

Tema 2: Curvas de Bézier

Leonardo Fernández Jambrina
E.T.S.I. Navales, Universidad Politécnica de Madrid
Avenida de la Memoria 4
E-28040-Madrid, España
Correo-e: leonardo.fernandez@upm.es

1. Introducción

Para el diseño asistido por ordenador es conveniente emplear representaciones sencillas de curvas y superficies, que involucren operaciones elementales, como sumas y multiplicaciones. Por tanto, el candidato más razonable a primera vista son las parametrizaciones polinómicas. Así podríamos representar curvas polinómicas de grado n como

$$c(t) = a_0 + a_1t + \cdots + a_nt^n, \quad t \in [0, 1],$$

donde cada coeficiente a_i es un punto del plano o del espacio, según que la curva sea plana o espacial.

Por ejemplo, la parábola de coeficientes $a_0 = (0, 0)$, $a_1 = (0, 1)$, $a_2 = (1, 1)$ está parametrizada por $c(t) = (t^2, t + t^2)$.

[Vídeo ¿curvas polinómicas?](#)

Esta representación tiene la ventaja de su sencillez. Sin embargo, no es muy práctica. Para empezar, la interpretación de los coeficientes hace referencia a los valores que toma la curva en las proximidades del punto inicial $c(0) = a_0$, ya que esencialmente son las derivadas de la parametrización en $t = 0$,

$$a_i = \frac{c^{(i)}(0)}{i!},$$

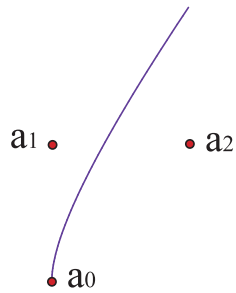


Figura 1: Parábola de coeficientes a_0, a_1, a_2

con lo cual no nos dan una idea clara del comportamiento global de la curva.

Por otra parte, si, como es frecuente en el diseño, queremos observar la curva desde otro punto de vista, rotándola, trasladándola, deformándola, transformándola en definitiva por una aplicación afín, el comportamiento de los coeficientes es de lo más variopinto. [Ejemplo](#).

Si realizamos una traslación por un vector \mathbf{v} , el coeficiente a_0 se ve trasladado por el mismo vector, mientras que el resto de coeficientes no se ve alterado.

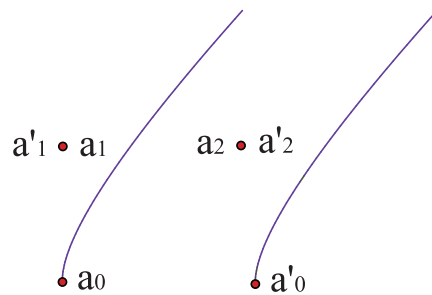


Figura 2: Parábola trasladada

En cambio, si realizamos una rotación con centro en a_0 , ocurre justo lo contrario, todos los coeficientes, excepto el a_0 , experimentan la rotación.

Como vemos, el comportamiento de los coeficientes bajo aplicaciones afines es complejo, debido a que, de todos los coeficientes, sólo a_0 es un punto, el origen de la curva, mientras que el resto son vectores, puesto que son derivadas de la parametrización.

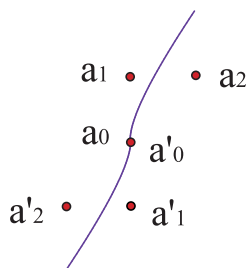


Figura 3: Parábola rotada

Además, esta representación no es muy estable frente a pequeños cambios. Imaginemos que tenemos una parametrización con sólo coeficientes a_0, a_1 no nulos, es decir, la correspondiente a un segmento de una recta. Cualquier pequeña fluctuación, por ejemplo la debida al hecho de que no trabajamos con aritmética exacta, daría lugar a coeficientes a_i pequeños, pero no nulos. Y una curva de grado n es cualitativamente muy distinta de una recta. [Ejemplo](#).



Figura 4: El coeficiente no nulo a_2 convierte el segmento en una parábola

Por todo ello, parece conveniente emplear una representación distinta de las curvas polinómicas.

2. Curvas de Bézier

[Vídeo de Curvas de Bézier](#)

Una base distinta para los polinomios de grado n o inferior, distinta de la canónica $\{1, t, \dots, t^n\}$, nos la proporcionan los polinomios de Bernstein, los mismos que se emplean en teoría de la aproximación para demostrar el Teorema de Weierstrass de aproximación uniforme de funciones continuas por polinomios. Su construcción es muy sencilla a partir de la fórmula del

binomio de Newton,

$$(a + b)^n = \sum_{i=0}^n \binom{n}{i} a^i b^{n-i}, \quad \binom{n}{i} = \frac{n!}{i!(n-i)!}.$$

Si tomamos $a = t$, $b = 1 - t$ en la expresión anterior, obtenemos

$$1 = (t + 1 - t)^n = \sum_{i=0}^n B_i^n(t), \quad B_i^n(t) := \binom{n}{i} t^i (1 - t)^{n-i}, \quad (1)$$

donde $B_i^n(t)$ es el polinomio i -ésimo de Bernstein de grado n .

Por ejemplo los polinomios de Bernstein de grado dos son

$$B_0^2(t) = (1 - t)^2, \quad B_1^2(t) = 2t(1 - t), \quad B_2^2(t) = t^2.$$

Estos polinomios forman una base alternativa $\{B_0^n(t), \dots, B_n^n(t)\}$ de los polinomios de grado n o inferior en una variable t y, frente a la base canónica, presentan la ventaja de ser todos del mismo grado.

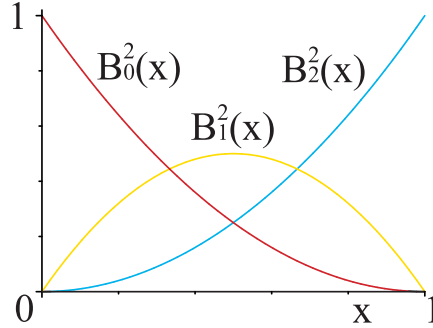


Figura 5: Polinomios de Bernstein de grado dos

La demostración de que forman base es sencilla. Echemos una ojeada al polinomio n -ésimo, $B_n^n(t) = t^n$. Coincide con el último polinomio de la base canónica. El anterior, $B_{n-1}^n(t) = n(t^{n-1} - t^n)$, involucra tan sólo dos polinomios de la base canónica. Y así sucesivamente el polinomio $B_i^n(t)$, por el término $(1 - t)^{n-i}$, tiene coeficientes no nulos sólo para los monomios t^i, \dots, t^n , por lo cual la matriz de cambio de base es triangular superior,

$$\begin{pmatrix} B_0^n(t) \\ \vdots \\ B_n^n(t) \end{pmatrix} = \begin{pmatrix} 1 & \cdots & (-1)^n \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ t^n \end{pmatrix},$$

y, por ello, su determinante es el producto de los términos de la diagonal principal, que corresponden al coeficiente del monomio de menor grado de cada polinomio $B_i^n(t)$, que es no nulo, $\binom{n}{i}$. Con lo cual, el determinante es no nulo y los polinomios de Bernstein forman base de los polinomios de grado n o inferior. \square

Podremos representar, pues, las curvas polinómicas de grado n como combinación de estos polinomios

$$c(t) = \sum_{i=0}^n c_i B_i^n(t), \quad t \in [0, 1], \quad (2)$$

donde todos los coeficientes c_i son puntos del plano o del espacio afín, según que la curva sea plana o espacial. A estos coeficientes los denominaremos **vértices del polígono de control**, $\{c_0, \dots, c_n\}$, de la curva de Bézier $c(t)$. Una curva de grado n tiene, pues, un polígono de control de $n + 1$ vértices.

Por ejemplo, una parábola con polígono de control dado por $c_0 = (0, 0)$, $c_1 = (1, 0)$, $c_2 = (1, 1)$ está parametrizada por $c(t) = (2t - t^2, t^2)$,

$$\begin{aligned} c(t) &= c_0 B_0^2(t) + c_1 B_1^2(t) + c_2 B_2^2(t) \\ &= (0, 0)(1 - t)^2 + (1, 0)2t(1 - t) + (1, 1)t^2 = (2t - t^2, t^2), \end{aligned}$$

para $t \in [0, 1]$.

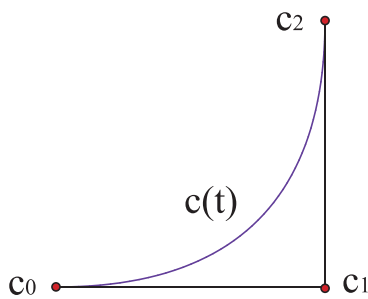


Figura 6: Parábola de polígono de control $\{c_0, c_1, c_2\}$

3. Propiedades de las curvas de Bézier

Aunque hayamos definido la curva en el intervalo $[0, 1]$, es posible utilizar otros intervalos. Sólo hace falta transformar el intervalo por una sencilla aplicación afín para que la parametrización esté definida en el intervalo $[a, b]$,

$$t(u) = \frac{u - a}{b - a}, \quad u \in [a, b],$$

de modo que $u = a$ corresponde a $t = 0$ y $u = b$, a $t = 1$. La nueva parametrización será, pues,

$$\tilde{c}(u) = c(t(u)) := c\left(\frac{u - a}{b - a}\right), \quad u \in [a, b].$$

Nótese que la parametrización $c(t)$ no ve el intervalo $[a, b]$, sino el $[0, 1]$. Es sólo un ajuste del usuario. [Ejemplo](#).

