - x_1)^m \\ 0 & 0 & \dots & 0 & 0 & (t_{m+2} - x_2)^m & \dots & & (t_{m+n-1} - x_2)^m \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & & (t_{m+n-1} - x_{n-1})^m \end{pmatrix}^T$$

Deste xeito tamén queda patente que a forma de expresar o spline é única, xa que o determinante desta matriz é:

$$\prod_{k=1}^{n-1} (t_{m+k} - x_k)^m \begin{vmatrix} 1 & t_0 & t_0^2 & \dots & t_0^m \\ 1 & t_1 & t_1^2 & \dots & t_1^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & t_m^2 & \dots & t_m^m \end{vmatrix},$$

o cal é distinto de cero xa que $t_{m+k} \neq x_k$ e o determinante da dereita é de tipo Vandermonde (e así non é nulo xa que $t_i \neq t_j$ para $i \neq j$, $i, j = 0, \dots, m$).

Para necesidades posteriores é importante coñecer as raíces (ceros) dos splines de $S_m(\Omega_n)$. Con este fin, debemos distinguir os subintervalos $[x_i, x_{i+1}]$ onde s é idénticamente nula, e os subintervalos onde iso non sucede.

Definición 1.8. Un punto $\xi \in [x_i, x_{i+1}) \subset [a, b]$, $0 \leq i \leq n-1$ denomínase cero esencial dun spline $s \in S_m(\Omega_n)$ se $s(\xi) = 0$, pero s non é idénticamente nula en $x \in [x_i, x_{i+1})$. Se $s(b) = 0$, entón b defínese como un cero esencial.

Por definición, se $[x_i, x_{i+j}]$ é un subintervalo de máxima lonxitude onde $s \equiv 0$, entón x_{i+j} é un cero esencial de s de multiplicidade m . En efecto, dado que $s \in \mathcal{C}^{m-1}([a, b])$, séguese que $s(x_{i+j}) = s'(x_{i+j}) = \dots = s^{(m-1)}(x_{i+j}) = 0$. Polo tanto tense o seguinte resultado:

Teorema 1.9 (Teorema dos ceros). *Un spline $s \in S_m(\Omega_n)$ pode ter como moito $m+n-1$ ceros esenciais en $[a, b]$, onde cada cero conta de acordo á súa multiplicidade.*

Demostración. Sexa r o número de ceros esenciais de s en $[a, b]$. Polo teorema de Rolle, $s^{(m-1)} \in S_1(\Omega_n)$ e ten polo menos $r - (m-1) = r - m + 1$ ceros esenciais. Agora o spline lineal a cachos $s^{(m-1)}$ é continuo e pode ter como moito n ceros esenciais en $[a, b]$. Concluimos entón que $r - m + 1 \leq n$, logo $r \leq m + n - 1$. \square

Corolario 1.10. *Esta cota $r \leq m + n - 1$ é óptima.*

Demostración. En efecto, basta ver que $r = m + n - 1$ nalgún caso. Facemos uso do exemplo presente en [4, p.232]:

$$s(x) = \left(\frac{x-a}{x_1-a} \right)^m + \sum_{i=1}^{n-1} b_i (x-x_i)_+^m,$$

cuxos coeficientes defínense de forma recursiva da seguinte forma:

$$b_i := \frac{1}{(x_{i+1} - x_i)^m} \left[(-1)^i - \left(\frac{x_{i+1} - a}{x_1 - a} \right)^m - \sum_{j=1}^{i-1} b_j (x_{i+1} - x_j)^m \right], \text{ para } i = 1, \dots, n-1.$$

Temos que $s(x_j) = (-1)^{j-1}$ para $j = 1, \dots, n$, logo s ten polo menos un cero en cada subintervalo (x_j, x_{j+1}) , $1 \leq j \leq n-1$. Ademais, $x = a$ é un cero de orde m . Deste xeito téñense $m+n-1$ ceros esenciais en $[a, b]$. \square

Observación 1.11. *O teorema dos ceros amosa que, respecto aos ceros esenciais dun spline, $s \in S_m(\Omega_n)$ compórtase como un polinomio do espazo \mathbb{P}_{m+n-1} de igual dimensión $(m+n)$.*

Seguidamente presentamos un corolario que establece unha mellor cota que empregaremos algo máis adiante.

Corolario 1.12. *Se un spline $s \in S_m(\Omega_n)$ é tal que $s(x) = 0$ para $x \in [x_0, x_p]$ e para $[x_q, x_n]$, $0 < p < q < n$ e $q-p \geq m+1$, e non é idénticamente nulo noutro subintervalo, entón o número r de ceros esenciais de s en (x_p, x_q) verifica a cota máis estrita seguinte:*

$$r \leq q - (p + m + 1).$$

Demostración. Sexa $\Omega_{(q-p)} := \{x_p, \dots, x_q\}$. Aplicando o teorema dos ceros ao spline $s|_{[x_p, x_q]}$ do espazo $S_m(\Omega_{(q-p)})$, temos que o número de ceros esenciais deste spline en $[x_p, x_q]$ verifica $r \leq q - p + m - 1$. Dado que $s(x_p) = s'(x_p) = \dots = s^{(m-1)}(x_p) = 0$ e $s(x_q) = s'(x_q) = \dots = s^{(m-1)}(x_q) = 0$, x_p e x_q son dous ceros de orde m . Deste xeito, s pode ter como moito $r \leq q - p + m - 1 - 2m = q - (p + m + 1)$ ceros en (x_p, x_q) . \square

1.3. B-splines

Na sección anterior demos unha base do espazo $S_m(\Omega_n)$ formada por polinomios de grao $\leq m$ e por funcións potencia “cortadas polo lado esquerdo” $q_{m,i}, i = 1, \dots, n-1$. Nesta sección presentamos unha base alternativa moito mellor adaptada ao cálculo do spline de interpolación que veremos no capítulo seguinte. Os B-spline foron introducidos por I. J. Schoenberg en 1946 (ver [8]) co nome de “Basic splines curves”. A vantaxe esencial desta base é que os B-splines teñen un soporte pequeno constituído por $m+1$ intervalos consecutivos. Para estudalos, consideremos a partición Ω_n como un subconxunto dunha partición infinita $\Omega_\infty = \{x_i\}_{i \in \mathbb{Z}}$ tal que verifique $x_i < x_{i+1}$, para todo $i \in \mathbb{Z}$; $\lim_{i \rightarrow -\infty} x_i = -\infty$ e $\lim_{i \rightarrow \infty} x_i = \infty$. Os nós x_{-1}, x_{-2}, \dots e x_{n+1}, x_{n+2}, \dots carecen de importancia práctica.

Teorema 1.13 (Existencia e unicidade de B-splines normalizados). *Para cada $i \in \mathbb{Z}$ existe un único spline $s \in S_m(\Omega_\infty)$ tal que*

$$s(x) = 0, \text{ si } x \in (-\infty, x_i) \cup [x_{i+m+1}, \infty)$$

e tal que verifica a seguinte condición de normalización:

$$\int_{-\infty}^{\infty} s(x) dx = \int_{x_i}^{x_{i+m+1}} s(x) dx = 1$$

Este spline denomínase B-spline, e denótase por $B_{m,i}^*$.

Demostración. Utilizando o teorema 1.7 e a expresión (1.1) para $n = q + 1$, onde $q \in \mathbb{Z}$ é un enteiro positivo aínda por determinar, se consideramos $x \in [x_{i-1}, x_{i+m+2}]$, posto que $s(x) = 0$ para todo $x < x_i$, é evidente que a parte polinómica de s vai ser idénticamente 0. Polo tanto, podemos escribir s da seguinte forma:

$$s(x) = \sum_{j=0}^q b_j (x - x_{i+j})_+^m.$$

Tomemos en particular $q = m + 1$. Sabendo que $s(x) = 0$ para $x \geq x_{i+m+1}$, e nesa parte os $(x - x_{i+j})_+^m = (x - x_{i+j})^m$, tense que os coeficientes b_0, \dots, b_{m+1} son solución do sistema de ecuacións:

$$\sum_{j=0}^{m+1} b_j (x - x_{i+j})^m = 0, \text{ para } x \geq x_{i+m+1}.$$

Facendo o desenvolvemento do binomio obtemos a seguinte expresión:

$$\sum_{j=0}^{m+1} b_j \left\{ \sum_{k=0}^m (-1)^k \binom{m}{k} x^{m-k} x_{i+j}^k = 0 \right\} = \sum_{k=0}^m \sum_{j=0}^{m+1} b_j (-1)^k \binom{m}{k} x^{m-k} x_{i+j}^k = 0.$$

Dado que a igualdade é certa para todo $x \geq x_{i+m+1}$, terase que os coeficientes de x son nulos; é dicir:

$$\sum_{j=0}^{m+1} b_j (-1)^k \binom{m}{k} x^{m-k} x_{i+j}^k = 0, \quad k = 0, \dots, m;$$

de onde deducimos que:

$$\sum_{j=0}^{m+1} b_j x_{i+j}^k = 0, \quad k = 0, \dots, m.$$

Polo tanto estamos ante un sistema de $m + 1$ ecuacións e $m + 2$ incógnitas. A última ecuación para que a solución sexa única vén imposta pola condición de normalización:

$$\begin{aligned} \int_{x_i}^{x_{i+m+1}} \sum_{j=0}^{m+1} b_j (x - x_{i+j})_+^m dx &= \sum_{j=0}^{m+1} \int_{x_{i+j}}^{x_{i+m+1}} b_j (x - x_{i+j})^m dx \\ &= \sum_{j=0}^{m+1} b_j \frac{(x_{i+m+1} - x_{i+j})^{m+1}}{m+1} = 1. \end{aligned}$$

Facendo un razoamento análogo ao anterior co desenvolvemento do binomio, esta igualdade pódese transformar en:

$$\sum_{j=0}^{m+1} b_j x_{i+j}^{m+1} = (-1)^{m+1} (m+1).$$

Xuntando todas as ecuacións e expresándoas en forma matricial, o sistema tería o seguinte aspecto:

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ x_i & x_{i+1} & \dots & x_{i+m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_i^m & x_{i+1}^m & \dots & x_{i+m+1}^m \\ x_i^{m+1} & x_{i+1}^{m+1} & \dots & x_{i+m+1}^{m+1} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \\ b_{m+1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ (-1)^{m+1}(m+1) \end{pmatrix}. \quad (1.2)$$

Claramente estamos ante un sistema con matriz de tipo Vandermonde, e polo tanto o sistema ten solución única. \square

Observación 1.14. *Percibamos que se na demostración anterior tomamos $q \leq m$, entón a condición $s(x) = 0$ en $(-\infty, x_i) \cup (x_{i+m+1}, +\infty)$ implica que $s \equiv 0$. En efecto, neste caso o sistema (1.2) ten polo menos unha columna menos, co que ten $(m+1)$ ecuacións homoxéneas con $(q+1) \leq (m+1)$ incógnitas b_0, \dots, b_q . Dado que a matriz de tipo Vandermonde ten rango máximo, dedúcese que $b_0 = b_1 = \dots = b_q = 0$ é a única solución, o que significa que non hai ningún spline de grao m (agás $s = 0$) cuxo soporte sexa un subconxunto propio do soporte do B-spline do mesmo grao. Dito doutra maneira, queda claro que $q+1 \geq m+1$*

Acabamos de probar que os B-splines existen e coa condición de normalización son únicos. A continuación imos estudar a súa definición formal e algunhas das súas propiedades. Recordamos o concepto de diferenzas divididas que nos facilitan a definición alternativa:

Definición 1.15. Dados $n+1$ puntos distintos x_0, x_1, \dots, x_n e os $n+1$ valores asociados $f(x_0), \dots, f(x_n)$; chámanse diferenzas divididas de primeiro orde aos n seguintes valores:

$$f[x_i, x_{i+1}] := \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}, \quad i = 0, 1, \dots, n-1.$$

Supoñendo que temos definidas as diferenzas divididas de orde $1, 2, \dots, m-1$, defínense as diferenzas divididas de orden m da seguinte forma:

$$f[x_i, \dots, x_{i+m}] = \frac{f[x_{i+1}, \dots, x_{i+m}] - f[x_i, \dots, x_{i+m-1}]}{x_{i+m} - x_i}, \quad i = 0, 1, \dots, n-m.$$

Teorema 1.16. *As diferenzas divididas verifican a seguinte igualdade:*

$$f[x_0, \dots, x_m] = \sum_{j=0}^m \frac{f(x_j)}{\prod_{\substack{i=0 \\ i \neq j}}^m (x_j - x_i)}.$$

Demostración. Probarémolo por indución sobre m . Para $m = 1$ é evidente:

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0}$$

Supoñamos entón que se verifica para $m = 1, \dots, r - 1$ e e probarémolo para $m = r$:

$$\begin{aligned}
 f[x_0, \dots, x_r] &= \frac{f[x_1, \dots, x_r] - f[x_0, \dots, x_{r-1}]}{x_r - x_0} \\
 &= \frac{1}{x_r - x_0} \left\{ \sum_{j=1}^r \frac{f(x_j)}{\prod_{\substack{i=1 \\ i \neq j}}^r (x_j - x_i)} - \sum_{j=0}^{r-1} \frac{f(x_j)}{\prod_{\substack{i=0 \\ i \neq j}}^{r-1} (x_j - x_i)} \right\} \\
 &= \frac{1}{x_r - x_0} \left\{ \frac{f(x_r)}{\prod_{i=1}^{r-1} (x_r - x_i)} + \sum_{j=1}^{r-1} \frac{(x_j - x_0)f(x_j) - (x_j - x_r)f(x_j)}{\prod_{\substack{i=0 \\ i \neq j}}^r (x_j - x_i)} - \frac{f(x_0)}{\prod_{i=1}^{r-1} (x_0 - x_i)} \right\} \\
 &= \frac{1}{x_r - x_0} \left\{ \frac{f(x_r) - f(x_0)}{\prod_{i=1}^{r-1} (x_r - x_i)} + \sum_{j=1}^{r-1} \frac{(x_r - x_0)f(x_j)}{\prod_{\substack{i=0 \\ i \neq j}}^r (x_j - x_i)} \right\} \\
 &= \sum_{j=0}^r \frac{f(x_j)}{\prod_{\substack{i=0 \\ i \neq j}}^r (x_j - x_i)}.
 \end{aligned}$$

□

Os seguintes son resultados clásicos de interpolación polinómica (ver [4] e [12])

Teorema 1.17 (Fórmula de Newton para o erro de interpolación). *Sexan x_0, x_1, \dots, x_n ($n + 1$) puntos distintos, $f(x_0), \dots, f(x_n)$ os valores asociados e $p_n \in \mathbb{P}_n = \langle 1, x, \dots, x^n \rangle$ o único polinomio de grao $\leq n$ tal que $p(x_i) = f(x_i)$, $i = 0, 1, \dots, n$. Tense:*

1.

$$p_n(x) = f(x_0) + f[x_0, x_1](x - x_0) + \dots + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1)\dots(x - x_{n-1}). \quad (1.3)$$

2.

$$p_n(x) = \sum_{i=0}^n y_i l_{n,i}(x), \quad (1.4)$$

onde $l_{n,i}$ é o polinomio de Lagrange de grao n tal que $l_{n,i}(x_j) = \delta_{ij}$, $0 \leq i, j \leq n$:

$$l_{n,i}(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \quad (1.5)$$

3. Para $x \neq x_i$, $i = 0, \dots, n$, tense a seguinte expresión para o erro de interpolación:

$$\mathcal{R}_n(x) = f(x) - p_n(x) = f[x_0, x_1, \dots, x_n, x](x - x_0)(x - x_1)\dots(x - x_n). \quad (1.6)$$

4. Sexa $[a, b]$ un intervalo que contén aos nós de interpolación x_0, x_1, \dots, x_n e $f \in \mathcal{C}^{n+1}([a, b])$ unha función. Entón, para todo $x \in [a, b]$, $x \neq x_i$, $i = 0, \dots, n$, existe $\xi_x \in (a, b)$ tal que:

$$f[x_0, x_1, \dots, x_n, x] = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}. \quad (1.7)$$

Este resultado é coñecido co nome de Cauchy-Peano.

