



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Técnicas de Bézier y B-splines aplicadas al modelado y animación gráficos

Autor/es

Javier Jiménez Santana

Director/es

JOSÉ MANUEL GUTIÉRREZ JIMÉNEZ

Facultad

Facultad de Ciencia y Tecnología

Titulación

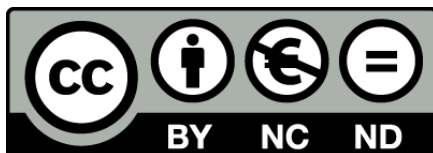
Grado en Matemáticas

Departamento

MATEMÁTICAS Y COMPUTACIÓN

Curso académico

2022-23



Técnicas de Bézier y B-splines aplicadas al modelado y animación gráficos, de
Javier Jiménez Santana

(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative
Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.

Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los
titulares del copyright.

© El autor, 2023

© Universidad de La Rioja, 2023

publicaciones.unirioja.es

E-mail: publicaciones@unirioja.es



UNIVERSIDAD DE LA RIOJA

Facultad de Ciencia y Tecnología

TRABAJO FIN DE GRADO

Grado en Matemáticas

Técnicas de Bézier y B-splines aplicadas al modelado y
animación gráficos

Realizado por:

Javier Jiménez Santana

Tutelado por:

José Manuel Gutiérrez Jiménez

Logroño, a 2 de julio de 2023

Índice general

Abstract / Resumen	1
Introducción	2
1 Polinomios de Bernstein y B-splines	5
1.1 Polinomios de Bernstein	6
1.2 Splines	9
1.3 La búsqueda de los splines básicos	14
1.4 Propiedades de los B-splines	21
1.5 El teorema de representación	27
2 Curvas de Bézier y B-splines	29
2.1 Curvas de Bézier: el algoritmo de De Casteljaou	29
2.2 Splines de Bézier	32
2.3 B-splines: el algoritmo de De Boor	34
2.4 NURBS	38
3 Superficies producto tensorial	39
3.1 Superficies B-spline	39
3.2 Parches de Bézier y NURBS	42
4 Introducción a la teoría de blossomes	43
4.1 El Principio del blossom	44
4.2 El teorema fundamental: curvas de Bézier	45
4.3 El teorema fundamental: curvas B-spline	47
5 Introducción a la subdivisión	49
5.1 Subdivisión de curvas de Bézier	49
5.1.1 Convergencia de la subdivisión de De Casteljaou	50
5.2 Subdivisión de curvas B-spline	51
5.2.1 B-splines uniformes	51
5.2.2 Subdivisión uniforme	53
5.2.3 Convergencia de subdivisión de curvas B-spline	58
6 Algunas aplicaciones de las técnicas de Bézier y B-splines	59
6.1 Diferentes utilidades de la suavidad	59
6.2 Rasterizado, edición e interpolación de imagen	61
6.3 Esquemas de subdivisión de superficies	63
Conclusiones	65
Bibliografía	66

Abstract / Resumen

Graphic modeling and animation may be one of the most breathtaking examples of Applied Mathematics, although we can find others much closer to our day-to-day life. Without going any further, the representation of this very text wouldn't be possible without Numerical Analysis. In this work we seek to study the two most popular paradigms used to produce geometric shapes in computers graphics: B-splines and Bézier curves. We will mainly focus on the B-spline theory and we will learn its relationship with the Bézier one through small brushstrokes of the theory of symmetric polynomials, known as blossoms, to culminate with the basics of subdivision algorithms.

El modelado y la animación gráfica deben de ser uno de los ejemplos más impresionantes de la matemática aplicada, si bien podemos encontrar otros muchos más cotidianos. Sin irnos tan lejos, la representación de este mismo texto no sería posible sin el análisis numérico. En el presente trabajo buscamos estudiar los dos paradigmas más populares utilizados para representar formas geométricas en computadores: los B-splines y las curvas de Bézier. Nos centraremos en la teoría de B-splines y conoceremos su relación con la de Bézier a través de pinceladas de la teoría de polinomios simétricos, conocidos como blossomes, para culminar con los fundamentos de los algoritmos de subdivisión.

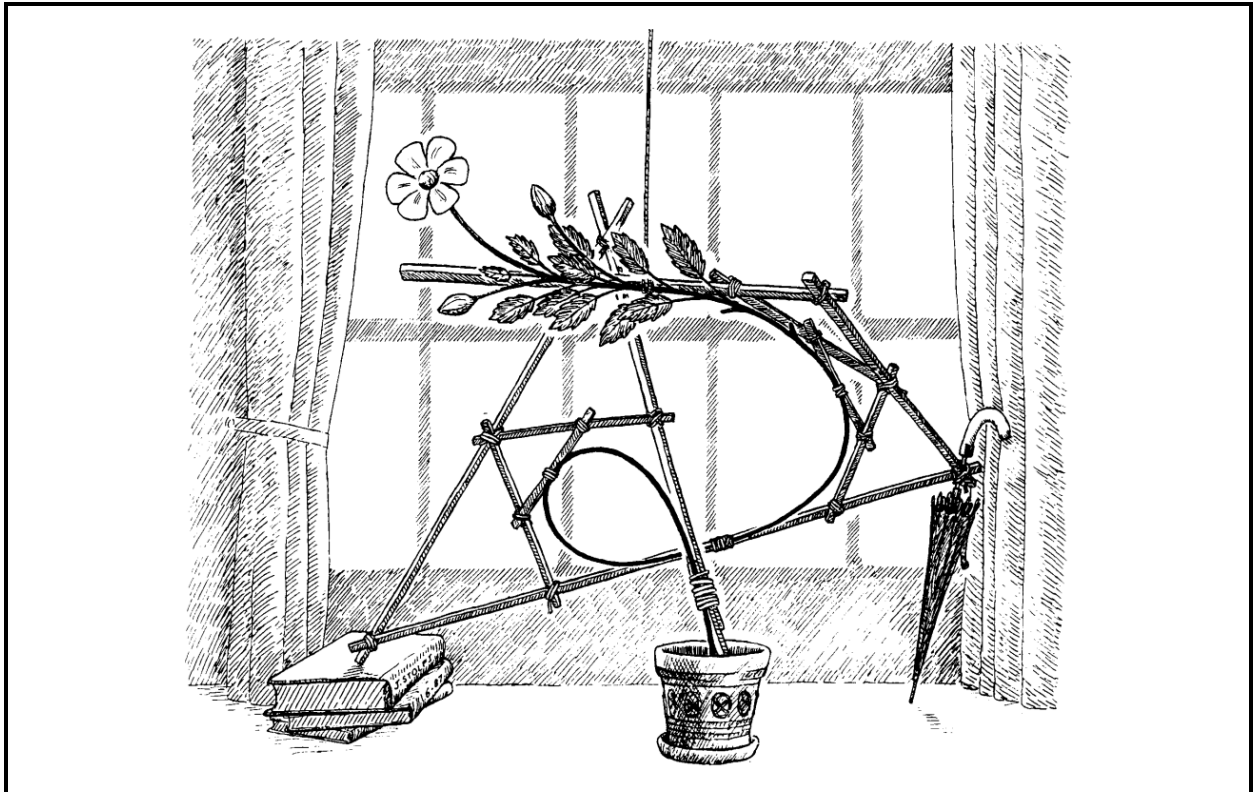


Figura 1: *La relación entre splines y blossomes*, ilustración de Jorge Stolfi (1987), tomada de *Blossoming: A Connect-the-Dots Approach to Splines*, [11].

Introducción

Los gráficos generados por ordenador son omnipresentes en la actualidad. No son pocos los ejemplos: los efectos especiales del cine, las películas de animación, los videojuegos, los diseños automovilísticos, aeronáuticos y arquitectónicos; o incluso los anuncios publicitarios y la medicina. El objetivo de este Trabajo Fin de Grado es precisamente conocer los fundamentos matemáticos de la computación gráfica, concretamente de la representación de formas geométricas, y dar una introducción a aspectos avanzados como los esquemas de subdivisión.

Por la propia naturaleza del diseño, el modelo matemático tiene que solucionar un problema fundamental: el poder representar una forma geométrica salida de la mente del diseñador, y de la que este no conoce representación matemática (o de la cual, quizá, ni siquiera exista una forma explícita). Clásicamente este problema se ha solucionado por dos vías o modelos matemáticos distintos. El más famoso, probablemente, sea las curvas de Bézier, seguidas de cerca por los B-splines. En este trabajo nos hemos centrado más en estos segundos, porque veremos que los B-splines generalizan las curvas de Bézier. De esta forma, podremos tener una panorámica completa de la cuestión.

Si bien nuestro interés en los splines radica en su aplicación gráfica, el contexto de su nacimiento se localiza temporalmente varios siglos antes de que existiera el primer ordenador. Traducido del inglés, spline significaría algo parecido a «listón elástico». Su etimología se remonta al uso que se daba a los listones de madera durante la Segunda Guerra Mundial para crear curvas, con las que se podían construir superficies para fabricar cascos de barcos o fuselajes de aviones. De hecho, podríamos remontarnos aún más, hasta los antiguos astilleros de barcos de vela, pues la técnica es originaria de la fabricación de cascos de navío. A los splines, una vez humedecidos en vapor de agua, se les podía dar forma mediante unos pesos, hasta que la curva que adquirían minimizaba su energía elástica.

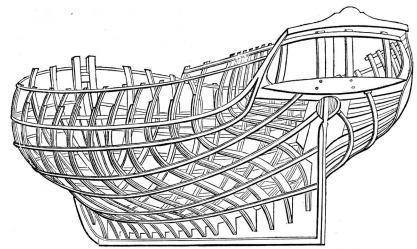


Figura 2: Casco de un bergantín del siglo XIX, formado por splines transversales.

Sin embargo, el contexto histórico-matemático de la aparición de los splines es muy diferente. El término spline es introducido por primera vez en 1946 en el célebre *Journal of Approximation Theory* por el matemático rumano Isaac Jacob Schoenberg, de cuyo comité editorial era miembro. En el siglo XX, el análisis numérico se enfrentaba al problema del fenómeno de Runge. Descubierto

en 1901 y nombrado en honor a su matemático descubridor, el fenómeno de Runge planteaba algo contraintuitivo a simple vista: seguir aumentando el grado del polinomio de interpolación empeoraba la aproximación. Los splines, que son concatenaciones de polinomios que obedecen ciertas reglas de continuidad, surgieron como una posible solución a este problema (entre otras, como por ejemplo los nodos de Chebyshev).

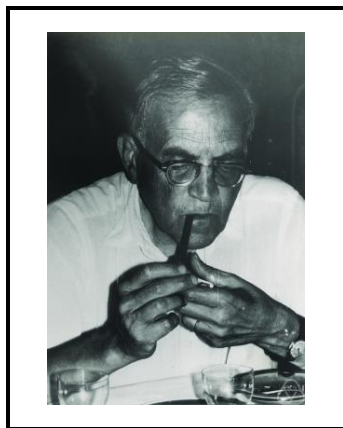


Figura 3: Fotoretrato de I. J. Schoenberg (1903-1990).

El primer ordenador de propósito general, el ENIAC (*Electronic Numerical Integrator And Computer*) fue desarrollado en 1943 en un esfuerzo combinado del Ejército de los EEUU con varios científicos, entre los que se encontraba Schoenberg. Los primeros gráficos generados por un ordenador a partir de ecuaciones matemáticas aparecieron apenas diez años más tarde, a través de las primeras máquinas fresadoras con control numérico y no a través de la pantalla de un ordenador, como pudiera uno creer.

Diferente es el caso del nacimiento de las curvas de Bézier, en los años 60. Nombradas en honor a su descubridor, la principal inquietud del ingeniero francés Pierre Bézier era facilitar el diseño de las carrocerías de los automóviles de Renault, empresa para la que trabajaba. También es conocido por haber patentado uno de los primeros sistemas de CAD (*Computer Assisted Design*) de la historia. De forma paralela en la marca rival Citroën, el físico Paul de Casteljaou hizo los mismos descubrimientos de forma independiente, pero no fueron publicados hasta la década de los 80, cuando las curvas de Bézier ya se habían extendido como la pólvora.

Los splines de Bézier llegaron para resolver un inconveniente al que los fabricantes industriales no habían logrado dar solución aún. El cambio en cualquier detalle de diseño de sus productos implicaba rehacer todos los dibujos desde cero, con el alto coste en tiempo y dinero que implicaba. Su propiedad de control local permite que los cambios en una curva no se propaguen a nivel global, justo lo que se buscaba. Los B-splines, que son una generalización suya y que comparten esta y más propiedades, en su caso, trascendieron su propósito original y también son utilizados como alternativa a las curvas de Bézier.

En este Trabajo Fin de Grado seguiremos el modelo [splines \rightarrow B-splines \rightarrow NURBS \rightarrow Superficies producto tensorial], de forma alternativa al más frecuente [polinomios de Bernstein \rightarrow curvas de Bézier \rightarrow Superficies producto tensorial], aunque veremos los aspectos más fundamentales de éste y cómo los dos esquemas están relacionados. Para ello, nos apoyaremos en una reformulación moderna de la teoría de polinomios simétricos denominada teoría de blossomes, propuesta por el matemático Lyle Ramshaw en 1987. Para concluir, haremos una incursión introductoria a los esquemas de subdivisión, que, por razones de alcance del trabajo, no podremos desarrollar más.



Figura 4: Fotoretrato de Pierre Bézier (1910-1999).



Figura 5: Fotoretrato de Paul de Casteljau (1930-2022).



Figura 6: Fotoretrato de Carl R. de Boor (1937-).

Entrando en detalle del contenido del trabajo:

- En el capítulo primero veremos los objetos más básicos (que algebraicamente también son básicos), los polinomios de Bernstein y los B-splines. Estudiaremos las propiedades fundamentales de los polinomios de Bernstein y veremos la propiedad física que cumplen los splines cúbicos, que comparten con sus homónimos de madera. Una vez sepamos qué es un spline, buscaremos una base para el espacio vectorial que conforman, es decir, unos splines básicos, que serán los B-splines.
- El capítulo segundo lo dedicaremos a construir las curvas de Bézier a partir de los polinomios de Bernstein, y (curvas) splines a partir de B-splines. Estudiaremos los algoritmos clásicos que se utilizan para calcularlas: el algoritmo de De Casteljau y de De Boor, respectivamente.
- Estudiadas las curvas, podremos construir superficies mediante productos tensoriales, con la cualidad de que seguirán siendo compatibles con los algoritmos ya vistos, cosa que veremos en el capítulo tercero.
- Seguidamente introduciremos la teoría de blossomes en el capítulo cuarto, que nos permitirá estudiar las relaciones geométricas implícitas en los algoritmos presentados con una herramienta de lo más interesante.
- Para terminar las matemáticas, el capítulo quinto lo dedicaremos a presentar la subdivisión para la construcción de curvas de Bézier y B-splines, como alternativa a los algoritmos clásicos estudiados en el tercer capítulo.
- Concluiremos con el capítulo sexto, puramente divulgativo, con el objetivo de ver algunas de las innumerables aplicaciones de todo lo visto teóricamente.

Capítulo 1

Polinomios de Bernstein y B-splines

Uno de los problemas básicos en el diseño es poder representar formas geométricas suaves (con formas nos referimos a curvas o superficies). Matemáticamente, la búsqueda de suavidad puede interpretarse como una búsqueda de continuidad diferenciable (C^k) o únicamente geométrica (G^k). Una de las mejores técnicas, por ser capaz de aportar los dos tipos de suavidad, son los splines. A muy grandes rasgos, entendemos por spline a una función polinomial paramétrica a trozos.

Con *a trozos*, nos referimos a un enfoque que permite modelar las diferentes regiones de la curva mediante unos respectivos sistemas de ecuaciones. Esto es muy conveniente a la hora de trabajar con splines, pues les aporta control local y cada región puede ser alterada sin afectar a las demás. De esta forma, cualquier cambio en la curva no es computacionalmente tan costoso.

Una curva *paramétrica* se denomina así por estar descrita en términos de parámetros auxiliares. Imaginemos que queremos modelar una forma geométrica p -dimensional contenida en un espacio ambiente q -dimensional, llamémoslo Q . La técnica paramétrica resuelve este problema inventando un espacio auxiliar P de dimensión p , el espacio paramétrico, y modeliza la forma S como la imagen $S = f(P)$ a través de una función $f : P \rightarrow Q$. En su lugar, en la técnica implícita inventamos un espacio auxiliar R de dimensión $q - p$ y S se modeliza como la preimagen $S = g^{-1}(z)$ a través de una función $g : R \rightarrow Q$ para un $z \in Q$ cualquiera. Este sería el caso de describir la circunferencia unidad como $x^2 + y^2 = 1$, en lugar de como $(\cos t, \sin t)$.

Por último, la función *polinomial* será el tipo de función sobre la que aplicaremos la técnica paramétrica. Los tres adjetivos que definen a un spline obedecen a las tres estrategias que hemos explicado. En este primer capítulo vamos a estudiar los elementos básicos, en el sentido algebraico, que nos permitirán representar estas funciones polinómicas más adelante.

La bibliografía fundamental para seguir este capítulo es [9] para polinomios de Bernstein y [7, 8] para B-splines. Para una mayor profundidad, recomiendo [5, 13], trabajos en los que me he basado para sacar los detalles más finos. No puedo olvidar mencionar el fantástico [11], que inspiró esta introducción de capítulo.

§ 1.1. Polinomios de Bernstein

La expresión de la unidad en términos del binomio de Newton induce la definición de unos polinomios con características muy interesantes:

$$1 = 1^k = (x + (1 - x))^k = \sum_{i=0}^k \binom{k}{i} x^i (1 - x)^{k-i}. \quad (1.1)$$

Estos polinomios fueron introducidos por primera vez por el matemático ucraniano Serguéi Bernstéin (1880-1968), en una demostración constructiva que dio del teorema de aproximación de Weierstrass. Dicho teorema, cuyo enunciado formal y demostración podemos encontrar en [3], afirma que cualquier función continua definida sobre un intervalo cerrado y acotado $[a, b]$ puede aproximarse tanto como se quiera por una función polinomial. Como los polinomios son una de las funciones más sencillas, son extremadamente prácticos a la hora de computarse, si bien el principal interés teórico de este resultado se encuentra en el campo de la interpolación.

Definición 1 (Polinomios de Bernstein). *Basándonos en la identidad (1.1), dado $0 \leq i \leq k$, definimos el i -ésimo polinomio de Bernstein de grado k como*

$$\mathcal{B}_i^k(x) = \binom{k}{i} x^i (1 - x)^{k-i}, \quad x \in [0, 1]. \quad (1.2)$$

La identidad (1.1) induce muchas propiedades, de entre las cuales cabe destacar que los polinomios de Bernstein constituyen una partición de la unidad. Las particiones de la unidad son un interesante concepto topológico, con la propiedad de que a veces permiten extender construcciones locales a la totalidad del espacio topológico. Existen numerosas definiciones equivalentes de este término, y por ello recomendamos una lectura introductoria como [19]. En el siguiente resultado presentamos la que mejor sirve a nuestros propósitos.

Proposición 1 (Propiedades de los polinomios de Bernstein). *Para todo entero $k \geq 0$ y para todo $0 \leq i \leq k$ se cumplen las siguientes propiedades:*

- (1) Simetría. *Se tiene $\mathcal{B}_i^k(x) = \mathcal{B}_i^k(1 - x)$, $x \in [0, 1]$.*
- (2) Positividad. *El polinomio $\mathcal{B}_i^k(x)$ es no negativo en el intervalo $[0, 1]$.*
- (3) Raíces. *Las únicas raíces de $\mathcal{B}_i^k(x)$ son $x = 0, 1$. Tienen por multiplicidad i y $k - i$ respectivamente.*
- (4) Partición de la unidad. *La familia de los polinomios de Bernstein de grado n constituye una partición de la unidad del intervalo $[0, 1]$, esto es, un conjunto R de funciones continuas no negativas*

$$f: [0, 1] \longrightarrow [0, 1]$$

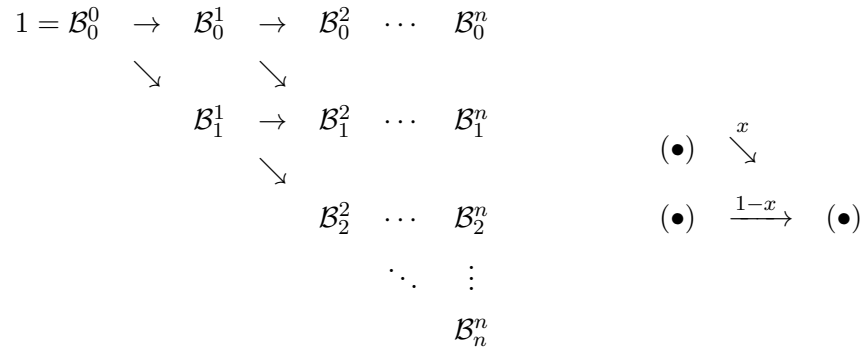
tales que $\sum_{f \in R} f(x) = 1$ para todo $x \in [0, 1]$.

(5) Independencia lineal. La familia de los polinomios de Bernstein de grado k es un sistema libre.

(6) Fórmula recursiva. Los polinomios de Bernstein satisfacen la siguiente relación de recurrencia:

$$\mathcal{B}_i^{k+1}(x) = x\mathcal{B}_{i-1}^k(x) + (1-x)\mathcal{B}_i^k(x), \quad x \in [0, 1], \quad (1.3)$$

donde $\mathcal{B}_i^k(x) = 0$ cuando $i < k$ o $i > k$ y $\mathcal{B}_0^0(x) = 1$.



Cuadro 1.1: Cálculo recursivo de los polinomios de Bernstein.

Demostración. Las propiedades de la (1) a la (4) son consecuencia inmediata de la definición y de la identidad (1.1).

Para probar la afirmación (5), como los números combinatorios, definidos en este contexto, son siempre mayores o iguales que uno, sin pérdida de generalidad basta probar que si $\lambda_0, \lambda_1, \dots, \lambda_k$ son números reales,

$$\sum_{i=0}^k \lambda_i x^i (1-x)^{k-i} = 0 \quad x \in (0, 1) \implies \lambda_0 = \lambda_1 = \dots = \lambda_k = 0. \quad (1.4)$$

Notemos que para $x = 0$ no tiene sentido estudiar la independencia lineal y por tanto, tampoco la anterior implicación, porque los polinomios de Bernstein se anulan (el vector nulo es por definición linealmente dependiente). Notemos que para $x = 1$ cuando $i \neq k$, tampoco. Si dividimos el término izquierdo de (1.4) entre $(1-x)^k$ obtenemos

$$\sum_{i=0}^k \lambda_i \left(\frac{x}{1-x} \right)^i = 0,$$

y con el cambio de variable $s = x/(1-x)$ se asevera la implicación, o bien porque s^i no se anula para $s \in (0, 1)$, o bien argumentando que porque $\{s^i \mid 0 \leq i \leq k\}$ es un sistema libre.

La propiedad (6) es consecuencia de la conocida fórmula de recurrencia de los números combinatorios,

$$\binom{k+1}{i} = \binom{k}{i-1} + \binom{k}{i},$$

de forma que

$$\mathcal{B}_i^{k+1}(x) = \binom{k+1}{i} x^i (1-x)^{k+1-i} = x \binom{k}{i-1} x^{i-1} (1-x)^{k-(i-1)} + (1-x) \binom{k}{i} x^i (1-x)^{k-i},$$

lo cual prueba (1.3). Notemos que esta fórmula es válida para $1 \leq i \leq k$, pero que se puede extender a los extremos $i = 0$ e $i = k + 1$ definiendo $\mathcal{B}_i^k(x) = 0$ cuando $i < k$ o $i > k$, tal y como hemos supuesto en la hipótesis. \square

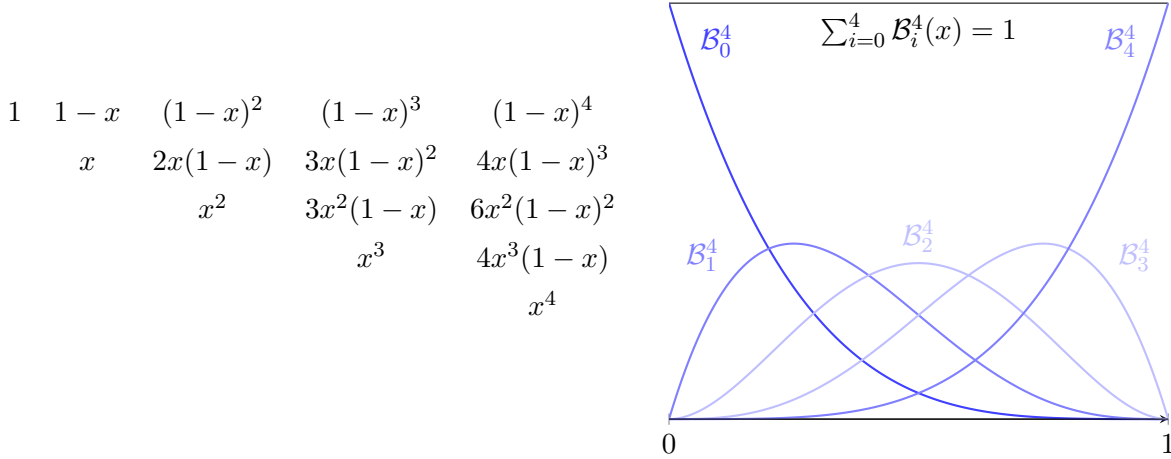


Figura 1.1: Partición de la unidad del intervalo $[0, 1]$ por los polinomios de Bernstein de grado 4.

Podemos generalizar la definición de los polinomios de Bernstein a cualquier intervalo $[a, b]$, $a < b$, con la transformación afín

$$u = a(1 - t) + bt, \quad u \in [a, b], \quad t \in [0, 1].$$

Recordemos que el espacio vectorial de los polinomios de grado menor o igual que k sobre el intervalo $[a, b]$ se define como el conjunto

$$\mathbb{P}^k([a, b]) = \left\{ a_0 + a_1x + a_2x^2 + \cdots + a_nx^k \mid a_i \in \mathbb{R}, x \in [a, b] \right\}. \quad (1.5)$$

Notemos que la propiedad (5) implica inmediatamente que el conjunto $\{\mathcal{B}_0^k, \mathcal{B}_1^k, \dots, \mathcal{B}_k^k\}$ constituye una base de $\mathbb{P}^k([a, b])$, esto es, cualquier polinomio de grado $\leq k$ sobre $[a, b]$ tiene una representación única en términos de los polinomios de Bernstein de grado k .

Todavía quedan muchas propiedades por conocer de los polinomios de Bernstein y que podemos encontrar en [9], de entre las que cabe mencionar que la expresión de la derivada y de la integral de una función polinómica en representación de Bernstein vienen dadas en términos de polinomios de Bernstein de un grado menos y de un grado más, respectivamente. Esta es una propiedad que tienen en común con los B-splines, resultado que sí probaremos.

§ 1.2. Splines

El presente desarrollo teórico está basado en los capítulos dedicados a splines de los libros [6, 7, 8]. Los splines ya los hemos presentado en la introducción del capítulo, pero todavía no los hemos definido debidamente. Si bien vamos a presentarlos en un contexto interpolatorio, que no es el objetivo de este trabajo, no podemos olvidar que es el marco teórico donde nació y donde más brilla la teoría de B-splines.

