

Curvas *spline*

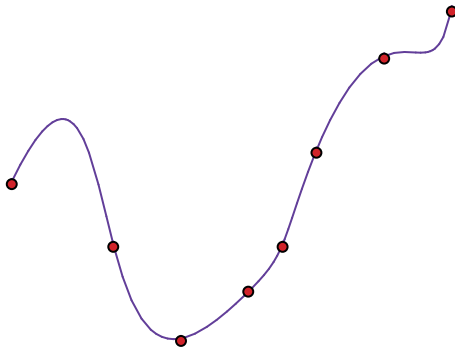
Leonardo Fernández Jambrina

Matemática Aplicada
E.T.S.I. Navales
Universidad Politécnica de Madrid
leonardo.fernandez@upm.es

11 de enero 2010

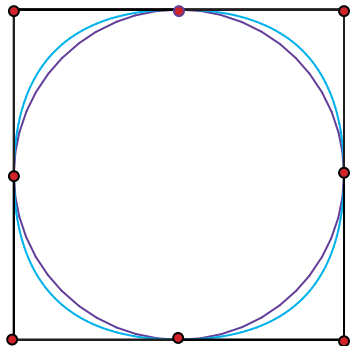
Máster universitario en Tecnologías y Sistemas de Comunicaciones





Las curvas polinómicas son demasiado rígidas. . .

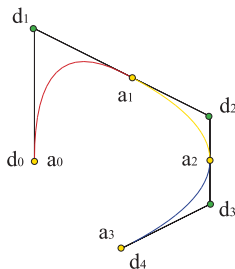





Permiten representar cónicas, pero aportan pocos grados de libertad nuevos frente a las curvas polinómicas.

Planteamiento

- La solución más práctica son las curvas polinómicas y racionales a trozos (NURBS o *spline*).
- Estudiaremos primero algunos casos particulares, parábolas a trozos de clase C^1 , cúbicas de clase C^2 a trozos.
- Después veremos el formalismo general de las curvas *spline*.
- Obviamente, no construiremos las uniones *a mano* como en los temas anteriores.



- Curvas polinómicas.
- Curvas racionales.
- **Curvas *spline*.** 
- Superficies de Bézier.
- Generación de superficies.

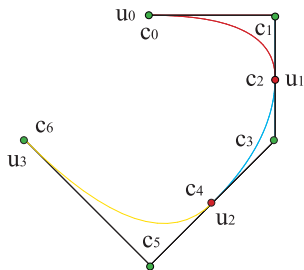


- 1 Motivación
- 2 Ejemplos de curvas *spline*
- 3 Curvas B-*spline*
- 4 Propiedades de las curvas *spline*
- 5 Algoritmo de inserción
- 6 Elevación del grado
- 7 Derivabilidad
- 8 Funciones B-spline o nodales
- 9 *Splines* racionales
- 10 Teorema de Holladay



Parábolas a trozos

- Consideremos una curva formada por N tramos parabólicos.
- Numeramos los vértices de modo que el vértice último de un polígono coincida con el primero del siguiente.
- Tenemos una lista de vértices $\{c_0, \dots, c_{2N}\}$, en la cual $\{c_0, c_1, c_2\}$ es el polígono del primer tramo, $\{c_2, c_3, c_4\}$ es el polígono del segundo, \dots , y $\{c_{2N-2}, c_{2N-1}, c_{2N}\}$, el del último.
- El primer tramo estará definido en $[u_0, u_1]$, el segundo en $[u_1, u_2]$, \dots , y el último en $[u_{N-1}, u_N]$. La curva estará definida en $[u_0, u_N]$.



- Las igualdades de las derivadas a ambos lados de las uniones

$$\frac{\Delta c_{2i-1}}{\Delta u_{i-1}} = \frac{\Delta c_{2i}}{\Delta u_i}, \quad i = 1, \dots, N-1,$$

se pierden si permitimos al usuario modificar *todos* los vértices.

- Si queremos mantener la continuidad de las derivadas, el *usuario* sólo deberá poder modificar vértices que no la alteren.
- Despejamos los vértices de las uniones,

$$c_{2i} = \frac{\Delta u_i}{\Delta u_{i-1} + \Delta u_i} c_{2i-1} + \frac{\Delta u_{i-1}}{\Delta u_{i-1} + \Delta u_i} c_{2i+1}.$$

- Conocidos los vértices “interiores”, c_{2i-1} y los **nudos** $\{u_0, \dots, u_N\}$, podemos reconstruir los vértices “exteriores”, c_{2i} .
- Hay que añadir los vértices extremos, c_0, c_{2N} , ya que no los define ninguna unión.
- Son los únicos datos de la curva que debemos facilitar al usuario para que los modifique libremente.

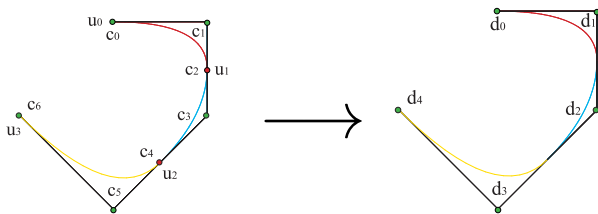


Parábolas B-spline

- Sólo es preciso dar los nudos $\{u_0, \dots, u_N\}$ y los vértices $\{c_0, c_1, \dots, c_{2i-1}, \dots, c_{2N-1}, c_{2N}\}$.
- Un total de $N + 2$ vértices ($2N + 1$ menos $N - 1$ condiciones en las uniones).
- Denotaremos como $\{d_0, \dots, d_{N+1}\}$ estos vértices,

$$d_0 := c_0, \quad d_1 := c_1, \quad \dots, \quad d_N := c_{2N-1}, \quad d_{N+1} := c_{2N}.$$

- A este conjunto de vértices lo denominaremos **polígono B-spline** de la curva a trozos.



- Sólo es preciso dar los nudos $\{u_0, \dots, u_N\}$ y los vértices $\{c_0, c_1, \dots, c_{2i-1}, \dots, c_{2N-1}, c_{2N}\}$.
- Un total de $N + 2$ vértices ($2N + 1$ menos $N - 1$ condiciones en las uniones).
- Denotaremos como $\{d_0, \dots, d_{N+1}\}$ estos vértices,

$$d_0 := c_0, \quad d_1 := c_1, \quad \dots, \quad d_N := c_{2N-1}, \quad d_{N+1} := c_{2N}.$$

- A este conjunto de vértices lo denominaremos **polígono B-spline** de la curva a trozos.