Dos de los vértices del polígono de control tienen una interpretación inmediata. A la vista de la expresión de los polinomios de Bernstein, (1), resulta que $B_i^n(0) = 0$, sea cual sea el grado n , salvo para $i = 0$, para el cual toma el valor $B_0^n(0) = 1$. Del mismo modo, $B_i^n(1) = 0$, salvo para $i = n$, que no tiene términos $(1 - t)$, por lo cual $B_n^n(1) = 1$.

Así pues, la curva pasa por los vértices c_0, c_n ,

$$c(0) = \sum_{i=0}^n c_i B_i^n(0) = c_0, \quad c(1) = \sum_{i=0}^n c_i B_i^n(1) = c_n. \quad \square$$

De hecho, son los únicos vértices del polígono por los que pasa la curva. [Ejemplo](#). Veremos más adelante que el resto de vértices están relacionados con las derivadas sucesivas de la parametrización.

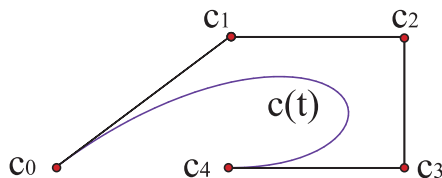


Figura 7: Una curva de Bézier siempre pasa por los vértices primero y último

El nombre de polígono de control hace referencia a que sirve para controlar la forma de la curva. Sin embargo, ese control no es local, ya que, desplazando un vértice, se mueve toda la curva, aunque principalmente la parte más próxima al vértice en cuestión. **Ejemplo.** Esto se debe a que el máximo del polinomio i -ésimo de Bernstein está en $t = i/n$,

$$0 = \frac{dB_i^n(t)}{dt} = \frac{n!}{i!(n-i)!} t^{i-1} (1-t)^{n-i-1} \{i(1-t) - (n-i)t\} \Rightarrow nt = i,$$

con lo cual la variación del vértice i afecta más a los valores de $c(t)$ en las proximidades del máximo. \square

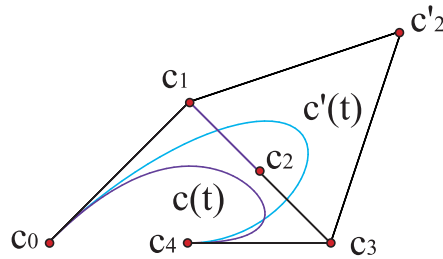


Figura 8: Control local: al mover el vértice c_2 , se deforma mayormente la parte más próxima a él

Esta propiedad mejorará sustancialmente cuando empleemos curvas polinómicas a trozos.

Una propiedad importante de la representación de Bézier, que es inmediata a partir de la fórmula (1), es que la suma de los polinomios de Bernstein de grado n es la unidad. Por tanto, la expresión (2) de la parametrización de la curva es una combinación baricéntrica de los vértices del polígono de control, tal como definimos en el capítulo anterior.

Esta propiedad se traduce en muchas otras importantes. Por ejemplo, la curva está contenida en la envolvente convexa del polígono de control. Así pues, como su nombre indica, el polígono nos proporciona una primera idea de por donde pasa la curva, lo cual puede ser muy útil, por ejemplo, para saber si dos curvas se cortan o no: si las envolventes de los polígonos no se cortan, las curvas no se cortan. **Ejemplo.**

Además, el hecho de que la expresión de la curva de Bézier sea una combinación baricéntrica facilita la transformación de la curva por una aplicación

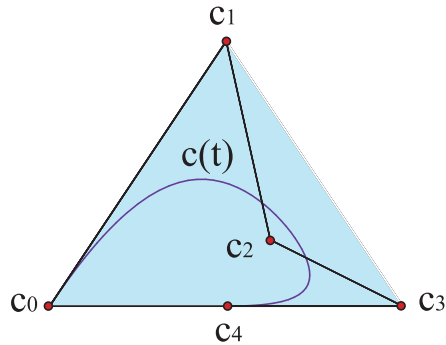


Figura 9: La curva está contenida dentro de la envolvente convexa de su polígono de control

afín f , ya que

$$f(c(t)) = f\left(\sum_{i=0}^n c_i B_i^n(t)\right) = \sum_{i=0}^n f(c_i) B_i^n(t),$$

es decir, la curva imagen tiene por polígono de control la imagen del polígono primitivo, $\{f(c_0), \dots, f(c_n)\}$. Por tanto, no es preciso calcular la imagen de cada punto para construir la curva imagen, sino sólo la imagen del polígono de control y recalculer la curva. [Ejemplo](#).

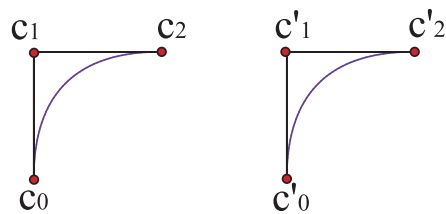


Figura 10: Para trasladar una curva de Bézier, basta trasladar su polígono de control

Finalmente, una propiedad sencilla, pero importante, de las curvas de Bézier es su **simetría**. Si invertimos el polígono de control, $\{c_n, \dots, c_0\}$, la gráfica de la curva es la misma que la correspondiente a $\{c_0, \dots, c_n\}$, sólo

que es recorrida en sentido inverso, de c_n en $t = 0$ a c_0 en $t = 1$. [Ejemplo](#). El motivo está en la propiedad de simetría de los números combinatorios,

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} = \binom{n}{n-i},$$

que implica la relación entre los polinomios de Bernstein, $B_i^n(1-t) = B_{n-i}^n(t)$, y la simetría de la parametrización de Bézier,

$$c(1-t) = \sum_{i=0}^n c_i B_i^n(1-t) = \sum_{i=0}^n c_i B_{n-i}^n(t) = \sum_{j=0}^n c_{n-j} B_j^n(t), \quad (3)$$

simplemente tomando $j = n - i$. \square

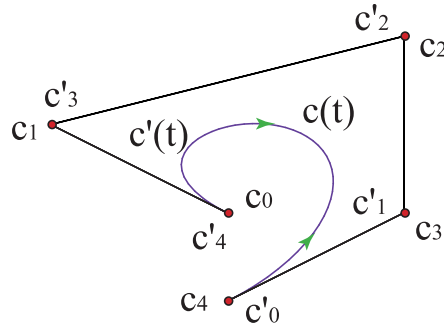


Figura 11: Para invertir el sentido de una curva de Bézier, basta invertir el orden de su polígono de control

4. Algoritmo de De Casteljaou

[Vídeo del Algoritmo de de Casteljaou](#)

Los polinomios de Bernstein son útiles para deducir las propiedades fundamentales de las curvas de Bézier, pero no proporcionan una manera eficiente de construirlas. La manera tradicional de trazarlas está basada en el algoritmo de De Casteljaou. Antes de introducirlo, recordemos la interpolación en $t \in [0, 1]$ entre dos puntos a, b ,

$$(1-t)a + tb,$$

de modo que para $t = 0$ obtenemos el primer punto y para $t = 1$, el segundo.

El algoritmo de De Casteljaou consiste simplemente en la aplicación reiterada de la interpolación al polígono de control de la curva de Bézier,

$$c_i^1(t) := (1 - t)c_i + tc_{i+1}, \quad i = 0, \dots, n - 1, \quad (4)$$

de modo que en cada paso disminuye el número de vértices en una unidad,

$$c_i^r(t) := (1 - t)c_i^{r-1}(t) + tc_{i+1}^{r-1}(t), \quad i = 0, \dots, n - r, \quad r = 1, \dots, n, \quad (5)$$

hasta llegar a un único punto en la iteración n -ésima. [Ejemplo](#). Dicho punto es precisamente el valor de la parametrización de la curva de Bézier en t ,

$$c(t) = c_0^n(t). \quad (6)$$

Veámoslo con un ejemplo. Consideremos el polígono de control de una parábola, $\{c_0 = (0, 0), c_1 = (1, 2), c_2 = (2, -1)\}$. Aplicar el algoritmo de De Casteljaou supone dos iteraciones:

$$\begin{aligned} c_0^1(t) &= (1 - t)c_0 + tc_1 = (t, 2t), & c_1^1(t) &= (1 - t)c_1 + tc_2 = (1 + t, 2 - 3t) \\ c_0^2(t) &= (1 - t)c_0^1(t) + tc_1^1(t) = (2t, 4t - 5t^2), \end{aligned}$$

que obviamente coincide con el valor que proporciona la construcción con polinomios de Bernstein, $c(t) = (2t, 4t - 5t^2)$.