Definición 2 (Spline). *Sean los puntos $a = t_0 < \dots < t_n = b$ y el número entero $k \geq 0$. Un spline de grado k con nodos en t_0, \dots, t_n es una función $S: [a, b] \rightarrow \mathbb{R}$ que verifica:*

- (1) en cada intervalo $[t_i, t_{i+1})$, S es un polinomio de grado $\leq k$ y
- (2) S tiene derivada de orden $k - 1$ continua en $[t_0, t_n]$, esto es, $S \in C^{(k-1)}([a, b])^1$.

Supongamos ahora que junto a nuestros puntos t_0, \dots, t_n tenemos una función f de la cual sólo conocemos los valores $f(t_i)$, $i = 0, \dots, n$. Pongamos que queremos interpolarla mediante una función spline S . ¿Qué debemos exigirle a S ? Veámoslo para grados $k = 0, 1$.

$k = 0$. La condición (ii) implica que un spline de grado 0 será una función constante a trozos, de la forma $S(x) = f(t_{i-1})$, $x \in [t_{i-1}, t_i)$, $i = 1, \dots, n$.

$k = 1$. Sea $x \in [t_i, t_{i+1})$. Como S es un spline de grado 1, tendremos que $S(x) := S_i(x) = a_i x + b_i$, para unos a_i, b_i reales. Si en su lugar tomamos $x \in [t_{i+1}, t_{i+2})$, entonces $S(x) := S_{i+1}(x) = a_{i+1} x + b_{i+1}$ para otros a_{i+1}, b_{i+1} . Si exigimos que S sea función interpoladora, entonces necesariamente

$$S(t_i) = f(t_i) \Rightarrow a_i x + b_i = f(t_i). \quad (1.6)$$

Ahora, como S es un spline de grado 1, debemos exigirle continuidad por la izquierda, pues $S \in C^0([a, b])$. Entonces

$$\lim_{x \rightarrow t_{i+1}^-} S_i(x) = S_{i+1}(t_{i+1}) \Rightarrow \lim_{x \rightarrow t_{i+1}^-} (a_i x + b_i) = f(t_{i+1}) \Rightarrow (a_i x + b_i) = f(t_{i+1}). \quad (1.7)$$

Resolviendo el sistema formado por (1.6) y (1.7), obtenemos

$$a_i = \frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i}, \quad b_i = \frac{f(t_i)t_{i+1} - f(t_{i+1})t_i}{t_{i+1} - t_i},$$

y nos queda que

$$S(x) = \frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i} x + \frac{f(t_i)t_{i+1} - f(t_{i+1})t_i}{t_{i+1} - t_i} = \frac{f(t_{i+1})(x - t_i) - f(t_i)(x - t_{i+1})}{t_{i+1} - t_i}.$$

¹Entenderemos por $C^{-1}([t_0, t_n])$ al espacio de funciones continuas a trozos sobre $[t_0, t_n]$.

Si agrupamos correctamente, teniendo en cuenta que $(x - t_{i+1}) = (x - t_i) + (t_i - t_{i+1})$, entonces

$$S(x) = f(t_i) + \frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i}(x - t_i), \quad (1.8)$$

donde $t_i \leq x \leq t_{i+1}$, $i = 0, \dots, n - 1$. La fórmula (1.8) se conoce como interpolación lineal. En el contexto de los gráficos a ordenador, suele aparecer como **Lerp**, abreviación de *linear interpolation*. \triangle

Como podemos imaginar, un spline de grado 2 interpolará parejas de nodos utilizando parábolas en lugar de rectas, pero no vamos a estudiarlos en este trabajo. Podemos encontrar solución al problema de interpolación por splines cuadráticos en [7]. Sin embargo, debido a su importancia, sí que vamos a prestar especial atención a los splines de orden 3, también llamados splines cúbicos.

Los splines cúbicos son los más empleados, gracias a que ofrecen un excelente ajuste y su cálculo no es excesivamente complejo. Vamos a ver en detalle su construcción y una pincelada de la teoría splines naturales. Podemos encontrar un desarrollo más profundo si hacemos una lectura del capítulo 6 de [8] o del capítulo 3 de [2].

Supongamos que tenemos $n + 1$ nodos $t_0 < \dots < t_n$, y que conocemos sus valores y_i , $i = 0, \dots, n$. Sea S nuestro spline de orden 3, que será de la forma

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

donde

$$S_j(x) = a_j + b_j(x - t_j) + c_j(x - t_j)^2 + d_j(x - t_j)^3, \quad j = 0, \dots, n - 1. \quad (1.9)$$

Sabiendo que $S \in C^2$, es decir, que S , S' , S'' son continuas, nos preguntamos: ¿son condiciones suficientes para definir un spline cúbico? De (1.9) se deducen $4n$ incógnitas (coeficientes). Por otro lado, cada intervalo $[t_j, t_{j+1}]$ nos aporta dos condiciones de interpolación en sus extremos, pues S_{j-1} y S_j son polinomios que interpolan el mismo valor de t_j , por lo tanto

$$S_{j+1}(t_{j+1}) = S_j(t_{j+1}), \quad j = 0, \dots, n - 2. \quad (1.10)$$

lo que resulta en $2n$ ecuaciones (condiciones). La continuidad de S' nos da una condición por cada nodo interior, pues

$$S'_{j+1}(t_{j+1}) = S'_j(t_{j+1}), \quad j = 0, \dots, n - 2. \quad (1.11)$$

que son otras $n - 1$ condiciones. Análogamente, la continuidad de S'' nos da

$$S''_{j+1}(t_{j+1}) = S''_j(t_{j+1}), \quad j = 0, \dots, n - 2; \quad (1.12)$$

que son también $n - 1$ condiciones. Por tanto contamos con $4n - 2$ ecuaciones para hallar $4n$ incógnitas y contamos con 2 grados de libertad.

Observemos que $S_j(t_j) = a_j$, $j = 0, \dots, n - 2$. Conociendo (1.10) y la expresión (1.9):

$$\begin{aligned} a_{j+1} &= S_{j+1}(t_{j+1}) = S_j(t_{j+1}) \\ &= a_j + b_j(t_{j+1} - t_j) + c_j(t_{j+1} - t_j)^2 + d_j(t_{j+1} - t_j)^3, \quad j = 0, \dots, n - 2. \end{aligned}$$

Si definimos $a_n = f(x_n)$ y el paso como $h_j = t_{j+1} - t_j$ llegamos a la ecuación

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3, \quad j = 0, \dots, n - 2. \quad (1.13)$$

Ahora derivamos (1.9) y procedemos de forma similar. Definimos el coeficiente $b_n = S'(t_n)$ y utilizamos (1.11) para obtener

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2, \quad j = 0, \dots, n - 2. \quad (1.14)$$

Por último definimos de forma conveniente $c_n = S''(t_n)/2$, que junto a (1.12) y la expresión de la derivada segunda de (1.9) nos da la ecuación

$$c_{j+1} = 2c_j + 6d_j h_j, \quad j = 0, \dots, n - 2. \quad (1.15)$$

Procedemos a despejar los d_j de la ecuación (1.15), $d_j = (c_{j+1} - c_j)/3h_j$, $j = 0, \dots, n - 2$; y los sustituimos en (1.13) y (1.14). Haciendo las cuentas deducimos

$$a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}(2c_j - c_{j+1}), \quad j = 0, \dots, n - 2, \quad \text{y} \quad (1.16)$$

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}), \quad j = 0, \dots, n - 2; \quad (1.17)$$

respectivamente. De (1.16) obtenemos los $b_j = (a_{j+1} - a_j)/h_j - (2c_j - c_{j+1})h_j^2/3$, de forma que si los reducimos en un índice y sustituimos en (1.17), llegamos finalmente al sistema lineal de ecuaciones

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1}), \quad j = 0, \dots, n - 2. \quad (1.18)$$

Hemos obtenido un sistema de $n - 1$ ecuaciones con n incógnitas. Para calcular nuestro spline cúbico, resolveríamos el sistema tomando como incógnitas los c_j , para luego calcular los b_j desde (1.17) y seguidamente los d_j a partir de (1.15). Sin embargo, ¿el sistema (1.18) tiene solución única?

Definición 3 (Spline cúbico natural). *Diremos S que un spline cúbico natural si, además de cumplir (1) y (2) de la definición de spline de orden 3, cumple la condición sobre la derivada segunda (3) $S''(t_0) = S''(t_n) = 0$, conocida como condición de frontera natural², donde t_0 y t_n son el primer y el último nodo que interpola S , respectivamente.*

Teorema 1 (Existencia y unicidad del spline cúbico natural). *Sea f una función definida sobre los nodos $a = t_0 < \dots < t_n = b$. Entonces existe un único spline cúbico natural S que interpola f en $[a, b]$.*

²Existen más condiciones de frontera, como por ejemplo las fronteras de Hermite, fronteras periódicas (ambas podemos encontrarlas en [7]) o fronteras condicionadas (en [2]).

Demostración. Las condiciones de frontera de S nos llevan a que $c_n = c_0 = 0$. Añadiendo estas dos ecuaciones al sistema (1.18), donde los c_j son nuestras incógnitas, la matriz de coeficientes nos queda

$$A = \begin{bmatrix} 1 & 0 & 0 & \dots & \dots & \dots & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & \dots & \dots & \dots & 0 \\ 0 & h_1 & 2(h_1 + h_2) & \dots & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 2(h_{n-3} + h_{n-2}) & h_{n-2} & \dots & 0 \\ \vdots & \vdots & \vdots & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} & \vdots \\ 0 & \dots & \dots & 0 & 0 & h_{n-1} & 1 \end{bmatrix}.$$

Es fácil ver que A es una matriz estrictamente diagonal dominante, es decir, tal que la magnitud de la entrada diagonal excede la suma de las magnitudes de todas las otras entradas de su fila. Es un resultado conocido de álgebra lineal que, un sistema lineal cuya matriz de coeficientes es una matriz de este tipo, tiene solución única³. \square

Nota 1. Los splines de orden $k \geq 2$ presentan una característica con la que no cuentan los splines de grado menor. Se trata del sobreajuste (respectivamente infraajuste), esto es, la aparición de valores interpolados mayores (menores) que el máximo (mínimo) de los valores conocidos.

Teorema 2 (Idoneidad del spline cúbico natural). *Sea S el spline cúbico natural que interpola los nodos $a = t_0 < \dots < t_n = b$. Sea f una función interpoladora cualquiera de t_0, \dots, t_n tal que su derivada segunda es continua en $[a, b]$. Entonces:*

$$\int_a^b [S''(x)]^2 dx \leq \int_a^b [f''(x)]^2 dx. \quad (1.19)$$

Demostración. Sean y_0, \dots, y_n los valores que toman los nodos t_0, \dots, t_n . Sea $g = f - S$. Entonces

$$g(t_i) = 0, \quad i = 0, \dots, n. \quad (1.20)$$

Como $f = S + g$, desarrollando el cuadrado y aplicando la linealidad de la integral tenemos que

$$\int_a^b [f''(x)]^2 dx = \int_a^b [S''(x)]^2 dx + \int_a^b [g''(x)]^2 dx + 2 \int_a^b S''(x)g''(x) dx.$$

La demostración estará completa si probamos que $\int_a^b [S''(x)g''(x)]^2 dx \geq 0$. Para ello, integramos

³El resultado mencionado es el Lema de Hadamard, del cual podemos encontrar una prueba en [21].

por partes tomando $u = S''$ y $dv = g''dx$.

$$\begin{aligned}
\int_a^b S''(x)g''(x) dx &= \sum_{i=1}^n \int_{t_{i-1}}^{t_i} S''(x)g''(x) dx \\
&= \sum_{i=1}^n \left(\left[S''(x)g'(x) \right]_{x=t_{i-1}}^{x=t_i} - \int_{t_{i-1}}^{t_i} S'''(x)g'(x) dx \right) \\
&= \sum_{i=1}^n \left(S''(t_i)g'(t_i) - S''(t_{i-1})g'(t_{i-1}) \right) - \sum_{i=1}^n c_i \int_{t_{i-1}}^{t_i} g'(x) dx \\
&= \left(S''(t_n)g'(t_n) - S''(t_0)g'(t_0) \right) - \sum_{i=1}^n c_i (g(t_i) - g(t_{i-1})) = 0.
\end{aligned}$$

En el penúltimo paso hemos utilizado que S''' es constante a trozos (supongamos que vale c_i en $[t_{i-1}, t_i]$). En el último paso, notamos que la primera suma es telescópica y deducimos que la expresión vale 0 porque sabemos que $S''(t_n) = S''(t_0) = 0$ (es spline natural) y utilizamos (1.20). \square

El spline cúbico natural genera la función interpoladora con derivada segunda continua en $[a, b]$ más suave posible. Esto se deduce del teorema anterior, como se puede ver en (1.19). En este contexto, suavidad se refiere a menor curvatura. Recordemos que dada una curva descrita por $y = f(x)$, su curvatura se define como la cantidad

$$|f''(x)| \left(1 + [f'(x)]^2 \right)^{-3/2}.$$

Si eliminamos el término no lineal asumiendo $|f'(x)| \ll 1$, notemos que $|f''(x)|$ es una buena aproximación de la curvatura. La interpretación mencionada de este teorema se puede justificar de dos formas. O bien interpretando que el promedio integral de la curvatura es el mínimo tomando S como función interpoladora, o bien interpretando que S es la función interpoladora con menor norma en $L_2([a, b])$.

Pero no sólo podemos extraer una interpretación geométrica del teorema 2. En mecánica, el momento flector de un listón homogéneo e isotrópico⁴ cuya forma viene dada por una función $y = f(x)$ es la cantidad

$$M(x) = \alpha \frac{f''(x)}{(1 + [f'(x)]^2)^{3/2}},$$

donde α es una constante física determinada. Asumiendo nuevamente que $|f'(x)| \ll 1$, obtenemos la aproximación $\alpha_1 \int_a^b [f''(x)]^2 dx$ de la energía de deformación $E(f) = \alpha_2 \int_a^b M^2(x) dx$. Es decir, si un listón de longitud $L = b - a$ toma la forma de $f(x) = S(x)$, éste tendrá la mínima energía elástica posible.

Concluimos este párrafo mencionando que la definición de spline cúbico natural es generalizable a grados impares de la forma $2m + 1$, $m \geq 1$. Los splines naturales de orden superior cuentan con

⁴Formado por un mismo material e igual en todas las direcciones del espacio.

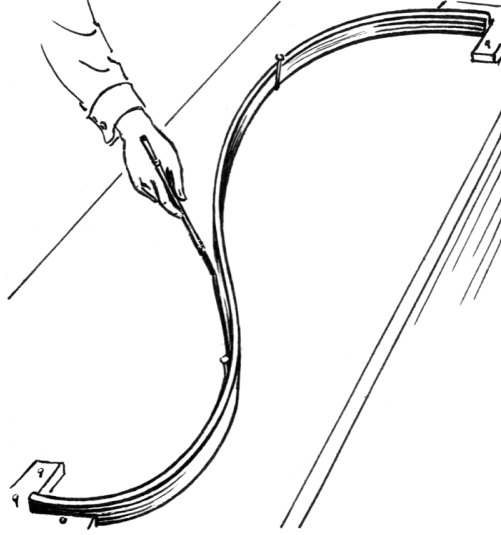


Figura 1.2: Listón de madera elástico o spline, con su forma determinada por unos clavos.

resultados completamente análogos a los vistos para el spline cúbico, desde su construcción, a su existencia única y propiedades de suavidad. Podemos encontrar estos resultados demostrados en el capítulo 6 de [8].

§ 1.3. La búsqueda de los splines básicos

Definimos el espacio de splines de grado $k \geq 0$ sobre $[t_0, t_n]$ como el conjunto

$$\mathbb{S}_n^k = \mathbb{S}^k([t_0, t_n]) := \{S \mid S \text{ es spline de grado } k \text{ definido sobre } [t_0, t_n]\}. \quad (1.21)$$

Es fácil comprobar que \mathbb{S}_n^k es \mathbb{R} -espacio vectorial. ¿Es posible encontrar, como hizo Bernstein con los polinomios, una base adecuada para representar cualquier spline? La respuesta es afirmativa, tal y como Schoenberg probó con su descubrimiento de los B-splines, abreviatura de *basis spline*. En esta sección estudiaremos la definición original en términos de diferencias divididas, siguiendo los pasos de [7] para luego encontrar y trabajar con la expresión recursiva descubierta por de Boor en [5], correspondiente a la definición moderna de los B-splines.

Lema 1. Una base para el espacio de los splines de grado $k \geq 0$ viene dada por el conjunto

$$\mathcal{L} = \left\{1, x, x^2, \dots, x^k, (x - t_1)_+^k, (x - t_2)_+^k, \dots, (x - t_{n-1})_+^k\right\}, \quad (1.22)$$

donde

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

es la llamada k -ésima potencia truncada de $x - a$. Por tanto $\mathbb{S}^k([t_0, t_n])$ es un espacio vectorial de dimensión $k + n$.

Demostración. Notemos que cualquier polinomio de grado k es automáticamente un spline de grado k si lo tomamos sobre $[t_0, t_n]$ (el recíproco no es cierto), luego las potencias x^i son splines de grado k sobre $[t_0, t_n]$. Es evidente que las potencias truncadas $(x - t_i)_+^k$ son de clase $C^{(k-1)}(\mathbb{R})$, luego todos los elementos de \mathcal{L} son splines de grado k sobre $[t_0, t_n]$ y así $\mathcal{L} \subseteq \mathbb{S}_n^k$.

Veamos que \mathcal{L} es libre y por tanto su expansión es un subespacio de \mathbb{S}_n^k de dimensión $n+k$. Sean $\lambda_1, \dots, \lambda_{n-1} \in \mathbb{R}$. Como las potencias x^i sabemos que son \mathbb{R} -linealmente independientes, basta que probemos que si

$$\sum_{i=1}^{n-1} \lambda_i (x - t_i)_+^k = 0, \quad \forall x \in [t_0, t_n],$$

entonces necesariamente $\lambda_1 = \dots = \lambda_{n-1} = 0$. En efecto, como las potencias truncadas no son la función nula y para $t_{n-1} < x \leq t_n$ se tiene que $(x - t_i)_+^k > 0$, necesariamente $\lambda_i = 0$ para $i = 1, \dots, n-1$. Luego $\dim_{\mathbb{R}}(\mathbb{S}_n^k) \geq k+n$.

Para ver la igualdad de dimensiones, veamos que cada elemento de \mathbb{S}_n^k se puede expresar como

$$S(x) = \sum_{i=0}^k a_i x^i + \sum_{i=1}^{n-1} b_i (x - t_i)_+^k, \quad x \in [t_0, t_n], \quad (1.23)$$

o lo que es lo mismo, el conjunto \mathcal{L} es generador. Comenzamos probando (1.23) en $[t_0, t_1]$. En este intervalo, todas las potencias truncadas que aparecen en (1.23) valen cero porque el índice de sumación es $i \geq 1$. Sabemos que S es un polinomio de grado a lo sumo k en $[t_0, t_1]$, llamémoslo S_0 . Por tanto, escogiendo apropiadamente sus coeficientes,

$$S_0(x) = \sum_{i=0}^k a_i x^i, \quad t_0 \leq x \leq t_1,$$

y la ecuación (1.23) es cierta en este intervalo. Notemos que esta expresión por sí sola determina de forma unívoca los a_i .

Si avanzamos al intervalo $[t_1, t_2]$, S también es otro polinomio de grado $\leq k$, digamos S_1 . Por la definición de k -spline (coincidencia de valor en los nodos y suavidad), tendremos

$$(S_1 - S_0)^{(r)}(t_1) = 0, \quad r = 0, \dots, k-1.$$

Como $(S_1 - S_0)$ es otro polinomio y vemos que tiene a t_1 como raíz de multiplicidad mayor o igual que k , pero es polinomio de grado a lo sumo k , concluimos que $(S_1 - S_0)^{(r)}(x) = b_1 (x - t_1)^k$, con lo que

$$S(x) = \sum_{i=0}^k a_i x^i + b_1 (x - t_1)_+^k, \quad x \in [t_0, t_2].$$

Repetiendo el mismo argumento para t_2, \dots, t_{n-1} habremos probado (1.23). \square

Generalmente se considera a I. J. Schoenberg como el padre de los splines, aunque quizá se trate en su lugar del padre adoptivo ([18]). El propio Schoenberg reconoció: «Los B-splines posiblemente ya eran conocidos para Hermite, y definitivamente por Peano [...] Laplace ya conocía los B-splines».

Teorema 3 (Existencia y unicidad de splines básicos, Schoenberg). *Con el objetivo de evitar definir condiciones de frontera, sea el conjunto infinito de nodos*

$$\Omega_\infty = \{t_i \mid t_i < t_{i+1}\}_{i \in \mathbb{Z}},$$

tales que $\lim_{i \rightarrow \infty} t_i = \infty$ y $\lim_{i \rightarrow -\infty} t_i = -\infty$. Entonces para cada $i \in \mathbb{Z}$ existe un único spline $S \in \mathbb{S}^k(\Omega_\infty)$ que cumple

$$S(x) = 0, \quad x \leq t_i, \quad t_{i+k+1} \leq x$$

y la siguiente condición de normalización

$$\int_{-\infty}^{+\infty} S(x) dx = \int_{t_i}^{t_{i+k+1}} S(x) dx = 1. \quad (1.24)$$

Demostración. Notemos que para $x \in [t_{i-1}, t_{i+k+1}]$ es claro que la representación de S en términos de la base \mathcal{L} no puede incluir ninguna potencia x^j , $j = 0, 1, \dots, k$ puesto que $S(x) = 0$ para los $x \leq t_i$. Entonces solo podemos escribir S en términos de las potencias truncadas, esto es, existen reales b_j , $j = 0, 1, \dots, m$ tales que

$$S(x) = \sum_{j=0}^m b_j (x - t_{i+j})_+^k, \quad (1.25)$$

para algún entero positivo m a determinar.

Tomando $m := k + 1$ y usando $S(x) = 0$ para $x \geq t_{i+k+1}$ mientras que simultáneamente se cumple $(x - t_{i+j})_+^k = (x - t_{i+j})^k$, veremos que los b_0, \dots, b_{k+1} son la solución única de un sistema de ecuaciones no singular de $k + 2$ incógnitas y $k + 2$ ecuaciones.

Con esta elección de m y con la consideración anterior, podemos deducir de (1.25) que

$$\sum_{j=0}^m b_j (x - t_{i+j})_+^k = 0, \quad x \geq t_{i+k+1}.$$

Desarrollando la anterior ecuación con el binomio de Newton y comparando los coeficientes de las potencias de x_j , $j = 0, 1, \dots, k$ obtenemos las primeras $k + 1$ ecuaciones del siguiente sistema,

$$\begin{aligned} b_0 &+ b_1 &+ \cdots &+ b_{k+1} &= 0, \\ b_0 t_i &+ b_1 t_{i+1} &+ \cdots &+ b_{k+1} t_{i+k+1} &= 0, \\ b_0 t_i^2 &+ b_1 t_{i+1}^2 &+ \cdots &+ b_{k+1} t_{i+k+1}^2 &= 0, \\ \vdots & & & & \vdots \\ b_0 t_i^k &+ b_1 t_{i+1}^k &+ \cdots &+ b_{k+1} t_{i+k+1}^k &= 0, \\ b_0 t_i^{k+1} &+ b_1 t_{i+1}^{k+1} &+ \cdots &+ b_{k+1} t_{i+k+1}^{k+1} &= (-1)^{k+1} (k + 1); \end{aligned} \quad (1.26)$$

donde la $(k + 2)$ -ésima ecuación se obtiene de la condición de normalidad.

En efecto, de (1.24) se puede hacer la siguiente cadena deductiva

$$\begin{aligned}
1 &= \int_{t_i}^{t_{i+k+1}} S(x) dx = \int_{t_i}^{t_{i+k+1}} \sum_{j=0}^{k+1} b_j (x - t_{i+j})_+^k dx = \sum_{j=0}^{k+1} \int_{t_i}^{t_{i+k+1}} b_j (x - t_{i+j})_+^k dx \\
&= \sum_{j=0}^{k+1} \int_{t_{i+j}}^{t_{i+k+1}} b_j (x - t_{i+j})^k dx = \sum_{j=0}^{k+1} \frac{b_j}{k+1} \left[(x - t_{i+j})^{k+1} \right]_{x=t_{i+j}}^{x=t_{i+k+1}} \\
&= \sum_{j=0}^{k+1} \frac{b_j}{k+1} (t_{i+k+1} - t_{i+j})^{k+1} = 1.
\end{aligned}$$

En [7] no se muestra este paso, pero veamos que con un poco de picardía podemos transformar

$$\sum_{j=0}^{k+1} b_j (t_{i+k+1} - t_{i+j})^{k+1} = k+1 \tag{1.27}$$

en la $(k+2)$ -ésima ecuación. Desarrollando el binomio de Newton en (1.27) obtenemos

$$\sum_{j=0}^{k+1} b_j \sum_{l=0}^{k+1} \binom{k+1}{l} t_{i+k+1}^{k+1-l} (-1)^l t_{i+j}^l = \sum_{l=0}^{k+1} (-1)^l t_{i+k+1}^{k+1-l} \binom{k+1}{l} \sum_{j=0}^{k+1} b_j t_{i+j}^l = k+1. \tag{1.28}$$

Pero las primeras $k+1$ ecuaciones de (1.26) implican que la suma que aparece en (1.28) se anula para todo l salvo $l = k+1$, esto es

$$\sum_{j=0}^{k+1} b_j t_{i+j}^{k+1} = b_0 t_i^{k+1} + b_1 t_{i+1}^{k+1} + \dots + b_{k+1} t_{i+k+1}^{k+1} = (-1)^{k+1} (k+1),$$

nuestra $(k+2)$ -ésima ecuación.

Para concluir la demostración basta que observemos que (1.26) es un sistema lineal no singular puesto que su determinante es de Vandermonde y por tanto, distinto de cero. Luego su solución existe y es única. \square

Nota 2. Hemos probado que los splines básicos existen, y que si los normalizamos son únicos.

Corolario 1 (Soporte minimal). *No existen splines no triviales de grado k cuyo soporte⁵ sea un subconjunto propio del soporte del spline definido en el teorema 3.*

Demostración. Tomemos $m \leq k$ en el teorema anterior. Veamos que necesariamente $S = 0$.

En efecto, la ecuación (1.25) queda

$$S(x) = \sum_{j=0}^k b_j (x - t_{i+j})_+^k = \sum_{j=0}^k b_j (x - t_{i+j})^k = 0, \quad x \geq x_{i+k},$$

⁵Por soporte de una función, entendemos el subconjunto de su dominio formado por los puntos donde la función no es nula.

luego el sistema (1.26) tiene una columna menos como mínimo. Así, del sistema original podemos rescatar m ecuaciones homogéneas para nuestras m incógnitas. Pero la matriz asociada a este nuevo sistema sigue siendo de Vandermonde, y por tanto no singular. Luego el sistema homogéneo

$$b_0 t_i^l + b_1 t_{i+1}^l + \dots + b_m t_{i+m}^l = 0, \quad l = 0, \dots, m;$$

tiene solución única y es $b_0 = \dots = b_m = 0$, con lo que hemos probado que $S = 0$. \square

Una vez probada la existencia y unicidad de los splines básicos, vamos a introducir el concepto de diferencia dividida. Se trata de una herramienta clásica del análisis numérico, que permite, si se tiene ya computado un polinomio de interpolación, calcular otro de un grado mayor reutilizando la expresión que ya se conoce. La definición original de Schoenberg se basa en diferencias divididas. Para conocer más sobre cómo se originaron y sus propiedades básicas, recomendamos una lectura del capítulo 5 de [7].