(fig402.mov)



- Sólo es preciso dar los nudos $\{u_0, \dots, u_N\}$ y los vértices $\{c_0, c_1, \dots, c_{2i-1}, \dots, c_{2N-1}, c_{2N}\}$.
- Un total de $N + 2$ vértices ($2N + 1$ menos $N - 1$ condiciones en las uniones).
- Denotaremos como $\{d_0, \dots, d_{N+1}\}$ estos vértices,

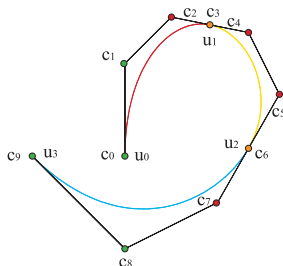
$$d_0 := c_0, \quad d_1 := c_1, \quad \dots, \quad d_N := c_{2N-1}, \quad d_{N+1} := c_{2N}.$$

- A este conjunto de vértices lo denominaremos **polígono B-spline** de la curva a trozos.
- Obsérvese que $d_0 = c[u_0, u_0]$, $d_1 = c[u_0, u_1]$, $d_2 = c[u_1, u_2]$,
 $d_N = c[u_{N-1}, u_N]$, $d_{N+1} = c[u_N, u_N]$.
- Se recupera la *simetría* de la definición sin añadimos **nudos auxiliares** $u_{-1} = u_0$, $u_{N+1} = u_N$, de modo que $d_0 = c[u_{-1}, u_0]$,
 $d_{N+1} = c[u_N, u_{N+1}]$.



Curvas cúbicas a trozos

- Consideremos una curva formada por N tramos cúbicos.
- Numeramos los vértices de modo que el vértice último de un polígono coincida con el primero del siguiente.
- Tenemos una lista de vértices $\{c_0, \dots, c_{3N}\}$, de los cuales $\{c_{3(i-1)}, c_{3i-2}, c_{3i-1}, c_{3i}\}$ corresponden al tramo i -ésimo, definido en el intervalo $[u_{i-1}, u_i]$. La curva entera está definida en $[u_0, u_N]$.



Cúbicas *spline*

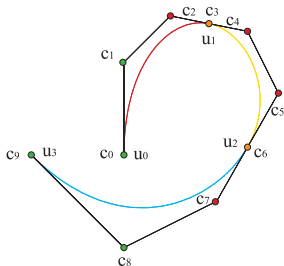
- La condición de ser de clase C^1 en u_i se traduce en

$$\frac{\Delta c_{3i-1}}{\Delta u_{i-1}} = \frac{\Delta c_{3i}}{\Delta u_i}, \quad i = 1, \dots, N-1,$$

- La condición de continuidad de las derivadas segundas,

$$\frac{\Delta^2 c_{3i-2}}{(\Delta u_{i-1})^2} = \frac{\Delta^2 c_{3i}}{(\Delta u_i)^2}, \quad i = 1, \dots, N-1,$$

es la misma para cualquier grado n .

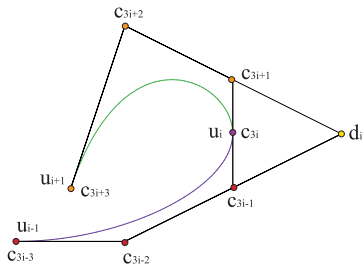


Vértices B-spline

- Aplicamos la condición C^2 a parábolas de polígonos $\{c_{3i-2}, c_{3i-1}, c_{3i}\}$, $\{c_{3i}, c_{3i+1}, c_{3i+2}\}$.
- Equivale a que provengan de una única parábola $b(u)$ de polígono $\{c_{3i-2}, d_i, c_{3i+2}\}$ en $[u_{i-1}, u_{i+1}]$ por restricción,

$$c_{3i-1} = b[u_{i-1}, u_i], \quad c_{3i} = b[u_i, u_i], \quad c_{3i+1} = b[u_i, u_{i+1}]$$

- Luego la condición C^2 equivale a que los lados $\overline{c_{3i-2}c_{3i-1}}$, $\overline{c_{3i+1}c_{3i+2}}$ se corten en un punto d_i en ciertas proporciones.

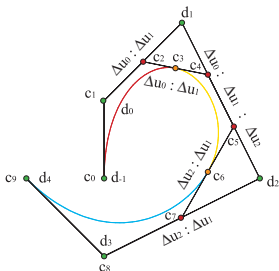


Cúbicas B-spline

- Tenemos fijados los vértices “exteriores” c_{3i} por la condición C^1 y los “interiores” c_{3i-1}, c_{3i+1} por d_i y la condición C^2 .
- Salvo los vértices iniciales c_0, c_1 y finales c_{3N-1}, c_{3N} , que no fija ninguna unión (se fijan con condiciones adicionales).
- **Datos:** nudos $\{u_0, \dots, u_N\}$, vértices $\{d_1, \dots, d_{N-1}\}$ y antiguos,

$$d_{-1} := c_0, \quad d_0 := c_1, \quad d_N := c_{3N-1}, \quad d_{N+1} := c_{3N}.$$

- $N + 3$ vértices = $3N + 1$ menos $2(N - 1)$ condiciones.
- Denominamos **polígono B-spline** de la curva a $\{d_{-1}, \dots, d_{N+1}\}$.

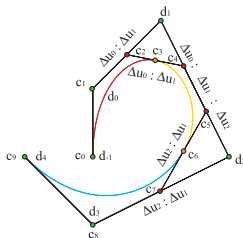


Interpolación

- Usamos las cúbicas *spline* para interpolar $N + 1$ puntos,

$$a_0 = c(u_0), \dots, a_N = c(u_N).$$

- Hay $N + 3$ vértices, pero $N + 1$ datos (2 grados de libertad).
- Lo usual es fijar las tangentes en los extremos vía c_1, c_{3N-1} :

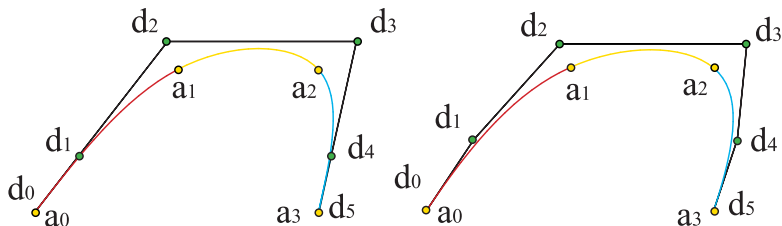


Interpolación

- Usamos las cúbicas *spline* para interpolar $N + 1$ puntos,

$$a_0 = c(u_0), \dots, a_N = c(u_N).$$

- Hay $N + 3$ vértices, pero $N + 1$ datos (2 grados de libertad).
- Lo usual es fijar las tangentes en los extremos vía c_1, c_{3N-1} :
 - Dar $c'(u_0), c'(u_N)$.
 - **Spline natural:** $c''(u_0) = 0 = c''(u_N)$.
 - **Tangentes de Bessel:** tangentes a parábolas $a_0 a_1 a_2, a_{N-2} a_{N-1} a_N$.



