

B- splines

Método para construir curvas polinomiales a trozos.

Ventajas: {

- el grado de la curva se escoge a priori.
- tienen carácter local: cada vértice de control va asociado a una única función de base, que tiene soporte local.

Datos: {

- orden de la curva k (la curva está formada por polinomios a trozos de grado k)
- vértices de control B_0, B_1, \dots, B_n (puntos del plano o del espacio).
- vector de nodos (knots) $[x_0, x_1, \dots, x_{n+k+1}]$, tales que $x_i \in \mathbf{R}$, $x_i \leq x_{i+1}$.

Notas: los nodos se pueden repetir.
los vértices se pueden repetir.

obs: $k \leq n$

B- splines

Definición: dados el orden k , los vértices de control B_0, \dots, B_n , y el vector de nodos $[x_0, \dots, x_{n+k+1}]$, definimos la curva B-spline asociada a estos datos como

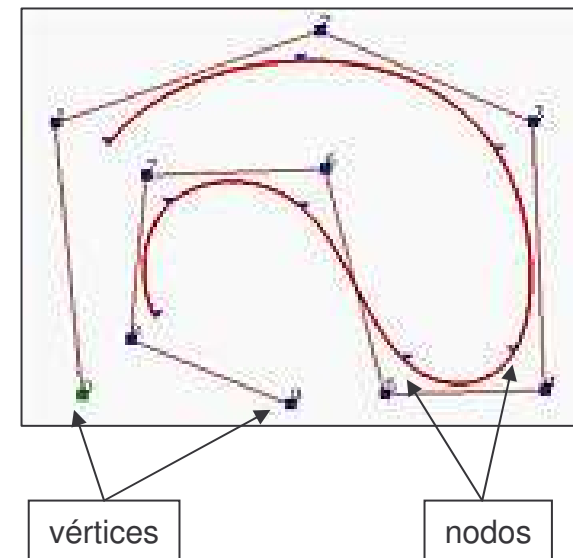
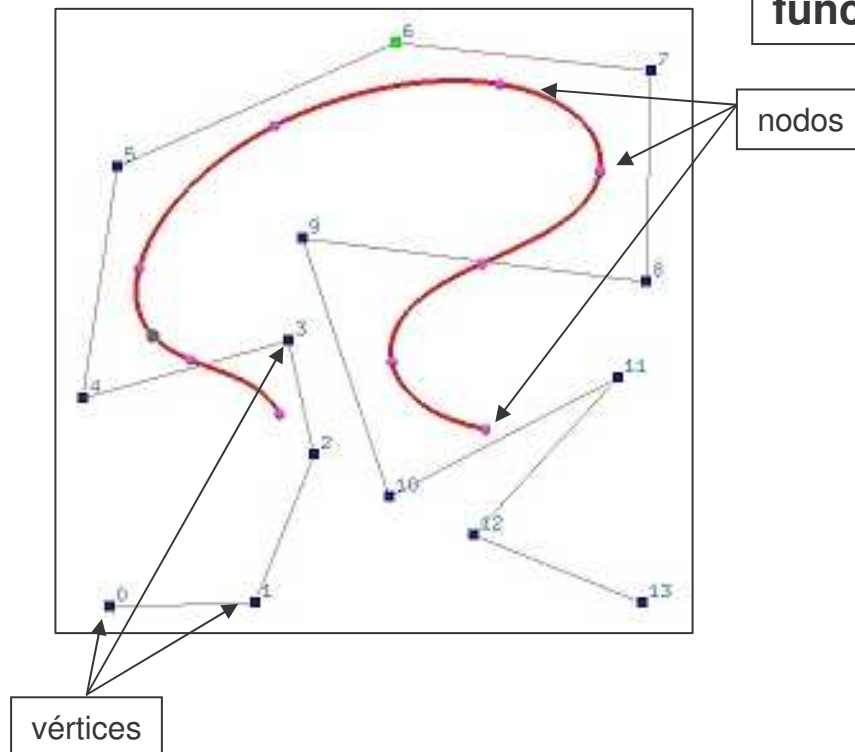
$$P(t) = \sum_{i=0}^n B_i N_i^k(t)$$

donde

$$t \in [x_k, x_{n+1}]$$

funciones de base

dominio de $P(t)$



Funciones de base

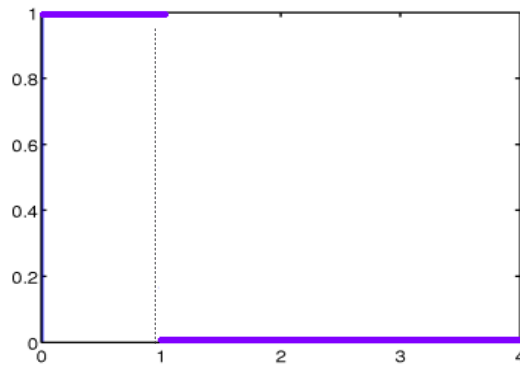
obs: si algún denominador es 0 se entiende que ese sumando es 0.