Definición 4 (Diferencia dividida). *Dada una función f definida sobre un conjunto de nodos t_i , definimos la diferencia dividida de primer orden en $\{t_i, t_{i+1}\}$ como el cociente*

$$[t_{i+1} \ t_i]f := \frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i}.$$

Definimos la diferencia finita de m -ésimo orden en $\{t_i, \dots, t_{i+m}\}$ como

$$[t_{i+m} \dots \ t_i]f := \frac{[t_{i+m} \dots \ t_{i+1}]f - [t_{i+m-1} \dots \ t_i]f}{t_{i+m} - t_i}.$$

Si no da lugar a confusión sobre qué función estamos calculando la diferencia finita, podemos prescindir de escribir f junto a los corchetes.

Las únicas propiedades que vamos a necesitar controlar sobre diferencias divididas son su simetría, esto es, que dada cualquier permutación σ de los nodos t_i , $[t_{\sigma(i+m)} \dots \ t_{\sigma(i)}] = [t_{i+m} \dots \ t_i]$, y la Regla de Leibniz. Esta última procedemos a enunciarla a continuación. Para una demostración, podemos consultar la pagina 191 de [7].

Teorema 4 (Regla de Leibniz). *Sea una función de la forma $f(x) = u(x)v(x)$ definida sobre un conjunto de nodos t_i . Entonces*

$$[t_{j+k} \dots \ t_j]f = \sum_{l=j}^{j+k} ([t_j \dots \ t_l]u)([t_l \dots \ t_{j+k}]v), \quad (1.29)$$

donde $[t_j]u := u(t_j)$ y $[t_{j+k}]v := v(t_{j+k})$.

Definición 5 (B-spline, definición de Schoenberg). *Sea el número entero $k \geq 0$. Sea $i \in \mathbb{Z}$. Definimos el i -ésimo B-spline de grado k como*

$$B_i^k(x) = (t_{i+k+1} - t_i)[t_i \dots \ t_{i+k+1}]g_k(\cdot, x), \quad k \geq 1, \quad (1.30)$$

$$B_i^0(x) = \begin{cases} 1 & \text{si } x \in [t_i, t_{i+1}), \\ 0 & \text{en caso contrario,} \end{cases}$$

donde $g_k(t, x) = (t - x)_+^k$.

Realmente en su definición, Schoenberg utiliza otra condición de normalización distinta a la de su teorema, esto es, difieren en una constante de normalización (la podremos calcular más adelante). Sin embargo, la definición útil, y que se suele encontrar cuando uno busca información de B-splines, es la definición recursiva que obtuvieron en 1971 de Boor, Cox y Mansfield, resultado publicado en [5]. A continuación vamos a deducir esta fórmula y las principales propiedades de los B-splines a partir de ella.

Sea $k \geq 1$. Aplicamos la regla de Leibniz a la potencia truncada $g_k(t, x)$. Para ello, la expresamos como producto de dos funciones:

$$g_k(t, x) = (t - x)_+^k = (t - x)(t - x)_+^{k-1} = f(t, x)g_{k-1}(t, x).$$

Es fácil comprobar que las diferencias divididas de orden mayor que dos de $f(t, x)$ se anulan sin más que aplicar la definición, con lo que nos queda

$$\begin{aligned} [t_i \dots t_{i+k+1}]g_k(\cdot, x) &= [t_{i+k+1} \dots t_i]g_k(\cdot, x) = \sum_{l=i}^{i+k+1} ([t_i \dots t_l]f(\cdot, x)) ([t_l \dots t_{i+k+1}]g_{k-1}(\cdot, x)) \\ &= f(t_i, x)[t_i \dots t_{i+k+1}]g_{k-1}(\cdot, x) + 1 \cdot [t_{i+1} \dots t_{i+k+1}]g_{k-1}(\cdot, x) \\ &\quad + \sum_{l=i+2}^{i+k+1} 0 \cdot ([t_l \dots t_{i+k+1}]g_{k-1}(\cdot, x)) \\ &= (t_i - x)[t_i \dots t_{i+k+1}]g_{k-1}(\cdot, x) + [t_{i+1} \dots t_{i+k+1}]g_{k-1}(\cdot, x). \end{aligned}$$

A continuación desarrollamos la primera diferencia dividida, con lo que se obtiene

$$\begin{aligned} [t_i \dots t_{i+k+1}]g_k(\cdot, x) &= (t_i - x) \left[\frac{[t_{i+1} \dots t_{i+k+1}]g_{k-1}(\cdot, x) - [t_i \dots t_{i+k}]g_{k-1}(\cdot, x)}{t_{i+k+1} - t_i} \right] \\ &\quad + [t_{i+1} \dots t_{i+k+1}]g_{k-1}(\cdot, x) \\ &= \left(\frac{t_i - x}{t_{i+k+1} - t_i} + 1 \right) [t_{i+1} \dots t_{i+k+1}]g_{k-1}(\cdot, x) \\ &\quad - \left(\frac{t_i - x}{t_{i+k+1} - t_i} \right) [t_i \dots t_{i+k}]g_{k-1}(\cdot, x). \end{aligned}$$

Con lo que tenemos

$$[t_i \dots t_{i+k+1}]g_k(\cdot, x) = \left(\frac{t_{i+k+1} - x}{t_{i+k+1} - t_i} \right) [t_{i+1} \dots t_{i+k+1}]g_{k-1}(\cdot, x) - \left(\frac{t_i - x}{t_{i+k+1} - t_i} \right) [t_i \dots t_{i+k}]g_{k-1}(\cdot, x).$$

Si multiplicamos por $(t_{i+k+1} - t_i)$ se sigue que

$$\begin{aligned}
B_i^k(x) &= (t_{i+k+1} - t_i)[t_i \dots t_{i+k+1}]g_k(\cdot, x) \\
&= (t_{i+k+1} - x)[t_{i+1} \dots t_{i+k+1}]g_{k-1}(\cdot, x) - (t_i - x)[t_i \dots t_{i+k}]g_{k-1}(\cdot, x) \\
&= \left(\frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} \right) (t_{i+k+1} - t_{i+1})[t_{i+1} \dots t_{i+k+1}]g_{k-1}(\cdot, x) \\
&\quad - \left(\frac{t_i - x}{t_{i+k} - t_i} \right) (t_{i+k} - t_i)[t_i \dots t_{i+k}]g_{k-1}(\cdot, x) \\
&= \left(\frac{x - t_i}{t_{i+k} - t_i} \right) B_i^{k-1}(x) + \left(\frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} \right) B_{i+1}^{k-1}(x).
\end{aligned}$$

Definición 6 (B-spline, definición de Cox-de Boor-Mansfield). *Sea el número entero $k \geq 0$. Sea $i \in \mathbb{Z}$. Definimos el i -ésimo B-spline de grado k como*

$$\begin{aligned}
B_i^k(x) &= V_i^k(x)B_i^{k-1}(x) + (1 - V_{i+1}^k(x))B_{i+1}^{k-1}(x), \quad k \geq 1, \\
B_i^0(x) &= \begin{cases} 1 & \text{si } x \in [t_i, t_{i+1}), \\ 0 & \text{en caso contrario,} \end{cases}
\end{aligned} \tag{1.31}$$

donde los coeficientes lineales son

$$V_i^k(x) = \frac{x - t_i}{t_{i+k} - t_i}, \quad 1 - V_{i+1}^k(x) = \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}}, \quad k \geq 1. \tag{1.32}$$

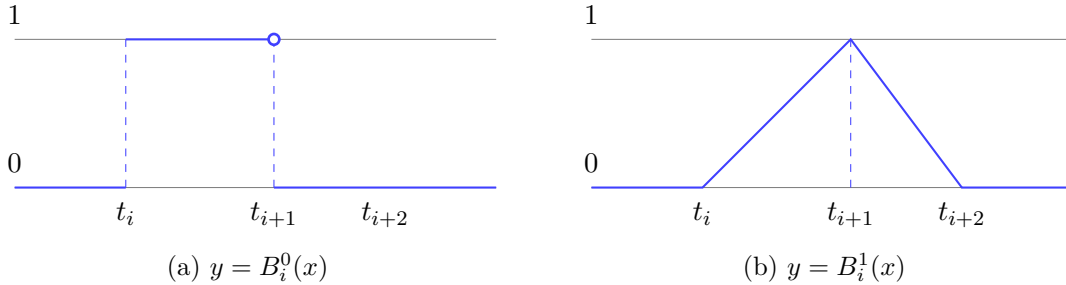


Figura 1.3: B-splines i -ésimos de grado 0 y 1.

Podemos extender la definición de B-spline de De Boor a una partición con nodos de multiplicidad mayor que uno, esto es, a un conjunto $\Omega_\infty = \{t_i \mid t_i \leq t_{i+1}\}_{i \in \mathbb{Z}}$. Para ello, si $t_i = t_{i+1} = \dots = t_{i+r}$, basta tomar la definición (1.31) y tener en cuenta la convención

$$B_i^{r-1} \equiv \frac{B_i^{r-1}}{t_{i+r} - t_i} = 0. \tag{1.33}$$

Sin embargo, a no ser que digamos lo contrario, vamos a estar trabajando con nodos simples. En tales condiciones y tomando como dominio el intervalo $[0, 1]$ con la partición $0 = t_1 = \dots = t_n < t_{n+1} = \dots = t_{2n} = 1$, entonces la recursión (1.31) coincide con la recursión (1.2) de los polinomios de Bernstein:

$$B_i^n(t) = \mathcal{B}_i^n(t), \quad t \in [0, 1].$$

¡Los B-splines son una generalización de los polinomios de Bernstein!

A pesar de esto, una desventaja de la representación B-spline, aparte de la mayor complejidad matemática, consiste en que las funciones básicas no admiten una expresión explícita, y cambian al variar el vector de nodos. De hecho, muchos programas de CAD, para procesar splines, transforman cada segmento al modelo de Bézier y así el procesado resulta más eficiente.

§ 1.4. Propiedades de los B-splines

Una vez descubierta una expresión recursiva, podemos estudiar de forma más sencilla las principales propiedades de los B-splines. En esta sección seguiremos la filosofía de [8], complementando los resultados con detalles más finos de [13]. Entre las propiedades más importantes que vamos a demostrar, cabe destacar el algoritmo de De Boor, que se trata del principal resultado que sustenta la teoría de curvas B-spline. La herramienta matemática a la que más vamos a recurrir será el principio de inducción, aunque también aparecerán pruebas directas.

Lema 2 (Soporte local). *Si $k \geq 1$ y $x \notin (t_i, t_{i+k+1})$, entonces $B_i^k(x) = 0$.*

Demostración. Trivial para los grados $k = 0, 1$. Lo probamos por inducción en $k \in \mathbb{N}$. Supongámoslo cierto para grado $k - 1$, con $k \geq 2$. Si $x \notin (t_i, t_{i+k+1})$, entonces en particular ni $x \notin (t_i, t_{i+k})$ ni $x \notin (t_{i+1}, t_{i+k+1})$. En consecuencia, por hipótesis de inducción, $B_i^{k-1}(x) = B_{i+1}^{k-1}(x) = 0$. Luego $B_i^k(x) = 0$ por la definición recursiva (1.31). \square

Lema 3 (Positividad). *Si $k \geq 0$ y $x \in (t_i, t_{i+k+1})$, entonces $B_i^k(x) > 0$.*

Demostración. Trivial para $k = 0, 1$. Usamos de nuevo inducción en $k \in \mathbb{N}$. Lo suponemos cierto para $k - 1$. Esta suposición junto al lema anterior implica que $B_i^k(x) \geq 0$ para toda $i \in \mathbb{Z}$ y $x \in \mathbb{R}$. Sea $x \in (t_i, t_{i+k+1})$, entonces los factores lineales en (1.31) son positivos y por hipótesis de inducción, $B_i^{k-1}(x), B_{i+1}^{k-1}(x) > 0$. Luego $B_i^k(x) > 0$. \square

Nota 3. El grado $k = 0$ es algo más especial, pues se tiene que $B_i^0(x) = 0$ para $x \notin [t_i, t_{i+1})$ y $B_i^0(x) = 1$ para $x \in [t_i, t_{i+1})$.

Lema 4 (Combinación lineal de B-splines). *Sin pérdida de generalidad, sean los escalares c_i , $i \in \mathbb{Z}$. Entonces*

$$\sum_{i=-\infty}^{\infty} c_i B_i^k = \sum_{i=-\infty}^{\infty} [c_i V_i^k + c_{i-1}(1 - V_i^k)] B_i^{k-1}, \quad (1.34)$$

donde los coeficientes V_i^k son los definidos en (1.32).

Demostración. Basta desarrollar la expresión (1.34) desde el miembro izquierdo y reagrupar las series como sigue:

$$\begin{aligned}
\sum_{i=-\infty}^{\infty} c_i B_i^k &= \sum_{i=-\infty}^{\infty} c_i \left[V_i^k B_i^{k-1} + (1 - V_i^k) B_{i+1}^{k-1} \right] \\
&= \sum_{i=-\infty}^{\infty} c_i V_i^k B_i^{k-1} + \sum_{i=-\infty}^{\infty} c_i (1 - V_i^k) B_{i+1}^{k-1} \\
&= \sum_{i=-\infty}^{\infty} c_i V_i^k B_i^{k-1} + \sum_{i=-\infty}^{\infty} c_{j-1} (1 - V_{j-1}^k) B_j^{k-1} \\
&= \sum_{i=-\infty}^{\infty} \left[c_i V_i^k + c_{i-1} (1 - V_i^k) \right] B_i^{k-1}.
\end{aligned}$$

□

Nota 4. Notemos que en el lema anterior, los coeficientes c_i podemos tomarlos como constantes, funciones, ¡e incluso puntos del plano o del espacio! Esto se debe a que en la prueba sólo utilizamos la definición recursiva de B-spline, la propiedad distributiva y un cambio de índices en la serie. Este hecho va tener mucha importancia, como vamos a ver ahora.

Algoritmo de De Boor

Vistas estas primeras propiedades podemos deducir un procedimiento numérico para trabajar con B-splines. Supongamos que, fijado un entero $k \geq 0$, tenemos una función f expresada como combinación lineal de los elementos de la sucesión $\{B_i^k(x) \mid x \in \mathbb{R}\}_{i \in \mathbb{Z}}$,

$$f(x) = \sum_{i=-\infty}^{\infty} C_i^k(x) B_i^k(x), \quad x \in \mathbb{R},$$

donde la sucesión de coeficientes $\{C_i^k(x)\}_{i \in \mathbb{Z}}$ ya es conocida. Dado un $x \in \mathbb{R}$, ¿existe alguna forma de calcular el valor $f(x)$ sin pasar por calcular primero los B-splines?

Nos inspiramos en la expresión del lema 4 y definimos unos nuevos coeficientes a partir de (1.32)

$$\begin{aligned}
C_i^{k-1}(x) &:= C_i^k(x) V_i^k(x) + C_{i-1}^k(x) (1 - V_i^k(x)) \\
&= \frac{1}{t_{i+k} - t_i} \left(C_i^k(x) (x - t_i) + C_{i-1}^k(x) (t_{i+k} - x) \right), \quad k \geq 1,
\end{aligned} \tag{1.35}$$

de forma que podemos calcularlos a partir de los $C_i^k(x)$ ya conocidos. Usando el lema 4 y la ecuación (1.35), notemos que

$$f(x) = \sum_{i=-\infty}^{\infty} C_i^k(x) B_i^k(x) = \sum_{i=-\infty}^{\infty} C_i^{k-1}(x) B_i^{k-1}(x), \quad x \in \mathbb{R},$$

y si reiteramos para $k - 1, \dots, 0$ se llega a

$$f(x) = \sum_{i=-\infty}^{\infty} C_i^0(x) B_i^0(x), \quad x \in \mathbb{R}.$$

Si identificamos el $m \in \mathbb{Z}$ para el cual $x \in [t_m, t_{m+1})$, por las propiedades de los B-splines de grado 0, tendremos

$$f(x) = C_m^0(x).$$

Este algoritmo, aplicado en el contexto de curvas splines, se conoce como algoritmo de De Boor (1972). Lo vamos a estudiar en mayor profundidad en el capítulo siguiente, pero veremos que es muy útil para probar algunos resultados. \triangle

$$\begin{array}{ccccccc} C_m^k & \rightarrow & C_m^{k-1} & \cdots & C_m^1 & \rightarrow & C_m^0 \\ C_{m-1}^k & \nearrow & C_{m-1}^{k-1} & \cdots & C_{m-1}^k & \nearrow & \\ \vdots & & \vdots & \ddots & & & \\ C_{m-k+1}^k & \rightarrow & C_{m-k+1}^{k-1} & & & & \\ C_{m-k}^k & \nearrow & & & & & \end{array}$$

Cuadro 1.2: Arreglo triangular asociado al algoritmo de De Boor.

Lema 5. Para todo entero $k \geq 0$ y toda $x \in \mathbb{R}$, $\sum_{i=-\infty}^{\infty} B_i^k(x) = 1$.

Demostración. Para probar el resultado, basta utilizar el algoritmo de De Boor con $C_i^k(x) = 1$ para toda $i \in \mathbb{Z}$, $x \in \mathbb{R}$. \square

Una vez vistas estas primeras propiedades básicas de B-splines, vamos a estudiar cualidades más interesantes, entre ellas el cálculo recursivo de la derivada y de la integral de B-splines. Introducimos a continuación los siguientes coeficientes

$$\alpha_i^k = \frac{1}{t_{i+k} - t_i},$$

cuya notación nos ahorrará bastante tinta a la hora de calcular la derivada.

Es trivial comprobar las siguientes relaciones entre los α_i^k y los V_i^k , definidos en (1.32):

$$\frac{d}{dx} V_i^k(x) = \alpha_i^k, \quad (1.36)$$

$$\alpha_i^k V_i^{k+1}(x) = \alpha_i^{k+1} V_i^k(x), \quad (1.37)$$

$$\alpha_{i+1}^k (1 - V_i^{k+1}(x)) = \alpha_i^{k+1} (1 - V_{i+1}^k(x)). \quad (1.38)$$

Utilizaremos estas fórmulas en la demostración del siguiente resultado, correspondiente a la derivada de un B-spline.

Lema 6 (Derivada de un B-spline). Para todo entero $k \geq 2$ y toda $x \in \mathbb{R}$,

$$\frac{d}{dx} B_i^k(x) = \left(\frac{k}{t_{i+k} - t_i} \right) B_i^{k-1}(x) - \left(\frac{k}{t_{i+k+1} - t_{i+1}} \right) B_{i+1}^{k-1}(x). \quad (1.39)$$

Para $k = 1$ la anterior expresión es cierta salvo en $x = t_i, t_{i+1}, t_{i+2}$.

Demostración. Vamos a demostrar (1.39) por inducción.

$k = 1$. Suponemos $x \neq t_i, t_{i+1}, t_{i+2}$.

$$\begin{aligned} \frac{d}{dx} B_i^1(x) &= \frac{d}{dx} \left(V_i^1 B_i^0 + (1 - V_{i+1}^1) B_{i+1}^0 \right) = \frac{d}{dx} V_i^1 B_i^0 + \frac{d}{dx} (1 - V_{i+1}^1) B_{i+1}^0 \\ &= \alpha_i^1 B_i^0 + (1 - \alpha_{i+1}^1) B_{i+1}^0 = \left(\frac{1}{t_{i+1} - t_i} \right) B_i^0 - \left(\frac{1}{t_{i+2} - t_{i+1}} \right) B_{i+1}^0. \end{aligned}$$

Notar que para $x = t_i, t_{i+2}$, $B_i^1(x) = 0$, con lo que la derivada es 0 y la fórmula no es válida. Por último, para $x = t_{i+1}$ se tiene $B_i^1(x) = 1$ y la derivada también es 0.

Suponemos

$$\frac{d}{dx} B_i^k(x) = k\alpha_i^k B_i^{k-1} - k\alpha_{i+1}^k B_{i+1}^{k-1}, \quad k \geq 2.$$

Veamos que es cierto para $k + 1$.

$$\begin{aligned} \frac{d}{dx} B_i^{k+1}(x) &= \frac{d}{dx} \left(V_i^{k+1} B_i^k + (1 - V_{i+1}^{k+1}) B_{i+1}^k \right) \\ &= \frac{d}{dx} \left[V_i^{k+1} B_i^k \right] + \frac{d}{dx} \left[(1 - V_{i+1}^{k+1}) B_{i+1}^k \right] \\ &= \left[\alpha_i^{k+1} B_i^k + V_i^{k+1} \frac{d}{dx} B_i^k \right] + \left[-\alpha_{i+1}^{k+1} B_{i+1}^k + (1 - V_{i+1}^{k+1}) \frac{d}{dx} B_{i+1}^k \right]. \end{aligned}$$

En la anterior cadena de igualdades hemos usado dos veces (1.36). Aplicando la hipótesis de inducción se sigue que

$$\begin{aligned} \frac{d}{dx} B_i^{k+1}(x) &= \alpha_i^{k+1} B_i^k + V_i^{k+1} \left[k\alpha_i^k B_i^{k-1} - k\alpha_{i+1}^k B_{i+1}^{k-1} \right] \\ &\quad - \alpha_{i+1}^{k+1} B_{i+1}^k + (1 - V_{i+1}^{k+1}) \left[k\alpha_{i+1}^k B_{i+1}^{k-1} - k\alpha_{i+2}^k B_{i+2}^{k-1} \right] \\ &= \alpha_i^{k+1} B_i^k + \left[k\alpha_i^k V_i^{k+1} B_i^{k-1} \right] \\ &\quad - \alpha_{i+1}^{k+1} B_{i+1}^k + \left[-k\alpha_{i+2}^k (1 - V_{i+1}^{k+1}) B_{i+2}^{k-1} \right] \\ &\quad + \left[-k\alpha_{i+1}^k V_i^{k+1} B_{i+1}^{k-1} + k\alpha_{i+1}^k (1 - V_{i+1}^{k+1}) B_{i+1}^{k-1} \right]. \end{aligned}$$

En el último miembro de la cadena de igualdades anterior, manipulamos los sumandos delimitados por corchetes de la siguiente forma:

$$\begin{aligned} k\alpha_i^k V_i^{k+1} B_i^{k-1} &= k\alpha_i^{k+1} V_i^k B_i^{k-1} \\ -k\alpha_{i+2}^k (1 - V_{i+1}^{k+1}) B_{i+2}^{k-1} &= -k\alpha_{i+1}^{k+1} (1 - V_{i+2}^k) B_{i+2}^{k-1} \\ -k\alpha_{i+1}^k V_i^{k+1} B_{i+1}^{k-1} + k\alpha_{i+1}^k (1 - V_{i+1}^{k+1}) B_{i+1}^{k-1} &= k\alpha_{i+1}^k (1 - V_i^{k+1}) B_{i+1}^{k-1} - k\alpha_{i+1}^k V_{i+1}^{k+1} B_{i+1}^{k-1} \\ &= k\alpha_i^{k+1} (1 - V_{i+1}^k) B_{i+1}^{k-1} - k\alpha_{i+1}^{k+1} V_{i+1}^k B_{i+1}^{k-1} \end{aligned}$$

Finalmente, reagrupando de forma conveniente y aplicando la definición se sigue

$$\begin{aligned}
\frac{d}{dx} B_i^{k+1}(x) &= \alpha_i^{k+1} B_i^k + k\alpha_i^{k+1} \left[V_i^k B_i^{k-1} + (1 - V_{i+1}^k) B_{i+1}^{k-1} \right] \\
&\quad - \alpha_{i+1}^{k+1} B_{i+1}^k - \alpha_{i+1}^{k+1} \left[V_{i+1}^k B_{i+1}^{k-1} + (1 - V_{i+2}^k) B_{i+2}^{k-1} \right] \\
&= \alpha_i^{k+1} B_i^k + k\alpha_i^{k+1} B_i^k - \alpha_{i+1}^{k+1} B_{i+1}^k - k\alpha_{i+1}^{k+1} B_{i+1}^k \\
&= (k+1)\alpha_i^{k+1} B_i^k - (k+1)\alpha_{i+1}^{k+1} B_{i+1}^k.
\end{aligned}$$

□

Lema 7 (Suavidad). *Si tomamos una partición de la recta real formada exclusivamente por nodos simples, entonces para todo entero $k \geq 1$, $B_i^k \in C^{k-1}(\mathbb{R})$. En caso contrario, si existen nodos de multiplicidad k se tiene que B_i^k es tan solo $n - k$ veces continuamente diferenciable sobre ellos, y de clase C^{k-1} en el resto del dominio.*

Demostración. Supongamos una partición por nodos simples. Para $k = 1$ es trivial ver que los $B_i^1 \in C^0(\mathbb{R})$. Si suponemos que es cierto para k , por el lema 6 se tiene $B_i^k \in C^k(\mathbb{R})$ porque su derivada es combinación lineal de $B_i^k, B_{i+1}^k \in C^{k-1}(\mathbb{R})$ por hipótesis de inducción. Por último, por el resultado del lema 6, es claro que los cocientes $k/(t_{i+k} - t_i)$ y $k/(t_{i+k+1} - t_{i+1})$ no están definidos si $t_i = \dots = t_{i+k}$ o $t_{i+1} = \dots = t_{i+k+1}$. En el resto del dominio, la demostración por inducción anterior se mantiene válida. □

Los resultados vistos hasta ahora tienen como consecuencia inmediata que los B-splines conforman una partición de la unidad de los números reales.

Corolario 2 (Partición de la unidad). *La sucesión de B-splines de grado $k \geq 1$ constituye una partición de la unidad de la recta real, esto es, un conjunto R de funciones continuas no negativas $f: \mathbb{R} \rightarrow [0, 1]$ tales que $\sum_{f \in R} f(x) = 1$ para todo $x \in \mathbb{R}$. Es más, dado $x \in [t_i, t_{i+1})$,*

$$\sum_{j=-\infty}^{\infty} B_j^k(x) = \sum_{j=i-k}^i B_j^k(x) = 1, \tag{1.40}$$

esto es, también es partición de la unidad de los intervalos nodales.

Demostración. La última afirmación es consecuencia inmediata del lema 2 (soporte compacto). □

A efectos de simplificar la demostración del siguiente resultado, vamos a deternos a hacer una observación. Conocida la derivada de un B-spline en el lema 6, podemos intentar obtener una expresión para la derivada de una combinación lineal de B-splines. A partir de esta obtendremos la expresión para la integral de un B-spline en términos de B-splines de un grado mayor.

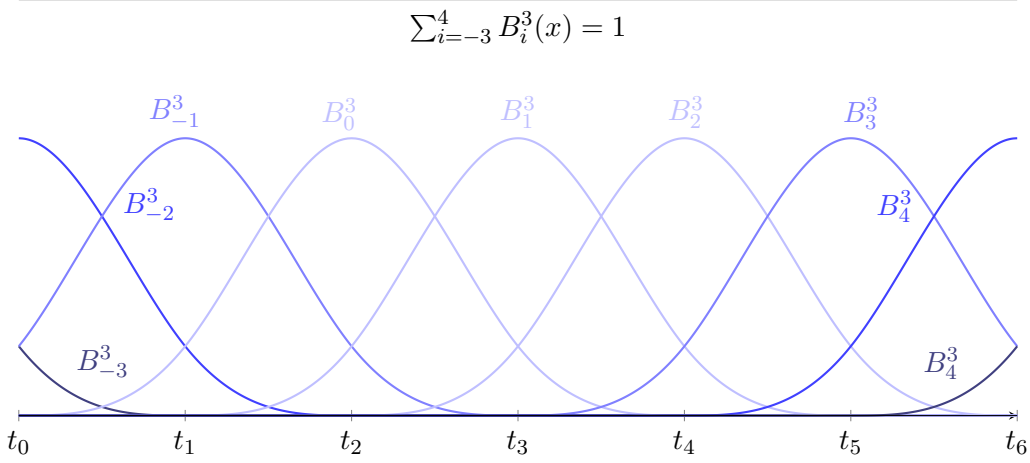


Figura 1.4: partición de la unidad del intervalo $[t_0, t_6]$ por B-splines de grado 3.