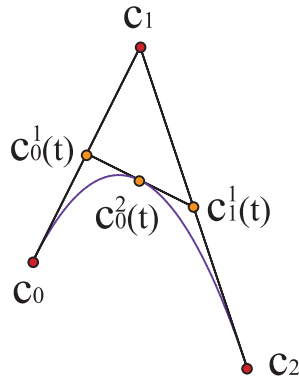


Figura 12: Algoritmo de De Casteljaou para una parábola

La manera de demostrar la equivalencia con la construcción por polinomios de Bernstein es por mera recursividad. Realmente demostraremos una fórmula más general,

$$c(t) = \sum_{i=0}^{n-r} c_i^{(r)}(t) B_i^{n-r}(t), \quad (7)$$

cuyo caso $r = 0$ es trivial. Supongamos que es cierta para $r - 1$,

$$c(t) = \sum_{i=0}^{n+1-r} c_i^{(r-1)}(t) B_i^{n+1-r}(t),$$

para demostrar la veracidad de (7) por inducción, sustituimos (5),

$$\begin{aligned} \sum_{i=0}^{n-r} c_i^{(r)}(t) B_i^{n-r}(t) &= \sum_{i=0}^{n-r} \left\{ (1-t)c_i^{(r-1)}(t) + t c_{i+1}^{(r-1)}(t) \right\} B_i^{n-r}(t) = \\ &= \sum_{i=0}^{n-r} c_i^{(r-1)}(t) (1-t) B_i^{n-r}(t) + \sum_{i=1}^{n+1-r} c_i^{(r-1)}(t) t B_{i-1}^{n-r}(t) = \\ &= \sum_{i=0}^{n+1-r} c_i^{(r-1)}(t) B_i^{n+1-r}(t) = c(t), \quad \square \end{aligned}$$

teniendo en cuenta la sencilla relación de recurrencia existente entre los polinomios de Bernstein,

$$(1-t)B_i^m(t) + tB_{i-1}^m(t) = B_i^{m+1}(t),$$

como se comprueba fácilmente.

Nótese que la fórmula (7) expresa que en el algoritmo de De Casteljaou se pueden suplir $n - r$ interpolaciones por los polinomios de Bernstein de grado $n - r$.

Las ventajas del algoritmo de De Casteljaou son su sencillez y el hecho de que involucra tan sólo sumas, ya que todos los términos son positivos, lo cual le confiere robustez a la hora de realizar cálculos con aritmética de coma flotante. [Ejemplo de parábola.](#) [Ejemplo de cúbica.](#)

5. Forma polar de una parametrización

Una construcción muy útil basada en el algoritmo de De Casteljau es el *blossom*, que denominaremos **forma polar** de la parametrización, debido a que está relacionado con esta del mismo modo que una forma cuadrática con su forma bilineal asociada, también llamada forma polar.

La idea es la siguiente. En vez de interpolar siempre para el mismo valor de t en los n pasos del algoritmo, interpolamos en el valor t_1 en el primer paso, en el valor t_2 en el segundo, \dots , y en el valor t_n en el último. Obtenemos así un punto que denotaremos $c[t_1, \dots, t_n]$. No se corresponde con ningún punto de la curva, salvo en el caso trivial en el que todos los t_i son iguales.

$$\begin{aligned} c_i^1[t_1] &:= c_i^1(t_1) = (1 - t_1)c_i + t_1c_{i+1}, \quad i = 0, \dots, n - 1, \\ c_i^r[t_1, \dots, t_r] &:= (1 - t_r)c_i^{r-1}[t_1, \dots, t_{r-1}] + t_rc_{i+1}^{r-1}[t_1, \dots, t_{r-1}], \\ c[t_1, \dots, t_n] &:= c_0^n[t_1, \dots, t_n], \quad i = 0, \dots, n - r, \quad r = 1, \dots, n. \end{aligned} \quad (8)$$

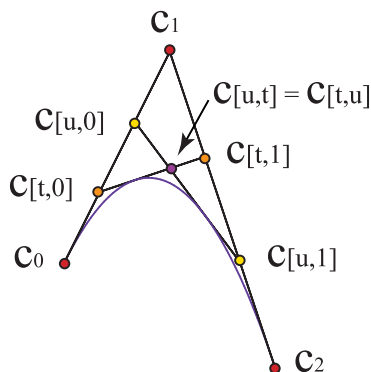


Figura 13: La forma polar es simétrica $c[t, u] = c[u, t]$.

Veámoslo con un ejemplo sencillo, el caso $n = 2$, para el polígono de control $\{c_0, c_1, c_2\}$:

$$c_0^1[t_1] = (1 - t_1)c_0 + t_1c_1, \quad c_1^1[t_1] = (1 - t_1)c_1 + t_1c_2,$$

$$\begin{aligned} c[t_1, t_2] &:= (1 - t_2)c_0^1[t_1] + t_2c_1^1[t_1] \\ &= (1 - t_1)(1 - t_2)c_0 + \{t_1(1 - t_2) + t_2(1 - t_1)\}c_1 + t_1t_2c_2. \end{aligned}$$

A la vista de la expresión anterior, deducimos que la forma polar es simétrica, es decir, $c[t_1, t_2] = c[t_2, t_1]$. No importa el orden de las interpolaciones, sólo los valores de t . Este resultado se generaliza a curvas de grado n haciendo uso del Teorema de Menelao. \square [Ejemplo](#).

Una propiedad importante de la forma polar es que permite reconstruir el polígono de la curva de Bézier, a diferencia de la parametrización, que sólo proporciona los vértices inicial y final. Lo vemos de manera sencilla interpolando en $t = 0$ y en $t = 1$ un polígono de control $\{c_0, \dots, c_n\}$

$$c_i^{(1)}(0) = c_i, \quad c_i^{(1)}(1) = c_{i+1}, \quad i = 0, \dots, n-1,$$

con lo cual en $t = 0$ el polígono interpolado es $\{c_0, \dots, c_{n-1}\}$ y en $t = 1$, $\{c_1, \dots, c_n\}$. Es decir, el efecto de interpolar en $t = 0$ es eliminar el último de los vértices y el de $t = 1$, eliminar el primero. Así pues, es trivial comprobar que, interpolando i veces en $t = 1$ y $n - i$ veces en $t = 0$, recuperamos el vértice i del polígono de control:

$$c[0^{<n-i>}, 1^{<i>}] = c_i, \quad a^{<s>} = \underbrace{a, \dots, a}_{s \text{ veces}}. \quad (9)$$

Este hecho puede parecer una trivialidad, ya que para calcular el polígono de control hemos de emplear la forma polar, que requiere a su vez el polígono de control. Sin embargo, es útil para obtener polígonos de control de curvas que se obtienen a partir de una dada, de la cual sí conocemos el polígono.

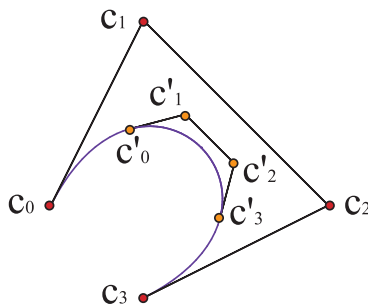


Figura 14: La forma polar nos permite obtener el polígono de una parte de una curva de Bézier

Por ejemplo, es útil para restringir el intervalo de definición de una curva de Bézier, $c(t)$, definida en el intervalo $[0, 1]$. Conocemos el polígono de control

$\{c_0, \dots, c_n\}$ de la curva primitiva, pero no el de su restricción, \tilde{c} , al intervalo $[a, b]$. **Ejemplo.** La parametrización de la curva \tilde{c} es fácil de obtener teniendo en cuenta que la relación entre el parámetro $u \in [a, b]$ y el parámetro $\tilde{t} \in [0, 1]$ de la curva trazada desde $c(a)$ hasta $c(b)$ es $u(\tilde{t}) = a(1 - \tilde{t}) + b\tilde{t}$. Así pues, $\tilde{c}(\tilde{t}) = c(u(\tilde{t}))$ y la forma polar, a su vez será

$$\tilde{c}[\tilde{t}_1, \dots, \tilde{t}_n] = c[a(1 - \tilde{t}_1) + b\tilde{t}_1, \dots, a(1 - \tilde{t}_n) + b\tilde{t}_n],$$

de donde inferimos que los vértices del polígono de control de la curva restringida serán

$$\tilde{c}_i = \tilde{c}[0^{\langle n-i \rangle}, 1^{\langle i \rangle}] = c[a^{\langle n-i \rangle}, b^{\langle i \rangle}]. \quad (10)$$

Este resultado es muy potente, ya que permite calcular el polígono de control de cualquier tramo de una curva de Bézier. Nótese que la curva final está definida de nuevo en el intervalo $[0, 1]$. **Ejemplo.**

Un ejemplo sencillo lo construimos a partir de la parábola de polígono de control $\{(0, 0), (1, 0), (0, 1)\}$. El polígono correspondiente a la parte de la curva definida en el intervalo $[0, 1/3]$ es $\{(0, 0), (1/3, 0), (4/9, 1/9)\}$,

$$\tilde{c}_0 = c[0, 0] = c_0 = (0, 0), \quad \tilde{c}_1 = c[0, 1/3] = \frac{2c_0 + c_1}{3} = (1/3, 0),$$

$$\tilde{c}_2 = c[1/3, 1/3] = \frac{4c_0 + 4c_1 + c_2}{9} = (4/9, 1/9).$$

Podría utilizarse esta expresión para extrapolar la curva fuera del intervalo $[0, 1]$ en lugar de restringirla, pero entonces surgirían los problemas de la aritmética de coma flotante, ya que aparecerían valores negativos en las interpolaciones.