FÓRMULA DE COX-DE BOOR:

$$N_i^0(t) = \begin{cases} 1 & \text{si } t \in [x_i, x_{i+1}] \\ 0 & \text{en el resto} \end{cases}$$
$$N_i^r(t) = \frac{(t-x_i)N_i^{r-1}(t)}{x_{i+r}-x_i} + \frac{(x_{i+r+1}-t)N_{i+1}^{r-1}(t)}{x_{i+r+1}-x_{i+1}} \quad r \geq 1$$

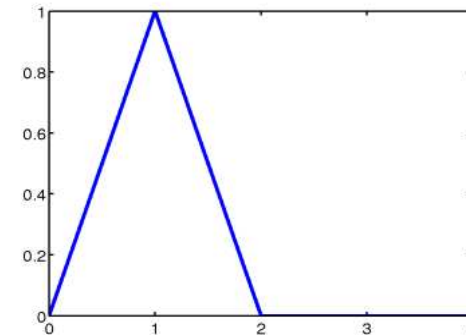
Ejemplos: vector de nodos $[x_0=0, x_1=1, x_2=2, \dots]$

k=0



$$B_0^0(x) = \begin{cases} 1 & x \in [0, 1), \\ 0 & \text{otherwise.} \end{cases}$$

k=1

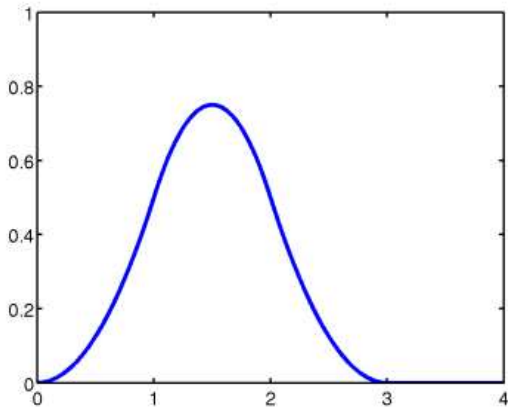


$$B_0^1(x) = \begin{cases} x & x \in [0, 1), \\ 2-x & x \in [1, 2), \\ 0 & \text{otherwise.} \end{cases}$$

Funciones de base

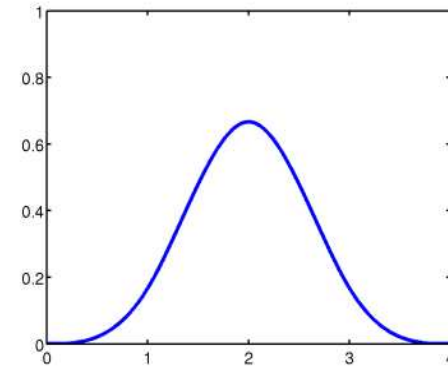
Ejemplos: vector de nodos $[x_0=0, x_1=1, x_2=2, \dots]$

k=2



$$B_0^2(x) = \begin{cases} \frac{1}{2}x^2 & x \in [0, 1), \\ -x^2 + 3x - \frac{3}{2} & x \in [1, 2), \\ \frac{1}{2}x^2 - 3x + \frac{9}{2} & x \in [2, 3), \\ 0 & \text{otherwise} \end{cases}$$

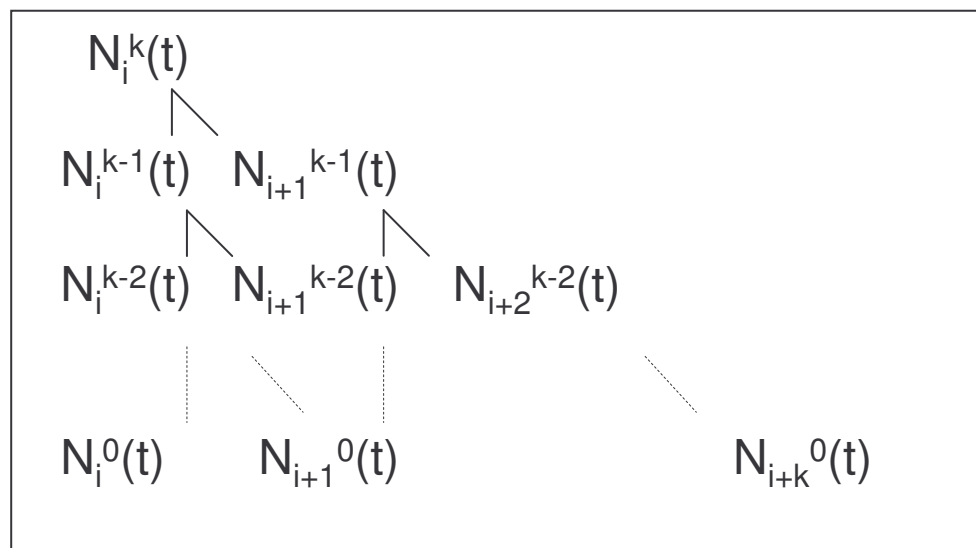
k=3



$$B_0^3(x) = \begin{cases} \frac{1}{6}x^3 & x \in [0, 1), \\ -\frac{1}{2}x^3 + 2x^2 - 2x + \frac{2}{3} & x \in [1, 2), \\ \frac{1}{2}x^3 - 4x^2 + 10x - \frac{22}{3} & x \in [2, 3), \\ -\frac{1}{6}x^3 + 2x^2 - 8x + \frac{32}{3} & x \in [3, 4), \\ 0 & \text{otherwise} \end{cases}$$

Relaciones de dependencia de las funciones de base

En el siguiente cuadro se exponen las relaciones de dependencia entre las funciones de base de distintos órdenes. En concreto, explicamos cómo calcular $N_i^k(t)$:



Como la función $N_i^k(t)$ depende de $N_i^0(t), N_{i+1}^0(t), \dots, N_{i+k}^0(t)$, su soporte coincidirá con la unión de los soportes de estas funciones de base, es decir,

el soporte de $N_i^k(t)$ es (x_i, x_{i+k+1})

(ver en los ejemplos)

Ejercicio: calcular y dibujar las funciones de base para $k=2$ y vector de nodos $[0, 1, 2, 3, 4, 5, 6]$.

Observar que cuando los nodos están equiespaciados, las funciones de base son copias trasladadas de la función $N_0^k(t)$.

Propiedad 1: $\sum_{i=0}^n N_i^k(t) = 1$ cuando $t \in [x_k, x_{n+1}]$

es decir, las funciones de base de orden k suman 1

(dem. por inducción)

Propiedad 2: $N_i^k(t) > 0 \Leftrightarrow t \in (x_i, x_{i+k+1})$ y en el resto valen 0

Propiedad 3: Para $k > 0$, $N_i^k(t)$ tiene un único valor máximo.