Sean c_i , $i \in \mathbb{Z}$, coeficientes constantes y un entero $k \geq 2$. Entonces

$$\begin{aligned} \frac{d}{dx} \left(\sum_{i=-\infty}^{\infty} c_i B_j^k(x) \right) &= \sum_{i=-\infty}^{\infty} c_i \frac{d}{dx} B_j^k(x) \\ &= \sum_{i=-\infty}^{\infty} c_i \left(\left(\frac{k}{t_{i+k} - t_i} \right) B_i^{k-1}(x) - \left(\frac{k}{t_{i+k+1} - t_{i+1}} \right) B_{i+1}^{k-1}(x) \right) \\ &= \sum_{i=-\infty}^{\infty} \left(\frac{c_i - c_{i-1}}{t_{i+k} - t_i} \right) B_i^{k-1}(x), \quad k \geq 2. \end{aligned} \quad (1.41)$$

Para $k = 1$ la anterior expresión es cierta salvo en $x = t_i, t_{i+1}, t_{i+2}$. △

Lema 8 (Integral de un B-spline). *Para todo entero $k \geq 2$ y toda $x \in \mathbb{R}$,*

$$\int_{-\infty}^x B_i^k(s) ds = \left(\frac{t_{i+k+1} - t_i}{k+1} \right) \sum_{j=1}^{\infty} B_j^{k+1}(x). \quad (1.42)$$

Demostración. Derivemos los dos miembros de la ecuación (1.42) y comprobemos que coinciden. Comenzamos estudiando el segundo miembro. Notemos que si definimos los siguientes coeficientes c_j como

$$c_j = \begin{cases} 0 & \text{si } j < i, \\ 1 & \text{si } j \geq i, \end{cases}$$

el segundo miembro se puede expresar como

$$\left(\frac{t_{i+k+1} - t_i}{k+1} \right) \sum_{j=-\infty}^{\infty} c_j B_j^{k+1}(x).$$

De esta forma, si derivamos teniendo en cuenta los c_j y la ecuación (1.41), llegamos a

$$\left(\frac{t_{i+k+1} - t_i}{k+1}\right) (k+1) \sum_{j=-\infty}^{\infty} \left(\frac{c_j - c_{j-1}}{t_{j+k} - t_j}\right) B_j^k(x) = (t_{i+k+1} - t_i) \left(\frac{1}{t_{i+k} - t_i}\right) B_i^k(x) = B_i^k(x).$$

La igualdad de derivadas se verifica utilizando el Teorema fundamental del cálculo en el primer miembro. Para completar la demostración basta integrar ambos miembros, aplicando el Teorema fundamental del cálculo en el primer miembro. En el segundo aparecerá una constante de integración, que se puede comprobar que es 0 tomando por ejemplo $x = t_i$, de forma que ambos miembros se anulen (ver lema 2). \square

Conociendo (1.42) ya estamos en condiciones de obtener la constante de normalización de la definición de Schoenberg, con lo que concluimos este párrafo:

$$\int_{-\infty}^{+\infty} B_i^k(s) ds = \int_{t_i}^{t_{i+k+1}} B_i^k(s) ds = \left(\frac{t_{i+k+1} - t_i}{k+1}\right) \sum_{j=1}^{\infty} B_j^{k+1}(t_{i+k+1}) = \frac{t_{i+k+1} - t_i}{k+1}. \quad (1.43)$$

En el último paso hemos aplicado que los B-splines son partición de la unidad. \triangle

§ 1.5. El teorema de representación

Ya probadas las principales propiedades de los B-splines, podemos por fin demostrar el principal resultado de teoría de B-splines. Vamos a probar que cualquier spline de grado k puede expresarse de forma única como combinación lineal de B-splines de grado k , o lo que es lo mismo, los B-splines de grado k forman una base del espacio vectorial de splines de grado k .

Sea una función real f y K un subconjunto de su dominio. Denotaremos por $f|_K$ a la restricción de f en K , es decir

$$(f|_K)(x) = f(x), \quad x \in K.$$

Vamos a trabajar con restricciones de B_i^k en los soportes (t_{k+j}, t_{k+j+1}) , que van a ser esencialmente polinomios restringidos a esos intervalos. Como norma, utilizaremos esta notación únicamente cuando busquemos enfatizar la restricción.

Lema 9. *El conjunto $\{B_j^k, B_{j+1}^k, \dots, B_{j+k}^k\}$ es linealmente independiente en (t_{k+j}, t_{k+j+1}) para todo entero j .*

Demostración. Nuevamente demostramos el lema por inducción. La base de inducción, $k = 0$, es trivial por ser un conjunto de un solo elemento. Supongamos que es cierto para $k - 1 \geq 0$. Sea la combinación lineal

$$S = \sum_{i=0}^k c_{j+i} B_{j+i}^k$$

tal que $S_{|(t_{k+j}, t_{k+j+1})} = 0$. ¿Son los $c_{j+i} = 0$ para todo $i = 0, \dots, k$?

De la ecuación (1.41) se deduce que, si definimos $c_{j+i} = 0$ para $i > k$,

$$0 = S'_{|(t_{k+j}, t_{k+j+1})} = \sum_{i=1}^{\infty} \left(\frac{c_{j+i} - c_{j+i-1}}{t_{j+i+k} - t_{j+i}} \right) B_i^{k-1}|_{(t_{k+j}, t_{k+j+1})}.$$

Pero por hipótesis de inducción el conjunto $\{B_j^{k-1}, B_{j+1}^{k-1}, \dots, B_{j+k}^{k-1}\}$ es linealmente independiente, y en particular $\{B_{j+1}^{k-1}, \dots, B_{j+k}^{k-1}\}$ también lo será, y forzosamente

$$c_0 = c_1 = \dots = c_k =: \lambda.$$

Entonces $S = \lambda \sum_{i=0}^k B_{j+i}^k = \lambda = 0$. En este último paso hemos utilizado el lema 5. Notemos que tomando $x \in (t_{k+j}, t_{k+j+1})$, los únicos B-splines que no se anulan sobre ese intervalo son precisamente los que aparecen en el sumatorio. \square

Lema 10. *El conjunto $\{B_{-k}^k, B_{-k+1}^k, \dots, B_{n-1}^k\}$ es linealmente independiente en (t_0, t_n) para todo entero $k \geq 0$ y $n \geq 1$.*

Demostración. Sea la combinación lineal $S = \sum_{i=-k}^{n-1} c_i B_i^k$ y suponer $S_{|(t_0, t_n)} = 0$. Por el lema 2, solo $B_{-k}^k, B_{-k+1}^k, \dots, B_0^k$ son distintos de cero en (t_0, t_1) . Por consiguiente,

$$0 = S_{|(t_0, t_1)} = \sum_{i=-k}^0 c_i B_i^k|_{(t_0, t_1)}.$$

Por el lema 9 (tomando por $j = -k$), $\{B_{-k}^k, B_{-k+1}^k, \dots, B_0^k\}$ es linealmente independiente en (t_0, t_1) . Luego $c_i = 0$, $-k \leq i \leq 0$. Supongamos, para obtener una contradicción, que no todos los c_i , $i \geq 0$, son cero. Tomamos j como el primer coeficiente para el cual c_j no se anula, $j \geq 1$. Entonces $(t_j, t_{j+1}) \subset (t_0, t_n)$. Pero por el lema 3, para cualquier $x_j \in (t_j, t_{j+1})$, obtenemos la contradicción

$$0 = S(x) = \sum_{i=-k}^{n-1} c_i B_i^k(x) = c_j B_j^k(x) \neq 0.$$

\square

Teorema 5 (Teorema de representación por B-splines). *Una base para el espacio $\mathbb{S}^k([t_0, t_n])$ es el conjunto*

$$\mathcal{B} = \left\{ B_i^k|_{[t_0, t_n]} \mid -k \leq i \leq n-1 \right\},$$

esto es, si S es un spline de grado $k \geq 0$ sobre $[t_0, t_n]$, entonces existen unos $c_i \in \mathbb{R}$ únicos tales que $S = \sum_{i=-k}^{n-1} c_i B_i^k$.

Demostración. Es evidente que $\mathcal{B} \subset \mathbb{S}_n^k$ puesto que los elementos de \mathcal{B} ya hemos probado que son splines de grado k sobre la sucesión infinita de nodos $\Omega_\infty = \{t_i \mid t_i < t_{i+1}\}_{i \in \mathbb{Z}}$ y por tanto, en particular sobre $t_0 < \dots < t_n$. Por el lema 10 el conjunto \mathcal{B} es libre y por tanto su expansión se corresponde con \mathbb{S}_n^k . \square

Corolario 3. *El conjunto $\mathcal{B} = \{B_l^k|_{[t_i, t_j]} \mid i-k \leq l \leq j-i-1\}$ es base del espacio $\mathbb{S}^k([t_i, t_j])$.*

Capítulo 2

Curvas de Bézier y B-splines

Una vez introducidos los elementos básicos en el primer capítulo, ¿cómo podemos construir curvas en el espacio con ellos? La solución es definir estas curvas como sumas ponderadas (realmente combinaciones convexas) de puntos del espacio que les darán la forma, y donde los pesos serán los polinomios de Bernstein o los B-splines.

Las principales fuentes teóricas de este capítulo la podemos encontrar en los capítulos 3, 5 y 6 de [9] y los contenidos más prácticos en los maravillosos vídeos [15, 16].

§ 2.1. Curvas de Bézier: el algoritmo de De Casteljaou

Las curvas más sencillas de representar por un ordenador son las polinomiales. Además, como hemos mencionado en el capítulo 1, el resultado de Weierstrass dice que, si nos restringimos a un intervalo $[a, b]$, podemos aproximar cualquier función cuanto queramos por polinomios, y por tanto, cualquier curva paramétrica por curvas polinómicas.

Definición 7 (Curva polinomial). *Diremos que una curva paramétrica $\mathbf{P}: [a, b] \rightarrow \mathbb{R}^n$ es una curva polinomial de grado $\leq k$ si sus componentes son polinomios de grado $\leq k$ sobre $[a, b]$, esto es, si*

$$\mathbf{P}(t) = (x(t), y(t)), \quad n = 2, \quad \text{ó} \quad \mathbf{P}(t) = (x(t), y(t), z(t)), \quad n = 3,$$

para $x, y, z \in \mathbb{P}^k([a, b])$, espacio definido en (1.5).

Podemos dar una definición equivalente en términos de los coeficientes de los polinomios componente, que constituyen puntos de \mathbb{R}^n . Sean $\mathbf{b}_0, \dots, \mathbf{b}_k \in \mathbb{R}^n$. Entonces una curva paramétrica $\mathbf{P}: [a, b] \rightarrow \mathbb{R}^n$ se dice curva polinomial de grado $\leq k$ si

$$\mathbf{P}(t) = \sum_{i=0}^k \mathbf{b}_i t^i, \quad t \in [a, b].$$

Recordemos que los polinomios de Bernstein de grado k constituyen una base para $\mathbb{P}^k([a, b])$. Por tanto, toda polinomial $\mathbf{B}(t)$ tiene una única representación en términos de la base de Bernstein, esto es existen $\mathbf{b}_0^k, \dots, \mathbf{b}_k^k \in \mathbb{R}^n$ únicos tales que:

$$\mathbf{B}(t) = \sum_{i=0}^k \mathbf{b}_i^k \mathcal{B}_i^k(t), \quad t \in [a, b]. \quad (2.1)$$

La representación (2.1) se denomina *representación de Bézier*, y a una curva polinomial en dicha representación, *curva de Bézier*. Los puntos \mathbf{b}_i^k se denominan *puntos de Bézier* y la poligonal que forman, *polígono de Bézier* asociado a \mathbf{B} .

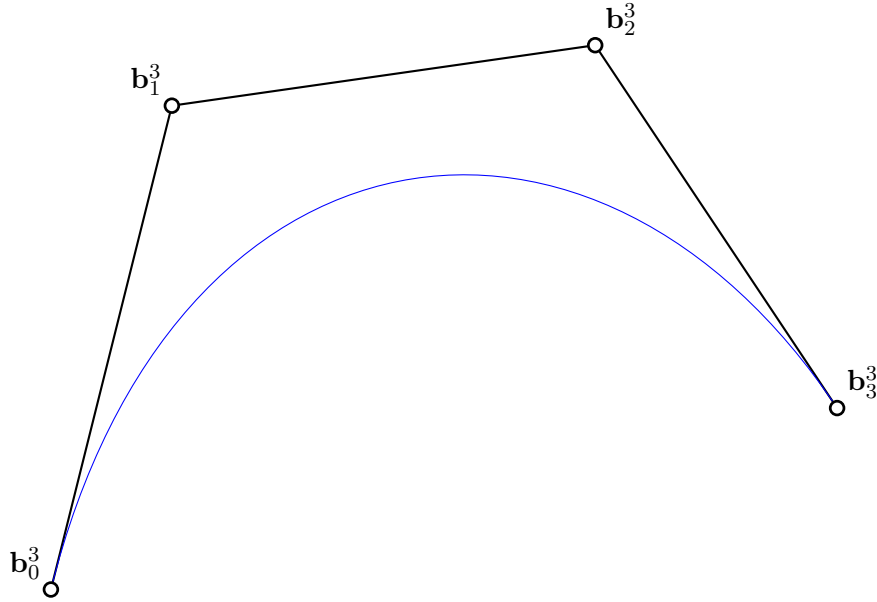


Figura 2.1: Una curva de Bézier cúbica (azul), junto a su polígono de Bézier (negro).

El cálculo recursivo de los polinomios de Bernstein, como hemos visto en el cuadro 1.1, induce un sencillo algoritmo para el cálculo de curvas polinómicas a través de su representación de Bézier. Por lo general se trabaja sobre el intervalo $[0, 1]$. Se conoce como algoritmo de De Casteljau, en honor a su inventor.

Algoritmo de De Casteljau

Sea $t \in [0, 1]$. Aplicando la definición recursiva de los polinomios de Bernstein, podemos reescribir (2.1) como

$$\mathbf{B}(t) = \sum_{i=0}^k \mathbf{b}_i^k t \mathcal{B}_{i-1}^{k-1}(t) + \sum_{i=0}^{k-1} \mathbf{b}_i^k (1-t) \mathcal{B}_i^{k-1}(t) = \sum_{i=0}^{k-1} \left((1-t) \mathbf{b}_i^k + t \mathbf{b}_{i+1}^k \right) \mathcal{B}_i^{k-1}(t).$$

Si ahora definimos

$$\begin{aligned} \mathbf{b}_i^{k-1}(t) &= (1-t) \mathbf{b}_i^k + t \mathbf{b}_{i+1}^k, \\ \mathbf{b}_i^{j-1}(t) &= (1-t) \mathbf{b}_i^j + t \mathbf{b}_{i+1}^j(t), \quad 1 \leq j \leq k-1, \end{aligned}$$

tenemos

$$\mathbf{B}(t) = \sum_{i=0}^k \mathbf{b}_i^k \mathcal{B}_i^k(t) = \sum_{i=0}^{k-1} \mathbf{b}_i^{k-1}(t) \mathcal{B}_i^{k-1}(t) = \dots = \mathbf{b}_0^0(t) \mathcal{B}_0^0(t) = \mathbf{b}_0^0(t).$$

¡Podemos evaluar la curva de Bézier en cualquier punto efectuando combinaciones convexas en lugar de calcular una expresión explícita de la curva! \triangle

$$\begin{array}{ccccccc}
 \mathbf{b}_0^n & \rightarrow & \mathbf{b}_0^{n-1} & \cdots & \mathbf{b}_0^1 & \rightarrow & \mathbf{b}_0^0 \\
 \mathbf{b}_1^n & \nearrow & \mathbf{b}_1^{n-1} & \cdots & \mathbf{b}_1^1 & \nearrow & \\
 \vdots & & \vdots & \ddots & & & \\
 \mathbf{b}_{n-1}^n & \rightarrow & \mathbf{b}_{n-1}^{n-1} & & & & \\
 \mathbf{b}_n^n & \nearrow & & & & &
 \end{array}
 \quad
 \begin{array}{ccc}
 (\bullet) & \xrightarrow{1-t} & (\bullet) \\
 (\bullet) & \nearrow^t &
 \end{array}$$

Cuadro 2.1: Arreglo triangular asociado al algoritmo de De Casteljaou.

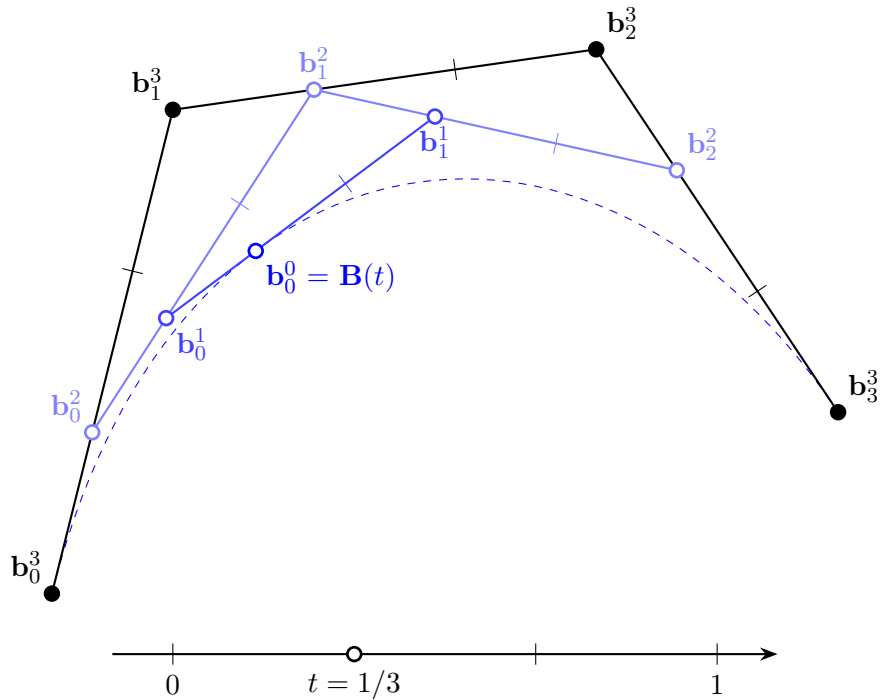


Figura 2.2: Algoritmo de De Casteljaou para la construcción de la figura 2.1 en $t = 1/3$.

Debido a su sencillez, las curvas de Bézier son el método más usado en la práctica para representar curvas en los ordenadores. Sin embargo, la representación de Bézier tiene unos cuantos inconvenientes. El primero es que no posee control local, una propiedad que sí tienen los B-splines. Esto significa que si desplazamos uno de los puntos de Bézier, toda la curva se ve afectada, como podemos ver en la figura 2.3. El segundo inconveniente es que para producir curvas algo más

complejas, necesitamos añadir puntos, y el grado de una curva de Bézier viene determinado por su número de puntos de Bézier. A medida que aumentamos el grado de la curva, el algoritmo de De Casteljau se vuelve cada vez menos eficiente. Además, como la curva solo interpola el punto inicial y final, se vuelve más complicado obtener una forma en concreto.

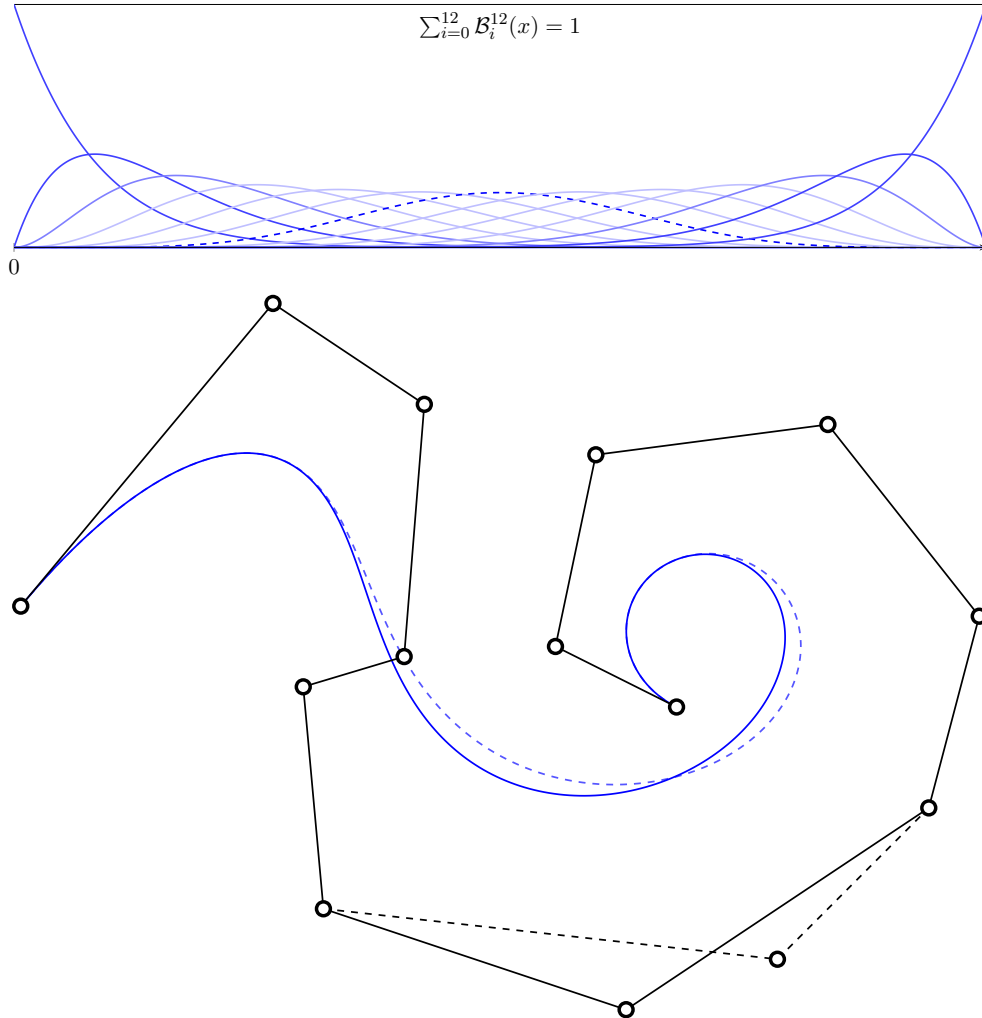


Figura 2.3: Una curva de Bézier de grado 12 junto a la partición de la unidad de los polinomios de Bernstein de grado 12. El séptimo punto de Bézier ha sido desplazado y podemos ver cómo su influencia sobre la curva es global en lugar de local, debido a su polinomio de Bernstein asociado.

§ 2.2. Splines de Bézier

Podemos obtener control local de forma artificial si construimos splines de Bézier, esto es, concatenaciones continuas de curvas de Bézier. En la práctica, este es el método que se utiliza para solventar el problema de las curvas de Bézier de grado alto.

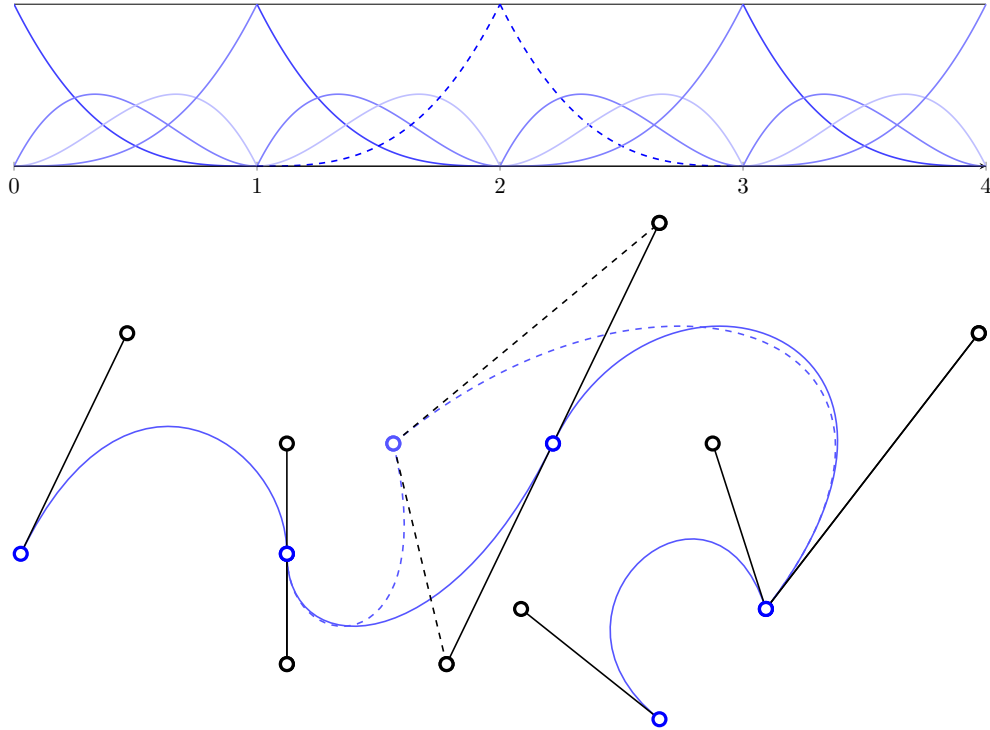


Figura 2.4: Un spline de Bézier de grado 3. Notemos que el punto de Bézier desplazado afecta a las dos curvas de las que forma parte, pero no al resto de la curva. ¡Hemos logrado control local! Podemos observar que tenemos libertad para crear curvas no regulares, con picos como el de esta curva.

El spline de Bézier más utilizado es el cúbico, gracias a su excelente comportamiento y la rapidez de su cálculo. En la figura 2.4 podemos observar un spline de Bézier cúbico, donde hemos desplazado uno de sus nodos. Notemos que ahora interpolamos un punto de Bézier cada tres puntos, y es mucho más fácil conseguir las figuras deseadas. Si observamos bien, en algunas juntas hemos alineado sus puntos anteriores y posteriores, de forma que equidisten de la junta (hemos alineado su segmento con el vector tangente en la junta). Se puede demostrar que así se obtiene continuidad C^1 en la junta. No debemos de confundir un spline de Bézier cúbico, donde aquí cúbico significa que cada trozo que lo conforma es una curva de Bézier cúbica, con un spline cúbico. Un spline cúbico logra continuidad C^2 en cada una de sus juntas, y un spline de Bézier cúbico solo tendrá la continuidad que nosotros le imponemos.

Sin embargo, imponer restricciones de continuidad tiene un alto precio. Cuantas más restricciones imponemos, más limitados estamos a la hora de crear figuras. Este fenómeno se conoce como *pérdida en cascada del control local*. En el caso de la figura 2.4, la continuidad C^1 obliga a que uno de los dos puntos que rodean a una junta esté determinado por la junta y el punto opuesto.