- Usamos las cúbicas *spline* para interpolar $N + 1$ puntos,

$$a_0 = c(u_0), \dots, a_N = c(u_N).$$

- Hay $N + 3$ vértices, pero $N + 1$ datos (2 grados de libertad).
- Lo usual es fijar las tangentes en los extremos vía c_1, c_{3N-1} :
 - Dar $c'(u_0), c'(u_N)$.
 - **Spline natural:** $c''(u_0) = 0 = c''(u_N)$.
 - **Tangentes de Bessel:** tangentes a parábolas $a_0 a_1 a_2, a_{N-2} a_{N-1} a_N$.

(fig415.mov)



Parábola auxiliar $b(u)$ v. cúbica $c(u)$

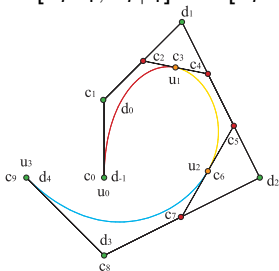
- En el nudo u_i , los vértices de la cúbica son

$$c_{3i-2} = c[u_{i-1}, u_{i-1}, \mathbf{u}_i], \quad c_{3i-1} = c[u_{i-1}, u_i, \mathbf{u}_i], \quad c_{3i} = c[u_i, u_i, \mathbf{u}_i], \\ c_{3i+1} = c[\mathbf{u}_i, u_i, u_{i+1}], \quad c_{3i+2} = c[\mathbf{u}_i, u_{i+1}, u_{i+1}].$$

- Y sobre la parábola,

$$c_{3i-2} = b[u_{i-1}, u_{i-1}], \quad c_{3i-1} = b[u_{i-1}, u_i], \quad c_{3i} = b[u_i, u_i], \\ c_{3i+1} = b[u_i, u_{i+1}], \quad c_{3i+2} = c[u_{i+1}, u_{i+1}].$$

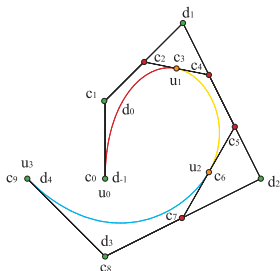
- Por coherencia, $d_i = b[u_{i-1}, u_{i+1}] = c[u_{i-1}, \mathbf{u}_i, u_{i+1}]$.



Vértices de la cúbica B-spline

- $d_{-1} := c_0 = c[u_0, u_0, u_0]$, $d_0 := c_1 = c[u_0, u_0, u_1]$.
- $d_i = c[u_{i-1}, u_i, u_{i+1}]$, $i = 1, \dots, N-1$.
- $d_N := c_{3N-1} = c[u_{N-1}, u_N, u_N]$, $d_{N+1} := c_{3N} = c[u_N, u_N, u_N]$.
- Si introducimos cuatro **nodos auxiliares** $u_{-2} := u_0$, $u_{-1} := u_0$, $u_{N+1} := u_N$, $u_{N+2} := u_N$, todos los vértices se evalúan sobre listas de nodos correlativos,

$$d_{-1} = c[u_{-2}, u_{-1}, u_0], \quad \dots \quad , d_{N+1} = c[u_N, u_{N+1}, u_{N+2}].$$



- Renumeramos para que las listas comiencen en cero siempre.
- Una curva C^{n-1} de grado n y N tramos tiene $n + N$ vértices, $\{d_0, \dots, d_L\}$, $L = n + N - 1$, el **polígono B-spline**.
- Y una **lista** de $2n + N - 1$ **nudos** $[u_0, \dots, u_K]$, $K = 2n + N - 2$.
- Los $n - 1$ primeros y últimos son **auxiliares**. Quedan $N + 1$ nudos.
- Se suelen tomar iguales respectivamente a u_n, u_{n+N-1} .
- La lista comienza y acaba con n nudos iguales. ¿IGES?
- La curva vive en $[u_{n-1}, u_{n+N-1}]$, $u_0 = \dots = u_{n-1}$, $u_{n+N-1} = \dots = u_K$.
- Bézier usa vértices $c_i = c[u_0^{<n-i>}, u_1^{<i>}]$ evaluados en dos nudos.
- La representación B-spline usa vértices $d_i = c[u_i, \dots, u_{i+n-1}]$ evaluados en una lista correlativa de nudos.



Algoritmo de De Boor para parábolas

¿Cómo se evalúa una parábola $c(u)$, $\{d_0, d_1, d_2\}$, $\{u_0, u_1, u_2, u_3\}$, $u \in [u_1, u_2]$ sin pasar por el polígono de control?

- Usando $u = \frac{u_2 - u}{u_2 - u_0}u_0 + \frac{u - u_0}{u_2 - u_0}u_2 = \frac{u_3 - u}{u_3 - u_1}u_1 + \frac{u - u_1}{u_3 - u_1}u_3$,

$$c[u_1, u] = \frac{u_2 - u}{u_2 - u_0}c[u_0, u_1] + \frac{u - u_0}{u_2 - u_0}c[u_1, u_2],$$

$$c[u, u_2] = \frac{u_3 - u}{u_3 - u_1}c[u_1, u_2] + \frac{u - u_1}{u_3 - u_1}c[u_2, u_3].$$

- Usando $u = \frac{u_2 - u}{u_2 - u_1}u_1 + \frac{u - u_1}{u_2 - u_1}u_2$,

$$c(u) = c[u, u] = \frac{u_2 - u}{u_2 - u_1}c[u_1, u] + \frac{u - u_1}{u_2 - u_1}c[u, u_2].$$



Algoritmo de De Boor para parábolas

¿Cómo se evalúa una parábola $c(u)$, $\{d_0, d_1, d_2\}$, $\{u_0, u_1, u_2, u_3\}$, $u \in [u_1, u_2]$ sin pasar por el polígono de control?

- Usando $u = \frac{u_2 - u}{u_2 - u_0} u_0 + \frac{u - u_0}{u_2 - u_0} u_2 = \frac{u_3 - u}{u_3 - u_1} u_1 + \frac{u - u_1}{u_3 - u_1} u_3$,

$$d_0^{(1)}(u) := \frac{u_2 - u}{u_2 - u_0} d_0 + \frac{u - u_0}{u_2 - u_0} d_1,$$

$$d_1^{(1)}(u) := \frac{u_3 - u}{u_3 - u_1} d_1 + \frac{u - u_1}{u_3 - u_1} d_2.$$

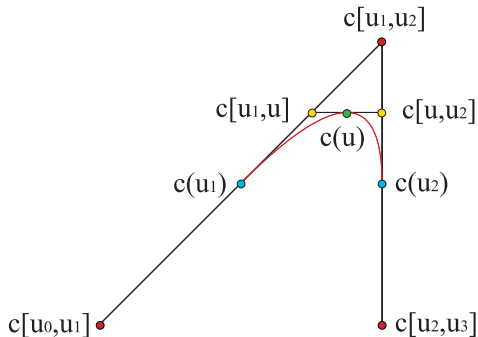
- Usando $u = \frac{u_2 - u}{u_2 - u_1} u_1 + \frac{u - u_1}{u_2 - u_1} u_2$,

$$c(u) = d_0^{(2)}(u) = \frac{u_2 - u}{u_2 - u_1} d_0^{(1)}(u) + \frac{u - u_1}{u_2 - u_1} d_1^{(1)}(u).$$



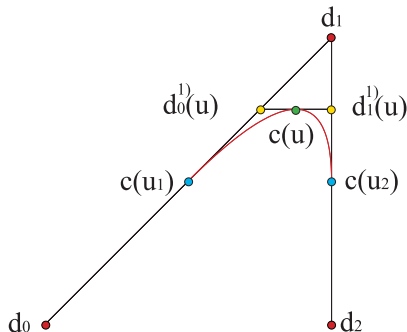
Algoritmo de De Boor para parábolas

¿Cómo se evalúa una parábola $c(u)$, $\{d_0, d_1, d_2\}$, $\{u_0, u_1, u_2, u_3\}$, $u \in [u_1, u_2]$ sin pasar por el polígono de control?