Vectores de nodos

Habitualmente se utilizan tres tipos de vectores de nodos:

- vectores de nodos uniformes o equiespaciados
- vectores de nodos "clamped" o agarrados
- vectores de nodos no uniformes (de ninguno de los tipos anteriores)

(en design mentor se usan closed knots, pero esto se verá más adelante)

Vectores de nodos uniformes

Los puntos x_i aparecen equiespaciados. En la práctica se suelen escribir de uno de los dos siguientes modos:

1. Empezando por cero y de modo que $x_i = x_{i-1} + 1$.

Ej: [0, 1, 2, 3, 4]

2. De forma que $x_0 = 0$, $x_{n+k+1} = 1$ y los demás estén repartidos en intervalos iguales entre 0 y 1.

Ej: [0, 0.25, 0.5, 0.75, 1]

Las funciones de base son copias trasladadas de la función $N_0^k(t)$:

$$N_i^k(t) = N_0^k(t-i), \text{ para } i > 0$$

¡ventaja en el cómputo de las funciones de base!

Vectores de nodos "clamped" o agarrados

Los nodos del inicio y del final del vector de nodos tienen multiplicidad $k+1$, y los nodos interiores están equiespaciados. Si el vector de nodos es $[x_0, \dots, x_{n+k+1}]$, se suele tomar:

$$x_i=0, \text{ para } 0 \leq i \leq k$$

$$x_i=i-k, \text{ para } k+1 \leq i \leq n$$

$$x_i=n-k+1, \text{ para } n+1 \leq i \leq n+k+1$$

Ejemplos:

- $k=1$: $[0, 0, 1, 2, 2]$

- $k=2$: $[0, 0, 0, 1, 1, 1]$

- $k=3$: $[0, 0, 0, 0, 1, 2, 3, 3, 3, 3]$

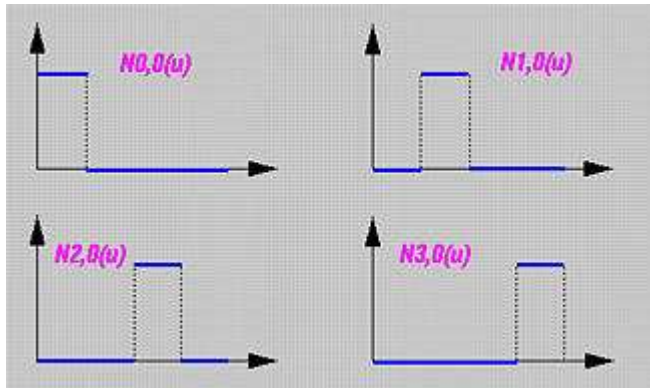
Caso particular: Si el vector de nodos es "clamped" y $k=n$, entonces las funciones de base coinciden con los polinomios de Bernstein.

Ejercicio: calcular $N_i^k(t)$ y observar que sale el polinomio de Bernstein $B_i^n(t)$ con $t \in [0,1]$. Nota: toma, por ej., $k = 2$ ó 3 .

Funciones de base con vectores de nodos uniformes

k=0 (grado 0)

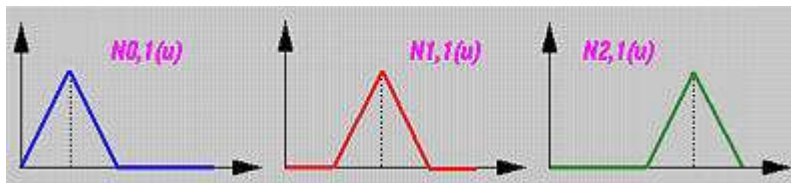
vector de nodos [0, 0.25, 0.5, 0.75, 1]



Func. de base	Soporte	Ecuación
$N_{0,0}(u)$	(0, 0.25)	1
$N_{1,0}(u)$	(0.25, 0.5)	1
$N_{2,0}(u)$	(0.5, 0.75)	1
$N_{3,0}(u)$	(0.75, 1)	1

k=1 (grado 1)

vector de nodos [0, 0.25, 0.5, 0.75, 1]

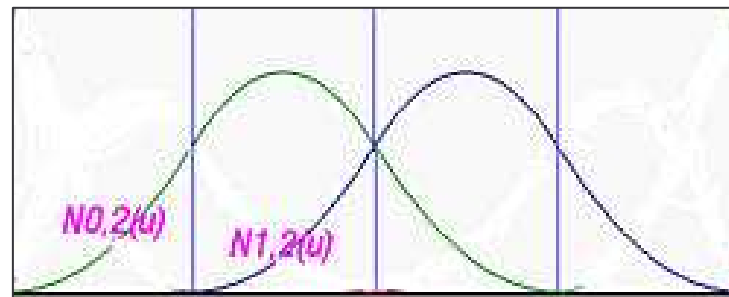


Func. de base	Soporte	Ecuación
$N_{0,1}(u)$	(0, 0.25)	$4u$
	[0.25, 0.5)	$2(1 - 2u)$
$N_{1,1}(u)$	(0.25, 0.5)	$4u - 1$
	[0.5, 0.75)	$3 - u$
$N_{2,1}(u)$	(0.5, 0.75)	$2(2u - 1)$
	[0.75, 1)	$4(1 - u)$

Funciones de base con vectores de nodos uniformes

k=2 (grado 2)

vector de nodos [0, 0.25, 0.5, 0.75, 1]