Podemos ilustrar esta última observación en el capítulo 2 de [15], con unas magníficas animaciones que lo ilustran a la perfección.

§ 2.3. B-splines: el algoritmo de De Boor

Los B-splines son una alternativa fundamental a las curvas de Bézier, porque no hace falta sacrificar libertad de diseño por continuidad. Además, cuentan con la propiedad de control local como consecuencia directa de su definición. Sin embargo, tal y como hemos definido hasta ahora los B-splines, en dimensión 1, no parecen muy útiles para diseñar ninguna forma. Igual que con los polinomios de Bernstein, para sacarles su máximo potencial vamos a recurrir a combinaciones convexas de puntos.

La bibliografía fundamental utilizada en esta sección es [9, 13], aunque hay parte de investigación propia. Podemos encontrar las pruebas de las propiedades análogas para curvas de Bézier en [3].

Definición 8 (Curva spline, curva B-spline). *Sea la partición $a = t_0 < \dots < t_n = b$ del intervalo $[a, b]$. Diremos que una curva paramétrica $\mathbf{S}: [a, b] \rightarrow \mathbb{R}^n$ es una curva spline (spline) de grado k si las componentes de \mathbf{S} son splines de grado k sobre $[a, b]$, esto es, si*

$$\mathbf{S}(u) = (x(u), y(u)), \quad n = 2, \quad \text{ó} \quad \mathbf{S}(u) = (x(u), y(u), z(u)), \quad n = 3,$$

para $x, y, z \in \mathbb{S}^k([a, b])$.

Diremos que una curva spline $\mathbf{S}: [a, b] \rightarrow \mathbb{R}^n$ de grado k es una curva B-spline (B-spline) si sus componentes vienen representadas en la base $\{B_{-k}^k, \dots, B_{n-1}^k\}$, esto es

$$\mathbf{S}(u) = \sum_{i=-k}^{n-1} \mathbf{C}_i^k B_i^k(u),$$

o lo que es lo mismo, si \mathbf{S} es una combinación convexa de los puntos \mathbf{C}_i^k a partir de los pesos B_i^k . Los $\mathbf{C}_{-k}^k, \dots, \mathbf{C}_{n-1}^k$ se denominan puntos de control de \mathbf{S} y determinan de forma unívoca a la curva B-spline. Diremos que la poligonal formada por sus puntos de control constituye su polígono de control.

La mayoría de los textos se refieren a las curvas B-spline como B-splines, y solo cuando se puede dar lugar a confusión, hablan de los B-splines básicos para referenciar la base vista en el capítulo 1. Análogamente, se suele entender que un spline es una curva spline. Sin embargo, según el contexto esto puede llevar a confusión porque existen splines multidimensionales, tal y como podemos encontrar en [9], y no tienen nada que ver con los objetos que estamos definiendo. Una vez advertido este detalle, vamos a seguir por comodidad la denotación clásica, y también nos referiremos por B-splines a este tipo de curvas.

Nota 5. La importancia de los puntos de control es tal, que si conocemos el espacio paramétrico que va a recorrer el B-spline y su grado, basta con dar un vector de puntos de control

$$(\mathbf{C}_{-k}^k, \dots, \mathbf{C}_{n-1}^k)$$

para definirla ¹.

¹De hecho, *Wolfram Mathematica* define los B-spline a partir de un vector de puntos de control. De forma parecida, define las curvas de Bézier a partir de un vector de puntos de Bézier.

El mejor comportamiento de un B-spline se obtiene si repetimos ciertos puntos de control. Si tomamos el primer y último punto de control $k + 1$ veces cada uno se puede lograr que la curva comience y acabe en dichos puntos. Es decir, si dada $\mathbf{S}(u)$ denotamos por $\mathbf{C}_0, \dots, \mathbf{C}_{n-k}$ a sus puntos de control diferentes, esto es, tomando el vector

$$(\mathbf{C}_0, \dots, \overset{k+1}{\mathbf{C}_0}, \mathbf{C}_1, \dots, \mathbf{C}_{n-k-1}, \overset{k+1}{\mathbf{C}_{n-k}}, \mathbf{C}_{n-k}),$$

entonces $\mathbf{S}(a) = \mathbf{C}_0$ y $\mathbf{S}(b) = \mathbf{C}_{n-k}$.

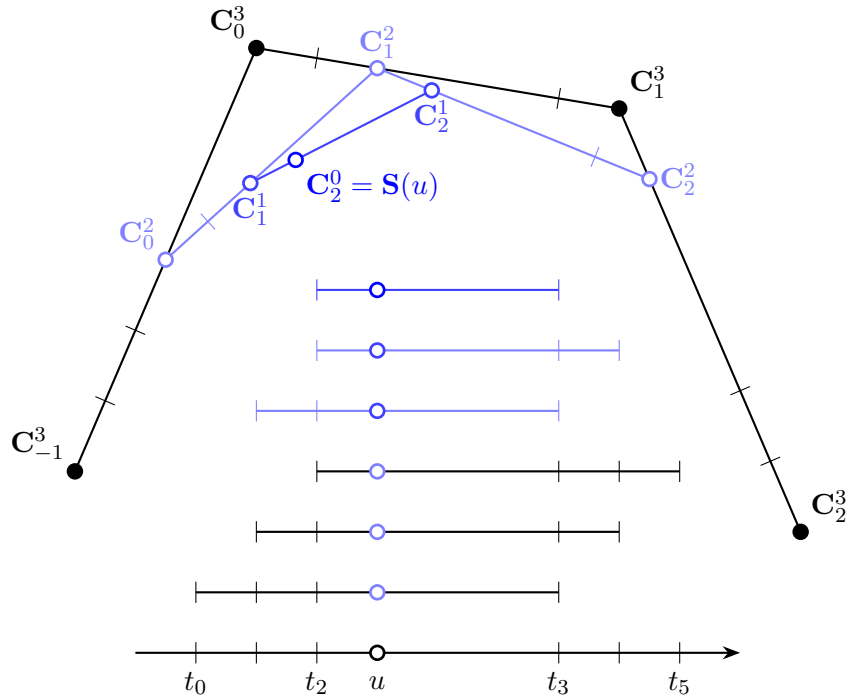


Figura 2.5: Algoritmo de De Boor para el B-spline cúbico $\mathbf{S}(u) = \sum_{i=-1}^2 \mathbf{C}_i^3 B_i^3(u)$ en $u \in [t_0, t_5]$. Sobre el espacio paramétrico, encontramos representadas las preimágenes de los segmentos que conforman las sucesivas iteraciones del algoritmo de De Boor. Podemos observar la influencia de cada nodo en los sucesivos cálculos.

Esta observación surgió a propósito de darme cuenta que en la mayoría de las referencias, el B-spline siempre interpolaba el primer y último punto de control. Pero, si por lo general los puntos de control no son interpolados, de alguna forma se tenía que poder forzar a la curva su paso por ellos. Para ello, utilicé el algoritmo de De Boor. En efecto, definamos $\alpha_u := (u - t_i)/(t_{i+k} - t_i)$ de forma que

$$\begin{aligned} \mathbf{C}_i^{k-1}(u) &= \alpha_u \mathbf{C}_i^k + (1 - \alpha_u) \mathbf{C}_{i-1}^k \\ \mathbf{C}_i^{j-1}(u) &= \alpha_u \mathbf{C}_i^j(u) + (1 - \alpha_u) \mathbf{C}_{i-1}^j(u), \quad 1 \leq j \leq k - 1. \end{aligned}$$

Tomando $u = t_0$, se tiene que $u \in [t_0, t_1)$. El arreglo triangular nos quedará

$$\begin{array}{rccccccc}
\mathbf{C}_0 = \mathbf{C}_0^k & \rightarrow & \mathbf{C}_0(\alpha_u + 1 - \alpha_u) = \mathbf{C}_0 & \cdots & \mathbf{C}_0 & \rightarrow & \mathbf{C}_0 \\
\mathbf{C}_0 = \mathbf{C}_{-1}^k & \nearrow & \mathbf{C}_0(\alpha_u + 1 - \alpha_u) = \mathbf{C}_0 & \cdots & \mathbf{C}_0 & \nearrow & \\
& \vdots & & & & & \ddots \\
\mathbf{C}_0 = \mathbf{C}_{-k+1}^k & \rightarrow & \mathbf{C}_0(\alpha_u + 1 - \alpha_u) = \mathbf{C}_0 & & & & \\
\mathbf{C}_0 = \mathbf{C}_{-k}^k & \nearrow & & & & &
\end{array}$$

Cuadro 2.2: Arreglo triangular para $m = 1$.

De forma análoga, el arreglo triangular para $u = t_n \in [t_n, t_{n+1})$ será

$$\begin{array}{rccccccc}
\mathbf{C}_{n-k} = \mathbf{C}_n^k & \rightarrow & \mathbf{C}_{n-k} & \cdots & \mathbf{C}_{n-k} & \rightarrow & \mathbf{C}_{n-k} \\
\mathbf{C}_{n-k} = \mathbf{C}_{n-1}^k & \nearrow & \mathbf{C}_{n-k} & \cdots & \mathbf{C}_{n-k} & \nearrow & \\
& \vdots & & & & & \ddots \\
\mathbf{C}_{n-k} = \mathbf{C}_{n-k+1}^k & \rightarrow & \mathbf{C}_{n-k} & & & & \\
\mathbf{C}_{n-k} = \mathbf{C}_{n-k}^k & \nearrow & & & & &
\end{array}$$

Cuadro 2.3: Arreglo triangular para $m = n$.

Notemos que para poder fijar \mathbf{C}_{n-k} tenemos que tomar un vector de puntos de control con una componente más, esto es, con $n + k + 1$ puntos de control. Esto se debe a la definición de los B-splines básicos en términos de intervalos cerrados por la izquierda y abiertos por la derecha. Realmente la representación de la curva $\mathbf{S}(u)$ en términos de la base $\{B_{-k}^k, \dots, B_{n-1}^k\}$ sigue siendo correcta, puesto que $B_i^n(u) = 0$ para todo $u \in [t_0, t_n]$. Sin embargo, para el cálculo de $S(t_n)$ a través del algoritmo de De Boor necesitamos conocer este punto de control, porque es un cálculo recursivo que necesita $k + 1$ nodos en la primera iteración. Notemos que para fijar otros puntos podríamos hacer un estudio similar. \triangle

Proposición 2 (Invarianza afín). *Los B-splines son invariantes bajo transformaciones afines, esto es, para aplicar una transformación afín a B-spline basta hacerlo sobre sus puntos de control.*

Demostración. Sea $n = 2$ o 3 . Sea $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^n$ una transformación afín, dada por $\phi(\mathbf{X}) = A\mathbf{X} + \mathbf{b}$ para una matriz $A \in \mathcal{M}_n(\mathbb{R})$ y un vector $\mathbf{b} \in \mathbb{R}^n$, para un $\mathbf{X} \in \mathbb{R}^n$. Sea el B-spline $\mathbf{S}(u) = \sum_{i=-k}^{n-1} \mathbf{C}_i^k B_i^k(u)$. Entonces

$$\begin{aligned}
\phi(\mathbf{S}(u)) &= \phi\left(\sum_{i=-k}^{n-1} \mathbf{C}_i^k B_i^k(u)\right) = A\left(\sum_{i=-k}^{n-1} \mathbf{C}_i^k B_i^k(u)\right) + \mathbf{b} = A\left(\sum_{i=-k}^{n-1} \mathbf{C}_i^k B_i^k(u)\right) + \mathbf{b} \sum_{i=-k}^{n-1} B_i^k(u) \\
&= \sum_{i=-k}^{n-1} B_i^k(u)(A\mathbf{C}_i^k + \mathbf{b}) = \sum_{i=-k}^{n-1} \phi(\mathbf{C}_i^k) B_i^k(u).
\end{aligned}$$

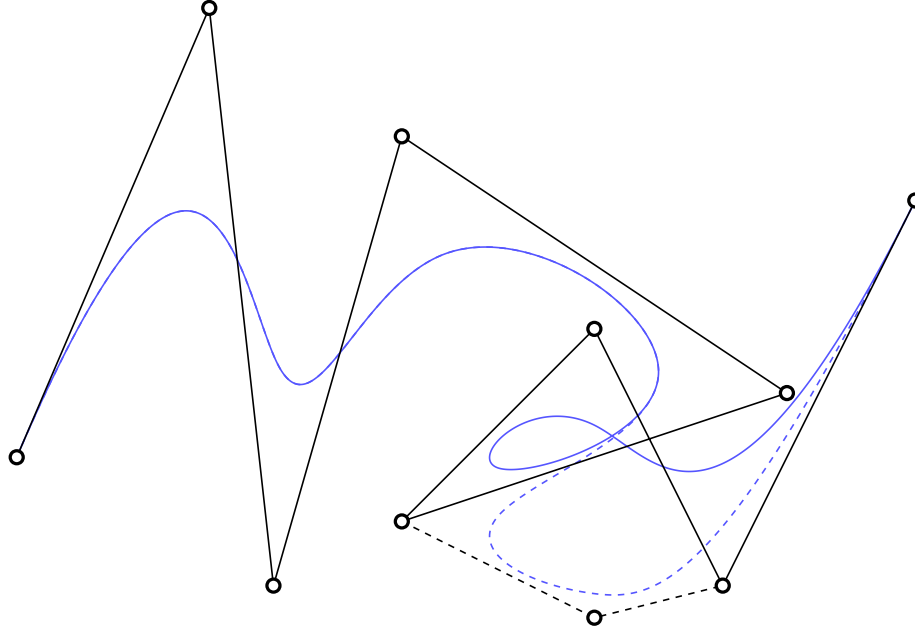


Figura 2.6: Un B-spline, junto con su polígono de control. Notemos que los puntos de control *atraen* la curva hacia ellos, cómo contamos con control local y cómo la curva solo interpola el punto inicial y final.

En el antepenúltimo paso hemos podido introducir el sumatorio que acompaña a \mathbf{b} utilizando la propiedad de partición de la unidad. \square

Proposición 3 (Propiedad fuerte de la clausura convexa). *Sea un B-spline de grado k sobre el intervalo $[a, b]$, con la partición $a = t_0 < \dots < t_n = b$. Entonces para $u \in [t_i, t_{i+1})$, el B-spline está contenido en la clausura convexa de los puntos de control $\mathbf{C}_{i-k}^k, \dots, \mathbf{C}_i^k$ (propiedad débil de la clausura convexa). Es más, la totalidad del B-spline está contenida en la clausura convexa formada por todos sus puntos de control.*

Demostración. Sea el B-spline $\mathbf{S}(u) = \sum_{j=-k}^{n-1} \mathbf{C}_j^k B_j^k(u)$. Sea $u \in [t_i, t_{i+1})$, entonces podemos reescribir la curva tomando la base sobre el intervalo nodal, prescindiendo de los B-splines que se anulan:

$$\mathbf{S}(u) = \sum_{j=i-k}^i \mathbf{C}_j^k B_j^k(u).$$

Tenemos que $\sum_{j=i-k}^i B_j^k(u) = 1$, luego $\mathbf{S}(u)$ es combinación convexa de los puntos de control, luego por definición está contenida en la clausura convexa de $\mathbf{C}_{i-k}^k, \dots, \mathbf{C}_i^k$. Tomando en general $u \in [a, b] = [t_0, t_n]$, por la propiedad de partición de la unidad, el resultado sigue siendo cierto y $\mathbf{S}(u)$ está contenida en la clausura convexa de todos sus puntos de control. \square

Proposición 4 (Control local). *Sea un B-spline de grado k sobre el intervalo $[a, b]$, con la partición $a = t_0 < \dots < t_n = b$ y dado por el vector de puntos de control $(\mathbf{C}_{-k}^k, \dots, \mathbf{C}_{n-1}^k)$. La traslación*

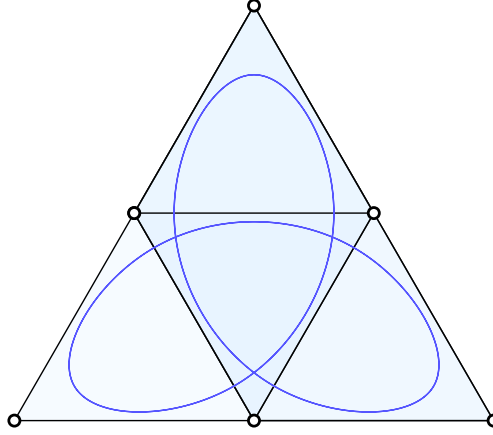


Figura 2.7: Curva B-spline cerrada generada por un polígono de control con forma de *trifuerza*. Observar que la curva yace contenida en la clausura convexa de sus puntos de control.

de un punto de control \mathbf{C}_i^k solo afecta a la curva en el intervalo $[t_i, t_{i+k+1})$. Como consecuencia, diremos que tenemos control local sobre la curva.

Demostración. Una curva B-spline es la suma ponderada de sus puntos de control, donde el peso asociado a \mathbf{C}_i^k es el B-spline B_i^k , que se anula para $u \notin [t_i, t_{i+k+1})$. Luego la influencia del punto de control \mathbf{C}_i^k se reduce al intervalo $[t_i, t_{i+k+1})$. \square

§ 2.4. NURBS

A pesar de la mejoría con respecto a las curvas de Bézier, los B-splines aún cuentan con ciertas limitaciones, relacionadas con la representación de ciertas figuras. Por ejemplo, con un B-spline no podemos representar una curva tan básica como la circunferencia. Las NURBS, abreviatura de *Non-Uniform Rational B-splines*, son las curvas que se utilizan en la práctica. Ofrecen la misma flexibilidad y aún mayor precisión que los B-splines. Una NURB definida en el intervalo $[a, b] = [t_0, t_n]$ tiene la forma

$$\mathbf{N}(u) = \frac{\sum_{j=-k}^{n-1} \mathbf{C}_j^k \omega_j B_j^k(u)}{\sum_{j=-k}^{n-1} \omega_j B_j^k(u)}, \quad u \in [a, b], \quad (2.2)$$

donde ω_j son unos pesos a elegir. El denominador es un factor de normalización, que vale uno si todos los pesos valen uno.

Las NURBS heredan todas las propiedades de los B-splines, salvo la de partición de la unidad, que solo la cumplen si todos los ω_j valen 1. Podemos encontrar mucha información práctica de las NURBS en [22], al igual que vídeos que ilustran su utilidad en el modelado gráfico como [17].

Capítulo 3

Superficies producto tensorial

Una de las formas más intuitivas de construir una superficie consiste en barrer una curva a lo largo del espacio, como con los cuerpos de revolución, que son engendrados girando una curva en torno a un eje. Fijados dos ejes y siguiendo una idea similar, pretendemos construir superficies tomando una curva sobre un eje y desplazando sus puntos de control siguiendo otra curva construida en el eje contrario. Este procedimiento se conoce como producto tensorial de dos curvas.

Supongamos que $\varphi_1, \dots, \varphi_r$ y ψ_1, \dots, ψ_s son aplicaciones lineales linealmente independientes. Consideremos los espacios vectoriales que generan, $X_r := \text{span}\langle \varphi_1, \dots, \varphi_r \rangle$ e $Y_s := \text{span}\langle \psi_1, \dots, \psi_s \rangle$. Definimos el producto tensorial de X_r e Y_s como el conjunto

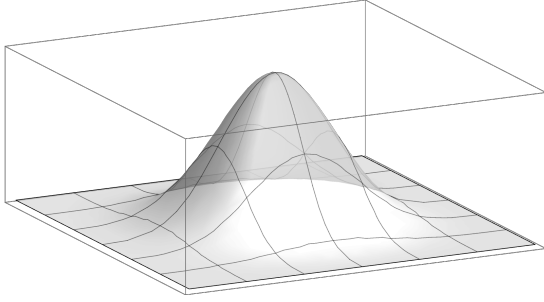
$$X_r \otimes Y_m := \text{span} \{ \varphi_i \psi_j \mid (\varphi_i \psi_j)(x, y) := \varphi_i(x) \psi_j(y), i = 1, \dots, r, j = 1, \dots, m \}. \quad (3.1)$$

Se puede probar que $X_r \otimes Y_m$ es espacio vectorial y que los elementos $\varphi_i \psi_j$ son linealmente independientes, con lo que es un espacio de dimensión $r \times m$.

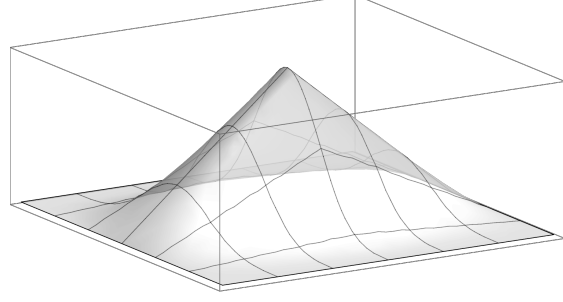
A diferencia del capítulo anterior, primero vamos a definir las superficies B-spline, en lugar de comenzar con superficies de Bézier. En la práctica, se utilizan unas u otras en función de los mismos inconvenientes que surgían con las curvas.

§ 3.1. Superficies B-spline

Esta sección está basada en las definiciones de [7, 9] y en los resultados de [13], aunque hemos hecho un esfuerzo propio en facilitar la notación. Debemos mencionar que parece que no existe una definición consensuada para la palabra *parche*, por lo menos en las referencias consultadas. Este término solo existe en el contexto de superficies de Bézier, pero por comodidad, lo hemos extendido a una definición para B-splines, buscando poder referirnos a los elementos básicos del producto tensorial (3.1) de alguna forma mediante un nombre propio.



(a) Parche básico de grado (3,3).



(b) Parche básico de grado (3,1).

Figura 3.1: Ejemplos de parches B-spline, representados como $(u, v, B_i^k B_j^l(u, v))$.

Definición 9 (Parche básico B-spline). Sean los intervalos $I = [a, b]$ y $J = [c, d]$, junto a las particiones $a = u_0 < u_1 < \dots < u_n = b$ y $c = v_0 < v_1 < \dots < v_m = d$. Consideremos los espacios de B-splines de grado k sobre I y de grado l sobre J , $\mathbb{S}^k(I) = \text{span}\langle B_{-k}^k, \dots, B_{n-1}^k \rangle$ y $\mathbb{S}^l(J) = \text{span}\langle B_{-l}^l, \dots, B_{m-1}^l \rangle$, respectivamente. Definimos el espacio de las superficies spline (B-spline según el contexto) como el producto tensorial

$$\mathbb{S}^k(I) \otimes \mathbb{S}^l(J) = \text{span} \left\{ B_i^k B_j^l \mid i = -k, \dots, n-1, j = -l, \dots, m-1 \right\}. \quad (3.2)$$

En este texto, denominaremos a los productos $B_i^k B_j^l$ como parches básicos B-spline de grado (k, l) .

Nota 6. Los elementos básicos $B_i^k B_j^l$ que hemos denominado parches básicos, ni el espacio producto tensorial que conforman, $\mathbb{S}^k \otimes \mathbb{S}^l$, cuentan con un nombre propio en las referencias que hemos consultado. Podemos de esta forma interpretar las superficies como combinaciones convexas de puntos del espacio, esta vez ponderadas por los parches B-spline. El nombre sin embargo, está inspirado en los parches de Bézier, que sí son un objeto definido en [9].

Lema 11 (Propiedades de los parches básicos B-spline).

- (1) Soporte local. Se tiene $B_i^k B_j^l(u, v) = 0$ si y solo si $(u, v) \notin (u_i, u_{i+k+1}) \times (v_j, v_{j+l+1})$. Si $k = 0$ o $l = 0$, el intervalo respectivo se toma cerrado por la izquierda.
- (2) Positividad. El parche $B_i^k B_j^l(u, v) > 0$ si y solo si $(u, v) \in (u_i, u_{i+k+1}) \times (v_j, v_{j+l+1})$. Si $k = 0$ o $l = 0$, el intervalo respectivo se toma cerrado por la izquierda.
- (3) Partición de la unidad. Los parches B-spline constituyen una partición de la unidad¹ de \mathbb{R}^2 , y en particular, de los rectángulos $[u_p, u_{p+1}) \times [v_q, v_{q+1})$, esto es

$$\sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} B_i^k B_j^l(u, v) = \sum_{i=p-k}^p \sum_{j=q-l}^q B_i^k B_j^l(u, v) = 1, \quad (u, v) \in [u_p, u_{p+1}) \times [v_q, v_{q+1})$$

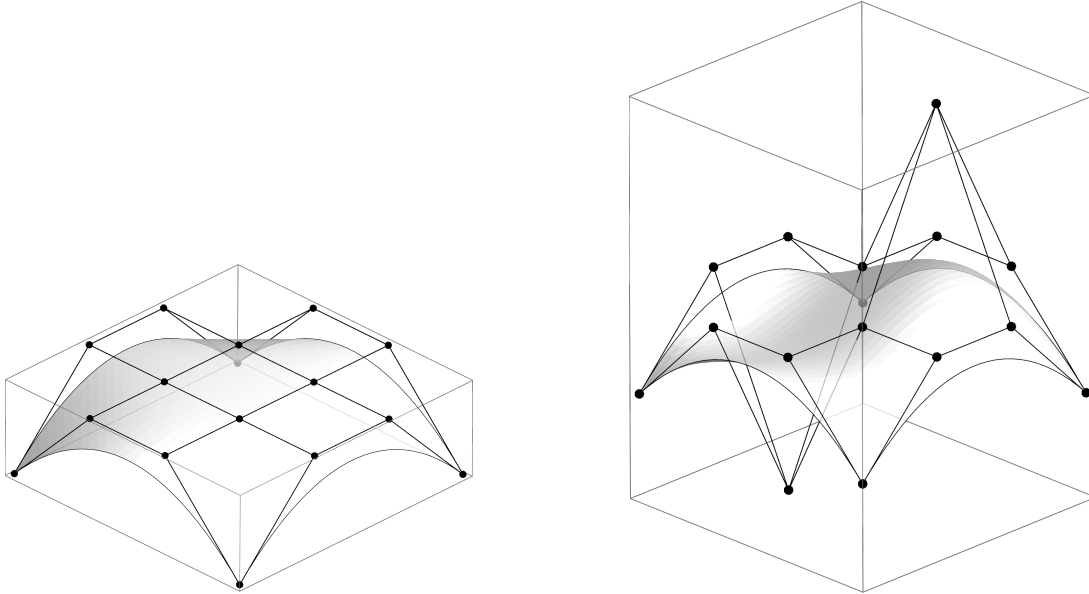
Demostración. Las propiedades (1), (2) y (3) son consecuencia directa de 2, 3 y de 5 respectivamente. \square

¹Observemos que no estamos atendiendo en esta definición a la continuidad del parche, como podemos encontrar en [13].

Definición 10 (Superficie B-spline). Una superficie $\mathbf{S}(u, v) \in \mathbb{S}^k(I) \otimes \mathbb{S}^l(J)$ diremos que es una superficie B-spline de grado (k, l) si viene representada en términos de la base de parches B-spline de grado (k, l) , es decir, si

$$\mathbf{S}(u, v) = \sum_{i=-k}^{n-1} \sum_{j=-l}^{m-1} \mathbf{C}_{i,j}^{k,l} B_i^k B_j^l(u, v),$$

para unos puntos de control $\mathbf{C}_{i,j}^{k,l}$. La matriz de puntos de control, $(\mathbf{C}_{i,j}^{k,l})$, determina de forma unívoca a la superficie \mathbf{S} . La malla formada por la unión de los puntos de control siguiendo los dos ejes u y v se denomina malla de control de \mathbf{S} .



(a) Superficie B-spline original.