Algoritmo de De Boor para parábolas

¿Cómo se evalúa una parábola $c(u)$, $\{d_0, d_1, d_2\}$, $\{u_0, u_1, u_2, u_3\}$, $u \in [u_1, u_2]$ sin pasar por el polígono de control?



Algoritmo de De Boor

¿Cómo se evalúa una curva $c(u)$ de un tramo de grado n , de polígono B-spline $\{d_0, \dots, d_n\}$ y lista de nudos $\{u_0, \dots, u_{2n-1}\}$, $u \in [u_{n-1}, u_n]$?

El **algoritmo de De Boor** consiste en la aplicación del algoritmo de De Casteljaou en la nueva base:

$$\begin{aligned}d_i^{(1)}(u) &:= c[u_{i+1}, \dots, u_{i+n-1}, u], \quad i = 0, \dots, n-1, \\ &= \frac{u_{i+n} - u}{u_{i+n} - u_i} d_i + \frac{u - u_i}{u_{i+n} - u_i} d_{i+1},\end{aligned}$$

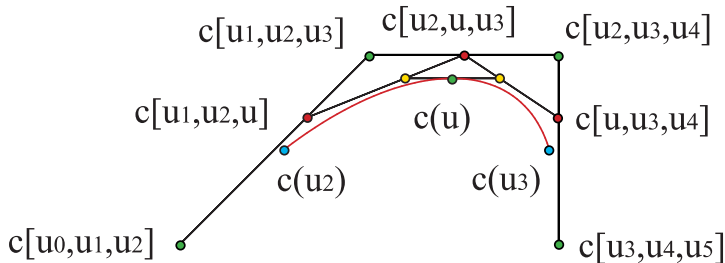
$$\begin{aligned}d_i^{(r)}(u) &:= c[u_{i+r}, \dots, u_{i+n-1}, u^{<r>}], \quad i = 0, \dots, n-r, \quad r = 1, \dots, n, \\ &= \frac{u_{i+n} - u}{u_{i+n} - u_{i+r-1}} d_i^{(r-1)}(u) + \frac{u - u_{i+r-1}}{u_{i+n} - u_{i+r-1}} d_{i+1}^{(r-1)}(u),\end{aligned}$$

$$d_0^{(n)}(u) := c[u^{<n>}] = \frac{u_n - u}{u_n - u_{n-1}} d_0^{(n-1)}(u) + \frac{u - u_{n-1}}{u_n - u_{n-1}} d_1^{(n-1)}(u).$$



Algoritmo de De Boor

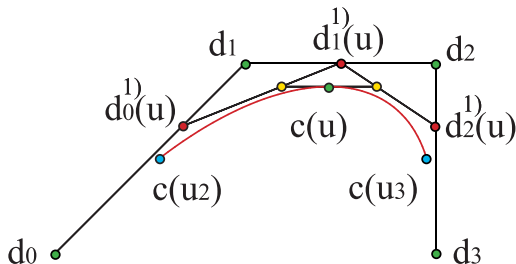
¿Cómo se evalúa una curva $c(u)$ de un tramo de grado n , de polígono B-spline $\{d_0, \dots, d_n\}$ y lista de nudos $\{u_0, \dots, u_{2n-1}\}$, $u \in [u_{n-1}, u_n]$?
El **algoritmo de De Boor** consiste en la aplicación del algoritmo de De Casteljaou en la nueva base:



Algoritmo de De Boor

¿Cómo se evalúa una curva $c(u)$ de un tramo de grado n , de polígono B-spline $\{d_0, \dots, d_n\}$ y lista de nudos $\{u_0, \dots, u_{2n-1}\}$, $u \in [u_{n-1}, u_n]$?

El **algoritmo de De Boor** consiste en la aplicación del algoritmo de De Casteljaou en la nueva base:



La polarización de una curva no es más que la aplicación del algoritmo de De Boor a un valor distinto en cada paso:

$$\begin{aligned}d_i^{(1)}[v_1] &:= c[u_{i+1}, \dots, u_{i+n-1}, v_1], \\ &= \frac{u_{i+n} - v_1}{u_{i+n} - u_i} d_i + \frac{v_1 - u_i}{u_{i+n} - u_i} d_{i+1}, \quad i = 0, \dots, n-1,\end{aligned}$$

$$\begin{aligned}d_i^{(r)}[v_1, \dots, v_r] &:= c[u_{i+r}, \dots, u_{i+n-1}, v_1, \dots, v_r] \quad i = 0, \dots, n-r, \\ &= \frac{u_{i+n} - v_r}{u_{i+n} - u_{i+r-1}} d_i^{(r-1)}[v_1, \dots, v_{r-1}] \\ &\quad + \frac{v_r - u_{i+r-1}}{u_{i+n} - u_{i+r-1}} d_{i+1}^{(r-1)}[v_1, \dots, v_{r-1}],\end{aligned}$$

$$\begin{aligned}d[v_1, \dots, v_n] &:= d_0^{(n)}[v_1, \dots, v_n] = \frac{u_n - v_n}{u_n - u_{n-1}} d_0^{(n-1)}[v_1, \dots, v_{n-1}] \\ &\quad + \frac{v_n - u_{n-1}}{u_n - u_{n-1}} d_1^{(n-1)}[v_1, \dots, v_{n-1}] = c[v_1, \dots, v_n].\end{aligned}$$

¡Sigue siendo simétrica!