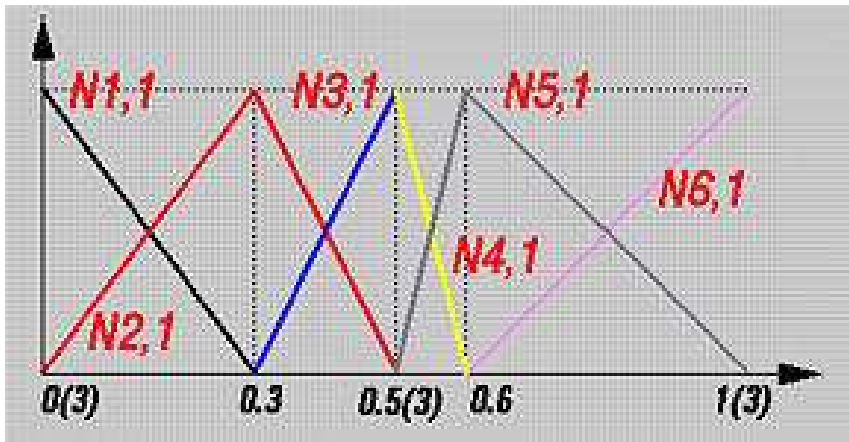


<i>Func. de base</i>	<i>Soporte</i>	<i>Ecuación</i>
$N_{0,2}(u)$	(0, 0.25)	$8u^2$
	[0.25, 0.5)	$-1.5 + 12u - 16u^2$
	[0.5, 0.75)	$4.5 - 12u + 8u^2$
$N_{1,2}(u)$	(0.25, 0.5)	$0.5 - 4u + 8u^2$
	[0.5, 0.75)	$-1.5 + 8u - 8u^2$
	[0.75, 1)	$8(1 - u)^2$

Funciones de base con vectores de nodos no uniformes

$k=1$ (grado 1)

vector de nodos:
 $[0, 0, 0, 0.3, 0.5, 0.5, 0.5, 0.6, 1, 1, 1]$
 (3) (3) (3)

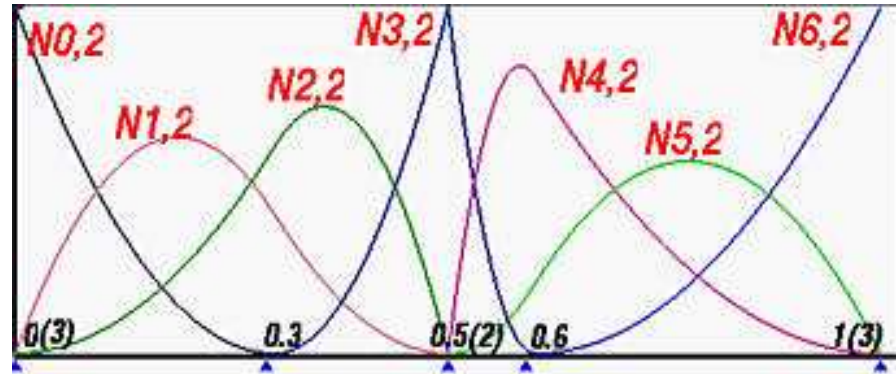


Func. de base	Soporte	Ecuación
$N_{0,1}(u)$	No	0
$N_{1,1}(u)$	$[0, 0.3)$	$1 - (10/3)u$
$N_{2,1}(u)$	$(0, 0.3)$	$(10/3)u$
	$[0.3, 0.5)$	$2.5(1 - 2u)$
$N_{3,1}(u)$	$(0.3, 0.5)$	$5u - 1.5$
$N_{4,1}(u)$	$[0.5, 0.6)$	$6 - 10u$
	$(0.5, 0.6)$	$10u - 5$
$N_{5,1}(u)$	$[0.6, 1)$	$2.5(1 - u)$
	$(0.6, 1)$	$2.5u - 1.5$
$N_{7,1}(u)$	No	0

Funciones de base con vector de nodos no uniforme

k=2 (grado 2)

vector de nodos:
 $[0, 0, 0, 0.3, 0.5, 0.5, 0.6, 1, 1, 1]$
 (3) (2) (3)



Func. de base	Soporte	Ecuación
$N_{0,2}(u)$	[0, 0.3)	$(1 - (10/3)u)^2$
$N_{1,2}(u)$	[0, 0.3)	$(20/3)(u - (8/3)u^2)$
	[0.3, 0.5)	$2.5(1 - 2u)^2$
$N_{2,2}(u)$	[0, 0.3)	$(20/3)u^2$
	[0.3, 0.5)	$-3.75 + 25u - 35u^2$
$N_{3,2}(u)$	[0.3, 0.5)	$(5u - 1.5)^2$
	[0.5, 0.6)	$(6 - 10u)^2$
$N_{4,2}(u)$	[0.5, 0.6)	$20(-2 + 7u - 6u^2)$
	[0.6, 1)	$5(1 - u)^2$
$N_{5,2}(u)$	[0.5, 0.6)	$12.5(2u - 1)^2$
	[0.6, 1)	$2.5(-4 + 11.5u - 7.5u^2)$
$N_{6,2}(u)$	[0.6, 1)	$2.5(9 - 30u + 25u^2)$

B-splines

Definición: dados el orden k , los vértices de control B_0, \dots, B_n , y el vector de nodos $[x_0, \dots, x_{n+k+1}]$, definimos la curva B-spline asociada a estos datos como