(b) Superficie alterada.

Figura 3.2: Una superficie B-spline junto con la misma superficie, a la que han sido desplazados dos puntos de control. Observemos por un lado, que contamos con control local, y por el otro, que las superficies interpolan los cuatro puntos de control localizados sobre los vértices del rectángulo $I \times J$, por razones de la implementación en *Wolfram Mathematica*.

Notemos que si disponemos los puntos de control formando una malla regular, fijado un v o un u , $\mathbf{S}(\cdot, v)$ y $\mathbf{S}(u, \cdot)$ son curvas B-spline de grado k y l respectivamente. Esta es la razón por la cual en la introducción al capítulo hablábamos de «barrer curvas» para formar superficies. De esta forma, podemos construir la superficie B-spline utilizando el algoritmo de De Boor recorriendo primero un eje y después el otro.

Si buscamos por algún motivo fijar el comportamiento de la superficie B-spline en su frontera, de forma que ésta cruce forzosamente por ejemplo, por los puntos de control $\mathbf{C}_{0,j}^{k,l}$, $\mathbf{C}_{n-k,j}^{k,l}$, para el eje U y $\mathbf{C}_{i,0}^{k,l}$, $\mathbf{C}_{i,m-l}^{k,l}$ para el eje V , bastaría tomarlos $(k+1)$ y $(l+1)$ veces respectivamente.

La estrecha relación de las superficies con las curvas tiene como consecuencia directa la herencia de sus propiedades, como vamos a enunciar en la siguiente y última proposición.

Proposición 5 (Propiedades de las superficies B-spline). *Sea una superficie B-spline de grado (k, l)*

$$\mathbf{S}(u, v) = \sum_{i=-k}^{n-1} \sum_{j=-l}^{m-1} \mathbf{C}_{i,j}^{k,l} B_i^k B_j^l(u, v), \quad (u, v) \in I \times J.$$

- (1) Invarianza afín. *La superficie \mathbf{S} es invariante bajo transformaciones afines, esto es, para aplicar una transformación afín a una superficie B-spline basta hacerlo sobre sus puntos de control.*
- (2) Propiedad fuerte de la clausura convexa. *Dado $u \in [u_p, u_{p+1}) \times [v_q, v_{q+1})$, la superficie \mathbf{S} está contenida en la clausura convexa de los puntos de control $\mathbf{C}_{p-k, q-l, q}^k, \dots, \mathbf{C}_{p, q}^k$. Es más, la totalidad de \mathbf{S} está contenida en la clausura convexa formada por todos sus puntos de control.*
- (3) Control local. *La traslación de un punto de control $\mathbf{C}_{p,q}^{k,l}$ solo afecta a la superficie en el rectángulo que lo contiene, esto es, en $[u_p, u_{p+1}) \times [v_q, v_{q+1})$. Como consecuencia, diremos que tenemos control local sobre la superficie.*

Demostración. Las propiedades (1), (2) y (3) son consecuencia directa de 2, 3 y de 4 respectivamente. □

§ 3.2. Parches de Bézier y NURBS

De forma completamente análoga a la definición de las superficies B-spline, podríamos construir las superficies de Bézier haciendo el producto tensorial $\mathbb{P}^k \otimes \mathbb{P}^l$ para los grados k y l deseados sobre el cuadrado unidad, $[0, 1]^2$. Estas construcciones se conocen como *parches de Bézier*. Al igual que con las curvas, en la práctica se construyen superficies spline de Bézier formando redes de parches, para obtener un control local y una mayor optimización de los cálculos. Las superficies de Bézier más utilizadas son las constituidas por parches bicúbicos ($k = l = 3$). La geometría de un parche bicúbico está completamente definida por tan solo 16 puntos de Bézier.

El caso de las NURBS no es diferente. Una superficie NURB se obtiene a partir del producto tensorial de dos curvas NURB en variables u y v independientes. Tienen por tanto la forma

$$\mathbf{S}(u, v) = \sum_{i=-k}^{n-1} \sum_{j=-l}^{m-1} \mathbf{C}_{i,j}^{k,l} R_{i,j}^{k,l}(u, v),$$

con las funciones

$$R_{i,j}^{k,l}(u, v) = \frac{\omega_{i,j} B_i^k(u) B_j^l(v)}{\sum_{p=-k}^{n-1} \sum_{q=-l}^{m-1} \omega_{p,q} B_p^k(u) B_q^l(v)}$$

como funciones racionales básicas y donde los $\omega_{i,j}$, $\omega_{p,q}$ son los diferentes pesos.

Capítulo 4

Introducción a la teoría de blossomes

Hasta el momento, la mayoría de los resultados vistos han sido probados por técnicas tradicionales como la inducción o la prueba directa, y los objetos matemáticos han ido surgiendo mediante pruebas constructivas. Tras el descubrimiento de los B-splines, entre los matemáticos de la época se hizo un esfuerzo en simplificar la notación y la prueba de los resultados.

Las explicaciones estándares de la teoría de curvas y superficies de Bézier y B-splines no son lo simples que deberían ser. Tienen preciosas figuras [...] que ilustran la geometría asociada a los algoritmos. ¡Pero las etiquetas de las figuras son un desastre! [...] No hay forma fácil [...] de deducir las relaciones geométricas a partir de ellas (L. Ramshaw, 1987).

Como punto de partida a dicho problema, supongamos que tenemos, por ejemplo, el polinomio $F(t) = 4t^3 - 3t^2 + 9t + 1$. ¿Existe alguna función $f = f(u, v, w)$ tal que $f(t, t, t) = F(t)$? Consideremos que f tiene la forma

$$f(u, v, w) = r_1uvw + r_2uv + r_3uw + r_4vw + r_5u + r_6v + r_7w + r_8, \quad r_i \in \mathbb{R},$$

con la que aparece el sistema compatible indeterminado

$$r_1 = 4, \quad r_2 + r_3 + r_4 = -3, \quad r_5 + r_6 + r_7 = 9, \quad r_8 = 1.$$

Si exigimos a f ser simétrica respecto a sus tres argumentos, esto es, que para cualquier permutación $(u_1 u_2 u_3)$ de u, v, w , se cumpla $f(u_1, u_2, u_3) = f(u, v, w)$, entonces $r_2 = r_3 = r_4 = -1$ y $r_5 = r_6 = r_7 = 3$, y la función queda

$$f(u, v, w) = 4uvw - uv - uw - vw + 3u + 3v + 3w + 1.$$

△

¿Cómo podemos llamar a una función así? ¿En dónde hemos visto antes este tipo de funciones, que nos suenan tanto? ¿Qué relación tiene esto con los B-splines y las curvas de Bézier?

§ 4.1. El Principio del blossom

Muchos algoritmos de Bézier y B-splines pueden describirse en términos de funciones como la anterior. Es más, toda la teoría clásica de Bézier y B-splines puede reescribirse completamente mediante funciones así, en una especie de procedimiento inverso. Este desarrollo teórico alternativo fue descubierto por de Casteljaou y desarrollado por Ramshaw a finales de los 80, una vez las bases de la teoría de curvas de Bézier y splines estaban ya asentadas. En este capítulo vamos a ver sólo la superficie de esta apasionante teoría, que podemos encontrar en [9, 11, 12].

Teorema 6 (Principio del blossom, interpretación afín). *Sean P y Q espacios afines de dimensión finita. Entonces las funciones polinómicas¹ $F: P \rightarrow Q$ de grado menor o igual que n son equivalentes a aplicaciones n -afines $F: P^n \rightarrow Q$ simétricas. En particular, dada una función cualquiera de sus tipos, existe una única función del otro tipo tal que*

$$F(t) = f(t, \dots, t).$$

A f la llamaremos blossom (flor, florecimiento o brote) multiafín de F y a F la llamaremos diagonal de f . Diremos que F florece (aflora, germina, blossomiza) en f y que f colapsa en F .

Demostración. No la vemos, pero podemos encontrarla en [11]. Veremos un resultado más débil restringido a curvas polinómicas en la siguiente sección. \square

El teorema anterior cuenta con una versión lineal. Se puede probar que si $F: P \rightarrow Q$ es homogénea de grado n , entonces florece en un blossom multilineal f . Es más, el caso particular de $n = 2$ y $Q = \mathbb{R}$ es un conocido resultado de álgebra lineal:

Las formas cuadráticas en un espacio vectorial V son equivalentes a formas bilineales. Si $f: V \times V \rightarrow \mathbb{R}$ es una forma bilineal simétrica, la aplicación $q: V \rightarrow \mathbb{R}$ dada por $q(u) = f(u, u)$ se denomina forma cuadrática asociada a f . Respectivamente, f se dice forma polar de q .

Históricamente, los blossomes se conocen como formas polares o polinomios simétricos asociados a una aplicación polinomial. *Blossom* es un término que inventó el propio Ramshaw buscando darle un nombre más poético a su generalización de las formas cuadráticas, tal y como admite: «[...] los matemáticos ya han sabido por mucho tiempo que existe una única forma bilineal simétrica asociada a cualquier forma cuadrática [...] de hecho, no hay nada de raro en transferir dicho resultado al caso no homogéneo [...]». Sin embargo, parece que en su siguiente trabajo regresó a la idea clásica de la forma polar, renombrando a este resultado como *Principio de polarización* en [12].

¹curvas polinómicas si nos restringimos a $P = \mathbb{R}$, $Q = \mathbb{R}^d$, $d = 2, 3$.

§ 4.2. El teorema fundamental: curvas de Bézier

En este capítulo solo vamos a ver una versión más débil del Principio del blossom para curvas de Bézier y B-splines. Veamos que a una curva polinomial $\mathbf{B}(t)$ de grado menor o igual que n se le puede asociar un único blossom $\mathbf{B}(t_1, \dots, t_n)$ que cuenta con las siguientes tres propiedades:

(I) Florecimiento. *La curva $\mathbf{B}(t)$ florece en $\mathbf{B}(t_1, \dots, t_n)$, es decir, coincide con su blossom en la diagonal.*

(II) Simetría. *Si $(s_1 \cdots s_n)$ es una permutación de t_1, \dots, t_n , entonces*

$$\mathbf{B}(s_1, \dots, s_n) = \mathbf{B}(t_1, \dots, t_n).$$

(III) Multiafinidad. *$\mathbf{B}(t_1, \dots, t_n)$ es n -afín, es decir, es afín en cada variable. Esto es, si $\alpha \in [0, 1]$,*

$$\mathbf{B}(\alpha u + (1 - \alpha)v, t_2, \dots, t_n) = \alpha \mathbf{B}(u, t_2, \dots, t_n) + (1 - \alpha) \mathbf{B}(v, t_2, \dots, t_n).$$

Para demostrar que cualquier polinomio en una variable tiene un blossom asociado, es suficiente encontrar las formas polares de los elementos de una base del espacio vectorial de los polinomios. Consideremos el intervalo $[a, b]$ y tomemos como base de $\mathbb{P}^n([a, b])$ al conjunto $\{A_i^n(x) \mid i = 0, 1, \dots, n\}$, donde $A_i^n(x)$ son los llamados monomios n -elementales

$$A_i^n(x) = \binom{n}{i} x^i, \quad i = 0, 1, \dots, n. \quad (4.1)$$

Si definimos los polinomios en n variables

$$A_i^n(x_1, \dots, x_n) := \sum_{1 \leq j_1 < \dots < j_i \leq n} x_{j_1} \cdots x_{j_i}, \quad i = 0, 1, \dots, n, \quad (4.2)$$

notemos que $A_i^n(x)$ es la diagonal de $A_i^n(x_1, \dots, x_n)$ los subíndices del sumatorio nos dan exactamente todas las combinaciones posibles de i elementos tomados de $\{x_1, \dots, x_n\}$. Si tomamos $x_j = x$, $j = 1, \dots, n$ se tiene $x_{j_1} \cdots x_{j_i} = x^i$ para cualesquiera $1 \leq j_1 < \dots < j_i \leq n$ y así

$$A_i^n(x_1, \dots, x_n) = \binom{n}{i} x^i = A_i^n(x).$$

Notemos que la propiedad de simetría se cumple trivialmente porque aunque cambiemos el orden de las variables en los productos, estamos tomando las mismas combinaciones (en las combinaciones el orden no importa). A los blossomes de (4.2) se les denomina *polinomios simétricos elementales*, y aparecen en la teoría de Galois como herramienta para demostrar el teorema de de Abel-Ruffini, que enuncia que las ecuaciones polinómicas generales de grado ≥ 5 no son resolubles por radicales.

La tercera propiedad, la multiafinidad, se prueba de forma trivial sin más que observar que, sin pérdida de generalidad, podemos hacer la comprobación sobre los monomios $x_{j_1} \cdots x_{j_i}$ de (4.2). \triangle

Si regresamos a las curvas polinómicas, esto quiere decir que si

$$\mathbf{B}(t) = \sum_{i=0}^n \mathbf{b}_i p_i(t),$$

para unos polinomios p_i de grado menor o igual que n , entonces podemos definir su blossom asociado como

$$\mathbf{B}(t_1, \dots, t_n) = \sum_{i=0}^n \mathbf{b}_i p_i(t_1, \dots, t_n),$$

donde $p_i(t_1, \dots, t_n)$ es el blossom de $p_i(t)$. Notemos que su diagonal puede ser de grado menor o igual que n , a pesar de ser un blossom en n variables.

¿Y si los p_i son los polinomios de Bernstein? Podemos inspirarnos en la deducción de la forma polar de los monomios n -elementales y deducir

$$\mathcal{B}_i^n(t_1, \dots, t_n) := \sum_{\substack{j_1 < \dots < j_i \\ k_1 < \dots < k_{n-i}}} t_{j_1} \cdots t_{j_i} (1 - t_{k_1}) \cdots (1 - t_{k_{n-i}}), \quad (4.3)$$

donde $(j_1 \cdots j_i k_1 \cdots k_{n-i})$ es una permutación de n elementos tomados de $\{1, \dots, n\}$. Realmente como el producto es conmutativo, acabamos sumando $\binom{n}{i}$ combinaciones de n elementos, tomadas del conjunto de las $n!$ posibles permutaciones de $\{1, \dots, n\}$. Con lo que al colapsar el blossom, se obtiene

$$\mathcal{B}_i^n(t, \dots, t) = \mathcal{B}_i^n(t).$$

Observemos que los blossomes de Bernstein heredan (más bien, generalizan) la relación de recurrencia (1.3) de los polinomios de Bernstein:

$$\mathcal{B}_i^{n+1}(t_0, \dots, t_n) = t_0 \mathcal{B}_{i-1}^n(t_1, \dots, t_n) + (1 - t_0) \mathcal{B}_i^n(t_1, \dots, t_n). \quad (4.4)$$

Teorema 7 (Teorema fundamental, interpretación para curvas de Bézier). *Para cada curva polinómica $\mathbf{B}(u)$ de grado menor o igual que n sobre $[a, b]$ existe un único blossom de n variables $\mathbf{B}(u_1, \dots, u_n)$ que satisface $\mathbf{B}(u, \dots, u) = \mathbf{B}(u)$. Es más, los puntos*

$$\mathbf{b}_i^n := \mathbf{B}(a, \dots, \overset{n-i}{a}, a, b, \dots, \overset{i}{b}), \quad i = 0, \dots, n \quad (4.5)$$

son exactamente los puntos de Bézier de $\mathbf{B}(u)$ sobre $[a, b]$.

Demostración. La existencia de la forma polar asociada a $\mathbf{B}(u)$ ya ha sido probada al comienzo de esta sección. La denotamos por $\mathbf{u}(u_1, \dots, u_n)$ y definimos los puntos

$$\mathbf{b}_i^{n-k} := \mathbf{B}(a, \dots, \overset{j}{a}, a, u_1, \dots, \overset{k}{u_k}, b, \dots, \overset{i}{b}), \quad i + j + k = n. \quad (4.6)$$

Como $\mathbf{b}_0^0 = \mathbf{B}(u_1, \dots, u_n)$ es simétrico y multiafín, puede calcularse a partir de los \mathbf{b}_i^n a partir de la relación de recurrencia

$$\mathbf{b}_i^{n-k-1} = (1 - t_{k+1}) \mathbf{b}_i^{n-k} + t_{k+1} \mathbf{b}_{i+1}^{n-k}, \quad k = 0, \dots, n-1, \quad (4.7)$$

donde $t_k := (u_k - a)/(b - a)$. Notemos que $t_k \in [0, 1]$.

En efecto, supongamos que $i + k + j = n$. Tenemos que $u_{k+1} = a(1 - t_{k+1}) + bt_{k+1}$, con lo que

$$\begin{aligned}
\mathbf{b}_i^{n-(k+1)} &= \mathbf{B}(a, \overset{j-1}{\cdot}, a, u_1, \overset{k}{\cdot}, u_k, u_{k+1}, b, \overset{i}{\cdot}, b) \\
&= \mathbf{B}(a, \overset{j-1}{\cdot}, a, u_1, \overset{k}{\cdot}, u_k, [a(1 - t_{k+1}) + bt_{k+1}], b, \overset{i}{\cdot}, b) \\
&= (1 - t_{k+1})\mathbf{B}(a, \overset{j-1}{\cdot}, a, u_1, \overset{k}{\cdot}, u_k, a, b, \overset{i}{\cdot}, b) + t_{k+1}\mathbf{B}(a, \overset{j-1}{\cdot}, a, u_1, \overset{k}{\cdot}, u_k, b, b, \overset{i}{\cdot}, b) \\
&= (1 - t_{k+1})\mathbf{B}(a, \overset{j}{\cdot}, a, u_1, \overset{k}{\cdot}, u_k, b, \overset{i}{\cdot}, b) + t_{k+1}\mathbf{B}(a, \overset{j-1}{\cdot}, a, u_1, \overset{k}{\cdot}, u_k, b, \overset{i+1}{\cdot}, b) \\
&= (1 - t_{k+1})\mathbf{b}_i^{n-k} + t_{k+1}\mathbf{b}_{i+1}^{n-k}.
\end{aligned}$$

Es más, si tomamos la diagonal principal en (4.7), entonces la relación de recurrencia se reduce al algoritmo de De Casteljaou. En consecuencia, como la representación de Bézier es única, los puntos \mathbf{b}_i^n son exactamente los puntos de Bézier de $\mathbf{B}(u)$ sobre $[a, b]$. Por tanto dos blossomes que coinciden sobre la diagonal, coinciden sobre los argumentos $(a, \overset{n-i}{\cdot}, a, b, \overset{n}{\cdot}, b)$ y por la relación (4.7) son idénticos. Luego $\mathbf{B}(u)$ tiene una única forma polar. \square

§ 4.3. El teorema fundamental: curvas B-spline

El teorema fundamental también cuenta con una versión para B-splines, que pone de manifiesto la relación geométrica entre los puntos de control y la curva a través de su blossom.

Teorema 8 (Teorema fundamental, interpretación para curvas B-spline). *Dada una curva B-spline $\mathbf{S}(u)$ de grado k sobre $a = t_0 < \dots < t_n = b$, para cada intervalo nodal $[t_i, t_{i+1})$ existe un único blossom de k variables $\mathbf{S}_i(u_1, \dots, u_k)$ que satisface $\mathbf{S}_i(u, \overset{k}{\cdot}, u) = \mathbf{S}(u)$ en $[t_i, t_{i+1})$. Es más, los puntos*

$$\mathbf{c}_j^k := \mathbf{S}_i(t_{j+1}, \dots, t_{j+k}), \quad j = i - k, \dots, i \quad (4.8)$$

son exactamente los puntos de control de $\mathbf{S}(u)$ sobre $[t_i, t_{i+1})$.

Demostración. Notemos que para $u \in [t_i, t_{i+1})$, $\mathbf{S}_i(u) := \mathbf{S}(u) = \sum_{j=i-k}^i \mathbf{c}_j^k B_i^k(u)$. Además, $\mathbf{S}_i(u)$ es una curva polinómica de grado menor o igual que k , luego la existencia y unicidad de su blossom asociado, que denotamos por $\mathbf{S}_i(u_1, \dots, u_k)$, está garantizada por el teorema fundamental.

Sea un entero r comprendido entre 0 y k . Definimos los puntos

$$\mathbf{c}_j^{k-r} := \mathbf{S}_i(t_{j+1}, \overset{k-r}{\cdot}, t_{j+k-r}, u_r, \overset{r}{\cdot}, u_r), \quad j = i - k, \dots, i, \quad (4.9)$$

para un $\alpha_{u_r} \in [0, 1]$ tal que

$$u_r = (1 - \alpha_{u_r})t_j + \alpha_{u_r}t_{j+k-r+1} \in [t_j, t_{j+k-r+1}).$$

Entonces, como \mathbf{S}_i es k -afín y simétrico, de (4.9) se tiene la recurrencia

$$\mathbf{c}_j^{k-(r+1)} = (1 - \alpha_{u_r})\mathbf{c}_{j-1}^{k-r} + \alpha_{u_r}\mathbf{c}_{k-r}^r, \quad \alpha_{u_r} = \frac{u - t_j}{t_{j+k-r+1} - t_k}, \quad (4.10)$$

y en particular se cumple

$$\mathbf{c}_j^k = \mathbf{S}_i(t_{j+1}, \dots, t_{j+k}), \quad \mathbf{c}_i^0 = \mathbf{S}_i(u_r). \quad (4.11)$$

Si nos fijamos bien en (4.11), si tomamos $u := u_r \in [t_i, t_{i+1})$, entonces $j = i$ y $r = k$, y la relación (4.10) coincide con la del algoritmo de De Boor, como podemos ver en (1.35) o en (2.3). Además puede utilizarse para evaluar cualquier curva polinomial de grado menor o igual que k $\mathbf{B}(u) := \mathbf{S}_i(u, \dots, u)$ sobre $[t_i, t_{i+1})$. \square

Dado que cualquier polinomio de grado n se puede considerar como un spline de grado n sobre cualquier secuencia de nodos, en particular podremos expresar los monomios elementales (4.1) como combinaciones lineales de B-splines sobre dicha secuencia de nodos. Habiendo deducido el blossom de los monomios elementales en (4.2), podemos aplicar el teorema fundamental a $A_i^n(u)$ si lo interpretamos como una polinomial. Entonces, para $u \in [t_i, t_{i+1})$,

$$A_i^n(u) = \sum_{j=i-k}^i \alpha_{i,j} B_i^n(u),$$

donde $\alpha_{i,j} := A_i^n(t_{j+1}, \dots, t_{j+n})$ son sus puntos de control. En consecuencia, si definimos una polinomial en base monomial, podemos generalizar la conocida como identidad de Marsden, como la siguiente ecuación

$$\begin{aligned} \mathbf{P}(u) &= \mathbf{a}_0 A_0^n(u) + \dots + \mathbf{a}_n A_n^n(u) \\ &= \sum_{j=0}^n (\mathbf{a}_0 \alpha_{0,j} + \dots + \mathbf{a}_n \alpha_{n,j}) B_j^n(u). \end{aligned} \quad (4.12)$$

En particular, aplicando (4.12) para la identidad en $[t_i, t_{i+1})$,

$$u = \frac{1}{n} n u = \frac{1}{n} \binom{n}{1} u = \frac{1}{n} A_1^n(u) = \sum_{j=i-k}^i \gamma_j B_j^n(u), \quad \gamma_j := \frac{\alpha_{1,j}}{n} = \frac{t_{j+1} + \dots + t_{j+n}}{n}. \quad (4.13)$$

Las γ_j que aparecen en (4.13) se denominan abscisas de Greville y aparecen de forma natural como los puntos de control del gráfico en \mathbb{R}^2 de una función spline $s(u) = \sum_j c_j B_j^n$. El gráfico $\mathbf{s}(u) = (u, s(u))$ tendrá por puntos de control a $\mathbf{c}_j = (\gamma_j, c_j)$.

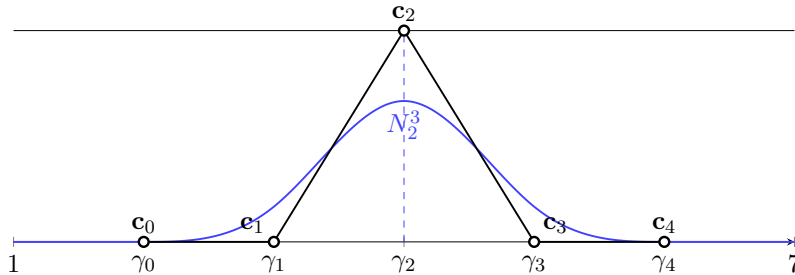


Figura 4.1: Polígono de control de la curva $y = N_2^3(u)$ dado por las abscisas de Greville, que coinciden con los nodos por estar en el caso uniforme.

Capítulo 5

Introducción a la subdivisión

En este capítulo vamos a cambiar el enfoque de la construcción de curvas mediante los algoritmos clásicos de De Casteljau y de De Boor. Un primer vistazo a curvas de Bézier o B-splines de grado alto nos puede llevar a conjeturar lo siguiente, los polígonos de Bézier y de control... ¿aproximan fielmente la curva conforme aumentamos el grado? En este capítulo vamos a contestar esa pregunta desde el paradigma de los esquemas de subdivisión. Si bien la subdivisión posee su nicho en las superficies, no vamos a poder verlas en profundidad en este trabajo y nos centraremos solamente en las curvas. Gracias a los resultados del capítulo 4, podremos demostrar su convergencia utilizando una herramienta con tanto juego como es la de los blossomes.

Cabe mencionar que existe otra estrategia parecida a la subdivisión, denominada *elevación de grado*, y cuyo refinamiento del polígono de Bézier o de control también converge a la curva en cuestión. Toda la información de este capítulo, junto con la elevación de grado, podemos encontrarla en [9], aunque también nos hemos basado algo en comentarios de [13].

§ 5.1. Subdivisión de curvas de Bézier

La relación de recurrencia obtenida del teorema fundamental (4.7), nos revela una importante propiedad del algoritmo de De Casteljau: su capacidad para generar curvas de subdivisión.

Recuperando el cuadro 2.1 con el arreglo triangular del algoritmo de De Casteljau, observemos que si queremos calcular el valor de una curva polinomial $\mathbf{B}(u)$ en $u = c \in [a, b]$, los puntos de Bézier de la fila superior y los de la diagonal inferior del arreglo son

$$\mathbf{b}_0^i = \mathbf{B}(a, \overset{n-i}{\dots}, a, \overset{i}{\dots}, c), \text{ y } \mathbf{b}_i^i = \mathbf{B}(c, \overset{n-i}{\dots}, c, \overset{i}{\dots}, b)$$

en el orden dado, para $i = 0, \dots, n$. Si nos fijamos, se encuentran calculados sobre los subintervalos $[a, c]$ y $[c, b]$ respectivamente. De esta forma, el cálculo de los puntos de Bézier sobre esta partición de $[a, b]$ se trata de un algoritmo de subdivisión.

En efecto, notemos que subdividir repetidamente la curva de esta forma, comenzando por ejemplo en $c \in [a, b]$ y continuando con $c_i := c + ih$, $i = 0, \dots, k$ para un paso h , genera una partición $a_0 \leq c_0 < \dots < c_k \leq b$ de su dominio. La curva de subdivisión asociada es la poligonal compuesta por la unión de los segmentos de los polígonos de Bézier sobre los subintervalos de la forma $[c_i, c_{i+1}]$. En general, constará de $kn + 1$ vértices.