- La polarización devuelve los valores correctos de los vértices del polígono,

$$d[u_i, \dots, u_{i+n-1}] = c[u_i, \dots, u_{i+n-1}] = d_i.$$

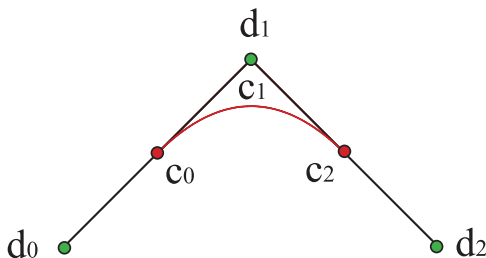
- Pasar de $\{c_0, \dots, c_n\}$ a $\{d_0, \dots, d_n\}$ es sólo un cambio de base de funciones. Pero la polarización sigue siendo la misma.
- Dado que $d_i = c[u_i, \dots, u_{i+n-1}]$ y $c_i = c[u_{n-1}^{<n-i>}, u_n^{<i>}]$, podemos pasar de una representación a otra.
- Si tenemos el polígono B-spline de una curva de grado n , $\{d_0, \dots, d_n\}$, su polígono de control vendrá dado por

$$c_i = c[u_{n-1}^{<n-i>}, u_n^{<i>}] = d[u_{n-1}^{<n-i>}, u_n^{<i>}].$$



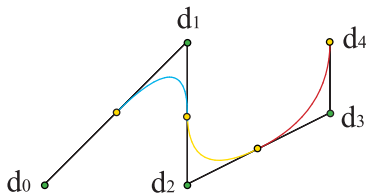
- Si tenemos el polígono B-spline de una curva de grado n , $\{d_0, \dots, d_n\}$, su polígono de control vendrá dado por

$$c_i = c[u_{n-1}^{<n-i>}, u_n^{<i>}] = d[u_{n-1}^{<n-i>}, u_n^{<i>}].$$



Algoritmo de De Boor de N tramos

- El algoritmo de De Boor se extiende a curvas de N tramos de grado n , polígono $\{d_0, \dots, d_{n+N-1}\}$ y nudos $\{u_0, \dots, u_{2n+N-2}\}$.
- 1º tramo: $[u_{n-1}, u_n]$, polígono $\{d_0, \dots, d_n\}$, nudos $\{u_0, \dots, u_{2n-1}\}$.
- 2º tramo: $[u_n, u_{n+1}]$, polígono $\{d_1, \dots, d_{n+1}\}$, nudos $\{u_1, \dots, u_{2n}\}$.
- i tramo: $[u_{n+i-2}, u_{n+i-1}]$, polígono $\{d_{i-1}, \dots, d_{n+i-1}\}$, nudos $\{u_{i-1}, \dots, u_{2n+i-2}\}$.
- N tramo: $[u_{n+N-2}, u_{n+N-1}]$, polígono $\{d_{N-1}, \dots, d_{n+N-1}\}$, nudos $\{u_{N-1}, \dots, u_{2n+N-2}\}$.
- Para evaluar la curva en $u \in [u_l, u_{l+1})$, se aplica el algoritmo a $\{\tilde{d}_0 := d_{l-n+1}, \dots, \tilde{d}_n := d_{l+1}\}$ en $\tilde{u}_0 := u_{l-n+1}, \dots, \tilde{u}_{2n-1} := u_{l+n}$.



Algoritmo de De Boor de N tramos

- El algoritmo de De Boor se extiende a curvas de N tramos de grado n , polígono $\{d_0, \dots, d_{n+N-1}\}$ y nudos $\{u_0, \dots, u_{2n+N-2}\}$.
- 1º tramo: $[u_{n-1}, u_n]$, polígono $\{d_0, \dots, d_n\}$, nudos $\{u_0, \dots, u_{2n-1}\}$.
- 2º tramo: $[u_n, u_{n+1}]$, polígono $\{d_1, \dots, d_{n+1}\}$, nudos $\{u_1, \dots, u_{2n}\}$.
- i tramo: $[u_{n+i-2}, u_{n+i-1}]$, polígono $\{d_{i-1}, \dots, d_{n+i-1}\}$, nudos $\{u_{i-1}, \dots, u_{2n+i-2}\}$.
- N tramo: $[u_{n+N-2}, u_{n+N-1}]$, polígono $\{d_{N-1}, \dots, d_{n+N-1}\}$, nudos $\{u_{N-1}, \dots, u_{2n+N-2}\}$.
- Para evaluar la curva en $u \in [u_l, u_{l+1})$, se aplica el algoritmo a $\{\tilde{d}_0 := d_{l-n+1}, \dots, \tilde{d}_n := d_{l+1}\}$ en $\tilde{u}_0 := u_{l-n+1}, \dots, \tilde{u}_{2n-1} := u_{l+n}$.
- La extensión a curvas racionales se realiza añadiendo una coordenada más a los vértices, $\mathbf{d}_i = (w_i, w_i d_i)$. Hay que incluir como dato la lista de pesos, $\{w_0, \dots, w_L\}$.



Propiedades antiguas

Sea una curva de grado n , polígono $\{d_0, \dots, d_L\}$ y nudos $\{u_0, \dots, u_K\}$ (y pesos $\{w_0, \dots, w_L\}$) definida en el intervalo $[u_{n-1}, u_{n+N-1}]$.

Se mantienen las siguientes propiedades por usar combinaciones baricéntricas:

- **Invariancia bajo transformaciones afines:** bajo una transformación f afín la curva $c(u)$ se convierte en $f(c(u))$ de polígono $\{f(d_0), \dots, f(d_L)\}$. Los pesos no cambian.
- **Invariancia bajo transformaciones proyectivas:** bajo una transformación f proyectiva la curva $c(u)$ se convierte en $f(c(u))$ de polígono vectorial $\{f(\mathbf{d}_0), \dots, f(\mathbf{d}_L)\}$, donde $\mathbf{d}_i = (w_i, w_i d_i)$
- **Invariancia bajo reparametrización afín:** podemos modificar nudos $\tilde{u}_i = au_i + b$ sin alterar la curva ($\tilde{u} = au + b$).
- **Extremos:** $d_0 = c(u_{n-1})$, $d_L = c(u_{n+N-1})$ si $u_0 = \dots = u_{n-1}$, $u_{n+N-1} = \dots = u_{2n+N-2}$.



Propiedades antiguas

Sea una curva de grado n , polígono $\{d_0, \dots, d_L\}$ y nudos $\{u_0, \dots, u_K\}$ (y pesos $\{w_0, \dots, w_L\}$) definida en el intervalo $[u_{n-1}, u_{n+N-1}]$.

Se mantienen las siguientes propiedades por usar combinaciones baricéntricas:

- **Extremos:** $d_0 = c(u_{n-1})$, $d_L = c(u_{n+N-1})$ si $u_0 = \dots = u_{n-1}$,
 $u_{n+N-1} = \dots = u_{2n+N-2}$.

(fig412.mov)

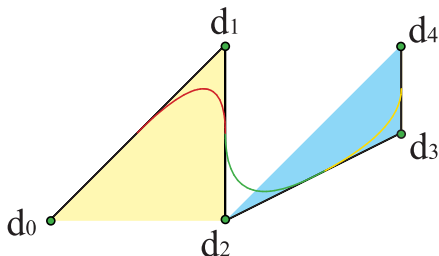


Propiedades mejoradas

Sea una curva de grado n , polígono $\{d_0, \dots, d_L\}$ y nudos $\{u_0, \dots, u_K\}$ (y pesos $\{w_0, \dots, w_L\}$) definida en el intervalo $[u_{n-1}, u_{n+N-1}]$.