La forma polar es una forma multiafín en el siguiente sentido. Si $\lambda + \mu = 1$, combinación baricéntrica de números, entonces

$$\begin{aligned} c[t_1, \dots, t_{i-1}, \lambda t_i + \mu s_i, t_{i+1}, \dots, t_n] &= \lambda c[t_1, \dots, t_{i-1}, t_i, t_{i+1}, \dots, t_n] \\ &+ \mu c[t_1, \dots, t_{i-1}, s_i, t_{i+1}, \dots, t_n] \end{aligned} \quad (11)$$

es decir, es afín en cada una de las variables.

La demostración es sencilla. Consideremos la interpolación i -ésima entre dos puntos a, b para dar el punto

$$(1 - (\lambda t_i + \mu s_i))a + (\lambda t_i + \mu s_i)b = \lambda((1 - t_i)a + bt_i) + \mu((1 - s_i)a + bs_i),$$

ya que $1 = \lambda + \mu$. \square

6. Elevación del grado

Vídeo de Derivación y elevación del grado

En algunos casos, nos puede interesar cambiar el grado de la curva para tener más grados de libertad a la hora de modificarla. Esto era muy fácil de realizar en la base canónica, bastaba añadir un término $0t^{n+1}$ y la parametrización formalmente sería de grado $n + 1$. La representación de Bézier también permite elevar el grado de manera sencilla. Basta multiplicar la parametrización por $1 = (1 - t + t)$,

$$\begin{aligned}
 c(t) &= \sum_{i=0}^n c_i B_i^n(t) = \sum_{i=0}^n c_i \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} (1-t+t) \\
 &= \sum_{i=0}^n c_i \frac{n!}{i!(n-i)!} (t^{i+1} (1-t)^{n-i} + t^i (1-t)^{n+1-i}) \\
 &= \sum_{i=0}^n c_i \left(\frac{i+1}{n+1} B_{i+1}^{n+1}(t) + \frac{n+1-i}{n+1} B_i^{n+1}(t) \right) \\
 &= \sum_{i=0}^{n+1} \left(\frac{i}{n+1} c_{i-1} + \frac{n+1-i}{n+1} c_i \right) B_i^{n+1}(t) = \sum_{i=0}^{n+1} c_i^1 B_i^{n+1}(t),
 \end{aligned}$$

e identificar los nuevos vértices, $\{c_0^1, \dots, c_{n+1}^1\}$,

$$c_i^1 = \left(1 - \frac{i}{n+1}\right) c_i + \frac{i}{n+1} c_{i-1}. \quad \square \tag{12}$$

Nótese que el vértice c_i^1 es una combinación baricéntrica de los vértices c_i, c_{i+1} con coeficientes positivos, luego está situado sobre el segmento $\overline{c_i c_{i+1}}$. Claramente $c_0^1 = c_0, c_{n+1}^1 = c_n$, como corresponde al hecho de que las curvas deben tener los mismos extremos. **Ejemplo. Ejemplo.**

Por ejemplo la curva cúbica de polígono $\{(0, 0), (1, 2), (2, 3), (4, 1)\}$ se puede expresar como una cuártica cuyo polígono de control está constituido por los vértices de coordenadas $\{(0, 0), (3/4, 3/2), (3/2, 5/2), (5/2, 5/2), (4, 1)\}$,

$$\begin{aligned}
 c_0^1 &= \left(1 - \frac{0}{4}\right) c_0 + \frac{0}{4} c_{-1} = c_0 = (0, 0), \\
 c_1^1 &= \left(1 - \frac{1}{4}\right) c_1 + \frac{1}{4} c_0 = \frac{3}{4}(1, 2) + \frac{1}{4}(0, 0) = \left(\frac{3}{4}, \frac{3}{2}\right),
 \end{aligned}$$

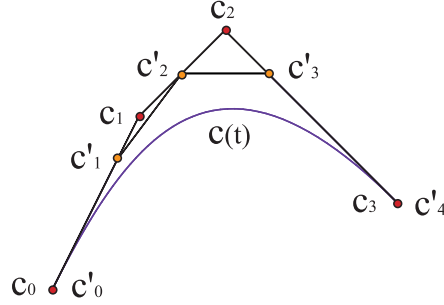


Figura 15: Los polígonos $\{c_0, \dots, c_3\}$, $\{c'_0, \dots, c'_4\}$ corresponden a la misma curva de Bézier

$$\begin{aligned} c_2^1 &= \left(1 - \frac{2}{4}\right) c_2 + \frac{2}{4} c_1 = \frac{1}{2}(2, 3) + \frac{1}{2}(1, 2) = \left(\frac{3}{2}, \frac{5}{2}\right), \\ c_3^1 &= \left(1 - \frac{3}{4}\right) c_3 + \frac{3}{4} c_2 = \frac{1}{4}(4, 1) + \frac{3}{4}(2, 3) = \left(\frac{5}{2}, \frac{5}{2}\right), \\ c_4^1 &= \left(1 - \frac{4}{4}\right) c_4 + \frac{4}{4} c_3 = c_3 = (4, 1). \end{aligned}$$

Este mismo procedimiento se puede emplear para expresar el polinomio lineal t formalmente como un polinomio de cualquier grado n , ya que $t = t(1 - t + t)^{n-1}$. Podemos pensar en los polinomios como “curvas” sobre la recta real, con vértices de una única coordenada. Así el polinomio t tiene por único polígono $\{c_0 = 0, c_1 = 1\}$, puesto que

$$c(t) = c_0 B_0^1(t) + c_1 B_1^1(t) = c_0(1 - t) + c_1 t = t.$$

Y si lo expresamos formalmente como un polinomio de grado n ,

$$\begin{aligned} t &= t(1 - t + t)^{n-1} = \sum_{i=0}^{n-1} \frac{(n-1)!}{i!(n-i-1)!} t^{i+1} (1-t)^{n-i-1} \\ &= \sum_{i=1}^n \frac{(n-1)!}{(i-1)!(n-i)!} t^i (1-t)^{n-i} = \sum_{i=1}^n \frac{i}{n} B_i^n(t) = \sum_{i=1}^n c_i^n B_i^n(t), \end{aligned}$$

podemos leer los vértices del polígono de control de grado n ,

$$c_i^n = \frac{i}{n}. \quad (13)$$

Esta propiedad se conoce como **precisión lineal** y no es más que una aplicación reiterada del algoritmo de elevación del grado. Su utilidad es clara cuando queremos expresar, no curvas paramétricas, de la forma $(c_x(t), c_y(t))$, sino funciones polinómicas $f(t)$ de grado n . Podemos ver estas últimas dentro del formalismo general viendo sus gráficas como curvas paramétricas de la forma $(t, f(t))$.

Al expresar el monomio t como “curva” de Bézier de grado formal n , podremos representar $c(t) = (t, f(t))$ por medio de un polígono de control $\{c_0, \dots, c_n\}$ de la forma $c_i = (i/n, c_{i,y})$. Es decir, los vértices están equiespaciados en su abscisa.

Como ejemplo, vamos a representar la función cúbica $f(x) = x^3$ en el intervalo $[0, 1]$. Como $f(x) = B_3^3(x)$, todas las ordenadas de los vértices del polígono de control, $\{c_0, \dots, c_3\}$ son nulas, menos $c_{3,y}$, que vale uno. Por tanto,

$$c_0 = (0, 0), \quad c_1 = (1/3, 0), \quad c_2 = (2/3, 0), \quad c_3 = (1, 1).$$

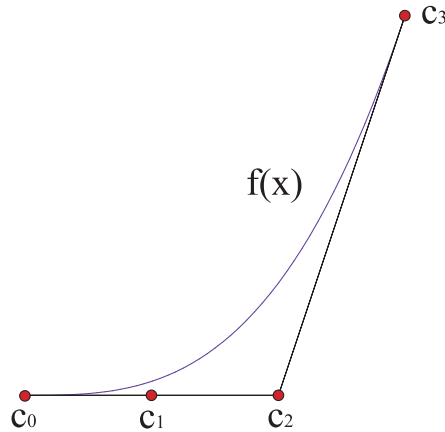


Figura 16: La gráfica de $f(x) = x^3$ vista como curva de Bézier

El algoritmo de elevación del grado se puede expresar de una manera elegante por medio de la forma polar:

$$c^1[t_1, \dots, t_{n+1}] = \frac{1}{n+1} \sum_{i=1}^{n+1} c[t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_{n+1}], \quad (14)$$

Este resultado expresa que la forma polar de grado $n + 1$ es la media de la forma polar de grado n evaluada en todos los valores t_1, \dots, t_{n+1} , eliminando uno de cada vez.