Con esta notación introducimos a seguinte definición:

Definición 1.18. O B-spline $B_{m,i}$ de grado m correspondente ao nó x_i no conxunto de nós Ω_∞ defínese como:

$$B_{m,i}(x) := (x_{i+m+1} - x_i)q_x^m[x_i, \dots, x_{i+m+1}],$$

onde

$$q_x^m : t \in \mathbb{R} \rightarrow q_x^m(t) = (t - x)_+^m = \begin{cases} (t - x)^m, & \text{se } t \geq x; \\ 0, & \text{se } t < x. \end{cases}$$

Para ver que $B_{m,i}$ é un spline basta con empregar a fórmula para as diferenzas divididas obtida na proposición anterior e polo tanto temos:

$$B_{m,i}(x) = (x_{i+m+1} - x_i) \sum_{j=i}^{i+m+1} \left(\prod_{\substack{l=i \\ l \neq j}}^{i+m+1} (x_j - x_l) \right)^{-1} (x_j - x)_+^m, \quad x \in \mathbb{R} \quad (1.8)$$

que polo teorema 1.7 pertence a $S_m(\Omega_n)$.

Nesta definición de $B_{m,i}$ emprégase unha normalización distinta da usada na proposición anterior, é dicir, que $\int_{-\infty}^{+\infty} B_{m,i} \neq 1$. A continuación probamos que agás un factor os splines $B_{m,i}$ coinciden cos $B_{m,i}^*$ definidos polo teorema 1.13.

Proposición 1.19. *Salvo un factor de normalización, o spline $B_{m,i}$ coincide co B-spline $B_{m,i}^*$.*

Demostración. En efecto, dado que $B_{m,i}^*$ ten soporte mínimo $[x_i, x_{i+m+1}]$ (visto na observación 1.14), basta ver que $B_{m,i}$ ten soporte en $[x_i, x_{i+m+1}]$. Consideremos $x < x_i$. Entón $q_x^m \in \mathbb{P}_m$ respecto a t , e a diferenza cociente de orde $m + 1$, $q_x^m[x_i, \dots, x_{i+m+1}] = 0$ pola propiedade (1.7) das diferenzas divididas. Por outra parte, se $x \geq x_{i+m+1}$, tense $q_x^m(t) = 0$. \square

Como consecuencia do teorema dos ceros, procedemos a ver que $B_{m,i}$ non se anula en (x_i, x_{i+m+1}) .

Proposición 1.20. $B_{m,i}(x) > 0$ para todo $x \in (x_i, x_{i+m+1})$.

Demostración. Polo teorema dos ceros sabemos que $B_{m,i}$ ten como moito $n + m - 1 \geq 2m$ ceros esenciais en $[x_i, x_{i+m+1}]$. Posto que os puntos x_i e x_{i+m+1} son ceros de orde m , xa que $B_{m,i} \in \mathcal{C}^{m-1}(\mathbb{R})$ e o spline é identicamente nulo en $(-\infty, x_i)$ e en $(x_{i+m+1}, +\infty)$, entón tense que non pode haber ningún outro cero en (x_i, x_{i+m+1}) . En particular, no intervalo (x_{i+m}, x_{i+m+1}) sabemos que $B_{m,i}(x) = (x_{i+m+1} - x_i) \left(\prod_{l=i}^{i+m} (x_{i+m+1} - x_l) \right)^{-1} (x_{i+m+1} - x)^m > 0$. \square

Se consideramos o espazo $S_m(\Omega_n)$ de splines de grado m definidos no intervalo $[x_0, x_n]$, da anterior proposición somos quen de afirmar que no conxunto de B-splines, unicamente as $n + m$ funcións $B_{m,-m}, \dots, B_{m,n-1}$ teñen valores distintos de 0 en $[x_0, x_n]$. Estas $n + m$ funcións serán as que conformen unha nova base de $S_m(\Omega_n)$, para o cal é suficiente establecer que son linealmente independentes.

Proposición 1.21. *Os B-splines $B_{m,-m}, \dots, B_{m,n-1}$ son linealmente independentes e, polo tanto, unha base do espazo $S_m(\Omega_n)$.*

Demostración. En efecto, a independencia lineal tense da ecuación:

$$s(x) = \beta_{-m}B_{m,-m}(x) + \dots + \beta_{n-1}B_{m,n-1}(x) = 0, \text{ para todo } x \in [x_0, x_n],$$

a cal se verifica unicamente se $\beta_{-m} = \dots = \beta_{n-1} = 0$. Para probalo, consideremos un nó adicional $x_{-m-1} < x_{-m}$. Sabemos que $s(x) = 0$ para $x \in [x_{-m-1}, x_{-m}]$. Se $s(x) = 0$ en $[x_0, x_1]$, entón facendo uso do corolario 1.12 aplicado ao intervalo $[x_{-m-1}, x_1]$, tense $q = 0$ e $p = -m$, co cal $q - p = m < m + 1$. Pola minimalidade do soporte dos B-splines recollida na observación 1.14, de aquí concluímos que $s(x) = 0$ para todo $x \in [x_{-m-1}, x_1]$.

Unha vez probado que $s(x) = 0$ en $[x_0, x_n]$ é equivalente a que $s(x) = 0$ en todo $[x_{-m}, x_n]$, estudar a independencia lineal dos B-splines $B_{m,-m}, \dots, B_{m,n-1}$ en $[x_{-m}, x_n]$ é análogo a estudala en $[x_0, x_n]$. Vexamos entón como probalo.

Comezando no extremo esquerdo, vemos que o único B-spline que non se anula en $[x_{-m}, x_{-m+1}]$ é $B_{m,-m}$, logo a ecuación $s(x) = 0$ pasa a $\beta_{-m}B_{m,-m}(x) = 0$, e así $\beta_{-m} = 0$. Procedendo de forma recursiva, para cada $i \in \{-m, \dots, n-1\}$, observamos que no subintervalo, $[x_i, x_{i+1}]$ somos quen de afirmar que $\beta_i = 0$, chegando así ao resultado. \square

Podemos reformular a proposición anterior do xeito equivalente seguinte:

Teorema 1.22 (Teorema de representación). *Para todo spline $s \in S_m(\Omega_n)$ no intervalo $[x_0, x_n]$ existe unha expresión única en termos de B-splines:*

$$s(x) = \sum_{i=-m}^{n-1} \alpha_i B_{m,i}(x), \quad \alpha_i \in \mathbb{R}, \quad \forall x \in [x_0, x_n]. \quad (1.9)$$

A continuación presentamos outras propiedades adicionais dos B-splines, que non demostraremos (ver [4])

Proposición 1.23. *Verifícase:*

1. *Os splines $B_{m,i}$ forman unha partición da unidade é dicir,*

$$\sum_{i \in \mathbb{Z}} B_{m,i}(x) = 1 \text{ para todo } x \in \mathbb{R}.$$

2.

$$B_{m,i}(x) = \frac{x - x_i}{x_{i+m} - x_i} B_{m-1,i}(x) + \frac{x_{i+m+1} - x}{x_{i+m+1} - x_{i+1}} B_{m-1,i+1}(x) \quad (1.10)$$

3.

$$B'_{m,i}(x) = m \left(\frac{B_{m-1,i}}{x_{i+m} - x_i} - \frac{B_{m-1,i+1}}{x_{i+m+1} - x_{i+1}} \right)$$

A fórmula recursiva (1.10) e a expresión directa (1.8) permiten xerar algoritmos moi rápidos e efectivos para os cálculos dos B-splines. O cálculo recursivo comezará co caso trivial dos B-splines constantes $B_{0,i}$ que se definen do seguinte xeito:

$$B_{0,i}(x) = \begin{cases} 1, & \text{se } x \in [x_i, x_{i+1}) \\ 0, & \text{noutro caso} \end{cases}$$

Na práctica resultaranos de conveniencia ter as expresións explícitas dos B-splines máis habituais, tanto en mallas arbitrarias como con nós equidistantes. Por ese motivo presentámolas a continuación. Poden verificarse con calquera das fórmulas dadas anteriormente.

Os B-splines lineais son funcións continuas, polinomios de grao ≤ 1 a cachos en $[x_i, x_{i+2}]$ e que se anulan fóra dese intervalo. A maiores han de verificar que $B_{1,i-1}(x) + B_{1,i}(x) = 1$ para todo $x \in [x_i, x_{i+1}]$. Con estas condicións tense:

$$B_{1,i}(x) = \begin{cases} 0, & \text{se } x < x_i, \\ \frac{x - x_i}{x_{i+1} - x_i}, & \text{se } x_i \leq x < x_{i+1}, \\ \frac{x_{i+2} - x}{x_{i+2} - x_{i+1}}, & \text{se } x_{i+1} \leq x < x_{i+2}, \\ 0, & \text{se } x \geq x_{i+2}, \end{cases}$$

Con nós equidistantes: $x_i = x_0 + ih$ ($h > 0$), os B-splines quedan:

$$B_{1,i}(x) = \frac{1}{h} \begin{cases} 0, & \text{se } x < x_i, \\ x - x_i, & \text{se } x_i \leq x < x_{i+1}, \\ x_{i+2} - x, & \text{se } x_{i+1} \leq x < x_{i+2}, \\ 0, & \text{se } x \geq x_{i+2}. \end{cases}$$

Observación 1.24. *Apréciase que con esta base de $S_1(\Omega_n)$, o único spline de grao 1, $s \in S_m(\Omega_n)$, que verifica $s(x_i) = f(x_i)$, $i = 0, \dots, n$ é:*

$$s(x) = \sum_{i=-1}^{n-1} f(x_i) B_{1,i}(x)$$

Deste xeito temos outra forma de resolver o problema de interpolación lineal a cachos.

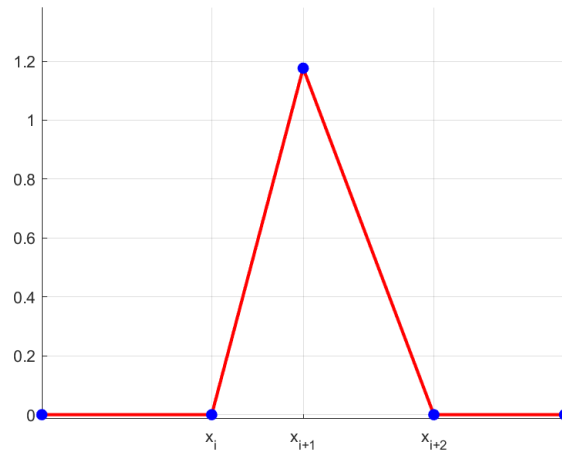


Figura 1.6: B-spline linear $B_{1,i}(x)$

Considerando por simplicidade os nós equidistantes, podemos empregar a fórmula de recorrência para obter $B_{2,i}$ a partir de $B_{1,i}$ e $B_{1,i+1}$. Alternativamente também podemos considerar que o suporte está contido em $[x_i, x_{i+3}]$, logo compõe-se de 3 cachos de parábola pegados de xeito que globalmente sexa clase 1. Fagámolo do xeito que o fagamos, chegamos á seguinte expresión:

$$B_{2,i}(x) = \frac{1}{2h^2} \begin{cases} 0, & \text{se } x < x_i, \\ (x - x_i)^2, & \text{se } x_i \leq x < x_{i+1}, \\ h^2 + 2h(x - x_{i+1}) - 2(x - x_{i+1})^2, & \text{se } x_{i+1} \leq x < x_{i+2}, \\ (x_{i+3} - x)^2, & \text{se } x_{i+2} \leq x < x_{i+3}, \\ 0, & \text{se } x \geq x_{i+3}. \end{cases}$$

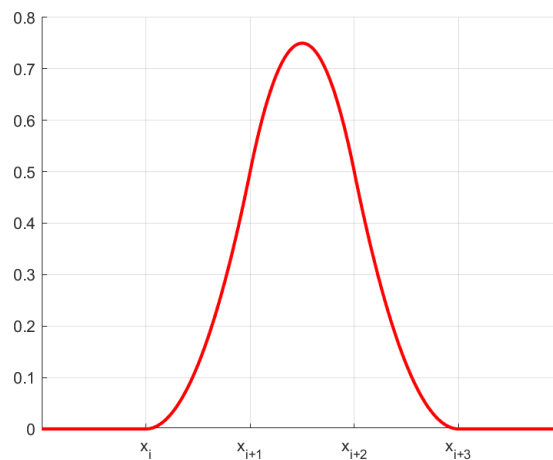


Figura 1.7: $B_{2,i}(x)$ con nós equidistantes.

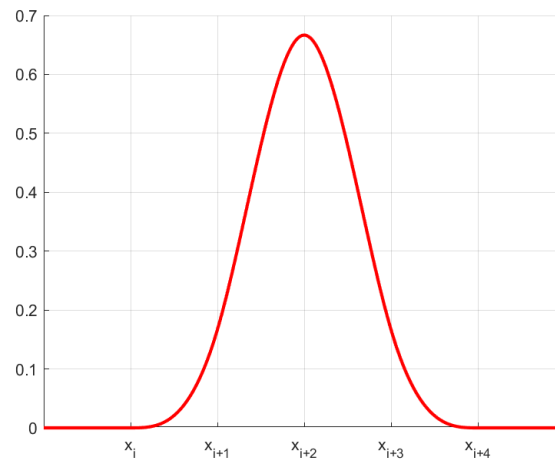


Figura 1.8: $B_{3,i}(x)$ com nós equidistantes.

Do mesmo xeito, o B-spline cúbico para nós equidistantes vén dado pola seguinte expresión:

$$B_{3,i}(x) = \frac{1}{6h^3} \begin{cases} 0, & \text{se } x < x_i, \\ (x - x_i)^3, & \text{se } x_i \leq x < x_{i+1}, \\ h^3 + 3h^2(x - x_{i+1}) + 3h(x - x_{i+1})^2 - 3(x - x_{i+1})^3, & \text{se } x_{i+1} \leq x < x_{i+2}, \\ h^3 + 3h^2(x_{i+3} - x) + 3h(x_{i+3} - x)^2 - 3(x_{i+3} - x)^3, & \text{se } x_{i+2} \leq x < x_{i+3}, \\ (x_{i+4} - x)^3, & \text{se } x_{i+3} \leq x < x_{i+4}, \\ 0, & \text{se } x \geq x_{i+4}. \end{cases} \quad (1.11)$$

Capítulo 2

Interpolación con splines cúbicos

Este capítulo abordaremos o problema de interpolación con splines cúbicos. Nesa liña, presentaremos a formulación do problema, probaremos a existencia e unicidade do spline de interpolación e daremos un algoritmo para o seu cálculo empregando B-splines. Ademais, programaremos este algoritmo en MATLAB e ilustraremos o seu uso con algúns exemplos e comparativas coa interpolación polinómica. Guiarémonos polas referencias [4] e [12]

2.1. Formulación do problema. Condicións nos extremos

Unha vez estudado o espazo $S_m(\Omega_n)$ no capítulo anterior, entendemos que debemos ter coitado á hora de formular problemas de interpolación con splines para garantir a unicidade da solución. Centrarémonos nos splines lineais, cadráticos e cúbicos, pois son os máis empregados na práctica, posto que empregando polinomios dun grao baixo, permiten obter unha función interpolante regular á vez que evitan os problemas derivados de empregar polinomios de grao elevado.

Dados os $n + 1$ nós $a = x_0 < x_1 < \dots < x_n = b$ da partición Ω_n e os valores asociados $y_i = f(x_i)$, $i = 0, \dots, n$, chámase spline de interpolación de grao $m \geq 1$ a unha función $s : [a, b] \rightarrow \mathbb{R}$ tal que:

$$s \in S_m(\Omega_n), \quad s(x_i) = f(x_i), \quad i = 0, \dots, n.$$

Dado que $S_m(\Omega_n)$ ten dimensión $(m + n)$ e lle engadimos $n + 1$ condicións, podemos deducir que para obter unha solución única no problema de interpolación, dalgunha forma deberemos impoñer outras restricións para facernos cargo dos $(m - 1)$ graos adicionais. Habitualmente estas condicións impoñense nos nós extremos, o que fai que os splines de grao impar sexan os máis empregados na práctica, posto que necesitan impoñer un número par de condicións, o que motiva

que se imponían de forma simétrica en cada nó extremo.