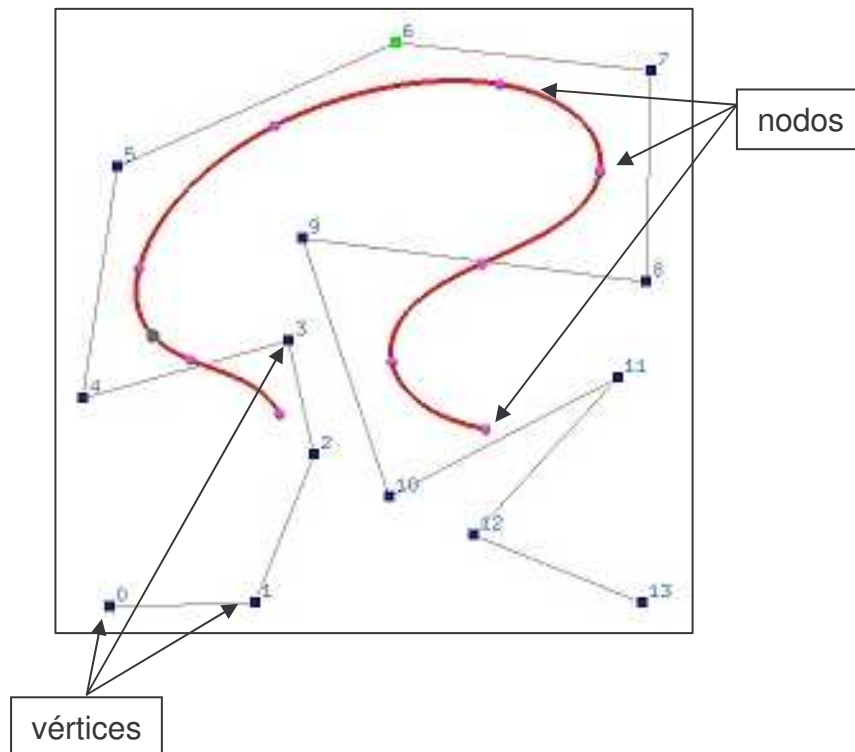
$$P(t) = \sum_{i=0}^n B_i N_i^k(t)$$

donde

$$t \in [x_k, x_{n+1}]$$

dominio de $P(t)$

(i en este intervalo, las funciones de base suman 1!!)



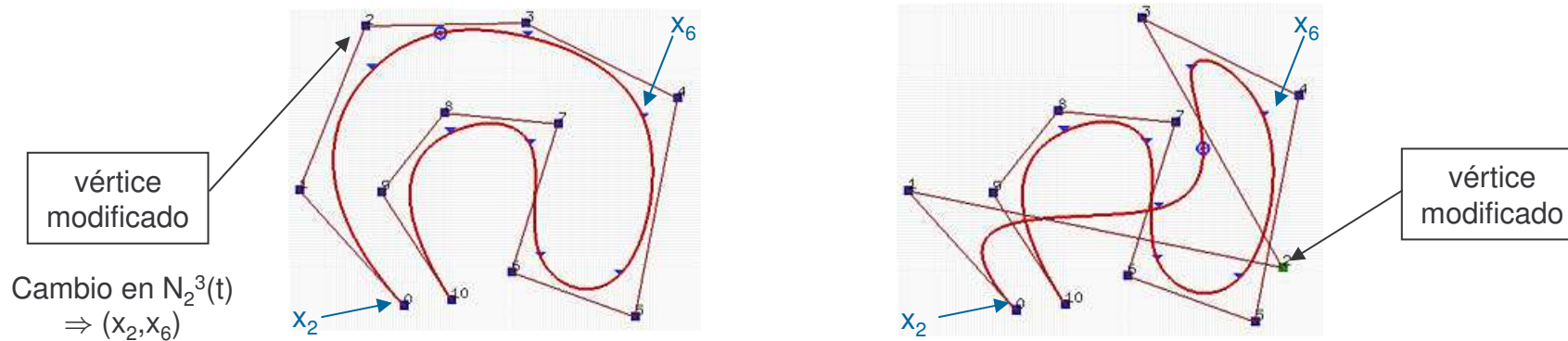
Propiedades de las curvas B-spline

1. Control local: como cada $N_i^k(t)$ tiene soporte local, al modificar un vértice de control tan sólo se modifica una porción de la curva.

Consecuencia de la Propiedad 2 de las funciones de base:

$$N_i^k(t) > 0 \Leftrightarrow t \in (x_i, x_{i+k+1})$$

(cada vértice está asociado a una única función de base)



Ej: $n=10, k=3, \text{ nodos}=[0, 0, \mathbf{0}, 0, 0.12, 0.25, \mathbf{0.37}, 0.5, 0.62, 0.75, 0.87, 1, 1, 1, 1]$
 $\mathbf{x_2}$ $\mathbf{x_6}$

2. Invarianza afín: La curva es una combinación convexa de sus vértices de control, así que respeta transformaciones afines.

Consecuencia de las Propiedades 1 y 2 de las funciones de base: las funciones de base son todas positivas y suman 1 en el dominio de la función $P(t) \Rightarrow P(t)$ es invariante por transformaciones afines por ser una combinación baricéntrica de sus vértices de control.

3. Envolverte convexa fuerte

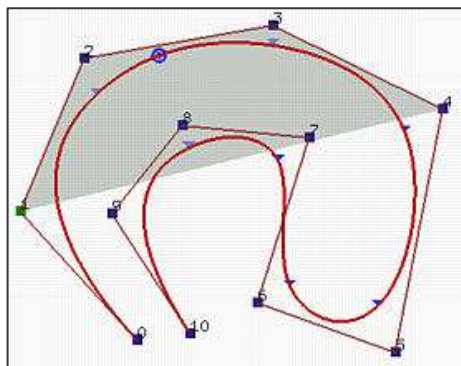
Cada punto de $P(t)$ pertenece a la envolverte convexa de $k+1$ vértices de control

De hecho, si $t_0 \in [x_i, x_{i+1}]$, entonces $P(t_0)$ se construye utilizando las funciones de base $N_{i-k}^k(t), \dots, N_i^k(t)$ (son las únicas con soporte en el intervalo (x_i, x_{i+1})):

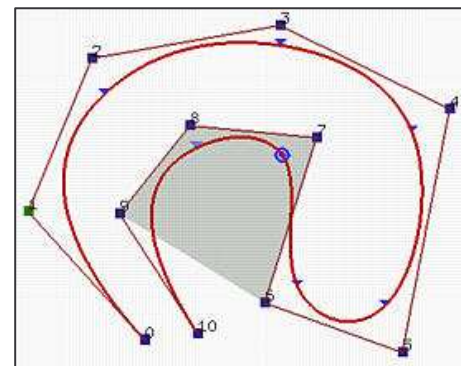
$$P(t_0) = \sum_{j=i-k}^i B_j N_j^k(t) \quad \text{si } t_0 \in [x_i, x_{i+1}]$$

Por tanto, $P(t_0)$ está contenido en la envolverte convexa de los vértices de control B_{i-k}, \dots, B_i .