Mejoran las siguientes propiedades:

- **Envolvente convexa:** *cada* tramo de la curva está contenido en el menor polígono convexo generado por *su* polígono.

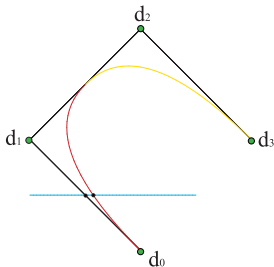


Propiedades mejoradas

Sea una curva de grado n , polígono $\{d_0, \dots, d_L\}$ y nudos $\{u_0, \dots, u_K\}$ (y pesos $\{w_0, \dots, w_L\}$) definida en el intervalo $[u_{n-1}, u_{n+N-1}]$.

Mejoran las siguientes propiedades:

- **Disminución de la variación:** Una recta corta siempre a *cada* tramo de la curva en a lo sumo tantos puntos como a *su* parte de polígono.



Propiedades mejoradas

Sea una curva de grado n , polígono $\{d_0, \dots, d_L\}$ y nudos $\{u_0, \dots, u_K\}$ (y pesos $\{w_0, \dots, w_L\}$) definida en el intervalo $[u_{n-1}, u_{n+N-1}]$.

Mejoran las siguientes propiedades:

- **Disminución de la variación:** Una recta corta siempre a *cada* tramo de la curva en a lo sumo tantos puntos como a *su* parte de polígono.

(fig434.mov)

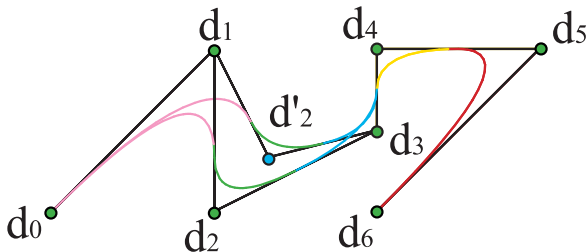


Propiedades mejoradas

Sea una curva de grado n , polígono $\{d_0, \dots, d_L\}$ y nudos $\{u_0, \dots, u_K\}$ (y pesos $\{w_0, \dots, w_L\}$) definida en el intervalo $[u_{n-1}, u_{n+N-1}]$.

Mejoran las siguientes propiedades:

- **Control local:** cada tramo depende sólo de los $n + 1$ vértices de su polígono. Mover d_i modifica tan sólo a un total de $n + 1$ tramos de la curva. Por ejemplo, el vértice inicial, d_0 , sólo afecta al primer tramo.



Propiedades mejoradas

Sea una curva de grado n , polígono $\{d_0, \dots, d_L\}$ y nudos $\{u_0, \dots, u_K\}$ (y pesos $\{w_0, \dots, w_L\}$) definida en el intervalo $[u_{n-1}, u_{n+N-1}]$.

Mejoran las siguientes propiedades:

- **Control local:** cada tramo depende sólo de los $n + 1$ vértices de su polígono. Mover d_i modifica tan sólo a un total de $n + 1$ tramos de la curva. Por ejemplo, el vértice inicial, d_0 , sólo afecta al primer tramo.

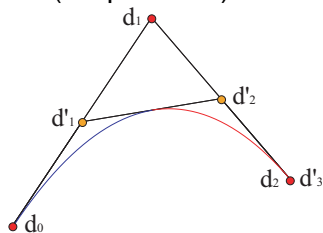
(fig415.mov)



Algoritmo de inserción de nudos

Sea una curva de grado n , polígono $\{d_0, \dots, d_L\}$ y nudos $\{u_0, \dots, u_K\}$.

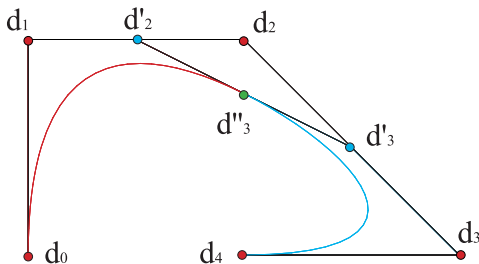
- Introducimos un nudo adicional \tilde{u} sin alterar la curva.
- Supone añadir un vértice, dividiendo un tramo.
- En la nueva lista $\{\tilde{u}_0, \dots, \tilde{u}_{2n+N-1}\}$, el nuevo nudo es el i -ésimo, $\{u_0, \dots, u_{i-1}, \tilde{u}, u_i, \dots, u_K\}$ (cambia la numeración sólo). Los nuevos tramos son $[\tilde{u}_{i-1}, \tilde{u}_i]$, $[\tilde{u}_i, \tilde{u}_{i+1}]$.
- Nuevos vértices (polarización): $\tilde{d}_j = d[\tilde{u}_j, \dots, \tilde{u}_{j+n-1}]$, $j = 0, \dots, n + N$.
- Sólo se alteran los n vértices en los que aparece \tilde{u} .
- El resto no se alteran (los primeros) o se reenumeran (los últimos).



Polígono B-spline v. Polígonos de control

Sea una curva de grado n , polígono $\{d_0, \dots, d_L\}$ y nudos $\{u_0, \dots, u_K\}$.

- Aplicamos el algoritmo de inserción de los nudos interiores u_i, u_{i+1} hasta que alcancen multiplicidad n .
- Obtenemos el polígono de control en el intervalo $[u_i, u_{i+1}]$.
- La lista de nudos del intervalo será $\{u_i^{<n>}, u_{i+1}^{<n>}\}$ que corresponde a una curva de Bézier de vértices $c_i = d[u_i^{<n-i>}, u_{i+1}^{<i>}]$.



Polígono B-spline v. Polígonos de control

Sea una curva de grado n , polígono $\{d_0, \dots, d_L\}$ y nudos $\{u_0, \dots, u_K\}$.

- Aplicamos el algoritmo de inserción de los nudos interiores u_i, u_{i+1} hasta que alcancen multiplicidad n .
- Obtenemos el polígono de control en el intervalo $[u_i, u_{i+1}]$.
- La lista de nudos del intervalo será $\{u_i^{<n>}, u_{i+1}^{<n>}\}$ que corresponde a una curva de Bézier de vértices $c_i = d[u_i^{<n-i>}, u_{i+1}^{<i>}]$.