Como podemos recuperar $c^1[t_1, \dots, t_{n+1}]$ a partir de combinaciones baricéntricas de $c_i^1 = c^1[0^{<n+1-i>}, 1^{<i>}]$, basta que demos­tre­mos la expresión anterior para los vértices del polígono de control. Teniendo en cuenta que en la expresión de c_i^1 aparece $t = 1$ i veces, deberá aparecer $1^{<i-1>}$ en la suma i veces también. Del mismo modo, $0^{<n-i>}$ deberá aparecer $n + 1 - i$ veces,

$$\begin{aligned} & \frac{1}{n+1} \{ i c^1[0^{<n+1-i>}, 1^{<i-1>}] + (n+1-i) c^1[0^{<n-i>}, 1^{<i>}] \} \\ &= \frac{i}{n+1} c_{i-1}^1 + \frac{n+1-i}{n+1} c_i^1 = c_i^1, \end{aligned}$$

con lo cual comprobamos que la fórmula es correcta, ya que coincide con la deducida previamente. \square

La expresión (14) se generaliza sin dificultad a r elevaciones de grado,

$$c^r[t_1, \dots, t_{n+1}] = \frac{n! r!}{(n+r)!} \sum_{1 \leq i_1 < \dots < i_r \leq n+1} c[t_1, \dots, t_{n+1} | t_{i_1}, \dots, t_{i_r}], \quad (15)$$

ya que existen $r!$ maneras distintas de ir eliminando los valores t_{i_1}, \dots, t_{i_r} . Los valores detrás de la barra son los que se omiten de la expresión. El factor $n!/(n+r)!$ aparece por el cálculo sucesivo de las medias de $n+1, \dots, n+r$ términos. \square

Para el caso particular de calcular un vértice del polígono de control, en la expresión de $c_i^r = c^r[0^{<n+r-i>}, 1^{<i>}]$ aparecerá el vértice $c_j = c[0^{<n-j>}, 1^{<j>}]$ tantas veces como maneras distintas de elegir $n - j$ ceros de un total de $n + r - i$, y j unos, de un total de i . Por tanto,

$$c_i^r = \sum_{j=j_0}^i c_j \frac{\binom{n+r-i}{n-j} \binom{i}{j}}{\binom{n+r}{n}} = \sum_{j=j_0}^i c_j \frac{\binom{n}{j} \binom{r}{i-j}}{\binom{n+r}{i}}, \quad (16)$$

donde el límite inferior de la suma, j_0 , es $i - r$, si este número es positivo, o cero, en caso contrario. Nótese que el coeficiente de c_j es una probabilidad hipergeométrica.

Si iteramos el proceso de elevación del grado, esencialmente estamos “redondeando” el polígono de control, recortándole las esquinas. En el límite

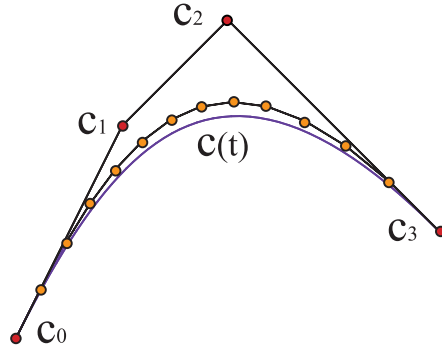


Figura 17: Al elevar el grado, el polígono de control tiende a la curva de Bézier

de infinitas elevaciones de grado, el polígono tiende a la forma de la curva, aunque no es un procedimiento eficiente para trazarla. [Ejemplo](#).

Una manera de verlo es tener en cuenta que la distribución hipergeométrica de probabilidad tiende a la binomial cuando r es muy grande,

$$\lim_{r \rightarrow \infty} \frac{\binom{n}{j} \binom{r}{i-j}}{\binom{n+r}{i}} = \binom{n}{j} t^j (1-t)^{n-j} = B_j^n(t),$$

si hacemos el límite de modo que $i/(n+r)$ tienda a t . Así pues,

$$\lim_{r \rightarrow \infty} c_i^r = \sum_{j=0}^n c_j B_j^n(t) = c(t).$$

como habíamos anunciado.

La elevación del grado no es la panacea universal para conseguir curvas más versátiles, ya que sabemos que los polinomios presentan mayor variación a medida que incrementamos el grado, con lo cual es probable que aparezcan oscilaciones no deseadas en la forma de la curva.

Una propiedad que se infiere directamente del proceso de elevación del grado es la **disminución de la variación**. Una recta r corta a una curva de Bézier, $c(t)$, en *a lo sumo* tantos puntos como a su polígono de control, $\{c_0, \dots, c_n\}$. La razón está clara. Al elevar el grado, recortamos las esquinas

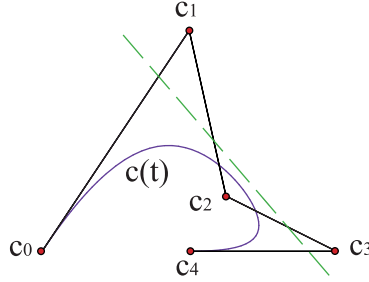


Figura 18: Disminución de la variación: la recta corta al polígono en cuatro puntos y en ninguno a la curva

al polígono de control y disminuimos, por tanto, las posibles intersecciones de la recta r con el polígono. Como hemos visto que la iteración de la elevación del grado tiene como límite la propia curva de Bézier, resulta que esta no puede ser cortada por la recta en más puntos que los que corta al polígono de control. \square [Ejemplo](#).

El problema inverso, la disminución del grado, como es obvio, no tiene en general solución, ya que una curva de grado $n + 1$ real no se puede expresar como una curva de grado n . La matriz del sistema de ecuaciones de la elevación del grado,

$$\frac{1}{n+1} \begin{pmatrix} n+1 & 0 & \cdots & 0 & 0 \\ 1 & n & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & n & 1 \\ 0 & 0 & \cdots & 0 & n+1 \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} c_0^1 \\ \vdots \\ c_{n+1}^1 \end{pmatrix},$$

no se puede invertir. Pero si modificamos el sistema $MC = C_1$ multiplicándolo por la matriz traspuesta de matriz de elevación del grado, obtenemos un sistema $M^tMC = M^tC_1$, cuya matriz M^tM es cuadrada, simétrica, definida positiva y, por tanto, invertible, $\tilde{C} = (M^tM)^{-1}M^tC_1$. La solución \tilde{C} , $\{c_0, \dots, c_n\}$, de este nuevo sistema es la mejor aproximación al problema de disminución del grado por mínimos cuadrados, tal como estudiaremos más adelante.

En el ejemplo de la figura [19](#) hemos aproximado la cuártica de polígono de control $\{(0, 0), (1, 2), (2, 3), (4, 1), (3, -1)\}$ por una cúbica de polígono de

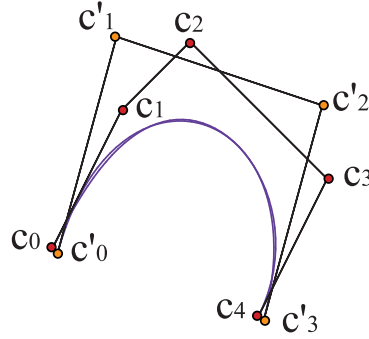


Figura 19: La disminución del grado no proporciona exactamente la misma curva

control $\{(1/14, -1/14), (13/14, 43/14), (55/14, 29/14), (43/14, -15/14)\}$,

$$M = \frac{1}{4} \begin{pmatrix} 4 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 4 \end{pmatrix}, \quad M^t M = \frac{1}{16} \begin{pmatrix} 17 & 3 & 0 & 0 \\ 3 & 13 & 4 & 0 \\ 0 & 4 & 13 & 3 \\ 0 & 0 & 3 & 17 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} 0 & 0 \\ 1 & 2 \\ 2 & 3 \\ 4 & 1 \\ 3 & -1 \end{pmatrix}, \quad M^t C_1 = \frac{1}{4} \begin{pmatrix} 1 & 2 \\ 7 & 12 \\ 16 & 9 \\ 16 & -3 \end{pmatrix},$$

$$\tilde{C} = (M^t M)^{-1} M^t C_1 = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \frac{1}{14} \begin{pmatrix} 1 & -1 \\ 13 & 43 \\ 55 & 29 \\ 43 & -15 \end{pmatrix}.$$

Obsérvese que los extremos no coinciden, aunque podríamos imponer esta condición como ecuaciones adicionales en la reducción del grado.