En consecuencia, toda la curva $P(t)$ está contenida en la unión de las envolvertes convexas de $k+1$ vértices de control consecutivos.

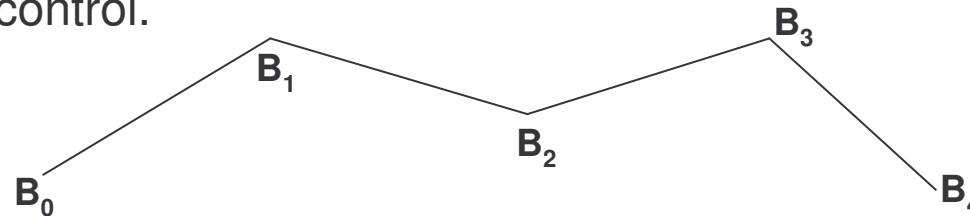


ejemplo de B-spline
con $k=3$



3. Envolverte convexa fuerte. Casos particulares

• Si $k=1$: cada $P(t_0)$ pertenece a la envolvente convexa de 2 vértices, es decir, a un segmento. Por tanto, $P(t)$ es la unión de los segmentos limitados por los vértices de control.

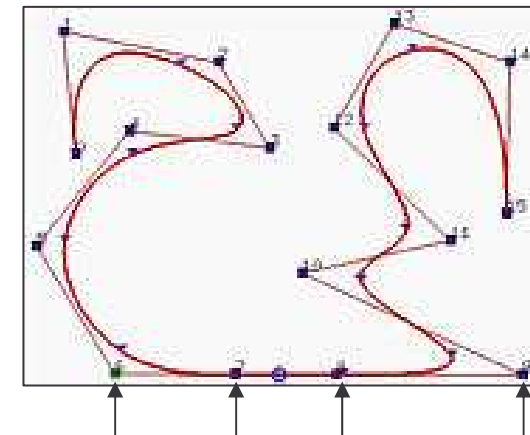


• Si todos los vértices de control están alineados: su envolvente convexa es el segmento que une B_0 y B_n .

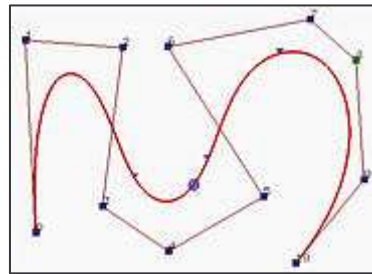
• Podemos obligar a que la curva pase por un determinado B_i , sin más que repetirlo k veces.

• La curva permite segmentos rectos en su interior. Basta tomar $k+1$ vértices de control consecutivos alineados.

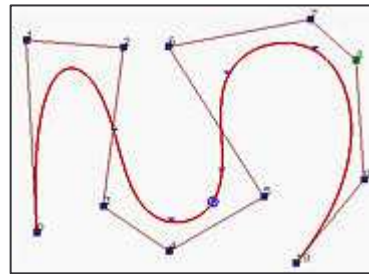
Ej. con $k=3$ y 4 vértices consecutivos



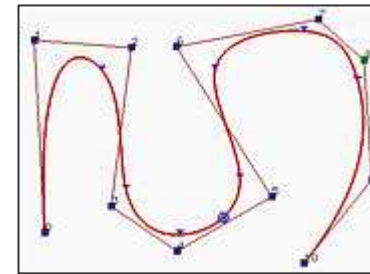
4. La curva $P(t)$ está formada a trozos por polinomios de grado k . Además, cuanto menor sea su grado, más se ajusta a los vértices de control.



$k=7$



$k=5$



$k=3$

5. Si los $k+1$ primeros nodos coinciden, la curva empieza en B_0
Si los $k+1$ últimos nodos coinciden, la curva termina en B_n

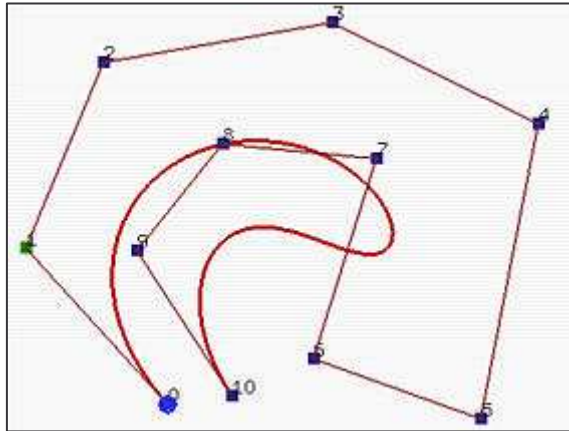
6. $P(t)$ es de clase C^∞ en los tiempos $t \in (x_i, x_{i+1})$ y en los nodos x_i su regularidad depende de la multiplicidad del nodo. Si un nodo x_i tiene multiplicidad m , $P(t)$ es de clase C^{k-m} en $t=x_i$

Casos particulares:

1. Vectores de nodos uniformes: $P(t)$ es de clase C^{k-1} en su dominio.
2. Vectores de nodos "clamped": $P(t)$ es de clase C^{k-1} en el interior de su dominio.