(fig417.mov)



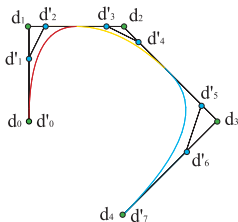
Elevación del grado

- La fórmula de elevación se puede usar con curvas *spline*,

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

- Hay que pasar de listas $\{u_0^{<n>}, u_1^{<n>}\}$ a $\{u_0^{<n+1>}, u_1^{<n+1>}\}$.
- Al elevar el grado, hay que incrementar la multiplicidad de los nudos interiores desde u_{n-1} hasta u_{n+N-1} en una unidad. Si son simples, acabamos con $2n + 2N$ nudos y $n + 2N$ vértices,

$$\{u_0, \dots, u_{n-1}, u_{n-1}, \dots, u_{n+N-1}, u_{n+N-1}, u_{n+N}, \dots, u_{2n+N-2}\},$$



Elevación del grado

- La fórmula de elevación se puede usar con curvas *spline*,

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

- Hay que pasar de listas $\{u_0^{<n>}, u_1^{<n>}\}$ a $\{u_0^{<n+1>}, u_1^{<n+1>}\}$.
- Al elevar el grado, hay que incrementar la multiplicidad de los nudos interiores desde u_{n-1} hasta u_{n+N-1} en una unidad. Si son simples, acabamos con $2n + 2N$ nudos y $n + 2N$ vértices,

$$\{u_0, \dots, u_{n-1}, u_{n-1}, \dots, u_{n+N-1}, u_{n+N-1}, u_{n+N}, \dots, u_{2n+2N-2}\},$$

(fig419.mov)

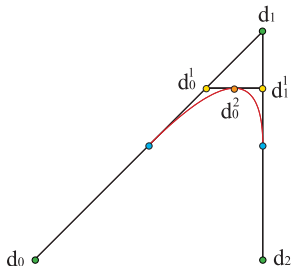


- La derivada de una curva *spline* en un tramo $[u_i, u_{i+1}]$

$$\frac{dc(u)}{du} = \frac{n}{u_{i+1} - u_i} \left(d_1^{n-1}(u) - d_0^{n-1}(u) \right),$$

está determinada por los vértices del último paso del algoritmo.

- Las derivadas superiores se definen por diferencias sucesivas.



- La derivada de una curva *spline* en un tramo $[u_i, u_{i+1}]$

$$\frac{dc(u)}{du} = \frac{n}{u_{i+1} - u_i} \left(d_1^{n-1}(u) - d_0^{n-1}(u) \right),$$

está determinada por los vértices del último paso del algoritmo.

- Las derivadas superiores se definen por diferencias sucesivas.

(fig432.mov)

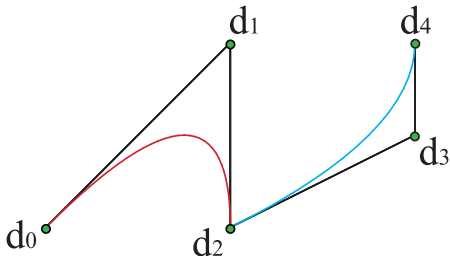


- Fijémosnos en la unión de dos tramos $[u_{i-1}, u_i]$, $[u_i, u_{i+1}]$ de una curva de grado n .
- Supongamos que los nudos interiores no están repetidos.
- El primero tiene nudos $\{u_{i-n}, \dots, u_{i+n-1}\}$ y polígono $\{d_{i-n}, \dots, d_i\}$
- El segundo, $\{u_{i-n+1}, \dots, u_{i+n}\}$, $\{d_{i-n+1}, \dots, d_{i+1}\}$.
- Los nudos y vértices comunes, $\{u_{i-n+1}, \dots, u_{i+n-1}\}$, $\{d_{i-n+1}, \dots, d_i\}$ proporcionan $n - 1$ derivadas iguales.
- Por tanto, si no hay nudos interiores repetidos, la curva es C^{n-1} .



Nudos repetidos

- Consideremos ahora la unión de dos tramos en u_i , nudo doble.
- La unión es de dos tramos $[u_{i-1}, u_i]$, $[u_{i+1}, u_{i+2}]$.
- El primero tiene nudos $\{u_{i-n}, \dots, u_{i+n-1}\}$ y polígono $\{d_{i-n}, \dots, d_i\}$
- El segundo, $\{u_{i-n+2}, \dots, u_{i+n+1}\}$, $\{d_{i-n+2}, \dots, d_{i+2}\}$.
- Los nudos y vértices comunes, $\{u_{i-n+2}, \dots, u_{i+n-1}\}$, $\{d_{i-n+2}, \dots, d_i\}$ proporcionan $n - 2$ derivadas iguales.
- La repetición de un nudo rebaja una unidad la derivabilidad en la unión



Nudos repetidos

- Consideremos ahora la unión de dos tramos en u_i , nudo doble.
- La unión es de dos tramos $[u_{i-1}, u_i]$, $[u_{i+1}, u_{i+2}]$.
- El primero tiene nudos $\{u_{i-n}, \dots, u_{i+n-1}\}$ y polígono $\{d_{i-n}, \dots, d_i\}$
- El segundo, $\{u_{i-n+2}, \dots, u_{i+n+1}\}$, $\{d_{i-n+2}, \dots, d_{i+2}\}$.
- Los nudos y vértices comunes, $\{u_{i-n+2}, \dots, u_{i+n-1}\}$, $\{d_{i-n+2}, \dots, d_i\}$ proporcionan $n - 2$ derivadas iguales.
- La repetición de un nudo rebaja una unidad la derivabilidad en la unión

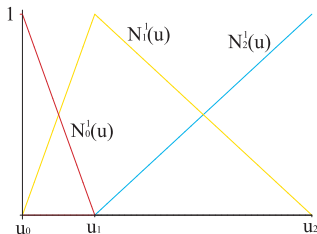
(fig421.mov)



- Expresar las curvas, $c(u) = \sum_{i=0}^L d_i N_i^n(u)$, con $\{N_0^n(u), \dots, N_L^n(u)\}$, funciones polinómicas de N trozos de grado n sobre $\{u_0, \dots, u_K\}$.
- Se llaman **funciones nodales** y son recursivas,

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u),$$

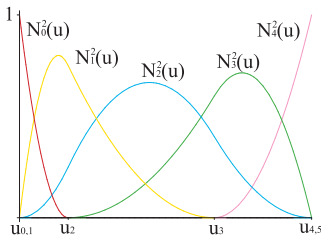
$$N_i^0(u) = \begin{cases} 1 & u \in [u_{i-1}, u_i] \\ 0 & u \notin [u_{i-1}, u_i] \end{cases} \quad \text{¿índices negativos?}$$



- Expresar las curvas, $c(u) = \sum_{i=0}^L d_i N_i^n(u)$, con $\{N_0^n(u), \dots, N_L^n(u)\}$, funciones polinómicas de N trozos de grado n sobre $\{u_0, \dots, u_K\}$.
- Se llaman **funciones nodales** y son recursivas,

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u),$$

$$N_i^0(u) = \begin{cases} 1 & u \in [u_{i-1}, u_i] \\ 0 & u \notin [u_{i-1}, u_i] \end{cases} \quad \text{¿índices negativos?}$$



- Expresar las curvas, $c(u) = \sum_{i=0}^L d_i N_i^n(u)$, con $\{N_0^n(u), \dots, N_L^n(u)\}$, funciones polinómicas de N trozos de grado n sobre $\{u_0, \dots, u_K\}$.
- Se llaman **funciones nodales** y son recursivas,