Posto que o caso lineal xa o tratamos (xa que non precisa de condicións adicionais), centrarémonos nos splines cúbicos. A continuación, presentamos tres maneiras de engadilas que resulten en problemas de interpolación ben definidos:

i) Interpolación con condicións de extremos Hermite/incrustados.

Atopar $s \in S_3(\Omega_n)$ tal que:

- $s(x_i) = f(x_i)$ para $i = 0, \dots, n$; e
- $s'(x_0) = f'(x_0) = y'_0$ e $s'(x_n) = f'(x_n) = y'_n$.

Neste caso o spline dise incrustado ou completo xeneralizando o caso $s'(x_0) = s'(x_n) = 0$ (xenuinamente incrustado).

ii) Interpolación con condicións de extremos naturais.

Atopar $s \in S_3(\Omega_n)$ tal que:

- $s(x_i) = f(x_i)$ para $i = 0, \dots, n$; e
- $s''(x_0) = f''(x_0) = y''_0$ e $s''(x_n) = f''(x_n) = y''_n$.

Neste caso o spline dise natural xeneralizando o caso $s''(x_0) = s''(x_n) = 0$ (xenuinamente natural).

iii) Interpolación con condicións de extremos periódicos.

Atopar $s \in S_3(\Omega_n)$ tal que:

- $s(x_i) = f(x_i)$ para $i = 0, \dots, n$; e
- $s'(x_0) = s'(x_n)$ e $s''(x_0) = s''(x_n)$.

Deseguido probamos que os problemas teñen solución única á vez que obtemos unha forma de calculala.

Teorema 2.1. *Os problemas de interpolación anteriores teñen solución única.*

Demostración. Dados $f(x_0, \dots, f(x_n))$, o problema de interpolación consiste en:

Atopar $s \in S_3(\Omega_n)$ que verifique $s(x_i) = f(x_i)$, $i = 0, \dots, n$,

ou o que é o mesmo,

$$\begin{aligned} s|_{[x_i, x_{i+1}]} &= s_i \in \mathbb{P}_3, \quad i = 0, \dots, n-1, \\ s(x_i) &= f(x_i), \quad i = 0, \dots, n, \\ s, s' \text{ e } s'' &\text{ continuas en } [x_0, x_n]. \end{aligned}$$

Esto pode ser rescrito da forma que segue:

$$\begin{aligned} s_i &\in \mathbb{P}_3, \quad i = 0, \dots, n-1, \\ s_0(x_0) &= f(x_0), \\ s_{i-1}(x_i) &= s_i(x_i) = f(x_i), \quad i = 1, \dots, n-1, \\ s_{n-1}(x_n) &= f(x_n), \\ s'_{i-1}(x_i) &= s'_i(x_i), \quad i = 1, \dots, n-1, \\ s''_{i-1}(x_i) &= s''_i(x_i), \quad i = 1, \dots, n-1. \end{aligned}$$

O anterior pódese escribir da seguinte forma empregando as derivadas:

Existen $y'_0, y'_1, \dots, y'_n, y''_0, y''_1, \dots, y''_n$ tales que:

$$\begin{aligned} s_i &\in \mathbb{P}_3, \quad i = 0, \dots, n-1, \\ s_0(x_0) &= f(x_0), \\ s_{i-1}(x_i) &= s_i(x_i) = f(x_i), \quad i = 1, \dots, n-1, \\ s_{n-1}(x_n) &= f(x_n), \\ \\ s'_i &\in \mathbb{P}_2, \quad i = 0, \dots, n-1, \\ s'_0(x_0) &= y'_0, \\ s'_{i-1}(x_i) &= s'_i(x_i) = y'_i, \quad i = 1, \dots, n-1, \\ s'_{n-1}(x_n) &= y'_n, \\ \\ s''_i &\in \mathbb{P}_1, \quad i = 0, \dots, n-1, \\ s''_0(x_0) &= y''_0, \\ s''_{i-1}(x_i) &= s''_i(x_i) = y''_i, \quad i = 1, \dots, n-1, \\ s''_{n-1}(x_n) &= y''_n. \end{aligned} \tag{2.1}$$

Queremos ver que ao engadir as condicións dos extremos a solución deste problema é única.

Engadimos a seguinte notación:

$$h_i = x_{i+1} - x_i, \quad \Delta f(x_i) = f(x_{i+1}) - f(x_i), \quad i = 0, \dots, n-1.$$

Coa fórmula interpolación de Lagrange podemos sacar da última condición de (2.1):

$$\begin{aligned} s''_i(x) &= \frac{x - x_{i+1}}{x_i - x_{i+1}} y''_i + \frac{x - x_i}{x_{i+1} - x_i} y''_{i+1} \\ &= -\frac{1}{h_i} (x - x_{i+1}) y''_i + \frac{1}{h_i} (x - x_i) y''_{i+1}, \quad 0 \leq i \leq n-1. \end{aligned}$$

Integrando dúas veces obtemos no intervalo $[x_i, x_{i+1}]$:

$$s_i(x) = -\frac{1}{6h_i} (x - x_{i+1})^3 y''_i + \frac{1}{6h_i} (x - x_i)^3 y''_{i+1} + c_i(x - x_i) - d_i(x - x_{i+1}), \tag{2.2}$$

onde c_i e d_i son constantes de integración que deben ser tomadas de xeito que s sexa continua e verifique $s_i(x_i) = f(x_i)$ e $s_i(x_{i+1}) = f(x_{i+1})$, $0 \leq i \leq n-1$. Substituíndo somos quen de chegar a:

$$c_i = \frac{f(x_{i+1})}{h_i} - \frac{y''_{i+1}h_i}{6},$$

$$d_i = \frac{f(x_i)}{h_i} - \frac{y''_i h_i}{6}.$$

Polo tanto (2.2) quédase en:

$$s_i(x) = -\frac{1}{6h_i}(x-x_{i+1})^3 y''_i + \frac{1}{6h_i}(x-x_i)^3 y''_{i+1}$$

$$+ \left(\frac{f(x_{i+1})}{h_i} - \frac{y''_{i+1}h_i}{6} \right) (x-x_i) \quad (2.3)$$

$$- \left(\frac{f(x_i)}{h_i} - \frac{y''_i h_i}{6} \right) (x-x_{i+1}), \quad 0 \leq i \leq n-1.$$

No anterior só prestamos atención a que o spline pasase polos puntos $f(x_i)$ en x_i e $f(x_{i+1})$ en x_{i+1} axustando as constantes de integración c_i e d_i . Non obstante, xa que nos estamos limitando a estudar o intervalo $[x_i, x_{i+1}]$, non sabemos se s' é continua; é dicir, $s'_{i-1}(x_i) = s'_i(x_i)$, $1 \leq i \leq n-1$.

Derivando s_i temos:

$$s'_i(x) = -\frac{y''_i}{2h_i}(x-x_{i+1})^2 + \frac{y''_{i+1}}{2h_i}(x-x_i)^2 + \frac{\Delta f(x_i)}{h_i} - \frac{h_i}{6}(y''_{i+1} - y''_i)$$

Para garantir a continuidade de s' bastaranos escoller y''_0, \dots, y''_n de xeito que

$$s'_{i-1}(x_i) = s'_i(x_i), \quad 1 \leq i \leq n-1$$

Da fórmula que temos para $s'_i(x)$ podemos ver que para $1 \leq i \leq n-1$

$$\frac{y''_i h_{i-1}}{2} + \frac{\Delta f(x_{i-1})}{h_{i-1}} - \frac{h_{i-1}}{6}(y''_i - y''_{i-1}) = -\frac{y''_i h_i}{2} + \frac{\Delta f(x_i)}{h_i} - \frac{h_i}{6}(y''_{i+1} - y''_i)$$

Agrupando segundo os y''_i , tense para $1 \leq i \leq n-1$:

$$h_{i-1}y''_{i-1} + 2(h_i + h_{i-1})y''_i + h_i y''_{i+1} = 6 \left[\frac{\Delta f(x_i)}{h_i} - \frac{\Delta f(x_{i-1})}{h_{i-1}} \right] \quad (2.4)$$

Nótese que estamos ante un sistema de $n-1$ ecuacións e $n+1$ incógnitas y''_0, \dots, y''_n . Vexamos agora se ao impoñer cada condición dos extremos o problema ten solución única:

i) Supoñamos condicións de extremos Hermite:

$$s'(x_0) = f'(x_0) = y'_0, \quad s'(x_n) = f'(x_n) = y'_n, \quad y'_0 \text{ e } y'_n \text{ dados.}$$

Substituíndo estas restricións na fórmula de s'_i temos:

$$s'(x_n) = s'_{n-1}(x_n) = \frac{y''_n}{2h_{n-1}}(x_n - x_{n-1})^2 + \frac{\Delta f(x_{n-1})}{h_{n-1}} - \frac{h_{n-1}}{6}(y''_n - y''_{n-1}),$$

$$s'(x_0) = s'_0(x_0) = -\frac{y''_0}{2h_0}(x_0 - x_1)^2 + \frac{\Delta f(x_0)}{h_0} - \frac{h_0}{6}(y''_1 - y''_0).$$

ao termo independente. Obtemos entón un sistema lineal de $n - 1$ incógnitas:

$$\begin{pmatrix} \gamma_1 & h_1 & & & & \\ h_1 & \gamma_2 & h_2 & & & 0 \\ & h_2 & \gamma_3 & h_3 & & \\ & & \ddots & \ddots & \ddots & \\ 0 & & & h_{n-3} & \gamma_{n-2} & h_{n-2} \\ & & & h_{n-2} & \gamma_{n-1} & \end{pmatrix} \begin{pmatrix} y_1'' \\ y_2'' \\ y_3'' \\ \vdots \\ y_{n-2}'' \\ y_{n-1}'' \end{pmatrix} = \begin{pmatrix} \tilde{b}_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-2} \\ \tilde{b}_{n-1} \end{pmatrix}, \quad (2.7)$$

onde se emprega a notación:

$$\gamma_i = 2(h_i + h_{i-1}); \quad b_i = 6 \left[\frac{\Delta f(x_i)}{h_i} - \frac{\Delta f(x_{i-1})}{h_{i-1}} \right], \quad 1 \leq i \leq n-1,$$

$$\tilde{b}_1 = b_1 - h_0 y_0'', \quad \tilde{b}_{n-1} = b_{n-1} - h_{n-1} y_n''.$$

Ao igual que antes volvemos ter unha matriz tridiagonal simétrica con diagonal dominante, polo tanto é non singular e ten solución única.

iii) Supoñamos condicións de extremos periódicos:

$$s'(x_0) = s'(x_n), \quad s''(x_0) = s''(x_n) = y_0'' = y_n''.$$

Ao igual que no anterior caso teremos:

$$s'(x_n) = s'_{n-1}(x_n) = \frac{y_n''}{2h_{n-1}}(x_n - x_{n-1})^2 + \frac{\Delta f(x_{n-1})}{h_{n-1}} - \frac{h_{n-1}}{6}(y_n'' - y_{n-1}''),$$

$$s'(x_0) = s'_0(x_0) = -\frac{y_0''}{2h_0}(x_0 - x_1)^2 + \frac{\Delta f(x_0)}{h_0} - \frac{h_0}{6}(y_1'' - y_0'').$$

Igualando ambas expresións:

$$\frac{h_{n-1}}{6} y_{n-1}'' + \frac{h_{n-1}}{3} y_n'' + \frac{\Delta f(x_{n-1})}{h_{n-1}} = -\frac{h_0}{3} y_0'' - \frac{h_0}{6} y_1'' + \frac{\Delta f(x_0)}{h_0};$$

é dicir,

$$\frac{1}{3}(h_{n-1} y_n'' + h_0 y_0'') = -\frac{h_{n-1}}{6} y_{n-1}'' - \frac{h_0}{6} y_1'' - \frac{\Delta f(x_{n-1})}{h_{n-1}} + \frac{\Delta f(x_0)}{h_0}$$

Polo tanto:

$$y_0'' = y_n'' = -\frac{3}{h_{n-1} + h_0} \left[\frac{\Delta f(x_{n-1})}{h_{n-1}} - \frac{\Delta f(x_0)}{h_0} \right] - \frac{1}{2(h_{n-1} + h_0)} [h_{n-1} y_{n-1}'' + h_0 y_1'']. \quad (2.8)$$

Substituíndo en (2.4) y_0'' na primeira ecuación e y_n'' na última ecuación de (2.4) obtemos o seguinte sistema lineal:

$$\begin{pmatrix} \gamma_1 & h_1 & 0 & \dots & 0 & \delta \\ h_1 & \gamma_2 & h_2 & \dots & 0 & 0 \\ 0 & h_2 & \gamma_3 & h_3 & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & h_{n-3} & \gamma_{n-2} & h_{n-2} \\ \delta & 0 & \dots & 0 & h_{n-2} & \gamma_{n-1} \end{pmatrix} \begin{pmatrix} y_1'' \\ y_2'' \\ y_3'' \\ \vdots \\ y_{n-2}'' \\ y_{n-1}'' \end{pmatrix} = \begin{pmatrix} \tilde{b}_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-2} \\ \tilde{b}_{n-1} \end{pmatrix}, \quad (2.9)$$

onde empregamos a seguinte notaci3n:

$$\begin{aligned}\gamma_1 &= 2(h_1 + h_0) - \frac{h_0^2}{2(h_{n-1} + h_0)}, \quad \gamma_i = 2(h_i + h_{i-1}), \quad i = 2, \dots, n-2, \\ \gamma_{n-1} &= 2(h_{n-1} + h_{n-2}) - \frac{h_{n-1}^2}{2(h_{n-1} + h_0)}, \quad \delta = -\frac{h_0 h_{n-1}}{2(h_{n-1} + h_0)}, \\ b_i &= 6 \left[\frac{\Delta f(x_i)}{h_i} - \frac{\Delta f(x_{i-1})}{h_{i-1}} \right], \quad 1 \leq i \leq n-1, \\ \tilde{b}_1 &= b_1 - \frac{3h_0}{h_{n-1} + h_0} \left[\frac{\Delta f(x_0)}{h_0} - \frac{\Delta f(x_{n-1})}{h_{n-1}} \right], \\ \tilde{b}_{n-1} &= b_{n-1} - \frac{3h_n}{h_{n-1} + h_0} \left[\frac{\Delta f(x_0)}{h_0} - \frac{\Delta f(x_{n-1})}{h_{n-1}} \right].\end{aligned}$$

A3nda que non 3 tridiagonal como nos casos anteriores, segue sendo unha matriz sim3trica con diagonal estritamente dominante, logo temos garantida a existencia e unicidade da soluci3n sen importar os valores de y_0, \dots, y_n . Vex3molo:

$$|\gamma_1| = \gamma_1 = 2(h_1 + h_0) - \frac{h_0^2}{2(h_{n-1} + h_0)} > h_1 + |\delta| = h_1 + \frac{h_0 h_{n-1}}{2(h_{n-1} + h_0)},$$

e analogamente tense:

$$|\gamma_{n-1}| = \gamma_{n-1} = 2(h_{n-1} + h_{n-2}) - \frac{h_{n-1}^2}{2(h_{n-1} + h_0)} > h_{n-2} + |\delta| = h_{n-2} + \frac{h_0 h_{n-1}}{2(h_{n-1} + h_0)}.$$

□

Corolario 2.2. *O 3nico spline c3bico $s \in S_3(\Omega_n)$ que verifica $s(x_i) = f(x_i)$, $i = 0, \dots, n$ e unha das condici3ns de contorno adicionais ten o seguinte aspecto:*

$$\begin{aligned}s_i(x) &= -\frac{y_i''}{6h_i}(x - x_{i+1})^3 + \frac{y_{i+1}''}{6h_i}(x - x_i)^3 + \left(\frac{f(x_{i+1})}{h_i} - \frac{y_{i+1}'' h_i}{6} \right) (x - x_i) \\ &\quad - \left(\frac{f(x_i)}{h_i} - \frac{y_i'' h_i}{6} \right) (x - x_{i+1}), \quad 0 \leq i \leq n-1,\end{aligned}$$

onde:

i) *En condici3ns de tipo Hermite, $s'(x_0) = y_0'$, $s'(x_n) = y_n'$: $(y_1'', \dots, y_{n-1}'')^T$ 3 a soluci3n do sistema lineal (2.6) e y_0'', y_n'' dados por (2.5).*

ii) *En condici3ns naturais, $s''(x_0) = y_0''$, $s''(x_n) = y_n''$: y_0'', y_n'' dados e $(y_1'', \dots, y_{n-1}'')^T$ 3 a soluci3n do sistema lineal (2.7).*

iii) *En condici3ns peri3dicas: $(y_1'', \dots, y_{n-1}'')^T$ 3 a soluci3n do sistema lineal (2.9) e y_0'', y_n'' dados por (2.8).*

2.2. Cálculo do spline cúbico de interpolación usando B-splines

Por simplicidade, centraremos en splines con nós equidistantes, ademais de termos que engadir nós adicionais $x_{-3} < x_{-2} < x_{-1} < x_0$. Deste xeito, $x_i = x_0 + hi$ con $h = \frac{x_n - x_0}{n}$, $i = -3, \dots, n$.