7. Derivadas

[Vídeo de Derivación y elevación del grado](#)

La derivada de la parametrización de una curva de Bézier, es decir, el campo tangente a la curva, se obtiene de manera sencilla gracias a las propiedades de los polinomios de Bernstein,

$$\begin{aligned}\frac{dB_i^n(t)}{dt} &= \frac{n!}{i!(n-i)!} (it^{i-1}(1-t)^{n-i} - (n-i)t^i(1-t)^{n-i-1}) \\ &= \frac{n!}{(i-1)!(n-i)!} t^{i-1}(1-t)^{n-i} - \frac{n!}{i!(n-i-1)!} t^i(1-t)^{n-i-1} \\ &= n (B_{i-1}^{n-1}(t) - B_i^{n-1}(t)),\end{aligned}$$

con lo cual la expresión de la derivada de la parametrización es

$$\begin{aligned}\frac{dc(t)}{dt} &= \sum_{i=0}^n c_i \frac{dB_i^n(t)}{dt} = n \sum_{i=0}^n c_i (B_{i-1}^{n-1}(t) - B_i^{n-1}(t)) \\ &= n \sum_{i=0}^{n-1} \Delta c_i B_i^{n-1}(t),\end{aligned}\tag{17}$$

haciendo un cambio de índices, $i \rightarrow i + 1$, en la primera suma y definiendo el vector diferencia como $\Delta c_i = c_{i+1} - c_i$. \square

Este es un resultado interesante, ya que muestra que la derivada de una parametrización de Bézier de grado n es otra parametrización de Bézier de grado $n - 1$ para una curva vectorial de polígono de control dado por $\{n\Delta c_0, \dots, n\Delta c_{n-1}\}$.

Como la curva pasa por los vértices inicial y final, tenemos que

$$\left. \frac{dc(t)}{dt} \right|_{t=0} = n\Delta c_0 = n(c_1 - c_0), \quad \left. \frac{dc(t)}{dt} \right|_{t=1} = n\Delta c_{n-1} = n(c_n - c_{n-1}),\tag{18}$$

resultado que proporciona una interpretación a los vectores definidos por las parejas de vértices iniciales y finales. Indican las tangentes en los extremos de la curva, un dato importante para el diseño. [Ejemplo](#).

Esta expresión se puede generalizar a derivadas superiores por simple iteración:

$$\frac{d^r c(t)}{dt^r} = \frac{n!}{(n-r)!} \sum_{i=0}^{n-r} \Delta^r c_i B_i^{n-r}(t) = \frac{n!}{(n-r)!} \sum_{i=0}^{n-r-s} \Delta^r c_i^s(t) B_i^{n-r-s}(t),$$

donde las diferencias de orden superior se definen de manera recurrente como $\Delta^s c_i = \Delta^{s-1} c_{i+1} - \Delta^{s-1} c_i$.

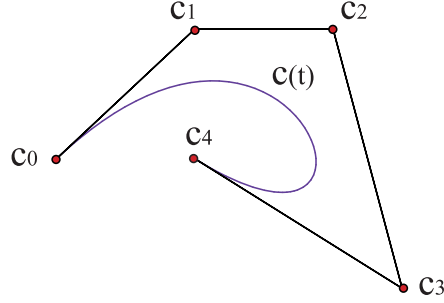


Figura 20: Los segmentos $\overline{c_0c_1}$, $\overline{c_3c_4}$ proporcionan las tangentes en c_0 y c_4 , respectivamente

Podemos asimilar las diferencias a “interpolaciones vectoriales”, es decir, a combinaciones vectoriales de puntos, usando el vector unitario \mathbf{e} de la recta real, que une 0 con 1. Así, para dos puntos, c_0, c_1 ,

$$c[\mathbf{e}] = c_1 - c_0.$$

Con esta notación, las derivadas se expresan como una forma polar mixta de puntos y vectores,

$$\frac{d^r c(t)}{dt^r} = \frac{n!}{(n-r)!} c[t^{\langle n-r \rangle}, \mathbf{e}^{\langle r \rangle}]. \quad (19)$$

En particular, la derivada primera,

$$\frac{dc(t)}{dt} = nc[t^{\langle n-1 \rangle}, \mathbf{e}] = n \left(c_1^{(n-1)}(t) - c_0^{(n-1)}(t) \right).$$

Un detalle importante que no debe pasar inadvertido es que en estas expresiones sólo aparece el grado en el factor multiplicativo, lo cual será decisivo a la hora de unir varios tramos de curvas de Bézier.

Supongamos que deseamos enlazar dos curvas de Bézier de grado n , definidas en los intervalos adyacentes $[u_0, u_1]$, $[u_1, u_2]$, con polígonos respectivos $\{c_0, \dots, c_n\}$, $\{\tilde{c}_0, \dots, \tilde{c}_n\}$. Si la curva es continua, deberemos tener $c(u_1) = \tilde{c}(u_1)$, es decir, $c_n = \tilde{c}_0$.

Si además queremos que la curva compuesta definida en el intervalo $[u_0, u_2]$ tenga una parametrización de clase C^1 , es decir, con tangente conti-

nua, deberemos exigir

$$\left. \frac{dc(u)}{du} \right|_{u=u_1} = \left. \frac{d\tilde{c}(u)}{du} \right|_{u=u_1},$$

y como la relación entre el parámetro $u \in [u_0, u_1]$ y el parámetro $t \in [0, 1]$ es, como vimos,

$$t(u) = \frac{u - u_0}{u_1 - u_0}, \quad \frac{dt(u)}{du} = \frac{1}{\Delta u_0}, \quad \Delta u_0 = u_1 - u_0,$$

por la regla de la cadena, $d/du = dt(u)/du \cdot d/dt$, usando la expresión deducida para la derivada de una curva de Bézier en los extremos, obtenemos una sencilla relación que expresa la diferenciabilidad de la parametrización:

$$\frac{\Delta c_{n-1}}{\Delta u_0} = \frac{\Delta \tilde{c}_0}{\Delta u_1}, \quad (20)$$

denotando $\Delta u_1 = u_2 - u_1$.

Nótese que el grado n ha desaparecido por cancelación en la expresión, con lo cual esta relación es independiente del grado de la curva. Es una simple condición geométrica sobre los vértices de los extremos adyacentes de las curvas que garantiza que la curva compuesta es de clase C^1 . [Ejemplo](#).

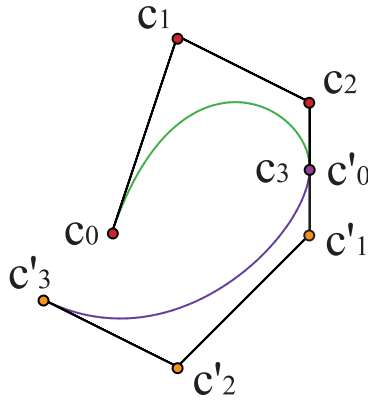


Figura 21: Los segmentos $\overline{c'_0c'_1}$, $\overline{c_2c_3}$ deben ser paralelos para que la curva compuesta tenga tangente continua

Esta expresión es bastante ilustrativa, ya que muestra que la condición sobre las derivadas de la parametrización se traduce en una condición sobre las diferencias de los vértices.

La condición para que la curva sea de clase C^r es fácil de deducir, a la vista de lo anterior,

$$\frac{\Delta^s c_{n-s}}{(\Delta u_0)^s} = \frac{\Delta^s \tilde{c}_0}{(\Delta u_1)^s}, \quad s = 0, \dots, r. \quad (21)$$

Echemos, finalmente, una ojeada a la condición para que sea de clase C^2 ,

$$\frac{\Delta^2 c_{n-2}}{(\Delta u_0)^2} = \frac{\Delta^2 \tilde{c}_0}{(\Delta u_1)^2}, \quad (22)$$

que, tal como se observa en la figura, implica que las prolongaciones de los segmentos $\overline{c_{n-2}c_{n-1}}$ y $\overline{\tilde{c}_2\tilde{c}_1}$ se cortan para definir un punto d ,

$$c_{n-1} + \frac{\Delta u_1}{\Delta u_0} \Delta c_{n-2} = d = \tilde{c}_1 - \frac{\Delta u_0}{\Delta u_1} \Delta \tilde{c}_1.$$

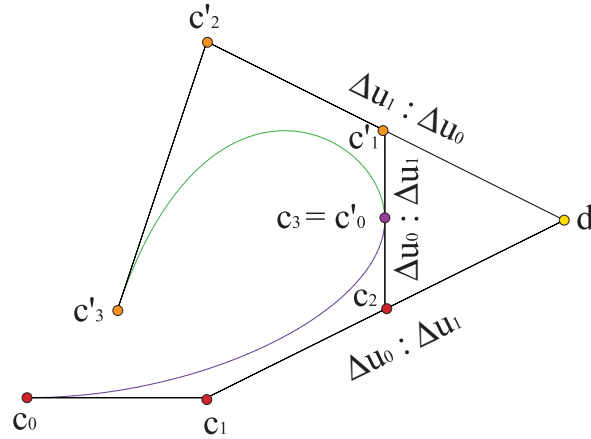


Figura 22: Curva de clase C^2 compuesta de dos tramos

Más información puede alcanzarse de otra manera. Podemos considerar los vértices $\{c_{n-2}, c_{n-1}, c_n\}$, $\{\tilde{c}_0, \tilde{c}_1, \tilde{c}_2\}$ como polígonos de control de dos

parábolas. Como las condiciones de diferenciabilidad son independientes del grado de la curva, las relaciones (20), (22) son aplicables exactamente igual a este caso, si exigimos que la curva compuesta por las dos parábolas sea de clase C^2 . Ahora bien, como las derivadas de una parábola son nulas a partir de la tercera, esto es tanto como exigir que la curva compuesta sea una única parábola, ya que todas las derivadas serían iguales en $u = u_1$.