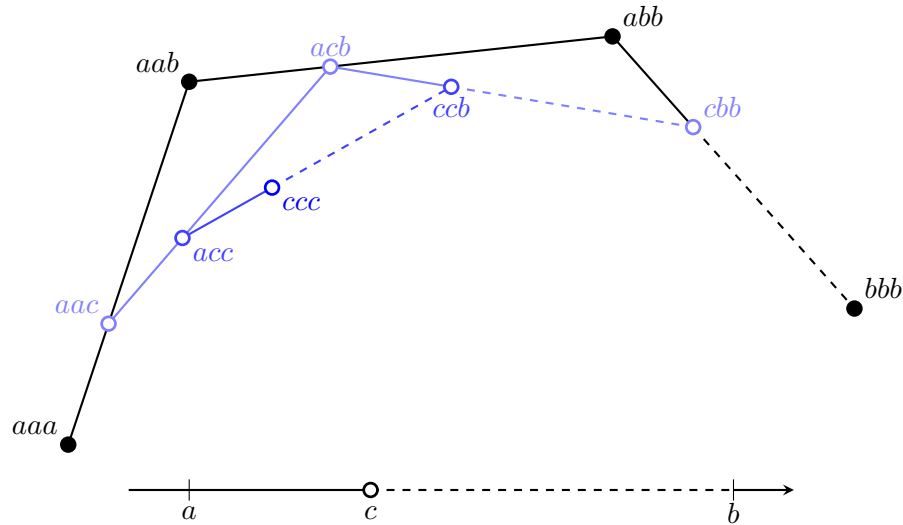


Figura 5.1: Interpretación del teorema fundamental de la subdivisión de una curva de Bézier cúbica en $c \in [a, b]$. Para determinar con una iteración más cómo evoluciona la curva de subdivisión en el dominio en líneas discontinuas, tenemos que tomar otro punto $c < d \leq b$.

5.1.1. Convergencia de la subdivisión de De Casteljaou

Vamos a demostrar la idea intuitiva de que cuanto menor sea la longitud de los subintervalos, mejor será la aproximación de la curva de subdivisión, viendo el siguiente resultado de convergencia, demostrable gracias a los blossommes.

Teorema 9 (Convergencia cuadrática del algoritmo de De Casteljaou). *Sea $\mathbf{B}(u)$ una curva de Bézier de grado n . Sean $\mathbf{b}_0, \dots, \mathbf{b}_n$ sus puntos de Bézier sobre el subintervalo $[c, c + nh]$ para un $c \in [a, b]$ y sean $c_i = c + ih$, $i = 0, \dots, n$, para un paso h que asegure $c + nh \leq b$. Consideremos una norma $\|\cdot\|$. Entonces existe una constante $M \geq 0$ que no depende de c tal que*

$$\max_i \|\mathbf{B}(c_i) - \mathbf{b}_i\| \leq Mh^2. \quad (5.1)$$

Demostración. Para probar el resultado, consideramos el blossom $\mathbf{B}(u_1, \dots, u_n)$ y tomamos su expansión de Taylor hasta orden 1 en un entorno de $\mathbf{c}_i := (c_i, \dots, c_i)$, para luego evaluarlo en $\mathbf{u} = (u_1, \dots, u_n) := (c, \overset{n-i}{\cdot}, c, c + nh, \overset{i}{\cdot}, c + nh)$.

Por el teorema de Taylor,

$$\mathbf{B}(\mathbf{u}) = \mathbf{c}_i + \sum_{j=1}^n \frac{\partial}{\partial u_j} \mathbf{B}(\mathbf{c}_i)(u_j - c_i) + O(\|\mathbf{u} - \mathbf{c}_i\|^2). \quad (5.2)$$

Así, sustituyendo \mathbf{u} en (5.2), teniendo en cuenta que $c - c_i = ih$ y aplicando las propiedades de la O de Landau,

$$\begin{aligned} \mathbf{b}_i &= \mathbf{B}(\mathbf{u}) \\ &= \mathbf{B}(\mathbf{c}_i) + \sum_{j=1}^{n-i} \frac{\partial}{\partial u_j} \mathbf{B}(\mathbf{c}_i)(c - c_i) + \sum_{j=n-i+1}^n \frac{\partial}{\partial u_j} \mathbf{B}(\mathbf{c}_i)(c + nh - c_i) \\ &\quad + O(\|(c - c_i, \dots, c - c_i, c + nh - c_i, \dots, c + nh - c_i)\|^2) \\ &= \mathbf{B}(\mathbf{c}_i) - ih \sum_{j=1}^{n-i} \frac{\partial}{\partial u_j} \mathbf{B}(\mathbf{c}_i) + (n - ih) \sum_{j=n-i+1}^n \frac{\partial}{\partial u_j} \mathbf{B}(\mathbf{c}_i) + O(h^2) \\ &= \mathbf{B}(\mathbf{c}_i) + O(h^2). \end{aligned}$$

La última igualdad es cierta porque todas las derivadas parciales son iguales debido a la propiedad de simetría de los blossomes.

Luego $\mathbf{b}_i - \mathbf{B}(\mathbf{c}_i) = O(h^2)$ y por la definición de Landau, existe una constante $M \geq 0$ tal que $\|\mathbf{b}_i - \mathbf{B}(\mathbf{c}_i)\| \leq Mh^2$. Si colapsamos el blossom, hemos terminado. \square

§ 5.2. Subdivisión de curvas B-spline

El cálculo de curvas B-spline se simplifica mucho si los nodos son simples y están uniformemente espaciados. Además, esta definición los dota de propiedades muy buenas de traslación y escalabilidad, que nos van a permitir idear algoritmos de subdivisión basados en cálculos matriciales. A continuación veremos el algoritmo de subdivisión uniforme, que se trata del más simple de toda una clase de algoritmos denominados algoritmos de subdivisión estacionaria.

5.2.1. B-splines uniformes

Como hemos mencionado, buscamos simplificar el trabajo con B-splines a partir de su definición, tomando los nodos equiespacios sobre \mathbb{Z} . Por diferenciar la notación entre los B-splines generales y los que vamos a ver, que denominaremos *B-splines uniformes*, vamos a recurrir a la letra N en lugar de la B para referirnos a estos, aunque cabe advertir que clásicamente a los B-splines se les ha denotado siempre con la N en el caso general, véase por ejemplo [5].

Trabajaremos sobre toda la recta real extendiendo el comportamiento local del intervalo $[0, 1]$. Definimos para todo $k \geq 0$, el B-spline uniforme

$$N^k(u) := B_0^k(u), \quad u \in [0, 1]. \quad (5.3)$$

De esta forma

$$N^0(u) = \begin{cases} 1 & \text{si } 0 \leq u \leq 1, \\ 0 & \text{en caso contrario,} \end{cases} = \mathcal{X}_{[0,1]}(u). \quad (5.4)$$

Con esta definición es fácil comprobar, teniendo en cuenta $t_{i+1} - t_i = 1$, que $N^k(u - j) = B_j^k(u)$, con lo que simplificamos la definición recursiva de De Boor a

$$N^k(u) = \frac{u}{n} N^{k-1}(u) + \frac{(n+1) - u}{n} N^{k-1}(u - 1), \quad (5.5)$$

y la expresión de la derivada a

$$\frac{d}{du} N^k(u) = N^{k-1}(u) - N^{k-1}(u - 1). \quad (5.6)$$

Nota 7. Notemos que de la identidad (1.43), tomando la definición uniforme (5.3) se deduce $\int_{-\infty}^{\infty} N^k(s) ds = 1$ y los B-splines uniformes coinciden con los que define Schoenberg en el teorema 3 si se toma como $\Omega_{\infty} = \mathbb{Z}$.

Definición por convolución

Veamos otra definición alternativa y muy interesante de cara al cálculo numérico, que es la definición de los B-splines uniformes en términos de una convolución. En efecto, teniendo en cuenta (5.4) y la derivada (5.6)

$$\begin{aligned} N^k(u) &= \int_0^u (N^{k-1}(v) - N^{k-1}(v-1)) dv = \int_0^u N^{k-1}(v) dv - \int_0^u N^{k-1}(v-1) dv \\ &= \int_0^u N^{k-1}(v) dv - \int_0^{u-1} N^{k-1}(t) dt \\ &= \left(\int_0^{u-1} N^{k-1}(v) dv + \int_{u-1}^u N^{k-1}(v) dv \right) - \int_0^{u-1} N^{k-1}(t) dt \\ &= \int_{u-1}^u N^{k-1}(v) dv = \int_0^1 N^{k-1}(u-t) dt \\ &= \int_{\mathbb{R}} N^{k-1}(u-t) \mathcal{X}_{[0,1]}(t) dt = (N^{k-1} * N^0)(u). \end{aligned}$$

Luego podemos calcular los B-splines uniformes con un algoritmo recursivo que integre numéricamente la siguiente expresión

$$N^k(u) = (N^{k-1} * N^0)(u), \quad u \in [0, 1]. \quad (5.7)$$

△

Dado un $i \in \mathbb{Z}$, podemos definir el resto de B-splines desplazando el definido en (5.3) como

$$N_i^k(u) := N^k(u - i), \quad k \geq 0. \quad (5.8)$$

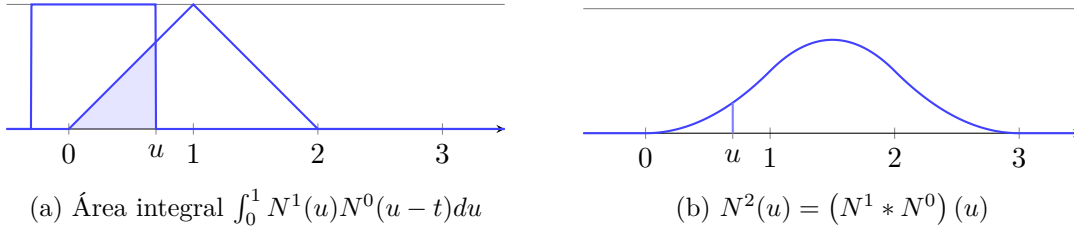


Figura 5.2: Construcción del B-spline N^2 por convolución de N^1 y N^0 como en (5.7). La conmutatividad es una conocida propiedad de las convoluciones y en este caso nos permite dar una interpretación gráfica muy interesante.

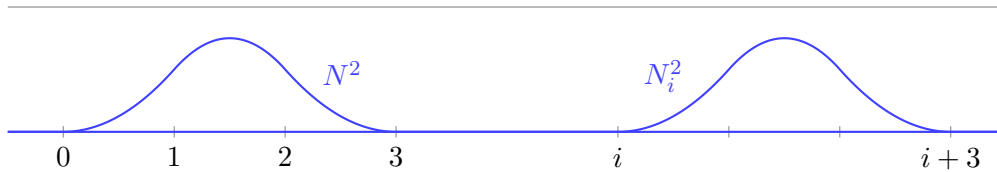


Figura 5.3: B-spline uniforme cuadrático trasladado i unidades a la derecha.

De forma parecida se definen los llamados B-splines uniformes $(1/2)$ -escalados como

$$M^k(u) := N^k(2u), \quad k \geq 0, \quad (5.9)$$

y sus trasladados $(1/2)$ unidades,

$$M_i^k(u) := M^k(u - i/2) = N^k(2u - i) = N_i^k(2u), \quad k \geq 0. \quad (5.10)$$

Notemos que los nodos de los M_i^k se encuentran sobre los puntos del conjunto $\frac{1}{2}\mathbb{Z} := \{\frac{1}{2}m \mid m \in \mathbb{Z}\}$. Este «refinamiento» es el motivo por el cual los B-splines escalados van a ser nuestra principal herramienta en la subdivisión uniforme, que vamos a ver a continuación.

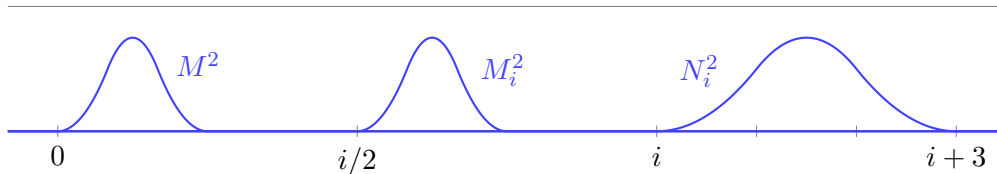


Figura 5.4: Comparativa entre los B-spline uniforme cuadrático $(1/2)$ -escalado, $(1/2)$ -escalado trasladado $i/2$ unidades a la derecha y uniforme trasladado i unidades a la derecha.

5.2.2. Subdivisión uniforme

Supongamos que tenemos el siguiente B-spline uniforme de grado k

$$\mathbf{S}(u) = \sum_{i=-k}^{n-1} \mathbf{c}_i^k N_i^k(u)$$

sobre la partición uniforme $a = 0 < 1 < \dots < i < \dots < n = b$, y que queremos representar esta misma curva sobre una partición refinada

$$a = 0 < \frac{1}{2} < \dots < i - \frac{1}{2} < i < i + \frac{1}{2} < \dots < n = b,$$

es decir, queremos imponer

$$\sum_{i=-k}^{n-1} \mathbf{b}_i^k M_i^k(u) := \mathbf{S}(u)$$

para unos coeficientes \mathbf{b}_i^k adecuados. ¿Cuál es la relación entre los \mathbf{c}_i^k y los \mathbf{b}_i^k ? Veámoslo.

$k = 0$. La igualdad queda sencilla de estudiar:

$$\mathbf{S}(u) = \sum_{i=0}^{n-1} \mathbf{c}_i^0 N_i^0(u) = \sum_{i=0}^{n-1} \mathbf{c}_i^0 \left(M_{2i}^0(u) + M_{2i+1}^0(u) \right),$$

donde la última igualdad se puede comprobar geoméricamente fácilmente. Así se deduce que

$$\mathbf{c}_i^0 = \mathbf{b}_{2i}^0 = \mathbf{b}_{2i+1}^0, \quad i = 0, \dots, n-1. \quad (5.11)$$

$k > 0$. Para este caso, la definición por convolución vista en (5.7) va a ser de gran utilidad. Primero notemos lo siguiente:

$$\begin{aligned} N_i^k(u) &= N^k(u-i) = \int_{-\infty}^{+\infty} N^{k-1}(u-i-t)N^0(s)dt \\ &= \int_{-\infty}^{+\infty} N_i^{k-1}(u-t)N^0(t)dt = \left(N_i^{k-1} * N^0 \right) (u). \end{aligned} \quad (5.12)$$

Para los B-splines escalados sucede algo muy parecido:

$$\begin{aligned} M_i^k(u) &= N_i^k(2u) = \int_{-\infty}^{+\infty} N_i^{k-1}(2u-t)N^0(t)dt = 2 \int_{-\infty}^{+\infty} N_i^{k-1}(2u-2s)N^0(2s)ds \\ &= 2 \int_{-\infty}^{+\infty} M_i^{k-1}(u-t)M^0(t)dt = 2 \left(M_i^{k-1} * M^0 \right) (u). \end{aligned} \quad (5.13)$$

Entonces, aplicando (5.12) y (5.13) podemos reescribir

$$\mathbf{S}(u) = \sum_{i=-k}^{n-1} \mathbf{c}_i^k N_i^k(u) = \sum_{i=-k}^{n-1} \mathbf{c}_i^k \left(N_i^{k-1} * N^0 \right) (u) = \int_{-\infty}^{+\infty} \sum_{i=-k}^{n-1} \mathbf{c}_i^k N_i^{k-1}(u-t)N^0(t)dt,$$

e imponiendo $\mathbf{c}_i^k N_i^{k-1} = \mathbf{b}_i^{k-1} M_i^{k-1}$, se sigue que

$$\begin{aligned} \mathbf{S}(u) &= \int_{-\infty}^{+\infty} \sum_{i=-k}^{n-1} \mathbf{b}_i^{k-1} M_i^{k-1}(u-t) \left[M_0^0(t) + M_1^0(t) \right] dt \\ &= \sum_{i=-k}^{n-1} \mathbf{b}_i^{k-1} \left[\int_{-\infty}^{+\infty} M_i^{k-1}(u-t)M^0(t)dt + \int_{-\infty}^{+\infty} M_i^{k-1}(u-t)M_1^0(t)dt \right]. \end{aligned} \quad (5.14)$$

El primer sumando que aparece en la expresión entre corchetes, por la identidad (5.13) es igual a $\frac{1}{2}M_i^k(u)$. El segundo sumando podemos desarrollarlo como sigue

$$\begin{aligned}\int_{-\infty}^{+\infty} M_i^{k-1}(u-t)M_1^0(t)dt &= \int_{-\infty}^{+\infty} M^{k-1}(u-(i+1)-(t-1))M^0(t-1)dt \\ &= \int_{-\infty}^{+\infty} M_{i+1}^{k-1}(u-s)M^0(s)ds \\ &= \frac{1}{2}M_{i+1}^k(u).\end{aligned}$$

Si sustituimos estas expresiones en (5.14) llegamos a

$$\begin{aligned}\mathbf{S}(u) &= \frac{1}{2} \sum_{i=-k}^{n-1} \mathbf{b}_i^{k-1} (M_i^k(u) + M_{i+1}^k(u)) = \frac{1}{2} \left(\sum_{i=-k}^{n-1} \mathbf{b}_i^{k-1} M_i^k(u) + \sum_{j=-k}^{n-1} \mathbf{b}_j^{k-1} M_{j+1}^k(u) \right) \\ &= \frac{1}{2} \left(\sum_{i=-k}^{n-1} \mathbf{b}_i^{k-1} M_i^k(u) + \sum_{i=-k+1}^n \mathbf{b}_{i-1}^{k-1} M_i^k(u) \right).\end{aligned}\tag{5.15}$$

Notemos que haber seguido un estudio riguroso de la curva nos está impidiendo continuar, porque no tenemos forma de combinar los dos sumatorios de (5.15). Supongamos que podemos añadir puntos de control a conveniencia a medida que los necesitemos para hacer los cálculos y que no nos importa alterar la definición de la curva \mathbf{S} conforme hagamos la subdivisión. Entonces podemos definir la curva de una forma menos restrictiva, como

$$\mathbf{S}(u) = \sum_{i \in I} \mathbf{c}_i N_i^k = \sum_{i \in I} \mathbf{b}_i^k M_i^k,$$

para un conjunto numerable I . Ahora es posible reescribir (5.15) como

$$\mathbf{S}(u) = \frac{1}{2} \sum_{i \in I} (\mathbf{b}_{i-1}^{k-1} + \mathbf{b}_i^{k-1}) M_i^k(u),\tag{5.16}$$

de lo que se deduce $\mathbf{b}_i^k = \frac{1}{2}(\mathbf{b}_{i-1}^{k-1} + \mathbf{b}_i^{k-1})$.

Como se puede hacer un estudio exactamente igual para los grados $j = 1, 2, \dots, k-1$, podemos definir como en (5.16) el resto de coeficientes,

$$\mathbf{b}_i^{j+1} = \frac{1}{2}(\mathbf{b}_{i-1}^j + \mathbf{b}_i^j), \quad j = 0, 1, \dots, k-1.\tag{5.17}$$

Este cálculo recursivo de los \mathbf{b}_i^k es conocido como algoritmo de Lane y Riesenfeld, o sencillamente como algoritmo de subdivisión uniforme.

Dado el polígono de control, primero se duplican todos los puntos de control y posteriormente se construye un nuevo polígono conectando los puntos medios repetidamente (Lane y Riesenfeld, 1980).

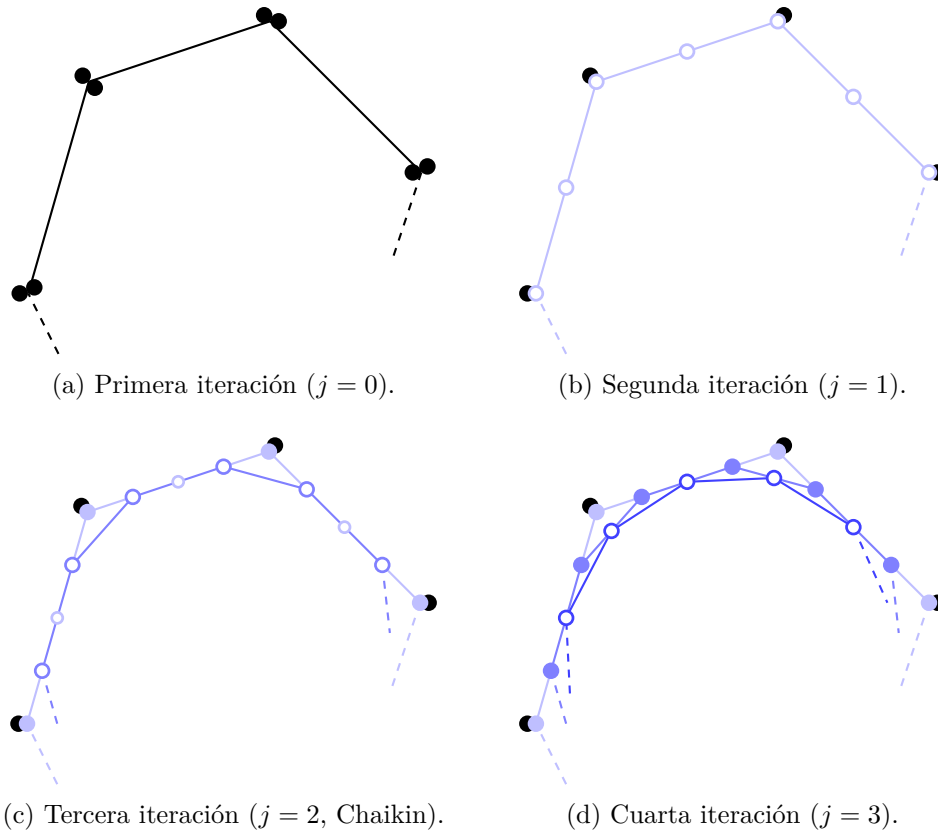


Figura 5.5: Algoritmo de Lane y Riesenfeld para una curva B-spline cúbica ($k = 3$). Notemos que conforme iteramos, necesitamos nuevos puntos para conocer el comportamiento en los extremos.

El anterior algoritmo utilizado para generar B-splines de grado $k = 2$ se conoce como algoritmo de Chaikin. Curiosamente es un algoritmo anterior al de Lane y Riesenfeld, descubierto en 1974. El estudio de Chaikin tenía un enfoque puramente geométrico, desligado completamente de la teoría de B-splines. Sin embargo, años más tarde, Riesenfeld probaría que el algoritmo de Chaikin generaba B-splines cuadráticos.

Definición 11 (Curva de subdivisión). *Dado un algoritmo de subdivisión, diremos que cada polígono de control generado en las sucesivas iteraciones de este, es una curva de subdivisión asociada al B-spline.*

Notemos que la curva de subdivisión es una aproximación de la curva original por splines lineales, como los vistos en (1.8). La subdivisión de B-splines es un proceso iterativo que busca generar más puntos de control a partir de los antiguos, de forma que la curva de subdivisión cada vez aproxime mejor la curva, que es lo que se conoce como refinamiento.

¿Y si queremos refinar aún más la secuencia de nodos? Por ejemplo, hasta $\frac{1}{4}\mathbb{Z}$ o en general, hasta $2^{-m}\mathbb{Z}$. Aquí nuestro enfoque analítico deja de ser útil. Veamos cómo describir el algoritmo de subdivisión uniforme con notación matricial.

Definimos las matrices infinitas

$$C := [\cdots \mathbf{c}_{i-1} \ \mathbf{c}_i \ \mathbf{c}_{i+1} \ \cdots], \quad B_k := [\cdots \mathbf{b}_{i-1}^k \ \mathbf{b}_i^k \ \mathbf{b}_{i+1}^k \ \cdots], \quad (5.18)$$

asociadas a los puntos de control \mathbf{c}_i y \mathbf{b}_i^k de una misma curva B-spline de grado k tomada sobre \mathbb{Z} y $\frac{1}{2}\mathbb{Z}$ respectivamente. Para las iteraciones necesarias para llegar desde la matriz C hasta B_k definimos

$$\begin{aligned} B_{j+1} &= B_j M, \quad j = 0, 1, \dots, k-1, \\ B_0 &= CD, \end{aligned} \quad (5.19)$$

donde C y M son las matrices infinitas

$$D := \begin{bmatrix} \ddots & & & & & & \\ & \ddots & & & & & \\ & & 1 & 1 & & & \\ & & & 1 & 1 & & \\ & & & & \ddots & \ddots & \\ & & & & & \ddots & \ddots \end{bmatrix}, \quad M := \frac{1}{2} \begin{bmatrix} \ddots & & & & & & \\ & \ddots & & & & & \\ & & 1 & 1 & & & \\ & & & 1 & 1 & & \\ & & & & 1 & 1 & \\ & & & & & \ddots & \ddots \end{bmatrix} \quad (5.20)$$

La multiplicación por D corresponde a la duplicación de los puntos de control iniciales, mientras que la multiplicación por M corresponde al cálculo de los puntos medios de pares consecutivos de puntos de control.

Ya estamos en condiciones de definir el operador asociado a la subdivisión uniforme. Si denotamos por M^j a la potencia j -ésima de M , entonces

$$S_k := DM^k$$

es el operador de subdivisión uniforme para las curvas B-spline de grado k . El procedimiento de subdivisión se puede repetir cuantas veces queramos, basta aplicar S_k varias veces. Por ejemplo, aplicando S_k dos veces, se obtiene la curva de subdivisión sobre $\frac{1}{4}\mathbb{Z}$ siguiente

$$\begin{aligned} \mathbf{S}(u) &= \sum_{i \in I} \mathbf{c}_i N_i^k(u) \\ &= \sum_{i \in I} \mathbf{b}_i^k N_i^k(2u) \\ &= \sum_{i \in I} \mathbf{a}_i^k N_i^k(4u), \end{aligned}$$

donde $[\cdots \mathbf{a}_{i-1}^k \ \mathbf{a}_i^k \ \mathbf{a}_{i+1}^k \ \cdots] = CS_k S_k$ nos da el polígono de control, y por tanto la curva de subdivisión.

A continuación veremos un resultado que tiene como consecuencia que la sucesión de curvas de subdivisión sobre 2^{-m} dada por $(CS_k^m)_{m \in \mathbb{N}}$ converge a la curva \mathbf{S} cuando m tiende a infinito.

5.2.3. Convergencia de subdivisión de curvas B-spline

Para concluir con el capítulo, vamos a probar que podemos generalizar el resultado (5.1) para cualquier algoritmo de subdivisión no estacionario de B-splines, tampoco necesariamente uniformes.

Teorema 10 (Convergencia cuadrática de subdivisión de curvas B-spline). *Sea la curva B-spline $\mathbf{S}(u)$ de grado n sobre la secuencia de nodos $(t_i)_{i \in I}$. Sean $[a, b]$ un intervalo, $h = \max\{t_{i+1} - t_i \mid [t_i, t_{i+1}] \subset [a, b]\}$ y $\gamma_i = (t_{i+1} + \dots + t_{i+n})/n$ las abscisas de Greville. Entonces existe una constante $M \geq$ que no depende de las abscisas de Greville tal que*

$$\max_i \|\mathbf{S}(\gamma_i) - \mathbf{c}_i\| \leq Mh^2. \quad (5.21)$$

Esto es, si a través de un proceso iterativo la secuencia de nodos se hace densa, la velocidad de convergencia de la curva de subdivisión asociada es cuadrática con relación al máximo de las distancias internodales.