Utilizando a base $\{B_{3,i} : i = -3, -2, \dots, n-1\}$, sabemos que s vén dado pola fórmula (1.9), en todos os casos as condicións de interpolación convértense en

$$\sum_{i=-3}^{n-1} \alpha_i B_{3,i}(x_j) = \sum_{i=j-3}^{j-1} \alpha_i B_{3,i}(x_j) = f(x_j) = y_j, \quad 0 \leq j \leq n \quad (2.10)$$

As correspondentes condicións de extremos son:

i) Con extremos Hermite:

$$\sum_{i=-3}^{n-1} \alpha_i B'_{3,i}(x_j) = \sum_{i=j-3}^{j-1} \alpha_i B'_{3,i}(x_j) = f'(x_j) = y'_j, \quad j = 0, n. \quad (2.11)$$

ii) Con extremos naturais:

$$\sum_{i=-3}^{n-1} \alpha_i B''_{3,i}(x_j) = \sum_{i=j-3}^{j-1} \alpha_i B''_{3,i}(x_j) = f''(x_j) = y''_j, \quad j = 0, n. \quad (2.12)$$

iii) Con extremos periódicos:

$$\begin{aligned} \sum_{i=-3}^{-1} \alpha_i B'_{3,i}(x_0) &= \sum_{i=n-3}^{n-1} \alpha_i B'_{3,i}(x_n), \\ \sum_{i=-3}^{-1} \alpha_i B''_{3,i}(x_0) &= \sum_{i=n-3}^{n-1} \alpha_i B''_{3,i}(x_n). \end{aligned} \quad (2.13)$$

Para establecer os sistemas de ecuacións (2.10), requirimos os valores de $B_{3,i}(x_j)$, $i = -3, \dots, n-1$, $j = 0, \dots, n$, así como os valores de $B'_{3,i}(x_j)$ e $B''_{3,i}(x_j)$ en $i = j-3, j-2, j-1$ cando $j = 0, n$. Utilizando a expresión (1.11) quedanos a seguinte táboa:

Agrupando as ecuación (2.10) e as correspondentes (2.11), (2.12) ou (2.13), obtemos un sistema lineal cuxo vector de incógnitas chamamos $\alpha := (\alpha_{-3}, \dots, \alpha_{n-1})^T \in \mathbb{R}^{n+3}$, que son os coeficientes do spline de interpolación na base de B-splines. Este vector α é a solución dun sistema lineal $B\alpha = b$, onde a matriz $B \in \mathcal{M}_{(n+3, n+3)}(\mathbb{R})$ e o vector $b \in \mathbb{R}^{n+3}$ teñen o seguinte aspecto:

iii) Con extremos periódicos:

$$B = \frac{1}{6} \begin{pmatrix} -\frac{3}{h} & 0 & \frac{3}{h} & 0 & \dots & 0 & \frac{3}{h} & 0 & -\frac{3}{h} \\ 1 & 4 & 1 & 0 & & & & & \\ & & \ddots & \ddots & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & & 0 & \\ & 0 & & & \ddots & \ddots & \ddots & & \\ & & & & & 1 & 4 & 1 & 0 \\ & & & & & & 1 & 4 & 1 \\ \frac{6}{h^2} & -\frac{12}{h^2} & \frac{6}{h^2} & 0 & \dots & 0 & -\frac{6}{h^2} & \frac{12}{h^2} & -\frac{6}{h^2} \end{pmatrix};$$

$$b = (0, f(x_0), f(x_1), \dots, f(x_{n-1}), f(x_n), 0)^T.$$

Utilizando as ideas anteriores, elaboramos un código en MATLAB para o cálculo e debuxo do spline de interpolación (ver Anexo). Con este código realizamos os exemplos da sección seguinte e do final do capítulo 3.

2.3. Exemplos

Comezamos amosando que cos splines obtemos unha boa aproximación da función do exemplo do capítulo 1 evitando o efecto Runge.

Exemplo 2.3. No intervalo $[-10, 10]$ consideramos a malla uniforme $x_0 < x_1 < \dots < x_n$, con $x_i = -10 + 20i/n$, $i = 0, \dots, n$. Sexa $f = 1/(1 + x^2)$.

En ambos casos vemos que obtemos splines sen comportamentos oscilatorios como si ocorría coa interpolación polinómica. No caso do spline con condicións de extremos periódicos non ten sentido comparalo coa función orixinal xa que esta non verifica as condicións de periodicidade. Aínda así vese que non é substancialmente peor aproximación da función orixinal posto que as derivadas primeira e segunda nos extremos son practicamente 0.

Outra vantaxe que teñen os splines é que permiten aproximar funcións non regulares por unha que si o sexa. Esta propiedade utilízase na práctica para o deseño de superficies aerodinámicas como carrocerías de vehículos, cascos de barcos, etc.

Exemplo 2.4. No intervalo $[-3, 3]$ consideramos unha malla uniforme, e sexa $f : \mathbb{R} \rightarrow \mathbb{R}$ a

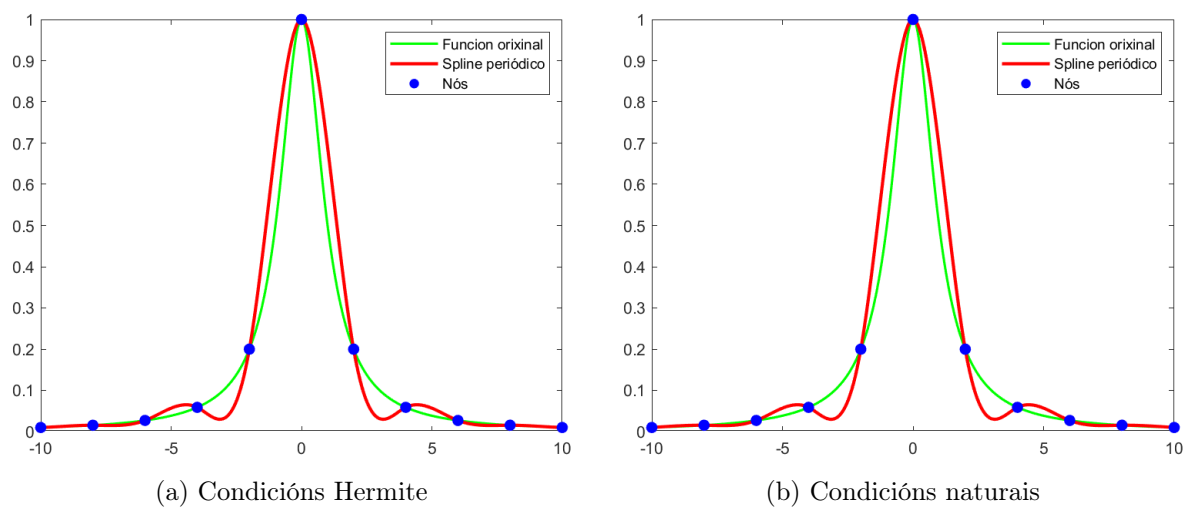


Figura 2.1: Interpolación con splines cúbicos

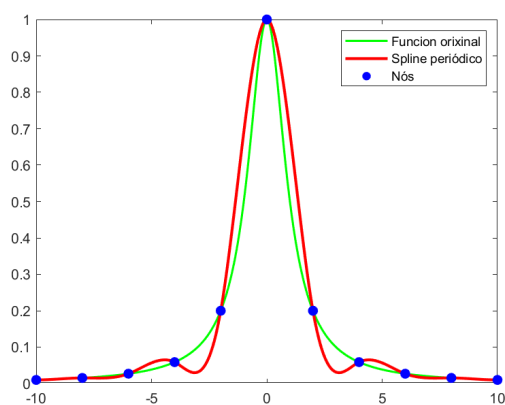


Figura 2.2: Spline interpolador con condições de contorno periódicas.

função que tem o seguinte aspecto:

$$f(x) = \begin{cases} 1, & \text{se } x < -2, \\ 3 + x, & \text{se } -2 \leq x < -1, \\ 2, & \text{se } -1 \leq x < 1, \\ 3 - x, & \text{se } 1 \leq x < 2, \\ 1, & \text{se } x \geq 2. \end{cases}$$

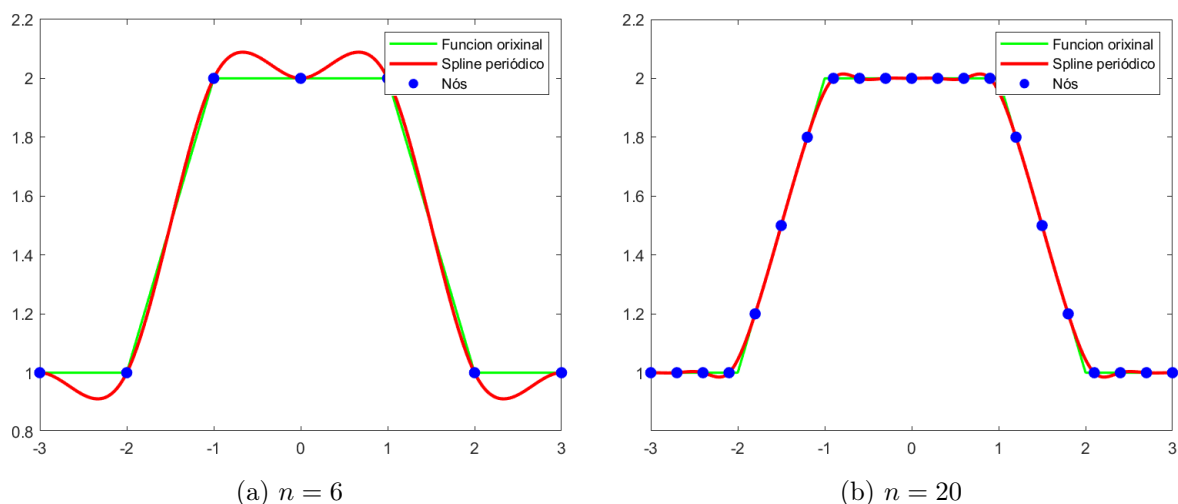


Figura 2.3: Spline interpolador con condicións de contorno Hermite

2.4. Problema xeral de interpolación con splines

Dado que o espazo $S_m(\Omega_n)$ ten dimensión $(m+n)$ é natural que nos preguntemos se é posible encontrar un spline de grao m , $s \in S_m(\Omega_n)$ que toma valores dados en $(n+m)$ puntos distintos do intervalo $[a, b]$: $y_j = f(\xi_j)$, $j = 1, \dots, n+m$, $\xi_j \in [a, b]$, $1 \leq j \leq n+m$. Este problema é equivalente ao sistema lineal de orde $(n+m)$ con incógnitas $(\alpha_{-m}, \dots, \alpha_{n-1})$:

$$\sum_{i=-m}^{n-1} \alpha_i B_{m,i}(x_j) = f(x_j) = y_j, \quad j = 1, \dots, n+m$$

En 1953 Schoenberg e Whitney probaron a existencia e unicidade se cada punto de interpolación ξ_j está no interior do soporte $B_{m,-m+j-1}$. O resultado é o seguinte:

Teorema 2.5 (Teorema xeral de interpolación con splines). *Fixados os nós $x_{-m} < \dots < x_0 < \dots < x_n < \dots < x_{m+n}$, existe un único spline $s \in S_m(\Omega_n)$ que pasa polos puntos $(\xi_1, y_1), \dots, (\xi_{m+n}, y_{m+n})$ se, e só se, $B_{m,-m+j-1}(\xi_j) \neq 0$, con $1 \leq j \leq n+m$.*

Demostración. Para $m = 0$ o resultado é obvio. Supoñamos $m \leq 1$. Empezamos probando que precisamos a condición $B_{m,-m+j-1}(\xi_j) \neq 0$. Sexa $s(x) = \sum_{i=-m}^{n-1} \alpha_i B_{mi}(x)$. Entón as condicións de interpolación imponen que

$$\sum_{i=-m}^{n-1} \alpha_i B_{mi}(\xi_j) = y_j, \quad j = 1, \dots, n+m.$$

Supoñamos que $B_{m,-m+j-1}(\xi_j) = 0$ para algún j . Entón ou ben $\xi_j \leq x_{-m+j-1}$ ou $x_j \leq \xi_j$. Do primeiro caso séguese que $B_{mi}(x) = 0$ para todo $x \leq \xi_j$ con $i \geq -m+j-1$. Á vista disto, as

primeiras j condicións de interpolación quedan:

$$\sum_{i=-m}^{-m+j-2} \alpha_i B_{mi}(\xi_k) = y_k, \quad k = 1, \dots, j$$

Estas son j ecuacións para as $(j-1)$ incógnitas $\alpha_{-m}, \dots, \alpha_{-m+j-2}$, logo non teñen solución para calquera termo independente da igualdade. No outro caso onde $x_j \leq \xi_j$, as últimas $n+m-(j-1)$ condicións de interpolación teñen a forma:

$$\sum_{i=-m+j}^{n-1} \alpha_i B_{mi}(\xi_k) = y_k, \quad j \leq k \leq n+m$$

Ao igual que antes, o número de ecuacións é maior có número de incógnitas, logo pode non haber solución.

Acabamos de probar que a condición $B_{m,-m+j-1}(\xi_j) \neq 0$, $1 \leq j \leq n+m$ é necesaria para que exista unha única solución. Vexamos agora que tamén é suficiente. En particular, demostraremos que cando se verifica esa condición, o problema de interpolación homoxéneo con $y_i = 0$, $i = 1, \dots, n+m$ só ten a solución trivial $s = 0$, logo os coeficientes que acompañan aos B-splines α_i han de ser todos 0.

Supoñamos que $B_{m,-m+j-1}(\xi_j) \neq 0$, $1 \leq j \leq n+m$, pero $s \neq 0$. Entón existe un intervalo $[x_p, x_q]$ onde, no caso de telos, s ten só ceros illados; mentres que $s(x) = 0$ en $[x_{p-1}, x_p]$ e en $[x_q, x_{q+1}]$. Do corolario 1.12 deducimos que $q-p \geq m+1$. Supoñamos que polo menos $q-(p+m)$ puntos de interpolación $\xi_p, \dots, \xi_{q-(m+1)}$ pertencen ao intervalo (x_p, x_q) . Dado que $s(\xi_j) = 0$ nestes puntos, podemos afirmar que s ten ceros illados neste intervalo. Pero aplicando o corolario antes mencionado aos nós $x_{p-1} < x_p < \dots < x_q < x_{q+1}$ danos a cota $r \leq q-(p+m+1)$, sendo r o número de ceros que o spline pode ter nese intervalo. Polo tanto $s(x) = 0$ para todo $x \in [x_p, x_q]$, e así $s = 0$ é a única solución do problema de interpolación homoxéneo. \square

Corolario 2.6. *O spline interpolador, $s \in S_m(\Omega_n)$ tal que $s(x_i) = f(x_i)$, $i = 0, \dots, n$, queda definido de forma unívoca se ademais coñecemos o seu valor en $m-1$ nós adicionais $s(\xi_i) = f(\xi_i)$, $n+2 \leq i \leq n+m$.*

Demostración. Consideremos $x_0 = \xi_1 < \xi_2 < \dots < \xi_{n+m} = x_n$, de xeito que os x_i forman parte desta nova partición. Temos que garantir que $B_{m,-m+j-1}(\xi_j) \neq 0$, $1 \leq j \leq n+m$. Pero como viamos no teorema anterior, isto equivale a que $\xi_j \leq x_{-m+j-1}$ ou $x_j \leq \xi_j$. O primeiro caso non se pode dar posto que só engadimos $m-1$ nós adicionais, e o segundo tampouco xa que $x_0 = \xi_1 < x_1$. \square

Terminamos esta sección con algúns exemplos de splines cúbicos de interpolación.