Por tanto, las dos parábolas se obtienen a partir de la curva compuesta por subdivisión de intervalo $[u_0, u_2]$ en dos subintervalos. Y dicha curva de grado dos tendrá un polígono de control $\{c_{n-2}, d, c_2\}$, lo que confirma la existencia del punto d . Dicha parábola estará parametrizada en el intervalo $[0, 1]$ como

$$b(t) = c_{n-2}B_0^2(t) + dB_1^2(t) + c_2B_2^2(t), \quad t = \frac{u - u_0}{u_2 - u_0},$$

y el valor del parámetro correspondiente a u_1 será $\lambda = \Delta u_0 / (\Delta u_0 + \Delta u_1)$.

Con estos datos podemos reobtener por subdivisión del intervalo en dos, $[0, \lambda]$, $[\lambda, 1]$, los vértices,

$$c_{n-1} = b[0, \lambda] = (1 - \lambda)c_{n-2} + \lambda d = \frac{\Delta u_1}{\Delta u_0 + \Delta u_1}c_{n-2} + \frac{\Delta u_0}{\Delta u_0 + \Delta u_1}d,$$

que muestra que los vectores $\mathbf{c}_{n-2}\mathbf{c}_{n-1}$ y $\mathbf{c}_{n-1}\mathbf{d}$ están en proporción $\Delta u_0 : \Delta u_1$. Del mismo modo,

$$\tilde{c}_1 = b[\lambda, 1] = (1 - \lambda)d + \lambda \tilde{c}_2 = \frac{\Delta u_1}{\Delta u_0 + \Delta u_1}d + \frac{\Delta u_0}{\Delta u_0 + \Delta u_1}\tilde{c}_2,$$

y los vectores $\mathbf{d}\tilde{\mathbf{c}}_1$ y $\tilde{\mathbf{c}}_1\tilde{\mathbf{c}}_2$ están también en relación $\Delta u_0 : \Delta u_1$. El resto de relaciones no aportan información nueva y podemos representar ya geométricamente la condición de ser de clase C^2 , tal como se aprecia en la figura.

8. Interpolación

Vídeo de Interpolación y aproximación de curvas

Las curvas de Bézier pueden utilizarse para interpolar entre varios puntos, conocidos los valores que les corresponden del parámetro t . Supongamos que

tenemos $n+1$ puntos $\{a_0, \dots, a_n\}$ y queremos obtener una curva $c(t)$ definida en el intervalo $[0, 1]$ que verifique

$$c(t_0) = a_0, \quad c(t_n) = a_n,$$

para unos valores t_0, \dots, t_n del parámetro.

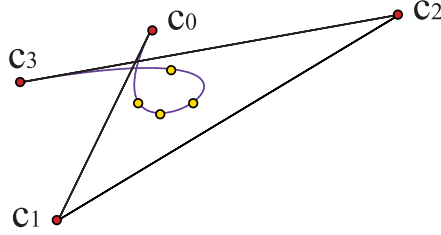


Figura 23: Cúbica interpolante de cuatro puntos

Para resolver el problema de interpolación, debemos encontrar el polígono de control de la curva de grado n que verifique

$$\sum_{i=0}^n c_i B_i^n(t_j) = a_j, \quad j = 0, \dots, n,$$

es decir, se trata de resolver el sistema lineal de $n+1$ ecuaciones

$$\begin{pmatrix} B_0^n(t_0) & \cdots & B_n^n(t_0) \\ \vdots & \ddots & \vdots \\ B_0^n(t_n) & \cdots & B_n^n(t_n) \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix}.$$

Este sistema está determinado y tiene solución única, como se comprueba, por ejemplo, expresándolo en una base más cómoda, la canónica,

$$\begin{pmatrix} 1 & \cdots & t_0^n \\ \vdots & \ddots & \vdots \\ 1 & \cdots & t_n^n \end{pmatrix} \begin{pmatrix} \tilde{c}_0 \\ \vdots \\ \tilde{c}_n \end{pmatrix} = \begin{pmatrix} \tilde{a}_0 \\ \vdots \\ \tilde{a}_n \end{pmatrix}$$

ya que el determinante de la matriz del sistema de ecuaciones lineales es el conocido determinante de Vandermonde,

$$\begin{vmatrix} 1 & \cdots & t_0^n \\ \vdots & \ddots & \vdots \\ 1 & \cdots & t_n^n \end{vmatrix} = (t_1 - t_0) \cdots (t_n - t_0) (t_2 - t_1) \cdots (t_n - t_1) \cdots (t_n - t_{n-1}),$$

que se anula si y sólo si al menos dos valores t_i, t_j son iguales, que no es nuestro caso. Luego el rango de la matriz del sistema es $n + 1$ y el sistema está bien determinado.

En notación matricial, nuestro sistema es $BC = A$, donde B es la matriz de los valores de los polinomios de Bernstein en t_0, \dots, t_n , y C, A son las matrices cuyas filas son las componentes, respectivamente, de los vértices del polígono de control y de los datos del problema. La solución formal sería $C = B^{-1}A$, aunque la manera eficiente de obtenerla será por algún algoritmo de resolución numérica. [Ejemplo](#).

Por ejemplo si queremos una curva cúbica que pase por los puntos $(-1, 0)$, $(1, -1)$, $(4, 0)$, $(2, 3)$ para los valores del parámetro t respectivos $1/5, 1/3, 1/2, 4/5$, obtenemos como vértices del polígono de control de la curva interpolante $\{(10/63, 137/21), (-223/27, -193/18), (614/27, 143/18), (-731/63, 38/21)\}$,

$$\begin{aligned}
 B &= \begin{pmatrix} B_0^3(1/5) & B_1^3(1/5) & B_2^3(1/5) & B_3^3(1/5) \\ B_0^3(1/3) & B_1^3(1/3) & B_2^3(1/3) & B_3^3(1/3) \\ B_0^3(1/2) & B_1^3(1/2) & B_2^3(1/2) & B_3^3(1/2) \\ B_0^3(4/5) & B_1^3(4/5) & B_2^3(4/5) & B_3^3(4/5) \end{pmatrix} \\
 &= \begin{pmatrix} 64/125 & 48/125 & 12/125 & 1/125 \\ 8/27 & 4/9 & 2/9 & 1/27 \\ 1/8 & 3/8 & 3/8 & 1/8 \\ 1/125 & 12/125 & 48/125 & 64/125 \end{pmatrix}, \quad A = \begin{pmatrix} -1 & 0 \\ 1 & -1 \\ 4 & 0 \\ 2 & 3 \end{pmatrix} \\
 C &= \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = B^{-1}A = \begin{pmatrix} 10/63 & 137/21 \\ -223/27 & -193/18 \\ 614/27 & 143/18 \\ -731/63 & 38/21 \end{pmatrix}.
 \end{aligned}$$

9. Aproximación

[Vídeo de Interpolación y aproximación de curvas](#)

Lo más común, sin embargo, no es que tengamos $n + 1$ puntos y queramos obtener una curva de grado n que pase por todos ellos, ya que, como ya se ha dicho, los grados altos presentan oscilaciones espúreas. Es más interesante el caso en el que tenemos $m + 1$ puntos y queremos obtener la curva de grado n que *más* se aproxime a nuestro conjunto de puntos. Luego matizaremos qué quiere decir “mejor aproximación”.

Si lo planteáramos como un problema de interpolación, no habría en general solución,

$$\begin{pmatrix} B_0^n(t_0) & \cdots & B_n^n(t_0) \\ \vdots & \ddots & \vdots \\ B_0^n(t_m) & \cdots & B_n^n(t_m) \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} a_0 \\ \vdots \\ a_m \end{pmatrix},$$

ya que obtendríamos un sistema sobredeterminado si, como es de esperar, $m > n$. Pero podemos obtener una *pseudoinversa* mediante el siguiente artificio. En vez de resolver el sistema $BC = A$, de $m + 1$ ecuaciones y $n + 1$ incógnitas, resolveremos el sistema $B^t B \tilde{C} = B^t A$, que es un sistema de $n + 1$ ecuaciones y $n + 1$ incógnitas. Como la matriz del nuevo sistema $B^t B$ es definida positiva y simétrica, tiene determinante positivo y el problema tiene solución única, $\tilde{C} = (B^t B)^{-1} B^t A$.