Demostración. La demostración es muy similar a la vista para curvas de Bézier. Consideremos un punto de control $\mathbf{c}_i = \mathbf{S}_r(t_{i+1}, \dots, t_{i+n})$, donde \mathbf{S}_r es la forma polar de \mathbf{S} restringida al intervalo $[t_r, t_{r+1})$ que contiene a γ_i . Como

$$\frac{\partial}{\partial u_1}(u, \dots, u) = \dots = \frac{\partial}{\partial u_n}(u, \dots, u) = \frac{d}{du} \mathbf{S}_r(u),$$

la expansión de Taylor de \mathbf{S}_r alrededor de $\Gamma_i := (\gamma_i, \dots, \gamma_i)$ tiene la forma

$$\begin{aligned} \mathbf{c}_i &= \mathbf{S}_r(t_{i+1}, \dots, t_{i+n}) = \mathbf{S}_r(\Gamma_i) + \sum_{j=1}^n \frac{\partial}{\partial u_j} \mathbf{S}_r(\Gamma_i) (t_{i+j} - \gamma_i) + O(h^2) \\ &= \mathbf{S}_r(\Gamma_i) + \frac{\partial}{\partial u_1} \mathbf{S}_r(\Gamma_i) \sum_{j=i+1}^{i+n} (t_j - \gamma_i) + O(h^2) \\ &= \mathbf{S}_r(\Gamma_i) + \frac{\partial}{\partial u_1} \mathbf{S}_r(\Gamma_i) \left[\sum_{j=i+1}^{i+n} t_j - n\gamma_i \right] + O(h^2) \\ &= \mathbf{S}_r(\Gamma_i) + O(h^2) \end{aligned}$$

Si aplicamos la definición de Landau y colapsamos la forma polar habremos probado la afirmación. \square

Con este resultado concluye el pequeño desarrollo teórico sobre esquemas de subdivisión de este trabajo, que por razones de extensión, hemos limitado a curvas. En el siguiente y último capítulo echaremos un vistazo, con un enfoque puramente divulgativo, a la subdivisión de superficies; que se trata de todo un campo teórico que a día de hoy se sigue investigando, y que es punto de encuentro para el análisis numérico, la topología y la geometría.

Capítulo 6

Algunas aplicaciones de las técnicas de Bézier y B-splines

Habiendo finalizado el desarrollo teórico de este Trabajo Fin de Grado, dedicaremos este último capítulo a conocer las principales aplicaciones que tienen las técnicas de Bézier y B-splines, desde un punto de vista puramente divulgativo. Hemos seguido de forma paralela el desarrollo histórico y teórico de ambos enfoques, centrándonos fundamentalmente en las propiedades de los B-splines y cómo resuelven algunos de los problemas que tienen las curvas de Bézier. Sin embargo, en la vida real ambos modelos coexisten, porque son útiles para diferentes cuestiones. Cabe mencionar que no son las únicas representaciones de curvas que se utilizan en la práctica, porque ninguna es perfecta en todas las circunstancias.

§ 6.1. Diferentes utilidades de la suavidad

A lo largo del trabajo hemos valorado mucho que las curvas nos puedan aportar libertad de diseño. Sin embargo, en algunos ámbitos o incluso en algunas industrias, la suavidad de la curva es mucho más importante que su aspecto, como podemos constatar en los ejemplos de [15, 22].

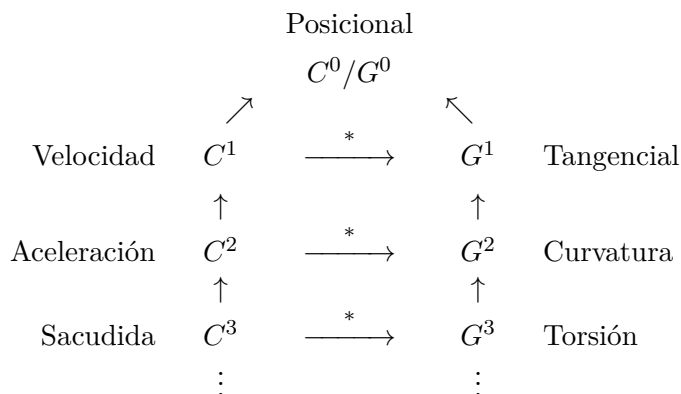
Este sería el caso, por ejemplo, de la industria de la robótica. Si tenemos un brazo robótico, interesaría poder programar su movimiento trazando una curva lo suficientemente suave como para que su velocidad, o incluso su aceleración, sean continuas. De esta forma, el brazo sufrirá lo menos posible si se producen movimientos bruscos, y además será más operativo. El mismo principio se puede aplicar en el diseño de carreteras o vías de tren.

Tal y como mencionamos en el capítulo 1, *suavidad* en matemáticas tiene múltiples significados, como por ejemplo la continuidad diferenciable, o paramétrica, que obedece a la existencia de las sucesivas derivadas de una curva. Quizá menos conocida sea la continuidad geométrica.

Siguiendo el ejemplo del movimiento de los brazos de un robot, nos interesaría, por ejemplo, que el vector velocidad (el vector tangente) tuviera una evolución «continua». Esto se conoce como movimiento tangencialmente continuo y motiva la definición de la continuidad geométrica G^1 , también llamada continuidad tangencial, en términos del vector tangente a una curva. Diremos que una curva $\gamma(t)$ tiene un punto de contacto G^1 en $t = t_0$ si los vectores tangentes a la curva en los dos segmentos de curva a la izquierda y a la derecha de t_0 son paralelos cuando $t \rightarrow t_0$.

La continuidad geométrica G^2 , o continuidad de curvatura, es una condición más fuerte. La curva $\gamma(t)$ tiene un punto de contacto G^2 en $t = t_0$ si la función curvatura de γ es continua en $t = t_0$. El concepto de continuidad geométrica se puede generalizar también a superficies. Este tipo de continuidad es vital en la industria, a la hora de fabricar superficies reflectantes. Una superficie reflectante, diseñada por ejemplo a partir de parches de Bézier, que cuente con discontinuidades en su curvatura, provocará reflejos no deseados y poco estéticos. Por ejemplo, algunos smartphones actuales tienen pantallas biseladas en sus extremos, que son diseñadas a partir del curioso *squircle*¹. O para el caso de curvas, el movimiento de una cámara de cine sobre raíles interesa que sea G^2 , porque si no se pueden provocar desenfoques, pues una cámara es muy sensible a la aceleración.

Se puede probar que si $\gamma(t)$ es regular, esto es, si el vector tangente no es nulo en ninguno de sus puntos, se da la siguiente relación entre la continuidad paramétrica y la geométrica.



Cuadro 6.1: Relación entre la continuidad paramétrica y la geométrica, * si la curva es regular.

Podemos dar una definición de contacto de orden $r > 2$ de forma general. Decimos que dos curvas $p(s)$ y $q(t)$, ambas r veces diferenciables en $s = t = x_0$, tienen un contacto de orden r en x_0 si $q(x_0) \neq 0$ y si existe una reparametrización $s = s(t)$ tal que $p(t)$ y $q(s(t))$ tienen derivadas continuas hasta orden r en x_0 .

Por desgracia, no podemos extendernos más en este tema, pero recomendamos encarecidamente revisar [15] para entender de una forma visual y animada todo lo que hemos contado en la sección. Concluimos este pequeño vistazo al mundo de la suavidad y sus aplicaciones con la tabla 6.2, con los splines más utilizados en la práctica y sus casos de uso. Si bien no los hemos visto, merecen una especial mención los splines de Hermite y los splines de Catmull-Rom, que podemos encontrar en dicha tabla, junto con unas referencias para quien quiera saber más acerca de ellos.

¹Un squircle o círculo cuadrado en español, es una forma intermedia entre círculo y cuadrado, de clase G^2 .

Splines	Grado	Continuidad	Tangentes	Interpolación	Casos de uso
Bézier	3	C^0/C^1	Manual	Algunos	Formas, tipografías y gráficos vectoriales
Hermite	3	C^0/C^1	Explícita	Todos	Animación, simulación de físicas e interpolación
Catmull-Rom	3	C^1	Automática	Todos	Animación y suavizado de trazos
B-spline	3	C^2	Automática	Ninguno	Formas, animaciones y caminos sensibles a la curvatura
Lineal	1	C^0	Automática	Todos	Representación de datos densos, interpolación y subdivisión de curvas

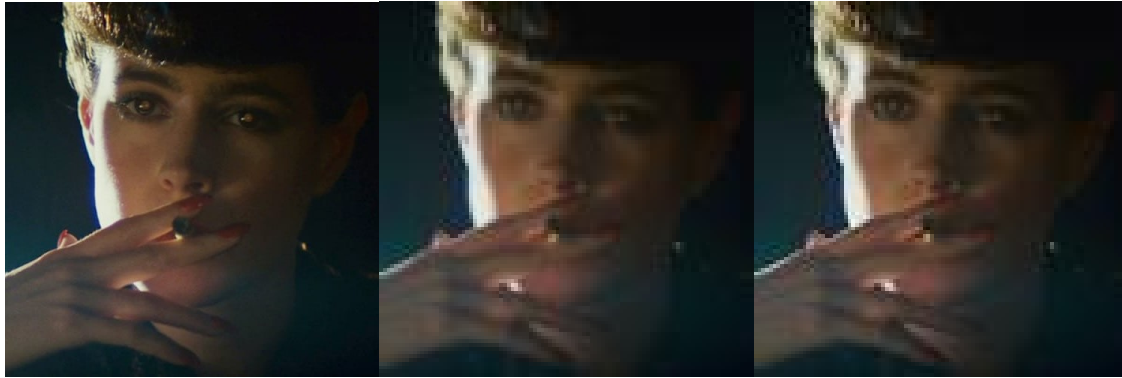
Cuadro 6.2: Tabla de los splines más utilizados, junto a su grado, continuidad, posibilidad de manipulación de sus vectores tangentes, puntos que interpolan y casos de uso. Para más información de los splines de Hermite y Catmull-Rom, contamos con [15, 23].

§ 6.2. Rasterizado, edición e interpolación de imagen

En matemáticas, las imágenes son modelizadas normalmente como aplicaciones $f: U \rightarrow V$, donde $U \subseteq \mathbb{R}^2$. Podríamos pensar que U es nuestra ventana a la realidad, y $f(x) \in V$ es el valor que corresponde en nuestra imagen para cada punto de $x \in U$. El rango V suele ser el espacio RGB, otro subconjunto de \mathbb{R}^3 , donde cada eje corresponde a un color: rojo, verde y azul. En el caso de las imágenes en blanco y negro, V se trata de un subconjunto de \mathbb{R}^+ .

Las imágenes digitales pueden ser vectoriales o rasterizadas. Estas primeras son definidas en términos geométricos y pueden representarse en cualquier resolución, si se conocen sus ecuaciones matemáticas. Las curvas de Bézier y los B-splines son unas de las herramientas que se utilizan para definir las imágenes vectoriales. Sin embargo, estas suelen estar muy limitadas a la hora de representar formas o coloraciones complejas. Un ejemplo de gráfico vectorial son las tipografías.

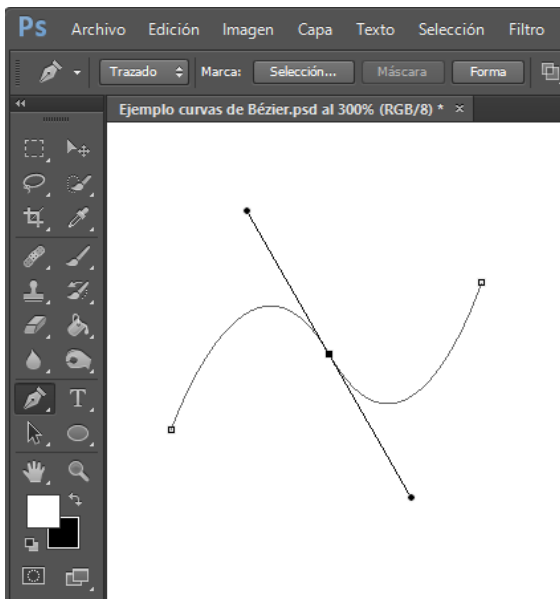
Las imágenes rasterizadas se corresponden a la discretización del dominio U , para poder ser computado por un ordenador a la hora de representar las imágenes. Como las pantallas tienen una resolución finita, todas las imágenes, son rasterizadas antes de poder mostrarse a través del monitor de un ordenador o de la pantalla de un teléfono móvil. Por ello es fundamental poder reescalar imágenes a mayores o menores resoluciones sin una pérdida de su calidad significativa y sin que se altere su reconocibilidad. Esta operación se conoce como interpolación de imagen. Dos de las tres técnicas más utilizadas en la práctica para interpolar imagen se basan en B-splines, la interpolación bilineal y la bicúbica, concretamente en superficies de grado $(1, 1)$ y $(3, 3)$ respectivamente.



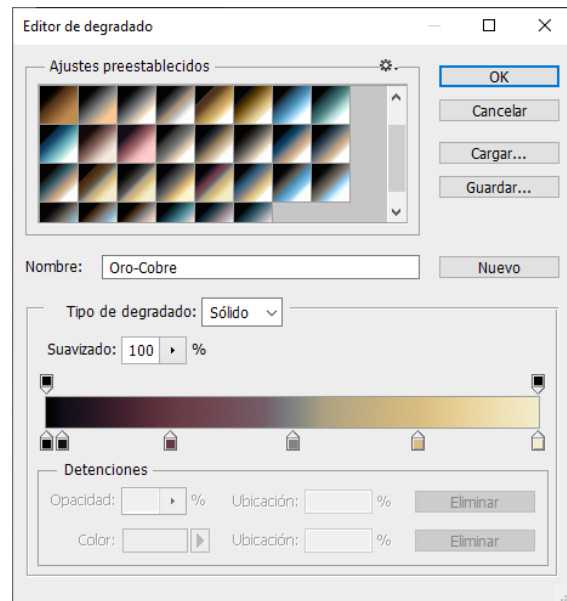
(a) Imagen original (500×500). (b) Bilineal (100×100). (c) Bicúbica (100×100).

Figura 6.1: Un fotograma de la película *Bladerunner* (1982) interpolado por splines lineales (interpolación bilineal) y por splines cúbicos (bicúbica). Hemos reducido la resolución de la imagen para que el efecto sea perceptible².

Los splines de Bézier se utilizan también en el contexto de la edición de imagen. Por ejemplo, en programas como *Photoshop* podemos encontrar la herramienta pluma de Bézier, que permite dibujar splines de Bézier con contactos G^1 . Y no solo eso, los gradientes o degradados de color también están basados en splines, mediante curvas definidas en el espacio RGB (figura 6.2).



(a) Herramienta pluma en *Photoshop*.



(b) Gradientes de color en *Photoshop*.

Figura 6.2: Algunos usos para curvas de Bézier en software de edición de imagen comercial.

²El programa utilizado para ello lo programé en Python en 3º de carrera para un trabajo de interpolación de la asignatura Métodos Numéricos.

§ 6.3. Esquemas de subdivisión de superficies

En el capítulo 5 vimos algunos de los algoritmos de subdivisión de curvas más sencillos y su convergencia, pero ninguno de superficies. El principal motivo es que los esquemas de subdivisión de superficies necesitan un desarrollo teórico tan extenso que no es adecuado para un Trabajo Fin de Grado. Posiblemente, el mejor contenido divulgativo sobre superficies de subdivisión lo podemos encontrar en el sitio web de *Pixar*, la compañía de animación, en [20]. Si por contra queremos estudiar los cimientos matemáticos de lo que nos cuentan, recomendamos ver [9, 10, 13].

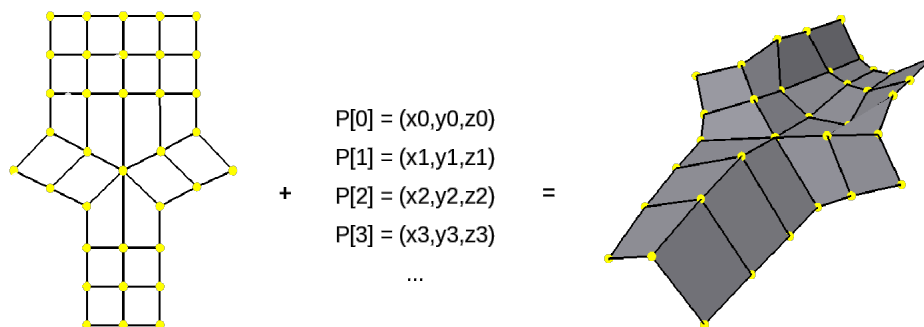


Figura 6.3: Una topología y unos puntos en el espacio determinan una superficie.

Si tomamos una malla formada por rectángulos en \mathbb{R}^2 y construimos encima nuestra malla de Bézier o de control, tendremos lo que se conoce como una topología rectangular. Si aplicamos la subdivisión de De Casteljaou o la uniforme, dependiendo de nuestra superficie, en las direcciones de los ejes, podremos construir superficies de subdivisión. Sin embargo, en la práctica se usan algoritmos como el de Catmull-Clark en [4], que permiten producir superficies límite más interesantes.

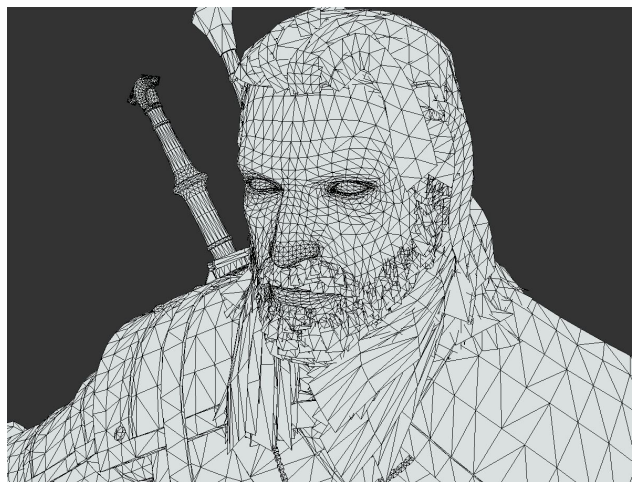


Figura 6.4: Modelo 3D de *Geralt de Rivia*, generado probablemente por *Loop Subdivision*.

Los modelados 3D suelen tener asociadas lo que se denominan topologías arbitrarias, donde se mezclan parches cuyas mallas son rectángulos, triángulos y otros polígonos. Por lo general las topologías son de Manifold esto es, 2-variedades topológicas, porque en caso contrario, la subdivisión da muchos problemas.

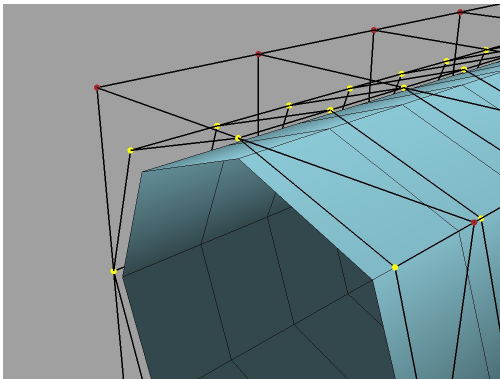
Pero no solo podemos encontrar modelos 3D en las películas de animación. Actualmente, los videojuegos implementan toda su geometría en tres dimensiones mediante superficies de subdivisión de topología triangular. Esto se debe a que las tarjetas gráficas de los ordenadores y de las consolas están especialmente diseñadas para computar triángulos, que son la figura geométrica más sencilla que existe.



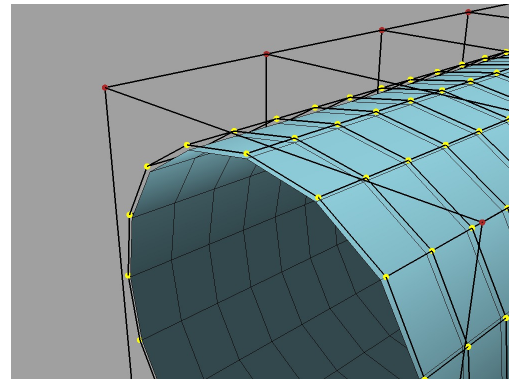
Figura 6.5: Modelo 3D de *Geralt de Rivia* aún más refinado y con texturas e iluminación.

La subdivisión de superficies, como sujeto de estudio de las matemáticas, es muy extenso y profundo. En él confluye el análisis numérico con el álgebra, la geometría y la topología. Esta última cobra especial importancia porque los esquemas deben trabajar con los puntos de la malla y sus valencias (número de incidencias), trazar nuevas aristas, unir parches (que pueden ser de diferente topología), representar agujeros, pliegues y un largo etcétera.

La teselación es otro concepto muy ligado a la subdivisión, y que no hemos comentado antes. El procedimiento por el cual se computa un número finito de puntos de la forma geométrica límite y se unen mediante figuras planas se denomina teselación, y es un proceso que puede hacer un ordenador directamente. Ambos procedimientos tienden a la misma forma límite.



(a) Teselación en la primera iteración.



(b) Teselación en la segunda iteración.

Figura 6.6: La subdivisión puede ser utilizada para teselar superficies, evaluando la forma geométrica únicamente en los puntos asociados a los puntos de la malla.

¿Entonces por qué se utiliza la subdivisión? Porque conforme refinamos la malla, aunque la superficie límite sea la misma, ganamos control sobre esta, al disponer de más puntos para modificarla a nuestro gusto. En la práctica, por tanto, suelen utilizarse ambas, tal y como vemos en la figura 6.6. Sin embargo, es importante conocer si el software de modelado implementa teselación o subdivisión a la hora de representar la forma final, porque cuando importamos un mismo modelo 3D en un programa o en otro, podemos obtener diferentes resultados, quizá para nuestro disgusto.

Conclusiones

Dedicaremos este apartado conclusivo para recapitular los objetivos logrados en este Trabajo Fin de Grado y para determinar qué frentes de conocimiento nos hemos abierto.

Como logros, podemos destacar:

- Se han estudiado las bases de los gráficos a ordenador desde dos puntos de vista complementarios, las curvas de Bézier y los B-splines. Si bien hemos hecho mayor hincapié en los segundos, hemos corroborado la necesidad práctica de los dos modelos, porque cada uno suple las carencias del otro. El trabajo comienza siendo un desarrollo matemático-histórico de análisis numérico clásico y evoluciona, una vez llegamos al aspecto gráfico, a un desarrollo más moderno haciendo uso de los polinomios simétricos.
- Hemos utilizado referencias académicas primarias siempre que ha sido posible. Véanse las referencias de De Boor o de Ramshaw ([5, 11, 12]) como ejemplo de publicaciones originales, o las de Kincaid y Hämmerlin ([7, 8]) como ejemplo de publicaciones derivadas. También hemos utilizado páginas web y vídeos de YouTube para documentar información más secundaria o para ejemplificar cuestiones mediante animaciones y recursos gráficos, respectivamente.
- La gran mayoría de los recursos gráficos del trabajo han sido realizados a mano, salvo los del capítulo último, que son propiedad de *Pixar* y de *CD Projekt RED*. Para ello, se ha efectuado un importante trabajo de aprendizaje del paquete TikZ de L^AT_EXy de *Wolfram Mathematica*, para poder elaborar gráficos de calidad y lograr un estilo propio y homogéneo.

Con unas bases matemáticas tan robustas como las vistas en este trabajo, estamos en posición de seguir profundizando por cualquiera de las siguientes ramas:

- Un estudio puramente numérico de la interpolación por splines y nuevos tipos de spline definidos a partir de las condiciones de frontera, como el de Hermite y Catmull-Rom.
- Un desarrollo de técnicas de Bézier más profundo, estudiando la derivada y la integral de una curva de Bézier, que al igual que los B-splines, vienen en términos de polinomios de Bernstein de grado un grado menor y mayor respectivamente. También podemos estudiar algoritmos de elevación de grado, inserción de nodos, y perseguir una conversión explícita entre representación de Bézier y B-spline. El libro de Paluszny, [9], es sin duda el mejor punto de partida, para el cual recomendamos tantear antes los blossomes con [11, 12].
- Estudiar cómo crear conexiones con la suavidad que deseemos y buscar algún estudio riguroso sobre las relaciones entre la continuidad paramétrica y la geométrica, su relación con el marco de Frenet y con el estudio más clásico de curvas y superficies. Recomendamos [9, 14].
- Introducirnos en las diferentes técnicas que conforman el problema, a día de hoy todavía abierto, de la subdivisión de superficies, para lograr un mayor entendimiento de la complejidad que hay detrás de los gráficos que vemos en nuestras películas y videojuegos favoritos. Para ello, contamos con numerosos libros como [1, 9, 10] o la tesis [13].

Bibliografía

- [1] L. ANDERSSON Y N. F. STEWART: *Introduction to the mathematics of subdivision surfaces*, Society for Industrial and Applied Mathematics, 2010.
- [2] R. L. BURDEN, D. J. FAIRES Y A. M. BURDEN: *Análisis numérico*, Cengage Learning, 2017.
- [3] A. BURGOS: *Esquemas de subdivisión y animación gráfica*. Trabajo Fin de Grado, Facultad de ciencia y tecnología, Universidad de La Rioja, 2021.
- [4] N. CARTER: *Introduction to the Mathematics of Computer Graphics*, American Mathematical Society, 2016.
- [5] C. DE BOOR: On Calculating With B-splines, *Journal of Approximation Theory* **6** (1972), 50–62.
- [6] W. GAUTSCHI: *Numerical Analysis*, Birkhäuser, 2012.
- [7] G. HÄMMERLIN Y K. HOFFMAN: *Numerical Mathematics*, Springer-Verlag, 1991.
- [8] D. KINCAID Y W. CHENEY: *Análisis numérico*, Addison-Wesley Iberoamericana, 1994.
- [9] M. PALUSZNY, B. WOLFGANG Y P. HARTMUT: *Métodos de Bézier y B-splines*, Universitätsverlag Karlsruhe, 2005.
- [10] J. PETERS Y U. REIF: *Subdivision Surfaces*, Springer, 2008.
- [11] L. RAMSHAW: *Blossoming: A Connect-the-Dots Approach to Splines*, Digital Equipment Corporation, 1987.
- [12] L. RAMSHAW: *Blossoms Are Polar Forms*, Digital Equipment Corporation, 1989.
- [13] B. RIECK: *Smoothness Analysis Of Subdivision Algorithms*. Tesis Doctoral, Ruprecht-Karls-Universität Heidelberg, 2010.
- [14] R. C. VELTKAMP: *Survey of Continuities of Curves and Surfaces*, Stichting Mathematisch Centrum, 1992.

Direcciones electrónicas

- [15] CANAL DE YOUTUBE DE FREYA HOLMÉR, *The Continuity of Splines*, 2022.
- [16] CANAL DE YOUTUBE DE FREYA HOLMÉR, *The Beauty of Bézier Curves*, 2022.
- [17] CANAL DE YOUTUBE DE JBENGHIALTD, *NURBS Part 1: An introduction to NURBS*, 2020.
- [18] DISCUSIÓN EN RESEARCHGATE, *Who first used B-spline?*, última vez visitada: 28/06/2022.
- [19] NLAB, *Partition of unity*, última vez visitada: 28/06/2022.
- [20] PÁGINA WEB DE PIXAR, *Subdivision Surfaces*, última vez visitada: 28/06/2022.
- [21] WIKIPEDIA, LA ENCICLOPEDIA LIBRE, *Matriz diagonal dominante*, última vez visitada: 28/06/2022.
- [22] WIKIPEDIA, LA ENCICLOPEDIA LIBRE, *Non-Uniform Rational B-splines*, última vez visitada: 28/06/2022.
- [23] WIKIPEDIA, LA ENCICLOPEDIA LIBRE, *Cubic Hermite spline*, última vez visitada: 28/06/2022.