Capítulo 3

Interpolación con splines cúbicos: análise do erro

Neste capítulo estamos interesados en estimar o erro $|f(x) - s(x)|$ cometido nun punto $x \neq x_j, j = 0, \dots, n$, sendo f unha función coñecida e $s \in S_3(\Omega_n)$ o spline de interpolación de f con respecto aos nós x_0, \dots, x_n . Seguiremos as referencias [4], [7] e [12].

Sexa $[a, b] \subset \mathbb{R}$ un intervalo acoutado e $\mathcal{C}([a, b])$ o espazo de funcións continuas en $[a, b]$. Neste espazo son ben coñecidas as seguintes normas:

- Norma de Chebyshev, norma uniforme ou norma do máximo:

$$\|f\|_\infty = \max_{x \in [a, b]} |f(x)|, \quad f \in \mathcal{C}([a, b]).$$

- Norma L_2 :

$$\|f\|_2 = \left(\int_a^b [f(t)]^2 \right)^{\frac{1}{2}}, \quad f \in \mathcal{C}([a, b]).$$

3.1. O erro en splines lineais

Para estudar o erro en splines cúbicos resulta conveniente comezar analizando primeiro o erro nos splines lineais. Ademais, resultará pertinente falar indistintamente en termos de aproximación de funcións como de interpolación porque neste apartado están estritamente ligados.

Dados $n + 1$ puntos distintos (x_i, y_i) , $0 \leq i \leq n$, existe un único polinomio $p \in \mathbb{P}_n$ que p pasa por todos eles. Porén, se pedimos que p sexa de grao $< n$, non podemos garantir que o polinomio percorra todos os puntos; cando facemos interpolación polinómica e aumentamos o

grao do polinomio non está garantida unha mellor aproximación pois xa sabemos que deriva en comportamentos oscilatorios non desexados.

Pola contra, na interpolación con splines temos a posibilidade de aumentar o número de nós para mellorar a aproximación. Neste capítulo constataremos como desta forma obteremos unha mellor converxencia. Empezamos considerando o problema de interpolar unha función nun intervalo $[x_i, x_{i+1}]$ por unha recta. Neste caso, os polinomios de Lagrange asociados $l_{1,i}$ e $l_{1,i+1}$ son:

$$l_{1,i}(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}}, \quad l_{1,i+1}(x) = \frac{x - x_i}{x_{i+1} - x_i}.$$

Séguese que para cada $0 \leq i \leq n - 1$, o spline lineal que interpola f pode ser escrito como

$$\tilde{s}(x) = f(x_i)l_{1,i}(x) + f(x_{i+1})l_{1,i+1}(x), \quad x \in [x_i, x_{i+1}].$$

Sexa $f \in \mathcal{C}([a, b])$. Posto que $l_{1,i}(x) + l_{1,i+1}(x) = 1$ tense:

$$\begin{aligned} f(x) - \tilde{s}(x) &= f(x)[l_{1,i}(x) + l_{1,i+1}(x)] - f(x_i)l_{1,i}(x) - f(x_{i+1})l_{1,i+1}(x) \\ &= l_{1,i}(x)[f(x) - f(x_i)] + l_{1,i+1}(x)[f(x) - f(x_{i+1})]; \end{aligned}$$

e tendo en conta que $l_{1,i}(x) \geq 0$ e $l_{1,i+1}(x) \geq 0$, temos:

$$|f(x) - \tilde{s}(x)| \leq \max\{|f(x) - f(x_i)|, |f(x) - f(x_{i+1})|\}.$$

Co anterior deducimos que:

$$\max_{x \in [x_i, x_{i+1}]} |f(x) - \tilde{s}(x)| \leq \max_{x \in [x_i, x_{i+1}]} \max\{|f(x) - f(x_i)|, |f(x) - f(x_{i+1})|\} \quad (3.1)$$

Definición 3.1. Dada unha función $f : [a, b] \rightarrow \mathbb{R}$, $\delta > 0$, definimos o módulo de continuidade de f como

$$\omega_f(\delta) := \sup_{\substack{|x-y| \leq \delta \\ x, y \in [a, b]}} |f(x) - f(y)|$$

Observación 3.2. Nótese que en funcións lipschitzianas $\omega_f(\delta) \leq K\delta < \infty$, onde K é a constante de Lipschitz.

Substituíndo co módulo de continuidade de f en (3.1):

$$\max_{x \in [x_i, x_{i+1}]} |f(x) - \tilde{s}(x)| \leq \omega_f(|x_{i+1} - x_i|).$$

Se denotamos por $h := \max_{i=0, \dots, n-1} |x_{i+1} - x_i|$ (o tamaño da malla), temos:

$$\|f - \tilde{s}\|_\infty \leq \omega_f(h)$$

Desta desigualdade deducimos que dada $f \in \mathcal{C}([a, b])$, o spline interpolador lineal converge de forma uniforme á función cando h tende a 0. Esta cota pódese empregar cando podemos coñecer

de forma explícita o módulo da continuidade. Por exemplo, se f é Hölder continua en $[a, b]$; é dicir, $|f(x) - f(z)| \leq K|x - z|^\alpha$, $\alpha \in (0, 1)$, ou f lipschitziana ($\alpha = 1$), temos a cota do erro:

$$\|f - \tilde{s}\|_\infty \leq Kh^\alpha$$

Isto quere dicir que cando temos unha función f lipschitziana, a sucesión de splines interpoladores converxe uniformemente a f de forma lineal respecto a h .

Sexa agora $f \in \mathcal{C}^1([a, b])$. Pola identidade de Newton (teorema 1.5) podemos ver que o erro no spline interpolador $\tilde{s} \in S_1(\Omega_n)$ podémolo escribir como:

$$f(x) - \tilde{s}(x) = (x - x_i)(x - x_{i+1})f[x_{i+1}, x_i, x], \quad x \in [x_i, x_{i+1}]; \quad (3.2)$$

onde

$$f[x_{i+1}, x_i, x] = \frac{1}{x_{i+1} - x_i} (f[x_{i+1}, x] - f[x_i, x])$$

e polo tanto queda:

$$f[x_{i+1}, x_i, x] = \frac{1}{x_{i+1} - x_i} \left(\frac{f(x) - f(x_{i+1})}{x - x_{i+1}} - \frac{f(x) - f(x_i)}{x - x_i} \right)$$

Polo teorema do valor medio, existen $\eta_i, \zeta_i \in (x_i, x_{i+1})$ tales que:

$$f'(\eta_i) = \frac{f(x) - f(x_{i+1})}{x - x_{i+1}} \text{ e } f'(\zeta_i) = \frac{f(x) - f(x_i)}{x - x_i}.$$

Do anterior podemos afirmar que:

$$|f(x) - \tilde{s}(x)| \leq \frac{x_{i+1} - x_i}{4} \max_{\eta, \zeta \in [x_i, x_{i+1}]} |f'(\eta) - f'(\zeta)|, \quad x \in [x_i, x_{i+1}]$$

Finalmente, chegamos a que:

$$\|f - \tilde{s}\|_\infty \leq \frac{h}{4} \omega_{f'}(h),$$

e así vemos que converxe uniformemente en $[a, b]$.

Por outra parte, apoiándonos na observación 3.4, cando f' é lipschitziana, obtemos converxencia cadrática, xa que da expresión anterior deducimos

$$\|f - \tilde{s}\|_\infty \leq \frac{K}{4} h^2.$$

Se $f \in \mathcal{C}^2([a, b])$, polo teorema 1.17 existe $\xi \in (x_i, x_{i+1})$ tal que

$$f[x_{i+1}, x_i, x] = \frac{1}{2} f''(\xi),$$

e substituíndo no erro (3.2), tendo en conta que:

$$|(x - x_i)(x - x_{i+1})| \leq \left| \left(\frac{1}{2}(x_{i+1} - x_i) - x_i \right) \left(\frac{1}{2}(x_{i+1} - x_i) - x_{i+1} \right) \right| = \frac{1}{4}(x_{i+1} - x_i)^2$$

obtense:

$$|f(x) - \tilde{s}(x)| \leq \frac{(x_{i+1} - x_i)^2}{8} \max_{t \in [x_i, x_{i+1}]} |f''(t)|, \quad x \in [x_i, x_{i+1}],$$

o que implica que:

$$\|f - \tilde{s}\|_\infty \leq \frac{h^2}{8} \|f''\|_\infty. \quad (3.3)$$

Esta expresión amósanos que non é posible obter unha converxencia mellor que a cadrática para unha función de clase dous f arbitraria.

Se $\tilde{s} \in S_1(\Omega_n)$ é o spline que interpola f nos nós, entón

$$\|\tilde{s}\|_\infty = \max_i |\tilde{s}(x_i)| = \max_i |f(x_i)| \leq \|f\|_\infty, \quad 0 \leq i \leq n.$$

Agora, se $s \in S_1(\Omega_n)$ é un spline arbitrario, entón como $(\tilde{s} - s)$ interpola a función $(f - s)$, séguese que:

$$\|\tilde{s} - s\|_\infty \leq \|f - s\|_\infty$$

De aquí podemos deducir que

$$\|f - \tilde{s}\|_\infty = \|(f - s) - (\tilde{s} - s)\|_\infty \leq \|f - s\|_\infty + \|\tilde{s} - s\|_\infty \leq 2\|f - s\|_\infty.$$

Denotando por $E_{S_1(\Omega_n)}(f) := \min_{s \in S_1(\Omega_n)} \|f - s\|_\infty$ obtemos a seguinte cota:

$$E_{S_1(\Omega_n)}(f) \leq \|f - \tilde{s}\|_\infty \leq 2E_{S_1(\Omega_n)}(f),$$

de onde extraemos que o spline lineal interpolador é un bo substituto da mellor aproximación uniforme de f en $S_1(\Omega_n)$.

3.2. Relación integral

Proposición 3.3 (Relación integral). *Sexa $f \in C^2([a, b])$ e sexa $s \in S_3(\Omega_n)$ un spline interpolador tal que a diferenza $d(x) := f(x) - s(x)$ satisfaga a condición de contorno:*

$$s''(a)d'(a) = s''(b)d'(b).$$

Entón verifícase a seguinte igualdade:

$$\int_a^b [f''(x)]^2 dx = \int_a^b [f''(x) - s''(x)]^2 dx + \int_a^b [s''(x)]^2 dx$$

Demostración. En efecto:

$$\int_a^b [f''(x) - s''(x)]^2 dx = \int_a^b [f''(x)]^2 dx + \int_a^b \{-2f''(x)s''(x) + [s''(x)]^2\} dx$$

Bastará entón probar que:

$$2 \int_a^b \{[s''(x)]^2 - f''(x)s''(x)\} dx = 0,$$

ou o que é o mesmo:

$$\int_a^b s''(x)d''(x) dx = 0.$$

Integrando por partes (tomando $u = s''(x)$ y $dv = d''(x)$), séguese:

$$\int_a^b s''(x)d''(x) dx = s''(b)d'(b) - s''(a)d'(a) - \int_a^b s'''(x)d'(x) dx = 0.$$

Nótese que pola condición de contorno que impoñiamos os dous primeiros termos canceláanse entre si, quedándonos só esa nova integral. Integrando novamente por partes ($u = s'''(x)$ e $dv = d'(x)$), tense:

$$\int_a^b s'''(x)d'(x) dx = s'''(b)d(b) - s'''(a)d(a) - \int_a^b s^{(4)}(x)d(x) dx.$$

Xa que $d(a) = d(b) = 0$ e $s \in S_3(\Omega_n)$ implica que $s^{(4)}(x) = 0$, chégase o resultado que buscabamos. \square

Observación 3.4. *Decatémonos de que ao impoñer a condición de contorno incluimos os tres casos de interpolación:*

- A condición $d'(a) = d'(b) = 0$ correspóndese cos splines con extremos Hermite;
- A condición $s''(a) = s''(b) = 0$ se correspóndese aos splines con extremos naturais xenuíños;
- A condición $d'(a) = d'(b)$ e $s''(a) = s''(b)$ correspóndese aos splines con extremos periódicos.

A relación integral sérvenos para ver como os splines non presentan un comportamento oscilatorio, xa que esta fai patente como os splines cúbicos son as curvas interpoladoras de clase 2 que minimiza a derivada segunda. Presentamos a continuación o resultado.

Proposición 3.5. *Sexa $f \in C^2([a, b])$ e sexa $s \in S_3(\Omega_n)$ o spline interpolador con respecto a unha das condicións de extremo anteriores. Sexa $g \in C^2([a, b])$ que verifique as mesmas condicións de extremo que s . Entón, tense que:*

$$\|s''\|_2 \leq \|g''\|_2.$$

Demostración. Se g verifica as mesmas condicións que s quere dicir que $g''(a)d'_g(a) = g''(b)d'_g(b)$, onde $d_g(x) := g(x) - s(x)$. Temos que ver entón que g tamén verifica a seguinte relación integral:

$$\int_a^b [g''(x)]^2 dx = \int_a^b [g''(x) - s''(x)]^2 dx + \int_a^b [s''(x)]^2 dx,$$

pero como para isto basta garantir que $d_g(a) = d_g(b) = 0$, e como iso é así xa estaría. Quitando o primeiro término á dereita da igualdade, tense o resultado. \square

Observación 3.6. *Esta propiedade pódese interpretar xeometricamente. Denotando por κ á curvatura dunha curva plana da forma $y = g(x)$:*

$$\kappa(x) = \frac{g''(x)}{(1 + [g'(x)]^2)^{3/2}}$$

Se asumimos $|g'(x)| \ll 1$ para $x \in [a, b]$, entón o valor de $\|\kappa(x)\|_2^2 \approx \int_a^b [g''(x)]^2 dx = \|g''\|_2^2$. Da anterior proposición somos quen de afirmar que o spline cúbico interpolador s minimiza a norma da curvatura $\|\kappa\|_2^2$ fronte ao resto de curvas $g \in \mathcal{C}^2([a, b])$ que pasan polos nós.

3.3. O erro en splines cúbicos

Co visto nas seccións anteriores xa somos quen de pasar a traballar con splines cúbicos. Necesitamos previamente probar dous lemas que vinculan o spline interpolador coa función interpolada.

Lema 3.7. *Sexa $f \in \mathcal{C}^2[a, b]$ e sexa $s \in S_3(\Omega_n)$ o spline interpolador cúbico con condicións de contorno Hermite. Entón s'' é a mellor aproximación de f'' en $S_1(\Omega_n)$ respecto á norma $\|\cdot\|_2$; é dicir,*

$$\|f'' - s''\|_2 \leq \|f'' - \tilde{s}\|_2, \text{ para todo } \tilde{s} \in S_1(\Omega_n), \quad (3.4)$$

ou o que é o mesmo,

$$\|f'' - s''\|_2 \leq E_{S_1(\Omega_n)}(f''). \quad (3.5)$$

Demostración. Para evitar confusións, dada $g \in \mathcal{C}^2([a, b])$, denotaremos ao spline cúbico con condicións de contorno Hermite que o interpola como s_g .

