

# Curvas de Bezier

---

- Origen en los trabajos de Bezier y de Casteljau.
- Bezier (ingeniero en Renault) desarrolla las curvas basándose en polinomios de Bernstein. 1966
- De Casteljau (ingeniero de Citroën) usa un desarrollo algorítmico. 1959
- Ambas teorías son equivalentes.
- El nombre de curvas de Bezier se debe a que Bezier publicó sus trabajos, mientras que de Casteljau los mantuvo como documentos internos.

# Curvas de Bezier. Bibliografía

---

Desarrollaremos este tema siguiendo:

- "Curves and Surfaces for CAGD. A practical guide", de Gerald Farin. Capítulos 3, 4 y 5.
- PROGRAMA "DESIGN MENTOR" Copyright (c) Ching-Kuang Shene, and John Lowther, Yuan Zhao, Yan Zhou 1997, 1998, subject to NSF Grant General Conditions (GC-1):

[www.cs.mtu.edu/~shene/NSF-2/DM2-BETA/index.html](http://www.cs.mtu.edu/~shene/NSF-2/DM2-BETA/index.html)

También es útil consultar los libros:

- "3-D Computer Graphics. A Mathematical Introduction with OpenGL", de S. R. Buss
- "Curvas y superficies para Modelado Geométrico", de J. M. Cordero y J. Cortés.

# Curvas de Bezier. De Casteljau

---

• Busca un modo de construir curvas que respete el principio de invarianza afín.

• **Principio de invarianza afín:** dados los puntos  $p_1, \dots, p_n$  del plano o del espacio afín y una transformación afín  $f$ , buscamos un método  $C(\ )$  para construir curvas a partir de los puntos  $p_1, \dots, p_n$  que cumpla

$$f(C(p_1, \dots, p_n)) = C(f(p_1), \dots, f(p_n))$$

• **Ejemplo básico:** si  $C(\ )$  es el segmento que une dos puntos, se cumple el principio de invarianza afín.  
Recordar  $C(p_1, p_2) = (1-t)p_1 + tp_2$ ,  $t \in [0, 1]$ .

# Algoritmo de de Casteljaou

- **Algoritmo de de Casteljaou:** basado en la repetición sucesiva del ejemplo básico (segmento entre dos puntos).

$$\begin{pmatrix} b_0^0 & b_0^1(t) & b_0^2(t) & \dots & b_0^n(t) \\ b_1^0 & b_1^1(t) & b_1^2(t) & & \\ b_2^0 & b_2^1(t) & & & \\ \vdots & \vdots & & & \\ b_{n-1}^0 & b_{n-1}^1(t) & & & \\ b_n^0 & & & & \end{pmatrix}$$

donde  $b_i^0$ ,  $i=0\dots n$ , son los puntos de partida y

$$b_i^r(t) = (1-t)b_i^{r-1}(t) + tb_{i+1}^{r-1}(t)$$

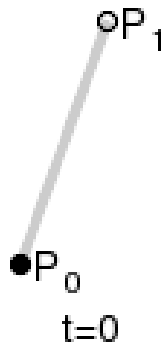
# La curva de Bezier (de Casteljau)

---

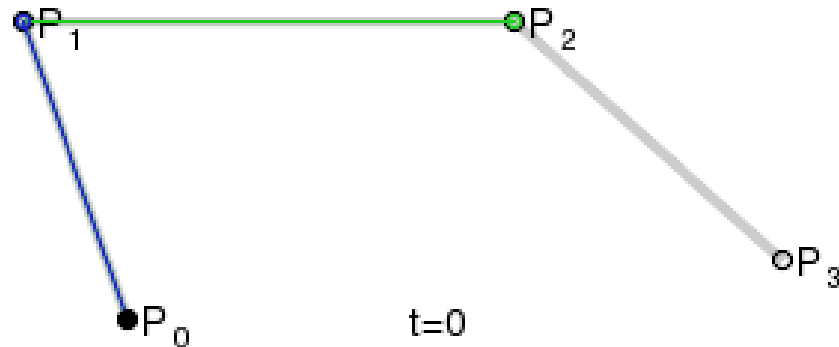
- El último paso del algoritmo anterior,  $b_0^n(t)$ ,  $t \in [0, 1]$ , se llama curva de Bezier y se denota  $B[b_0, \dots, b_n; t]$ .
- Los puntos  $b_0, \dots, b_n$  se llaman vértices de control
- El polígono formado por los vértices de control se llama polígono de control.