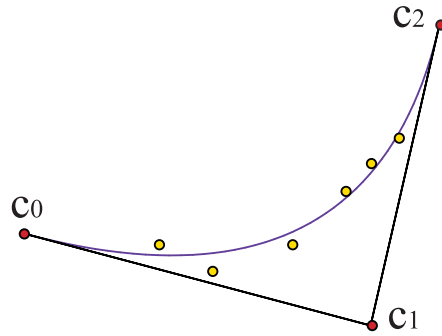


Figura 24: Parábola aproximante de seis puntos

Ahora bien, ¿qué hemos resuelto realmente? Porque el problema de interpolación sabemos a ciencia cierta que no tiene solución. Lo que hemos resuelto es el problema de aproximación por mínimos cuadrados, es decir, la desviación de la parametrización de la curva solución $c(t)$ de los valores $a_i = c(t_i)$ es mínima. Denotando por $\| \cdot \|$ la longitud de un vector y por $\langle \cdot, \cdot \rangle$, el producto escalar y desarrollando la longitud $\|v - w\|^2 = \|v\|^2 + \|w\|^2 - 2\langle v, w \rangle$,

$$L(c_0, \dots, c_n) : = \sum_{i=0}^m \|c(t_i) - a_i\|^2 = \sum_{i=0}^m \left\| \sum_{j=0}^n c_j B_j^n(t_i) - a_i \right\|^2$$

$$\begin{aligned}
&= \sum_{i=0}^m \left\{ \left\langle \sum_{j=0}^n c_j B_j^n(t_i), \sum_{k=0}^n c_k B_k^n(t_i) \right\rangle + \langle a_i, a_i \rangle \right. \\
&\quad \left. - 2 \left\langle \sum_{k=0}^n c_k B_k^n(t_i), a_i \right\rangle \right\},
\end{aligned}$$

el extremo de la desviación se obtiene derivando respecto a cada componente de las variables c_k e igualando a cero,

$$0 = 2 \sum_{i=0}^m \left\{ \sum_{j=0}^n c_j B_j^n(t_i) - a_i \right\} B_k^n(t_i),$$

que en notación matricial es precisamente $B^t B C - B^t A$, con lo cual la construcción heurística del comienzo de la sección proporciona la curva de grado n mejor aproximación por mínimos cuadrados a los datos del problema. [Ejemplo](#).

Por ejemplo si queremos hacer pasar una parábola por los puntos $(-1, 0)$, $(1, -1)$, $(4, 0)$, $(2, 3)$ para los valores del parámetro t respectivos $0, 1/3, 2/3, 1$, tendremos más datos que incógnitas y el problema no tendrá solución, ya que en general sólo podemos forzar a una curva de grado dos a pasar por tres puntos. Pero podemos obtener una mejor aproximación con vértices del polígono de control de la curva interpolante $\{(-13/10, 0), (5, -3), (23/10, 3)\}$,

$$B = \begin{pmatrix} B_0^2(0) & B_1^2(0) & B_2^2(0) \\ B_0^2(1/3) & B_1^2(1/3) & B_2^2(1/3) \\ B_0^2(2/3) & B_1^2(2/3) & B_2^2(2/3) \\ B_0^2(1) & B_1^2(1) & B_2^2(1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 4/9 & 4/9 & 1/9 \\ 1/9 & 4/9 & 4/9 \\ 0 & 0 & 1 \end{pmatrix},$$

$$B^t B = \frac{1}{81} \begin{pmatrix} 98 & 20 & 8 \\ 20 & 32 & 20 \\ 8 & 20 & 98 \end{pmatrix}, \quad A = \begin{pmatrix} -1 & 0 \\ 1 & -1 \\ 4 & 0 \\ 2 & 3 \end{pmatrix}, \quad B^t A = \frac{1}{9} \begin{pmatrix} -1 & -4 \\ 20 & -4 \\ 35 & 26 \end{pmatrix}$$

$$\tilde{C} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = (B^t B)^{-1} B^t A = \begin{pmatrix} -13/10 & 0 \\ 5 & -3 \\ 23/10 & 3 \end{pmatrix}.$$

10. Elección de los nudos

En los problemas de interpolación y aproximación que hemos tratado hasta el momento, hemos dado por sentado que conocemos una sucesión de nudos, $\{u_0, \dots, u_m\}$, que determinan el “momento” en el cual la curva pasa por los puntos dato del problema, $c(u_i) = a_i$.

Este planteamiento es el adecuado cuando en un experimento vamos obteniendo datos para distintos valores de, por ejemplo, el tiempo, es decir, cuando interesa la parametrización, la ordenación temporal de los puntos de la gráfica.

Sin embargo, el caso común en el diseño es distinto. En el diseño importa exclusivamente la gráfica de la curva, es decir, su forma, no la manera en la que esta es recorrida. La parametrización es un artificio nuestro para describirla y poco más.

Por tanto, lo más frecuente es que no dispongamos de una sucesión de valores u_i privilegiada, sino tan sólo de un conjunto de puntos, y tal vez tangentes, por los cuales queremos que pase nuestra curva.

Una elección ingenua consistiría en elegir la **partición uniforme**, es decir, aquella en la que las diferencias $\Delta u_i = k$ son iguales. Nótese que, por la invariancia bajo transformaciones afines del parámetro, $\tilde{u} = au + b$, no importa ni el origen de la partición, ni la separación entre nudos.

Esta partición puede ser útil cuando los datos están distribuidos de manera homogénea, pero, obviamente, no será útil cuando la separación entre puntos sea muy dispar.

Una segunda solución al problema consiste en suponer que la curva está parametrizada aproximadamente por su longitud de arco, s ,

$$s(u) = \int_a^u du \|c'(u)\|, \quad (23)$$

y tomar como estimación grosera de esta medida la **longitud de la cuerda** entre los puntos dato,

$$\Delta u_i = k \|\Delta a_i\|, \quad (24)$$

donde k es una constante de escala.

Esta solución del problema mejora los resultados de la partición uniforme, pero fracasa si la u está alejada de la longitud de arco o si esta difiere mucho de la longitud de la cuerda entre puntos. Esto sucede, por ejemplo, si la curva presenta mucha variación.

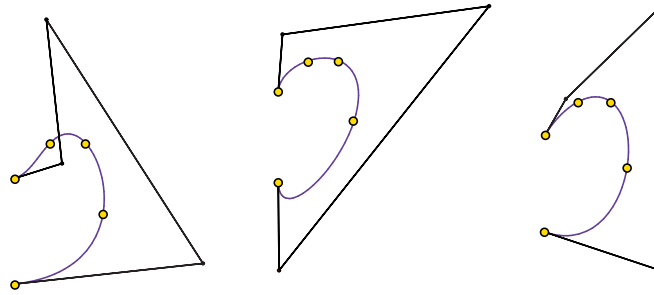


Figura 25: Curva interpolante de cinco puntos usando partici3n uniforme, por longitud de cuerda y aproximaci3n centr3peta

Otra parametrizaci3n es la llamada **centr3peta**, ya que modeliza una aproximaci3n de aceleraci3n centr3peta suave,

$$\Delta u_i = k \sqrt{\|\Delta a_i\|}, \quad (25)$$

que proporciona buenos resultados. De esta manera tratamos de evitar variaciones bruscas de las fuerzas de inercia a lo largo de la curva.

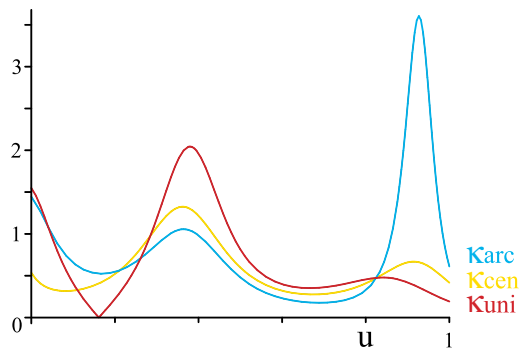


Figura 26: Gr3fica de la curvatura de las curvas interpolantes

A falta de otros m3todos para elegir los nudos de la partici3n del intervalo, la aproximaci3n centr3peta suele dar mejores resultados. En el ejemplo de la figura 25 vemos que la partici3n uniforme introduce una inflexi3n esp3rea al comienzo de la curva, que desaparece con los otros dos m3todos. En cambio, la aproximaci3n por la longitud de la cuerda introduce una curvatura extra3a

en el tramo final, tal como se refleja en el pico final de la gráfica 26. En esta misma gráfica observamos que la curvatura que presenta menor variación en este caso la correspondiente a la aproximación centrípeta.