Dado $\tilde{s} \in S_1(\Omega_n)$, definimos a función $\sigma(x) := \int_a^b (x-t)_+ \tilde{s}(t) dt$. Entón $\sigma'' = \tilde{s}$, logo $\sigma \in S_3(\Omega_n)$, o que significa que $s_\sigma = \sigma$. Aplicando a relación integral da proposición 3.3 temos:

$$\|g''\|_2^2 = \|g'' - s_g''\|_2^2 + \|s_g''\|_2^2$$

Se $f = g + \sigma$, isto quere dicir que $g = f - \sigma$, e como $s_{(f-\sigma)} = s_f - s_\sigma$, podemos escribir:

$$\|f'' - \sigma''\|_2^2 = \|f'' - \sigma'' - (s_f'' - s_\sigma'')\|_2^2 + \|s_f'' - s_\sigma''\|_2^2,$$

de onde podemos concluír que:

$$\|f'' - s_f''\|_2^2 \leq \|f'' - \tilde{s}\|_2^2,$$

para todo $\tilde{s} \in S_1(\Omega_n)$, de onde $s_f'' = s''$ é a mellor aproximación de f'' en $S_1(\Omega_n)$. \square

Consideremos $f \in \mathcal{C}^4([a, b])$, a cal pasa por $(n + 1)$ puntos $(x_0, f(x_0)), \dots, (x_n, f(x_n))$, e sexa $s \in S_3(\Omega_n)$ o spline cúbico interpolador de f con condicións de contorno Hermite. Dado un intervalo $[x_i, x_{i+1}]$ con $0 \leq i \leq n - 1$, podemos aplicar a identidade de Newton a $d := f - s$, co que obtemos:

$$\begin{aligned} d(x) &= d(x_i) + (x - x_i)d[x_i x_{i+1}] + (x - x_i)(x - x_{i+1})\frac{d''(\xi)}{2} \\ &= (x - x_i)(x - x_{i+1})\frac{d''(\xi)}{2}, \quad \xi \in (x_i, x_{i+1}); \end{aligned}$$

onde empregamos que $d(x_i) = 0$ e $d[x_i, x_{i+1}] = 0$. Da anterior igualdade deducimos que

$$|d(x)| \leq \frac{h^2}{8} \max_{t \in [x_i, x_{i+1}]} |d''(t)|, \quad \forall x \in [x_i, x_{i+1}].$$

Como isto se verifica para todo subintervalo, tendo en conta (3.5), chegamos a:

$$\|f - s\|_\infty \leq \frac{h^2}{8} \|f'' - s''\|_\infty \leq \frac{h^2}{2} E_{S_1(\Omega_n)}(f'')$$

Aplicando a cota do erro (3.3) a $f'' \in \mathcal{C}^2([a, b])$, temos:

$$E_{S_1(\Omega_n)}(f'') \leq \frac{h^2}{8} \|f^{(4)}\|_\infty,$$

co cal os splines cúbicos interpoladores de tipo Hermite con nós equidistantes verifican:

$$\|f - s\|_\infty \leq \frac{h^4}{16} \|f^{(4)}\|_\infty.$$

Co anterior acabamos de probar o teorema de estimación do erro:

Teorema 3.8. *Sexa $f \in \mathcal{C}^4([a, b])$ e $s \in S_3(\Omega_n)$ o único spline cúbico de tipo Hermite que interpola a f . Entón:*

$$\|f - s\|_\infty \leq \frac{h^4}{16} \|f^{(4)}\|_\infty.$$

Esta estimación é óptima en canto a orde do mesmo, pero a constante $\frac{1}{16}$ pode ser mellorada. De feito, tense o seguinte resultado que pode verse en [7]

Teorema 3.9 (Hall, 1968). *Sexa $f \in \mathcal{C}^4([a, b])$ e $s \in S_3(\Omega_n)$, o único spline cúbico de tipo Hermite que interpola a f . Entón:*

$$\begin{aligned} \|f - s\|_\infty &\leq \frac{5h^4}{384} \|f^{(4)}\|_\infty, \\ \|f' - s'\|_\infty &\leq \left[\frac{\sqrt{3}}{216} + \frac{1}{24} \right] h^3 \|f^{(4)}\|_\infty. \end{aligned}$$

Resultaranos máis doado probar o seguinte resultado, que non é tan potente posto que só chega á orde $h^{3/2}$, pero é válido para $f \in \mathcal{C}^2([a, b])$.

Teorema 3.10. *Sexa $f \in \mathcal{C}^2([a, b])$ e $s \in S_3(\Omega_n)$, o único spline cúbico de tipo Hermite que interpola a f . Entón:*

$$\|f - s\|_\infty \leq \frac{1}{2}h^{3/2}\|f''\|_2,$$

$$\|f' - s'\|_\infty \leq h^{1/2}\|f''\|_2.$$

Demostración. Sexa $d = f - s$. Verifícase que $d(x_j) = 0$, $j = 0, 1, \dots, n$, de modo que polo teorema de Rolle existen $\xi_1, \xi_2, \dots, \xi_n$, con $\xi_i \in (x_{i-1}, x_i)$ tal que $d'(\xi_i) = 0$, $i = 1, \dots, n$. Ademais, se fixamos $\xi_0 = x_0 = a$ e $\xi_{n+1} = x_n = b$, tense por definición de s que $d'(\xi_0) = d'(\xi_n) = 0$. Supoñamos que $t \in [x_{i-1}, x_i]$ verifica $\|d'\|_\infty = |d'(t)|$. Entón, se temos en conta que:

$$d'(t) = \int_{\xi_i}^t d''(x) dx,$$

facendo uso da desigualdade de Cauchy-Schwartz séguese:

$$\begin{aligned} \|d'\|_\infty = |d'(t)| &\leq \int_{\xi_i}^t |d''(x)| dx \leq \left[\int_{\xi_i}^t |d''(x)|^2 dx \right]^{1/2} \left[\int_{\xi_i}^t dx \right]^{1/2} \\ &\leq \left[\int_a^b |d''(x)|^2 dx \right]^{1/2} |t - \xi_i|^{1/2} \leq \|d''\|_2 h^{1/2}. \end{aligned}$$

Con isto podemos reescribir a relación integral da seguinte maneira:

$$\|f''\|_2^2 = \|f'' - s''\|_2^2 + \|s''\|_2^2 = \|d''\|_2^2 + \|s''\|_2^2,$$

co cal vemos que:

$$\|d''\|_2 \leq \|f''\|_2.$$

Polo tanto, aplicando esta cota na anterior desigualdade:

$$\|d'\|_\infty \leq \|f''\|_2 h^{1/2}$$

Sexa agora $t \in [x_{i-1}, x_i]$ de forma que $\|d\|_\infty = |d(t)|$, existen dúas posibilidades:

$$|x_{i-1} - t| \leq \frac{h}{2} \text{ ou } |x_i - t| \leq \frac{h}{2}.$$

Supoñamos a segunda (no outro caso o razoamento é análogo). Entón:

$$d(t) = \int_{x_j}^t d'(x) dx,$$

de onde obtemos:

$$\|d\|_\infty = |d(t)| \leq \int_{x_i}^t |d'(x)| dx \leq \|d'\|_\infty |x_i - t| \leq \frac{h}{2} \|d'\|_\infty \leq \frac{1}{2} \|f''\|_2 h^{3/2}.$$

□

Corolario 3.11. *Sexa $\Omega_n = \{a = x_0^{(n)} < x_1^{(n)} < \dots < x_n^{(n)} = b\}$ unha partición con $n + 1$ nós equidistantes ($x_i = x_0 + ih_n$, $i = 0, \dots, n$; $h_n = (b - a)/n$). Sexa $f \in \mathcal{C}^2([a, b])$ e $s^{(n)} \in S_3(\Omega_n)$, o único spline cúbico de tipo Hermite que interpola a f nos nós de Ω_n . Entón:*

$$\lim_{n \rightarrow +\infty} \|s^{(n)} - f\|_\infty = 0,$$

$$\lim_{n \rightarrow +\infty} \|[s^{(n)}]' - f'\|_\infty = 0.$$

Demostración. Só necesitamos ter en consideración as seguintes desigualdades:

$$\lim_{n \rightarrow +\infty} \|s^{(n)} - f\|_\infty \leq \lim_{n \rightarrow +\infty} \frac{1}{2} \|f''\|_2 h_n^{3/2} = 0,$$

$$\lim_{n \rightarrow +\infty} \|[s^{(n)}]' - f'\|_\infty \leq \lim_{n \rightarrow +\infty} \|f''\|_2 h_n^{1/2} = 0.$$

□

Observación 3.12 (Estimacións na norma $\|\cdot\|_2$). *En [7, p.109] podemos ver que para $f \in \mathcal{C}^2([a, b])$:*

$$\|f - s\|_2 \leq 16h^2 \|f''\|_2,$$

$$\|f' - s'\|_2 \leq 4h \|f''\|_2.$$

3.4. Consecuencias da relación integral

Á vista dos resultados obtidos na sección 3.2, unha área onde os splines poden resultar unha ferramenta moi boa é o deseño do trazado das vías do tren, posto que estas precisan ter pouca curvatura para minimizar a perda de velocidade. Ademais, impondo condicións de contorno naturais xenuínas obteremos que ao principio e ao final da vía non teñen curvatura, o cal é coherente coa vida real.

Para ilustralo, compararemos o trazado da liña de tren rexional que conecta Vigo e A Coruña. Esta pasa polas estacións de Vigo-Urzaiz, Redondela, Arcade, Pontevedra, Vilagarcía de Arousa, Padrón-Barbanza, Santiago de Compostela, Cerceda-Meirama e A Coruña. Coñecendo as coordenadas xeográficas das estacións somos quen de situalas de forma precisa no mapa á vez que as empregaremos como nós para o cálculo do spline. Empregando o algoritmo que implementamos en MATLAB somos quen de comparalo co trazado real sobre un mapa de Galicia coa liña real (ver coordenadas e código no Anexo).

Nótese que os splines son funcións, e polo tanto o sistema de coordenadas escollido afecta á curva que obtemos. O único que necesita verificar un sistema de coordenadas para que sexa válido é que a proxección sobre o eixo de abscisas manteña a orde das estacións na vida real.

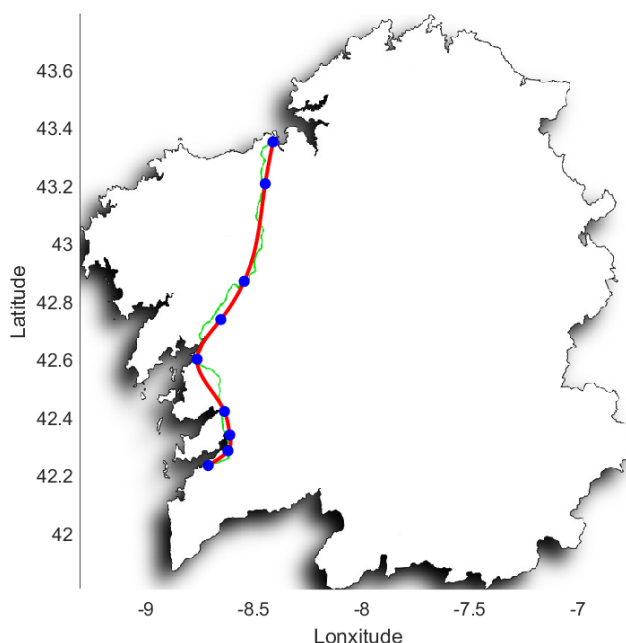


Figura 3.1: Trazado da liña rexional Vigo-Coruña empregando splines cúbicos e condición de contorno naturais

Por simplicidade, neste exemplo consideraremos como eixo de abscisas a latitude e como eixo de ordenadas a lonxitude. Malia que non o fagamos, isto motivaría a considerar todos os eixes posibles e ver cal se adapta mellor ao trazado.

Malia que o spline poida minimizar a lonxitude de vía, evidentemente o seu cálculo non atende á situación xeográfica, como poden ser desniveis excesivos ou a presenza de costa.

A maiores, podemos pensar que o trazado dunha vía rexional non presta tanta atención á perda de velocidade en comparación á rede de alta velocidade. Polo tanto, imos estudar tamén o trazado da liña de tren de alta velocidade Madrid-Figueras. Con este fin debuxamos nun mapa de España o trazado ferroviario. Esta vía pasa polas estacións Madrid-Atocha, Guadalajara-Yebes, Calatayud, Zaragoza-Delicias, Lleida-Pirineos, Campo de Tarragona, Barcelona-Sants, Xirona e Figueras-Vilafant. Procedendo de xeito análogo ao anterior (ver coordenadas e código no Anexo), só que neste caso tomando como eixo de abscisas a lonxitude e como eixo de ordenadas a latitude, podemos ver que se adapta bastante ben o trazado. Non obstante, si que destaca a diferenza presente no tramo entre as estacións de Zaragoza e Lleida. Mentres que o noso spline alcanza un máximo local, tras pasar Zaragoza a vía desvíase cara o sur. Podemos comprobar que o lugar onde está o noso spline cruza directamente a Serra de Alcubierre, o que motiva que no deseño final se realice o desvío pola cunca do Ebro.

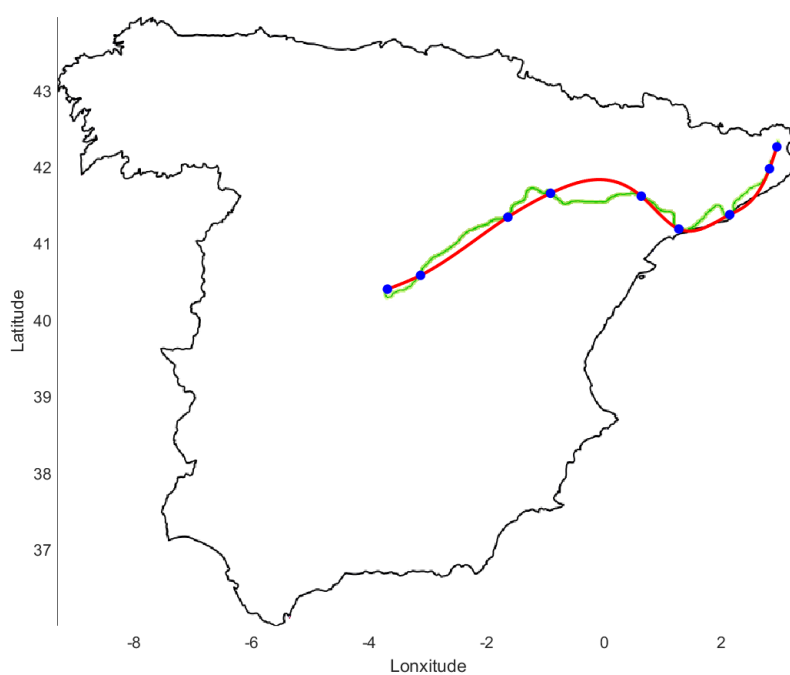


Figura 3.2: Trazado da liña de alta velocidade Madrid-Figueras empregando splines cúbicos e condición de contorno naturais

Anexo A

Códigos de MATLAB

Código para as figuras do exemplo 1.1:

```
n=10; % n=15
x(1)=-10;
x(n+1)=10;
x=linspace(x(1),x(n+1),n+1);
y = f(x);

N=1000;
x_trozos=linspace(x(1),x(n+1),N);
y_trozos = zeros(size(x_trozos));

% Debuxar a función orixinal
g = f(x_trozos);
plot(x_trozos, g, 'g-', 'DisplayName', 'Función orixinal', 'LineWidth', 2);
hold on

% Calcular e debuxar o polinomio interpolador
p = polyfit(x, y, n);
y_fit = polyval(p, x_trozos);
plot(x_trozos, y_fit, 'r-', 'DisplayName', 'Interpolación polinómica', ...
'LineWidth', 2);

% Debuxar os nós
scatter(x, y, 50, 'b', 'filled', 'DisplayName', 'Nós');
```

```
legend('show');  
hold off
```

Código para as figuras do exemplo 1.2

```
n=10;  
k=1; % k=2 orde dos polinomios  
x(1)=-10;  
x(n+1)=10;  
x=linspace(x(1),x(n+1),n+1);  
y = f(x);  
  
N=1000;  
x_trozos=linspace(x(1),x(n+1),N);  
y_trozos = zeros(size(x_trozos));  
  
% Debuxar a función orixinal  
g = f(x_trozos);  
plot(x_trozos, g, 'g-', 'DisplayName', 'Función orixinal', 'LineWidth', 2);  
hold on  
  
% Calcular e debuxar o polinomio interpolador a cachos  
y_cacho = y(1:k+1);  
p = polyfit(x(1:k+1), y_cacho, k);  
x_cachito = linspace(x(1), x(k+1),N);  
y_fit = polyval(p,x_cachito);  
plot(x_cachito, y_fit, 'r-', 'LineWidth', 2, 'DisplayName', 'Interpolación a cachos');  
for j=2:floor(n/k)  
    %y_cacho = f(x((j-1)*k+1:j*k+1));  
    y_cacho = y((j-1)*k+1:j*k+1);  
    p = polyfit(x((j-1)*k+1:j*k+1), y_cacho, k);  
    x_cachito = linspace(x((j-1)*k+1), x(j*k+1),N);  
    y_fit = polyval(p,x_cachito);  
    plot(x_cachito, y_fit, 'r-', 'LineWidth', 2, 'HandleVisibility', 'off');  
end  
  
% Debuxar os nós  
nodos = scatter(x, y, 50, 'b', 'filled', 'DisplayName', 'Nós');
```

```
legend('show');  
xlabel('');  
ylabel('');  
grid off  
hold off
```