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u),$$

$$N_i^0(u) = \begin{cases} 1 & u \in [u_{i-1}, u_i) \\ 0 & u \notin [u_{i-1}, u_i) \end{cases} \quad \text{¿índices negativos?}$$

(fig430.mov)

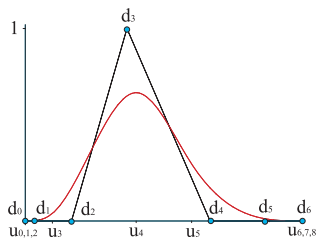


Propiedades de las funciones nodales

- **Partición de la unidad:** $\sum_{i=0}^L N_i^n(u) \equiv 1$. Y son positivas.
- **Base:** son linealmente independientes y forman base.
- **Polígono de control de $N_i^n(u)$:** $\{d_0^i, \dots, d_{n+N-1}^i\}$, donde

$$d_j^i = (\xi_j, \delta_j^i), \quad \xi_i = \frac{1}{n} \sum_{j=i}^{i+n-1} u_j \text{ (abscisas de Greville).}$$

- **Soporte mínimo:** la función $N_i^n(u)$ se anula fuera de $[u_{i-1}, u_{i+n}]$ ($n + 1$ intervalos).

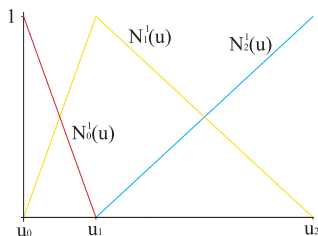


Propiedades de las funciones nodales

- **Partición de la unidad:** $\sum_{i=0}^L N_i^n(u) \equiv 1$. Y son positivas.
- **Base:** son linealmente independientes y forman base.
- **Polígono de control de $N_i^n(u)$:** $\{d_0^i, \dots, d_{n+N-1}^i\}$, donde

$$d_j^i = (\xi_j, \delta_j^i), \quad \xi_i = \frac{1}{n} \sum_{j=i}^{i+n-1} u_j \text{ (abscisas de Greville).}$$

- **Soporte mínimo:** la función $N_i^n(u)$ se anula fuera de $[u_{i-1}, u_{i+n}]$ ($n+1$ intervalos).

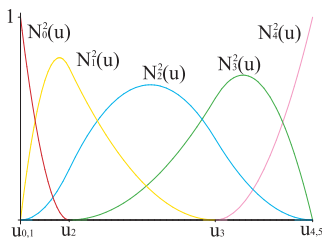


Propiedades de las funciones nodales

- **Partición de la unidad:** $\sum_{i=0}^L N_i^n(u) \equiv 1$. Y son positivas.
- **Base:** son linealmente independientes y forman base.
- **Polígono de control de $N_i^n(u)$:** $\{d_0^i, \dots, d_{n+N-1}^i\}$, donde

$$d_j^i = (\xi_j, \delta_j^i), \quad \xi_i = \frac{1}{n} \sum_{j=i}^{i+n-1} u_j \text{ (abscisas de Greville).}$$

- **Soporte mínimo:** la función $N_i^n(u)$ se anula fuera de $[u_{i-1}, u_{i+n}]$ ($n + 1$ intervalos).

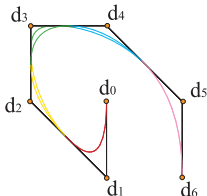


Splines racionales

- Aplicar el algoritmo de De Boor a curvas racionales supone introducir el peso (inserción de nudos, elevación del grado...).
- Una curva de grado n a N trozos tiene vértices, $\{d_0, \dots, d_{n+N-1}\}$, pesos, $\{w_0, \dots, w_{n+N-1}\}$, nudos, $\{u_0, \dots, u_{2n+N-2}\}$,

$$c(u) = \frac{\sum_{i=0}^{n+N-1} w_i d_i N_i^n(u)}{\sum_{i=0}^{n+N-1} w_i N_i^n(u)}$$

- Modificar un peso altera tan sólo a $n + 1$ tramos de la curva.



- Aplicar el algoritmo de De Boor a curvas racionales supone introducir el peso (inserción de nudos, elevación del grado...).
- Una curva de grado n a N trozos tiene vértices, $\{d_0, \dots, d_{n+N-1}\}$, pesos, $\{w_0, \dots, w_{n+N-1}\}$, nudos, $\{u_0, \dots, u_{2n+N-2}\}$,

$$c(u) = \frac{\sum_{i=0}^{n+N-1} w_i d_i N_i^n(u)}{\sum_{i=0}^{n+N-1} w_i N_i^n(u)}.$$

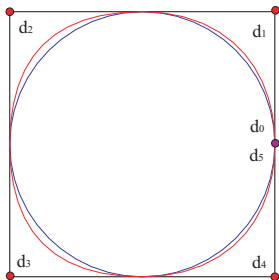
- Modificar un peso altera tan sólo a $n + 1$ tramos de la curva.

(fig426.mov)



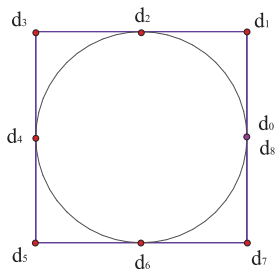
Falsas circunferencias B-spline

- Una circunferencia de 4 tramos debería tener polígono $\{(R, 0), (R, R), (-R, R), (-R, -R), (R, -R), (R, 0)\}$, pesos $\{1, 1/\sqrt{2}, 1/\sqrt{2}, 1/\sqrt{2}, 1/\sqrt{2}, 1\}$ y nudos $[0, 0, 1, 2, 3, 4, 4]$.
- El problema está en los pesos $w_1 = w_2 = w_3 = w_4$, que producen tramos parabólicos.



Correctas circunferencias B-spline

- La solución correcta es usar nudos dobles para trabajar con *splines* C^0 (9 vértices, 10 nudos).
- Una circunferencia de 4 tramos tiene polígono $\{(R, 0), (R, R), (0, R), (-R, R), (-R, 0), (-R, -R), (0, -R), (R, -R), (R, 0)\}$, pesos $\{1, 1/\sqrt{2}, 1, 1/\sqrt{2}, 1, 1/\sqrt{2}, 1, 1/\sqrt{2}, 1\}$ y nudos $[0, 0, 1, 1, 2, 2, 3, 3, 4, 4]$.
- Las tangentes son continuas no obstante.



- *Spline* = junquillo.
- La posición del junquillo, descrita por una curva $c(s)$, viene determinada por el principio de mínima energía elástica,

$$E[c] = \int_a^b ds \kappa^2(s) = \int_a^b ds \|\ddot{c}(s)\|^2,$$

en la parametrización por longitud de arco.

- Linealizando,

$$\tilde{E}[c] = \int_a^b du \|c''(u)\|^2.$$

- El *spline* cúbico de clase C^2 , $c(u)$, minimiza \tilde{E} .