7. Las curvas de Bezier son casos particulares de los B-splines.

Cuando se usan vectores de nodos de tamaño $2k$, los $k+1$ primeros nodos valen 0 y los $k+1$ últimos nodos valen 1, la curva $P(t)$ es realmente la curva de Bezier de los vértices de control.



$$k=10, n=10$$

$$\text{nodos}=[0^{<11>}, 1^{<11>}]$$

(es una curva de Bezier)

Curvas B-spline cerradas ("closed knots")

REPETICIÓN DE PUNTOS DE CONTROL

Supongamos que queremos construir una curva cerrada de grado k , con n puntos de control B_0, \dots, B_n .

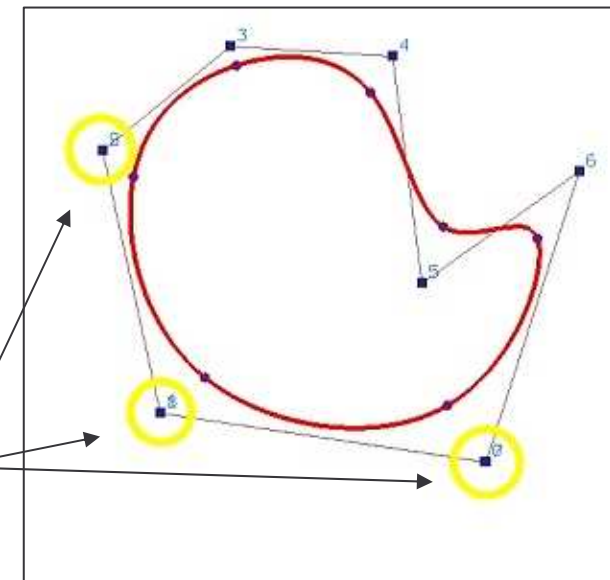
- imponemos que el vector de nodos sea uniforme entre $x_0=0$ y $x_{n+k+1}=1$.
- igualamos los últimos k puntos de control con los primeros k puntos de control:

$$B_{n+k+1}=B_0, B_{n-k+2}=B_1, \dots, B_n=B_{k-1}.$$

La curva se cierra en $t=x_k$.

Ejercicio: da las ecuaciones de una curva cerrada con $k=2$ y $n=4$.

puntos
repetidos



en el dibujo, $k=3$

Diseño de B-splines con Design-Mentor

Introducimos por pantalla los puntos de control y escogemos el grado k de la curva (por defecto el grado es 3). Para el vector de nodos, sólo nos dan 3 opciones:

1. "Open knots": utiliza un vector de nodos uniforme o equispaciado.
2. "Clamped knots": utiliza un vector de nodos "clamped", con lo que la curva empieza en el primero de los puntos de control y termina en el último.
3. "Closed knots": utiliza la técnica vista en la transparencia anterior para cerrar un B-spline, es decir, toma un vector de nodos uniforme y repite al final de la secuencia de puntos de control los k primeros puntos de control.

Obs: puedes modificar el vector de nodos moviendo los nodos en la línea de tiempos que aparece a la dcha. de la pantalla.

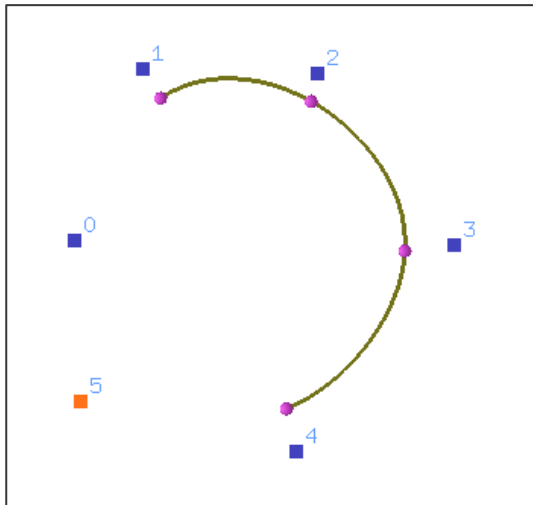
(Veamos ejemplos de estas elecciones en la transparencia siguiente)

Ejemplos

En los tres dibujos tenemos:

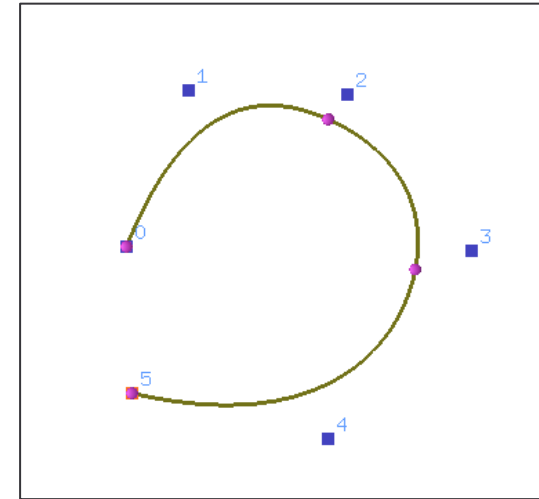
- 6 puntos ($n=5$)
- grado 3 ($k=3$)

vector de nodos uniforme

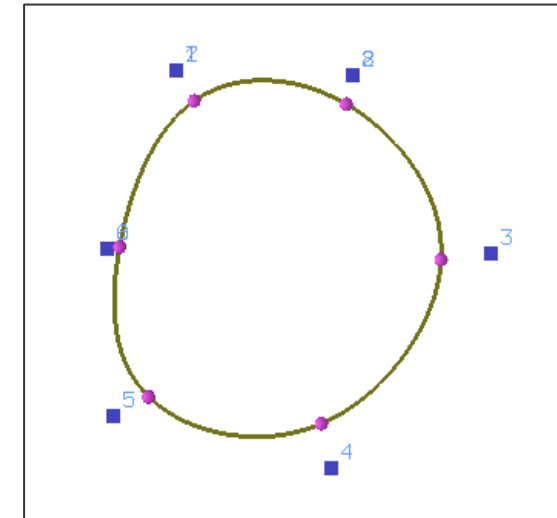


Ejercicio: dibuja tus iniciales con B-splines. Escoge los grados y los vectores de nodos (equiespaciados, "clamped", no uniformes) que consideres.