Código para a figura do exemplo 1.6

```
close all  
% Definimos a función  
f = @(x) (x >= 0) .* x.^3;  
  
% Crear un rango de valores de x para la evaluación  
x = linspace(-5,5,1000);  
y = f(x);  
  
% Dibujar as funcións  
figure;  
hold on  
plot(x, y, 'r-', 'Linewidth', 2);  
plot(x-3, y, 'r-', 'Linewidth', 2);  
  
xticks = [-3, 0];  
xticklabels = {'x_{i}', 'x_{i+1}'};  
  
% Agregar os nomes das funcións  
set(gca, 'XTick', xticks, 'XTickLabel', xticklabels);  
text(0, 75, 'q_{m,i}', 'FontSize', 12, 'Color', 'r', 'FontWeight', 'bold');  
text(2.5, 75, 'q_{m,i+1}', 'FontSize', 12, 'Color', 'r', 'FontWeight', 'bold');  
  
hold off
```

Código para a figura do exemplo 1.3 e 1.6

```
n=8;  
k=1;
```

```

x=linspace(1,6,n+1);
y = [2,1,1.5,3,0,2,0.5,1,3];

N=1000;
x_trozos=linspace(2,6,N);
y_trozos = zeros(size(x_trozos));

figure;
hold on;

% Calculamos e debuxamos o spline lineal
y_cacho = y(1:k+1);
p = polyfit(x(1:k+1), y_cacho, k);
x_cachito = linspace(x(1), x(k+1),N);
y_fit = polyval(p,x_cachito);
plot(x_cachito, y_fit, 'r-', 'LineWidth', 2);
for j=2:floor(n/k)
    %y_cacho = f(x((j-1)*k+1:j*k+1));
    y_cacho = y((j-1)*k+1:j*k+1);
    p = polyfit(x((j-1)*k+1:j*k+1), y_cacho, k);
    x_cachito = linspace(x((j-1)*k+1), x(j*k+1),N);
    y_fit = polyval(p,x_cachito);
    plot(x_cachito, y_fit, 'r-', 'LineWidth', 2);
end

% Debuxamos os nós
nodos = scatter(x, y, 50, 'b', 'filled');
legend('hide');
xlabel('');
ylabel('');
grid off
hold off

```

Código para o cálculo dos splines cúbicos das figuras 1.4, 2.1, 2.2 e 2.3.

```

% Este programa calcula o spline cúbico interpolador que pasa polos
% nós (x(1), y(1)), ..., (x(n+1), y(n+1)).
% Os datos introdúcense en data

```

```

% Se  $y(i) = f(x(i))$  (efe = 1), a función indicárase no arquivo f
% En condicións de contorno Hermite (bound = 1), se se ten a
% derivada de f (defe = 1) indicárase no arquivo df
% En condicións de contorno naturais (bound = 3), se se ten a
% derivada segunda de f (d2efe = 1) indicárase no arquivo d2f
%
% En caso de ter unha malla uniforme, chama á función SPLINEcalc1,
% onde a matriz do problema depende só da lonxitude da malla (h).
% En caso contrario, chamará a SPLINEcalc2, onde a matriz é
% calculada empregando as fórmulas do capítulo 2 axudándose da
% función Bspline2
tic
clc
clear
close all
%
data
%
if unif == 1
    [coeff] = SPLINEcalc1(x,h,n,bound,y,deri,deri2);
    plotter1
else
    d1=x(2)-x(1);
    d2=x(n+1)-x(n);
    x=[x(1)-3*d1, x(1)-2*d1, x(1)-d1, ...
        x, x(n+1)+d2, x(n+1)+2*d2, x(n+1)+3*d2];
    [coeff] = SPLINEcalc2(x,n,bound,y, deri,deri2);
    plotter2
end
%
fprintf('\n\n')
toc

```

% Neste arquivo introdúcense os datos

% Número de intervalos (numero de nodos -1):

n = 10; % $x(0) = a < x(1) < \dots < x(n - 1) < x(n) = b.$

```
% Malla uniforme (unif = 1) ou non (unif outra ousa)
unif = 1;

if unif == 1
    x(1) = -10; % Extremo esquerdo a
    x(n+1) = 10; % Extremo dereito b
    x = linspace(x(1),x(n+1),n+1);
    % Lonxitude da malla
    h = (x(n+1)-x(1))/n;
else
    % Indícase o vector x
    % x=[1,2,3,4]
    if length(x) ~= n+1
        %error('%-i',length(x), 'distinto de ', n+1);
        error(['A lonxitude de x debe ser igual a n+1.' ...
            ' Lonxitude de x: %d, n+1: %d'], length(x), n+1);
    end
end

end

% Condicións de contorno
bound = 3;
% bound = 1: Hermite
% bound = 2: natural
% bound = 3: periódicas

% Amosamos por consola as condicións de contorno
switch bound
    case 1
        bound_text = 'Hermite';
    case 2
        bound_text = 'naturais';
    case 3
        bound_text = 'periódicas';
    otherwise
        error('Valor non válido. Debe ser 1, 2 o 3.');
```

```
end
fprintf('\nCondicións de contorno: %s.', bound_text);
```

```
% y(i) = f(x(i)) (efe = 1) ou non (efe = outra cousa)?
efe = 1;
fprintf('\n\nOpción para solución exacta: efe = %-i', efe);
if (efe == 1)
    fprintf(' (f coñecida).\n');
    y = f(x);
else
    fprintf(' (f non coñecida).\n');
    %Ponemos aqui o vector y de dimension n+1
    %y = [2,1,1.5,3,0,2,0.5,1,3];
    if length(y) ~= n+1
        error(['A lonxitude de y debe ser igual á de x.' ...
            ' Lonxitude de y: %d, Lonxitude de x: %d'], ...
            length(y), length(x));
    end
end

defe = 1;
d2efe = 1;
% Inicializamos os vectores coas derivadas en x0 e xn
% para evitar erros
deri=[0,0];
deri2=[0,0];
if bound == 1
    fprintf('Opción para a derivada: defe = %-i', defe)
    if defe ~= 1
        fprintf(' (df non coñecida).\n\n');
        deri(1)=0;
        deri(2)=0;
    else
        fprintf(' (df coñecida).\n\n');
        deri(1)=df(x(1));
        deri(2)=df(x(n+1));
    end
elseif bound == 2
    fprintf('Opción para a derivada segunda: d2efe = %-i', d2efe)
    if d2efe ~= 1
```

```

    fprintf(' (d2f non coñecida).\n\n');
    deri2(1)=-3;
    deri2(2)=-10;
    else
    fprintf(' (d2f coñecida).\n\n');
    deri2(1)=d2f(x(1));
    deri2(2)=d2f(x(n+1));
    end
elseif bound == 3
    fprintf('Opcion para a derivada: defe = %-i', defe)
    if defe ~= 1
    fprintf(' (df non coñecida).');
    else
    fprintf(' (df coñecida).');
    deri(1)=df(x(1));
    deri(2)=df(x(n+1));
    end
    fprintf('\nOpción para a derivada segunda: d2efe = %-i', d2efe)
    if d2efe ~= 1
    fprintf(' (d2f non coñecida).\n\n');
    else
        fprintf(' (d2f coñecida).\n\n');
        deri2(1)=d2f(x(1));
        deri2(2)=d2f(x(n+1));
    end
    if deri(1) ~= deri(2) || deri2(1) ~= deri2(2)
    error(['f non verifica as condicións de periodicidade. ' ...
        ' Elexir outra condición de contorno'])
    end
end
end

```

```

% Este arquivo contén a función que seguen os y(i)
% En caso de non coñecela deberemos introducir os valores
% directamente no arquivo data
function vf = f(x)
% vf = 0.*x;
vf = 2.*x+x.^2;

```

```

%vf = 2.*x;
%vf = cos(x);
%vf = 1./(1+x.^2);
end

```

```

% Este arquivo contém a derivada de f
% Requerirase cando f sexa coñecida e en condicións de contorno
% Hermite
% En caso de non coñecela deberemos introducir os valores de
% df(x(0)) e df(x(n+1)) no arquivo data
function vdf = df(x)
% vdf = 0.*x;
vdf = 2+2.*x;
%vdf = 2+0.*x;
%vdf = -sin(x);
%vdf = -(2.*x)/(1+x.^2).^2;
end

```

```

% Este arquivo contém a derivada de f
% Requerirase cando f sexa coñecida e en condicións de contorno
% naturais
% En caso de non coñecela deberemos introducir os valores de
% d2f(x(0)) e d2f(x(n+1)) no arquivo data
function vd2f = d2f(x)
%vd2f = 0.*x;
vd2f = 2+0.*x;
%vd2f = 0.*x;
%vd2f = -1.*cos(x);
%vd2f = -2.*((1+x.^2)-4.*x.^2)./(1+x.^2).^3;
end

```

```

% Este arquivo resolve o sistema lineal de splines con nós
% equidistantes
function [coeff] = SPLINEcalc1(x,h,n,bound,y,deri,deri2)

```

```

if bound == 1 % condiciones Hermite
    main_diag = 4 * ones(n+3, 1);
    upper_diag = 1 * ones(n+2, 1);
    lower_diag = 1 * ones(n+2, 1);
    B = diag(main_diag) +diag(upper_diag,1) +diag(lower_diag,-1);
    c1 = 3/h;
    B(1,1:3) = [-c1,0,c1];
    B(n+3,n+1:n+3) = [-c1,0,c1];
    B=B/6;

    vectb = [deri(1),y,deri(2)].';

    coeff=B\vectb;
elseif bound == 2 % condiciones naturales
    main_diag = 4 * ones(n+3, 1);
    upper_diag = 1 * ones(n+2, 1);
    lower_diag = 1 * ones(n+2, 1);
    B = diag(main_diag) +diag(upper_diag,1) +diag(lower_diag,-1);
    c2 = 6/h^2;
    B(1,1:3) = [c2,-2*c2,c2];
    B(n+3,n+1:n+3) = [c2,-2*c2,c2];
    B=B/6;

    vectb = [deri2(1),y,deri2(2)].';

    coeff=B\vectb;
else % condiciones periódicas
    upper_diag = 1 * ones(n+3, 1);
    main_diag = 4 * ones(n+2, 1);
    lower_diag = 1 * ones(n+1, 1);
    B = diag(main_diag,-1) +diag(upper_diag) +diag(lower_diag,-2);
    c1 = 3/h; c2 = 6/h^2;
    B(1,1:3) = [-c1,0,c1];
    B(1,n+1:n+3) = [c1,0,-c1];
    B(2,1:3) = [c2,-2*c2,c2];
    B(2,n+1:n+3) = [-c2,2*c2,-c2];
    B=B/6;

```

```

    vectb = [0,0,y].';

    coeff=B\vectb;
end
end

```

```

% Este archivo contén a función do B-spline para mallas uniformes
function Bspln = Bspline1(x0,i,h,x)
h2=h^2;
h3=h^3;
xi0=x0+i*h;
xi1=x0+(i+1)*h;
xi2=x0+(i+2)*h;
xi3=x0+(i+3)*h;
xi4=x0+(i+4)*h;
% Definir os cachos da función
if x >= xi0 && x < xi1
    % Cacho 1
    Bspln = ((x-xi0)^3)/(6*h3);
elseif x >= xi1 && x < xi2
    % Cacho 2
    Bspln = (h3+3*h2*(x-xi1)+3*h*(x-xi1)^2-3*(x-xi1)^3)/(6*h3);
elseif x >= xi2 && x < xi3
    % Cacho 3
    Bspln = (h3+3*h2*(xi3-x)+3*h*(xi3-x)^2-3*(xi3-x)^3)/(6*h3);
elseif x >= xi3 && x <= xi4
    % Cacho 4
    Bspln = ((xi4-x)^3)/(6*h3);
else Bspln= 0.*x;
end
end

```

```

% Este arquivo grafica o spline calculado en SPLINEcalc1.m
% Cando f é coñecida tamén é debuxada en verde para comparala co
% spline
N=1000; %N e a cantidade de puntos a amosar

```

```

x_trozos=linspace(x(1),x(n+1),N);
y_trozos = zeros(size(x_trozos));

for k = 1:N
for j = 1:n
if x_trozos(k) >= x(j) && x_trozos(k) <= x(j+1)
    y_trozos(k) = coeff(j)*Bspline1(x(1),j-4,h,x_trozos(k))+...
        coeff(j+1)*Bspline1(x(1),j-3,h,x_trozos(k))+...
        coeff(j+2)*Bspline1(x(1),j-2,h,x_trozos(k))+...
        coeff(j+3)*Bspline1(x(1),j-1,h,x_trozos(k));
    break;
end
end
end

figure;
% Se f é conhecida debúxaaa tamén
if efe == 1
    g = f(x_trozos);
    plot(x_trozos, g, 'g-', 'DisplayName', 'Funcion orixinal', ...
        'LineWidth', 1.5);
    err = max(abs(y_trozos-g));
    fprintf('Erro maximo: max(abs(f-s)) = %f6', err);
end
hold on;
% Debuxamos o spline
plot(x_trozos, y_trozos, 'r-', 'LineWidth', 2, 'DisplayName', ...
    'Spline interpolador' );
scatter(x, y, 50, 'b', 'filled', 'DisplayName', 'Nodos');

%legend('show');
%title('Titulo');
xlabel('');
ylabel('');
grid off;
hold off;

```

%Este archivo resuelve o sistema lineal de splines con nós arbitrarios

```
function [coeff] = SPLINEcalc2(x,n,bound,y,deri,deri2)

B = zeros(n+3,n+3);

if bound == 1 % condicións Hermite
    for i=1:n+1
        B(i+1,i) = Bspline2(x,i,x(i+3));
        B(i+1,i+1) = Bspline2(x,i+1,x(i+3));
        B(i+1,i+2) = Bspline2(x,i+2,x(i+3));
    end

    [~, produ] = Bspline2(x,1, x(4));
    B(1,1) = -3*(produ(1)*(x(1)-x(4))^2 + ...
        + produ(2)*(x(2)-x(4))^2 + produ(3)*(x(3)-x(4))^2);
    [~, produ] = Bspline2(x,3, x(4));
    B(1,3) = -3*(produ(1)*(x(3)-x(4))^2);

    [~, produ] = Bspline2(x,n+1, x(n+4));
    B(n+3,n+1) = -3*(produ(1)*(x(n+1)-x(n+4))^2 + ...
        + produ(2)*(x(n+2)-x(n+4))^2 + produ(3)*(x(n+3)-x(n+4))^2);
    [~, produ] = Bspline2(x,n+3, x(n+4));
    B(n+3,n+3) = -3*(produ(1)*(x(n+3)-x(n+4))^2);

    vectb = [deri(1),y,deri(2)].';

    coeff=B\vectb;
elseif bound == 2 % condicións naturais
    for i=1:n+1
        B(i+1,i) = Bspline2(x,i,x(i+3));
        B(i+1,i+1) = Bspline2(x,i+1,x(i+3));
        B(i+1,i+2) = Bspline2(x,i+2,x(i+3));
    end

    [~, produ] = Bspline2(x,1, x(4));
    B(1,1) = 6*(produ(1)*(x(1)-x(4)) + ...
        + produ(2)*(x(2)-x(4)) + produ(3)*(x(3)-x(4)));
```

```

[~, produ] = Bspline2(x,2, x(4));
B(1,2) = 6*(produ(1)*(x(2)-x(4)) + produ(2)*(x(3)-x(4)));
[~, produ] = Bspline2(x,3, x(4));
B(1,3) = 6*(produ(1)*(x(3)-x(4)));

[~, produ] = Bspline2(x,n+1, x(n+4));
B(n+3,n+1) = 6*(produ(1)*(x(n+1)-x(n+4)) + ...
    + produ(2)*(x(n+2)-x(n+4)) + produ(3)*(x(n+3)-x(n+4)));
[~, produ] = Bspline2(x,n+2, x(n+4));
B(n+3,n+2) = 6*(produ(1)*(x(n+2)-x(n+4)) + produ(2)*(x(n+3)-x(n+4)));
[~, produ] = Bspline2(x,n+3, x(n+4));
B(n+3,n+3) = 6*(produ(1)*(x(n+3)-x(n+4)));

vectb = [deri2(1),y,deri(2)].';

coeff=B\vectb;
else % condições periódicas

for i=1:n+1
    B(i+2,i) = Bspline2(x,i,x(i+3));
    B(i+2,i+1) = Bspline2(x,i+1,x(i+3));
    B(i+2,i+2) = Bspline2(x,i+2,x(i+3));
end

% Derivada primeira na primeira fila
[~, produ] = Bspline2(x,1, x(4));
B(1,1) = -3*(produ(1)*(x(1)-x(4))^2 + ...
    + produ(2)*(x(2)-x(4))^2 + produ(3)*(x(3)-x(4))^2);
[~, produ] = Bspline2(x,3, x(4));
B(1,3) = -3*(produ(1)*(x(3)-x(4))^2);

[~, produ] = Bspline2(x,n+1, x(n+4));
B(1,n+1) = 3*(produ(1)*(x(n+1)-x(n+4))^2 + ...
    + produ(2)*(x(n+2)-x(n+4))^2 + produ(3)*(x(n+3)-x(n+4))^2);
[~, produ] = Bspline2(x,n+3, x(n+4));
B(1,n+3) = 3*(produ(1)*(x(n+3)-x(n+4))^2);

% Derivada segunda na segunda fila