# Ejemplos de curvas de Bezier

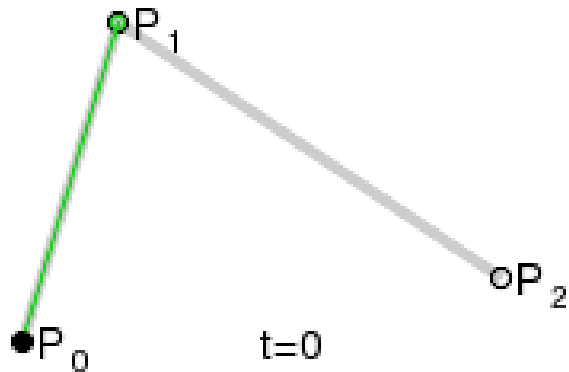
---



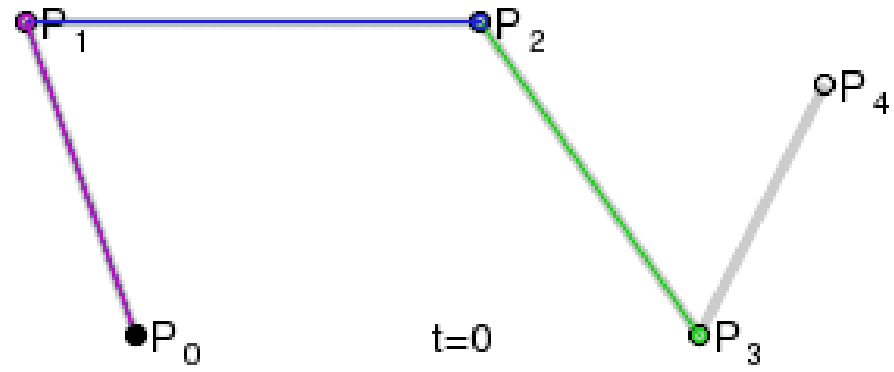
Bezier con 2 puntos



!!!Bezier con 4 puntos!!!



Bezier con 3 puntos



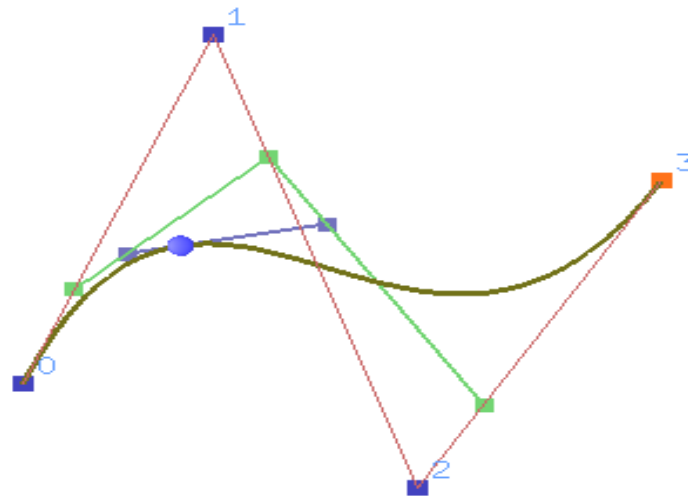
Bezier con 5 puntos

# Algoritmo de de Casteljaou

---

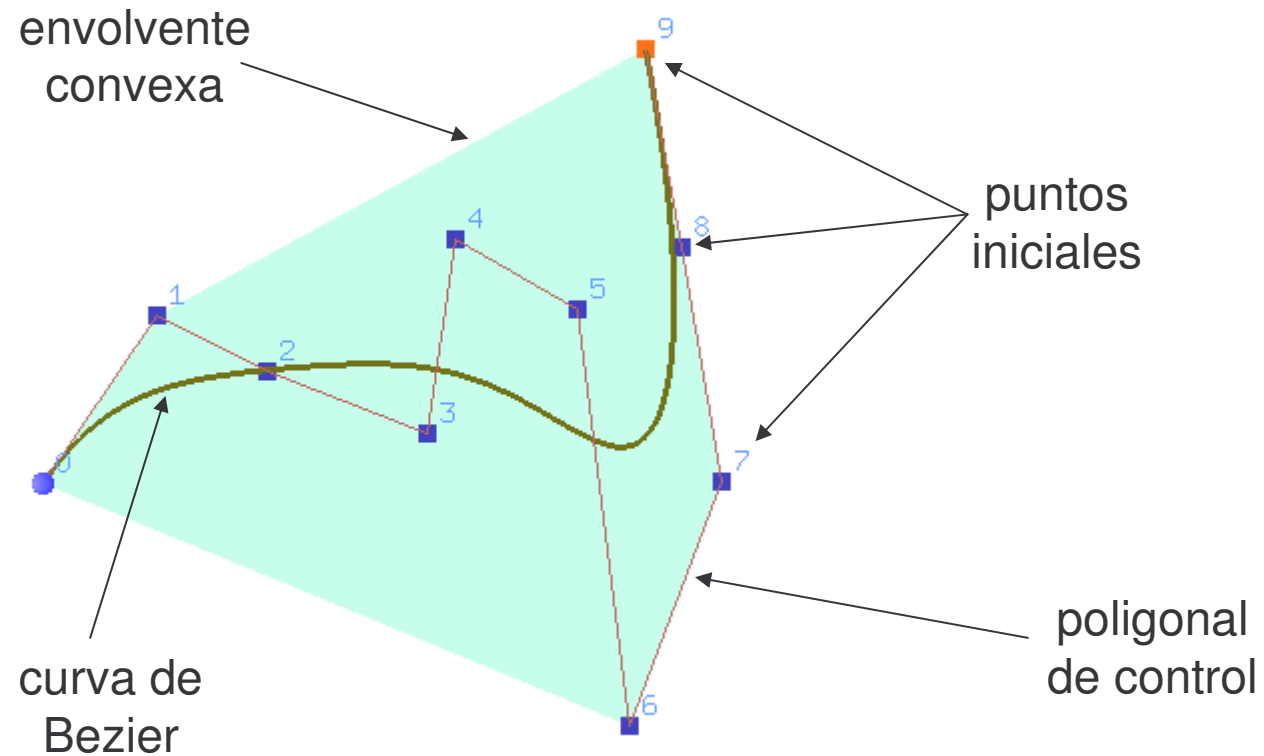
Ver el funcionamiento del  
Algoritmo de de Casteljaou con  
el programa “Design Mentor”