vector de nodos "clamped"



vector de nodos cerrado



observar que se repiten $k=3$ puntos

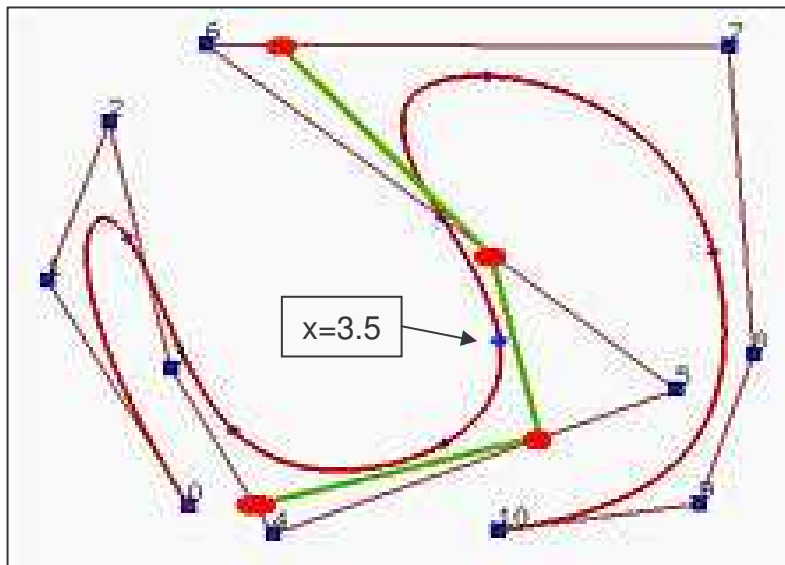
Inserción de nodos (1 nodo)

Idea: insertar un nodo en la lista de nodos sin que cambie la forma de la curva

Fórmula:

número de nodos = número de vértices de control + grado + 1
 (núm. de nodos = $(n+1) + (k+1)$)

Consecuencia: se necesitan más vértices de control (uno más). De hecho, hay que recalcular varios vértices de control, y añadir otro nuevo.



($k=4$, originalmente nodos "clamped")
 nodos: (0 0 0 0 0 1 2 3 4 5 6 7 7 7 7 7)



Si queremos añadir el nodo x tal que $x_m \leq x < x_{m+1}$, hay que modificar los vértices de control $B_{m-k+1}, \dots, B_{m-1}$, y sustituirlos por: Q_{m-k+1}, \dots, Q_m que se hallan con la fórmula:

$$Q_i = (1-a_i)B_{i-1} + a_i B_i, \text{ donde}$$

$$a_i = (x-x_i)/(x_{i+k}-x_i)$$

obs: para los nuevos Q_i se usan B_{m-k}, \dots, B_m

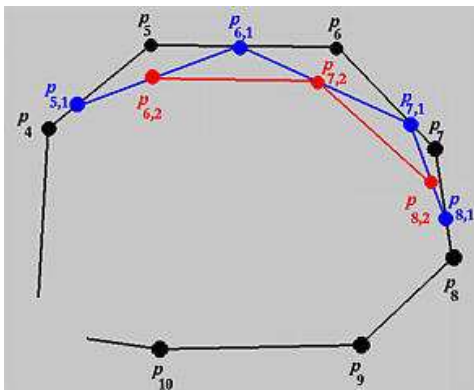
Inserción de nodos (1 nodo)

- Si el nodo insertado coincide con un nodo (simple) ya existente, al usar la fórmula anterior se obtiene $Q_m = B_{m-1}$ (porque en ese caso $a_m = 0$)
- Si el nodo insertado coincide con un nodo (múltiple de multiplicidad s) ya existente, al usar la fórmula anterior se obtienen

$$Q_m = B_{m-1}, Q_{m-1} = B_{m-2}, \dots, Q_{m-s+1} = B_{m-s}$$

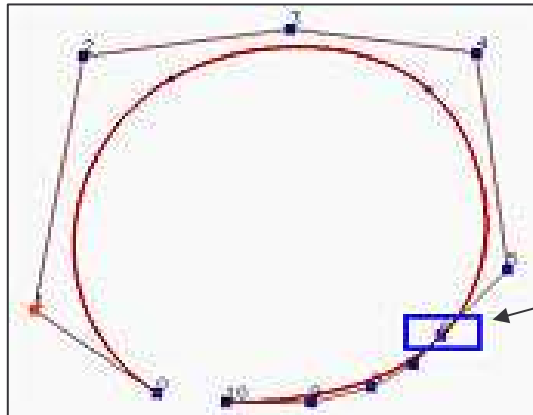
- Para insertar un nodo múltiples veces se puede optar por insertarlo de modo repetido según las fórmulas anteriores. Otra opción es utilizar directamente un algoritmo para inserción múltiple de nodos (algoritmo tipo "corner cutting"):

<http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/spline/B-spline/multiple-time.html>



← se pueden calcular directamente los nuevos nodos, o bien ir paso a paso...

Algoritmo de De Boor



Idea:

la inserción de un nodo x tantas veces como el grado k del B-spline nos permite calcular el punto $P(x)$ de la curva B-spline.

Trabajo:

El algoritmo de De Boor proporciona un método para calcular puntos sobre la curva B-spline. Se basa en la inserción de nodos múltiples.

Lee las secciones:

- Inserting a knot multiple times
- De Boor's Algorithm

de www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/notes.html

y haz un resumen de las mismas.