```

```

    [~, produ] = Bspline2(x,1, x(4));
    B(2,1) = 6*(produ(1)*(x(1)-x(4)) + ...
        + produ(2)*(x(2)-x(4)) + produ(3)*(x(3)-x(4)));
    [~, produ] = Bspline2(x,2, x(4));
    B(2,2) = 6*(produ(1)*(x(2)-x(4)) + produ(2)*(x(3)-x(4)));
    [~, produ] = Bspline2(x,3, x(4));
    B(2,3) = 6*(produ(1)*(x(3)-x(4)));

    [~, produ] = Bspline2(x,n+1, x(n+4));
    B(2,n+1) = -6*(produ(1)*(x(n+1)-x(n+4)) + ...
        + produ(2)*(x(n+2)-x(n+4)) + ...
        + produ(3)*(x(n+3)-x(n+4)));
    [~, produ] = Bspline2(x,n+2, x(n+4));
    B(2,n+2) = -6*(produ(1)*(x(n+2)-x(n+4)) + ...
        + produ(2)*(x(n+3)-x(n+4)));
    [~, produ] = Bspline2(x,n+3, x(n+4));
    B(2,n+3) = -6*(produ(1)*(x(n+3)-x(n+4)));

    vectb = [0,0,y].';

    coeff=B\vectb;
end
end

```

% Este archivo contén a función do B-spline para mallas arbitrarias

```
function [Bspln2,produ] = Bspline2(malla,i,x)
```

```
nod=malla(i:i+4);
```

% nod é un vector de 5 elementos que contén

% xi, xi+1, xi+2, xi+3, xi+4

```
for i=1:5
```

```
for j=1:5
```

```
    D(i,j)= nod(i)-nod(j);
```

```
end
```

```
end
```

```

produ(1) = -4/(prod(D(1,2:5)));
produ(2) = -4/(D(2,1)*prod(D(2,3:5)));
produ(3) = -4/(prod(D(3,1:2))*prod(D(3,4:5)));
produ(4) = -4/(prod(D(4,1:3))*prod(D(4,5)));

if x > nod(1) && x <= nod(2)
    % Cacho 1
    Bspln2 = produ(1)*(nod(1)-x)^3;
elseif x > nod(2) && x <= nod(3)
    % Cacho 2
    Bspln2 = produ(1)*(nod(1)-x)^3 + produ(2)*(nod(2)-x)^3;
elseif x > nod(3) && x <= nod(4)
    % Cacho 3
    Bspln2 = produ(1)*(nod(1)-x)^3 + produ(2)*(nod(2)-x)^3 + ...
        produ(3)*(nod(3)-x)^3;
elseif x > nod(4) && x < nod(5)
    % Cacho 4
    Bspln2 = produ(1)*(nod(1)-x)^3 + produ(2)*(nod(2)-x)^3 + ...
        +produ(3)*(nod(3)-x)^3 + produ(4)*(nod(4)-x)^3;
else Bspln2 = 0.*x;
end
end

```

```

% Este archivo grafica o spline calculado en SPLINEcalc2.m
% Cando f é coñecida tamén é debuxada en verde para comparala co
% spline
N=1000; %N é a cantidade de puntos a amosar
x_trozos=linspace(x(4),x(n+4),N);
y_trozos = zeros(size(x_trozos));

for k = 1:N
for j = 1:n
if x_trozos(k) >= x(j+3) && x_trozos(k) <= x(j+4)
    y_trozos(k) = coeff(j)*Bspline2(x,j, x_trozos(k))+...
        coeff(j+1)*Bspline2(x, j+1, x_trozos(k))+...
        coeff(j+2)*Bspline2(x, j+2, x_trozos(k))+...
        coeff(j+3)*Bspline2(x, j+3, x_trozos(k));

```

```

        break;
    end
end
end

figure;
% Se f é conhecida debúxaa tamén
if efe == 1
    g = f(x_trozos);
    plot(x_trozos, g, 'g-', 'DisplayName', 'Funcion orixinal', ...
         'LineWidth', 1.5);
    err = max(abs(y_trozos-g));
    fprintf('Erro maximo: max(abs(f-s)) = %f6', err);
end
hold on;
% Debuxamos o spline
plot(x_trozos, y_trozos, 'r-', 'LineWidth', 2, 'DisplayName', ...
     'Spline interpolador');
scatter(x(4:n+4), y, 50, 'b', 'filled', 'DisplayName', 'Nodos');

legend('show');
title('Titulo');
xlabel('');
ylabel('');
grid off;
hold off;

```

Código para o B-spline do exemplo 1.7

```

N=1000; %N é a cantidade de puntos a amosar
x_trozos=linspace(1,6,N);
y_trozos = zeros(size(x_trozos));

% Definir os nós
x2 = 2; x3 = 3;
x4 = 4; x5 = 5;

```

```

y_trozos = zeros(size(x_trozos));

% Definir o B-spline a cachos
for i = 1:length(x_trozos)
    if x_trozos(i) >= x2 && x_trozos(i) < x3
        y_trozos(i) = ((x_trozos(i)-x2).^2)/2;
    elseif x_trozos(i) >= x3 && x_trozos(i) < x4
        y_trozos(i) = (1+2*(x_trozos(i)-x3)- ...
            2*(x_trozos(i)-x3)^2)/2;
    elseif x_trozos(i) >= x4 && x_trozos(i) < x5
        y_trozos(i) = ((x5-x_trozos(i)).^2)/2 ;
    end
end

% Debuxar o B-spline
figure;
plot(x_trozos, y_trozos, 'r-', 'LineWidth', 2);
xlabel('');
ylabel('');

% Marcar a partição
xticks([x2,x3,x4,x5]);
xticklabels({'x_{i}', 'x_{i+1}', 'x_{i+2}', ...
    'x_{i+3}'});
grid on;

```

Código para o B-spline do exemplo 1.8

```

N=1000; %N é a quantidade de pontos a amosar
x_trozos=linspace(-2,7,N);
y_trozos = zeros(size(x_trozos));

% Definir os nós
x1 = -1; x2 = 0;
x3 = 1; x4 = 2;
x5 = 3; x6 = 4;
x7=5;

```

```

y_trozos = zeros(size(x_trozos));

% Definir o B-spline a cachos
for i = 1:length(x_trozos)
    if x_trozos(i) >= x2 && x_trozos(i) < x3
        y_trozos(i) = ((x_trozos(i)-x2)^3)/6;
    elseif x_trozos(i) >= x3 && x_trozos(i) < x4
        y_trozos(i) = (1+3*(x_trozos(i)-x3)+ ...
            3*(x_trozos(i)-x3)^2-3*(x_trozos(i)-x3)^3)/6;
    elseif x_trozos(i) >= x4 && x_trozos(i) < x5
        y_trozos(i) = (1+3*(x5-x_trozos(i))+ ...
            3*(x5-x_trozos(i))^2-3*(x5-x_trozos(i))^3)/6;
    elseif x_trozos(i) >= x5 && x_trozos(i) < x6
        y_trozos(i) = ((x6-x_trozos(i))^3)/6 ;
    end
end

% Debuxar o B-spline
figure;
plot(x_trozos, y_trozos, 'r-', 'LineWidth', 2);
xlabel('');
ylabel('');

% Marcar a partición
xticks([x1,x2,x3,x4,x5,x6]);
xticklabels({'x_{i-1}', 'x_{i}', 'x_{i+1}', 'x_{i+2}', ...
    'x_{i+3}', 'x_{i+4}', 'x_{i+5}'});
grid on;

```

Código para a figura 3.1

```

clc
clear
close all
% Coordenadas das estacións (latitudes e lonxitudes)
latitudes = [42.23528779944515,...

```

```
42.28617763956931, ...
42.34058465739847, ...
42.42177185995426, ...
42.60205580860679, ...
42.73903479089577, ...
42.870768987916534, ...
43.20775395140882, ...
43.35213283765025];
lonxitudes = [-8.710878048510406, ...
-8.61942068594061, ...
-8.610781036802935, ...
-8.63563582241349, ...
-8.762048125176715, ...
-8.65247623866276, ...
-8.544273551819652, ...
-8.446764257055719, ...
-8.410805104720742];

% Coordenadas da imaxe
latlim = [41.80738711862975, 43.79134745288148];
lonlim = [-9.301481092521062, -6.734293989874599];

% Cargar a imaxe de fondo
backgroundImage = imread('galicia2.png');

% Crear o mapa de base
figure;
hold on;
image('CData', flipud(backgroundImage), 'XData', lonlim, ...
      'YData', latlim);

% Configurar o sistema de coordenadas
set(gca, 'YDir', 'normal', 'XLim', lonlim, 'YLim', latlim);

x=latitudes; y=lonxitudes; n=8; efe=0;

d1=x(2)-x(1);
d2=x(n+1)-x(n);
```

```

x=[x(1)-3*d1, x(1)-2*d1, x(1)-d1, ...
   x, x(n+1)+d2, x(n+1)+2*d2, x(n+1)+3*d2];
[coeff] = SPLINEcalc2(x,n,2,y, [0,0],[0,0]);
N=1000; %N é a quantidade de pontos a amosar
x_trozos=linspace(x(4),x(n+4),N);
y_trozos = zeros(size(x_trozos));

for k = 1:N
for j = 1:n
if x_trozos(k) >= x(j+3) && x_trozos(k) <= x(j+4)
    y_trozos(k) = coeff(j)*Bspline2(x,j, x_trozos(k))+...
                 coeff(j+1)*Bspline2(x, j+1, x_trozos(k))+...
                 coeff(j+2)*Bspline2(x, j+2, x_trozos(k))+...
                 coeff(j+3)*Bspline2(x, j+3, x_trozos(k));
    break;
end
end
end

hold on;
% Debuxamos o spline
plot(y_trozos, x_trozos, 'r-', 'LineWidth', 2, 'DisplayName', ...
     'Spline interpolador');
scatter(y, x(4:n+4), 50, 'b', 'filled', 'DisplayName', 'Nós');

xlabel('Lonxitude');
ylabel('Latitude');

```

Código para a figura 3.2

```

clc
clear
close all
% Coordenadas das estações (latitudes e lonxitudes)
latitudes = [40.405970321074044, ...
             40.586440657215455, ...
             41.346465440457195, ...

```

```
41.65884167181193, ...
41.62108205461517, ...
41.19196393060805, ...
41.379657640534695, ...
41.97940221304221, ...
42.26482900130191];
lonxitudes = [-3.6898356256805034, ...
-3.126123233346957, ...
-1.63837289085496, ...
-0.9118115815730244, ...
0.6331116502637246, ...
1.2727592115100763, ...
2.139904118310508, ...
2.816776662789238, ...
2.942907653221494];

% Coordenadas da imaxe
latlim = [36.0072077834539, 43.96134745288148];
lonlim = [-9.301481092521062, 3.3183139960052612];

% Cargar a imaxe de fondo
backgroundImage = imread('espania3.png');

% Crear o mapa de base
figure;
hold on;
image('CData', flipud(backgroundImage), 'XData', lonlim, ...
      'YData', latlim);

% Configurar o sistema de coordenadas
set(gca, 'YDir', 'normal', 'XLim', lonlim, 'YLim', latlim);

x=lonxitudes; y=latitudes; n=8; efe=0;

d1=x(2)-x(1);
d2=x(n+1)-x(n);
x=[x(1)-3*d1, x(1)-2*d1, x(1)-d1, ...
   x, x(n+1)+d2, x(n+1)+2*d2, x(n+1)+3*d2];
```

```

[coeff] = SPLINEcalc2(x,n,2,y, [0,0],[0,0]);
N=1000; %N e a quantidade de pontos a amosar
x_trozos=linspace(x(4),x(n+4),N);
y_trozos = zeros(size(x_trozos));

for k = 1:N
for j = 1:n
if x_trozos(k) >= x(j+3) && x_trozos(k) <= x(j+4)
    y_trozos(k) = coeff(j)*Bspline2(x,j, x_trozos(k))+...
        coeff(j+1)*Bspline2(x, j+1, x_trozos(k))+...
        coeff(j+2)*Bspline2(x, j+2, x_trozos(k))+...
        coeff(j+3)*Bspline2(x, j+3, x_trozos(k));
    break;
end
end
end

hold on;
% Debuxamos o spline
plot(x_trozos, y_trozos, 'r-', 'LineWidth', 2, 'DisplayName', ...
    'Spline interpolador');
scatter(x(4:n+4), y, 50, 'b', 'filled', 'DisplayName', 'Nós');

% Calcular e debuxar o spline interpolador
xx = linspace(min(lonxitudes), max(lonxitudes), 1000);
cs = csape(lonxitudes, latitudes, 'second');
yy = ppval(cs, xx);
plot(xx, yy, 'r-', 'LineWidth', 2);

% Agregar puntos das estacións
scatter(lonxitudes, latitudes, 'b', 'filled', ...
    'DisplayName', 'Nós');

xlabel('Lonxitude');
ylabel('Latitude');

```

Bibliografía

- [1] Ahlberg, J. H., Nilson, E. N., Walsh, J. L. *The theory of splines and their applications*, Academic Press (1967).
- [2] Boyer, C. B., *Historia de la matemática*, Alianza (1996).
- [3] De Boor, C., *A practical guide to splines*, Springer-Verlag (1987).
- [4] Hammerlin, G. e Hoffmann, K-H . *Numerical mathematics*, Springer-Verlag (1991).
- [5] Micula, G., Micula, S., *Handbook of splines*, Kluwer Academic Publishers (1999).
- [6] Nürnberger, G., *Approximation by spline functions*, Springer-Verlag (1989).
- [7] Prenter, P.M., *Splines and variational methods*, John Wiley & Sons (1975).
- [8] Schoenberg, I. J., *Contributions to the problem of approximation of equidistant data by analytic functions*, Quart. Appl. Math, 45-99 e 112-141, Brown University (1946).
- [9] Schoenberg, I.J., *Cardinal spline interpolation*, SIAM (1973).
- [10] Schumaker, L., *Spline functions: basic theory*, John Wiley & Sons (1981).
- [11] Stoer, J., Bulirsch, R., *Introduction to Numerical Analysis*, Springer-Verlag (1993).
- [12] Viaño, J. M., Burguera, M., *Lecciones de métodos numéricos. 3, Interpolación*, Tórculo Edicións (2000).