Copyright (c) Ching-Kuang Shene, and John Lowther,  
Yuan Zhao, Yan Zhou 1997, 1998,  
subject to NSF Grant General Conditdions (GC-1)



# Propiedades de las curvas de Bezier

---



# Propiedades de las curvas de Bezier

---

- La curva de Bezier  $B[b_0, \dots, b_n; t]$  con  $n+1$  vértices de control tiene grado  $n$ .
- $B[b_0, \dots, b_n; t]$  es una combinación convexa de sus vértices de control.
- Cumple el principio de invarianza afín.
- Está contenida en la envolvente convexa de los vértices de control.
- Cumple la propiedad de interpolación en los extremos.
- Estable bajo reparametrizaciones de  $t$ .

# Justificación de las propiedades

---

- Las curvas de Bezier se definen de modo recursivo a partir de dos curvas de Bezier que involucran menos puntos de control.

$$B[b_0, \dots, b_n; t] = (1-t)B[b_0, \dots, b_{n-1}; t] + t B[b_1, \dots, b_n; t]$$

- Se puede razonar por inducción gracias a esta propiedad.
- Las reparametrizaciones son un cambio de variable.

COMPROBAMOS LAS PROPIEDADES USANDO EL PROGRAMA “DESIGN MENTOR” Copyright (c) Ching-Kuang Shene, and John Lowther, Yuan Zhao, Yan Zhou 1997, 1998, subject to NSF Grant General Conditions (GC-1)

# Teoría de Blossoms

---

- La  $T^a$  de Blossoms generaliza el alg. de de Casteljau.
- El Blossom de  $b_0, \dots, b_n$  se denota  $b[t_1, \dots, t_n]$  y se calcula utilizando el algoritmo de de Casteljau (donde en cada paso se introduce un nuevo parámetro  $t_r$ ).

$$\left( \begin{array}{ccccccc}
 b_0^0 & b_0^1(t_1) & b_0^2(t_1, t_2) & \dots & b_0^n(t_1, \dots, t_n) \\
 b_1^0 & b_1^1(t_1) & b_1^2(t_1, t_2) & & & & \\
 b_2^0 & b_2^1(t_1) & & & & & \\
 \vdots & \vdots & & & & & \\
 b_{n-1}^0 & b_{n-1}^1(t_1) & & & & & \\
 b_n^0 & & & & & & 
 \end{array} \right)$$

b[t<sub>1</sub>, ..., t<sub>n</sub>]

donde  $b_i^r(t_1, \dots, t_r) = (1 - t_r)b_i^{r-1}(t_1, \dots, t_{r-1}) + t_r b_{i+1}^{r-1}(t_1, \dots, t_{r-1})$

# Ventaja de la T<sup>a</sup> de Blossoms

---

- Conociendo el Blossom  $b[t_1, \dots, t_n]$  se recupera toda la matriz de de Casteljau.

$$b_i^r(t) = b[0^{<n-r-i>}, 1^{<i>}, t^{<r>}]$$

(n-r-i) veces 0      i veces 1      r veces t

(escribir un ejemplo)

- La curva de Bezier es  $b[t, \dots, t]$ .
- 
- n veces

# Polinomios de Bernstein

---

- Polinomios de Bernstein  $\{B_0^n(t), B_1^n(t), \dots, B_n^n(t)\}$   
de grado  $n$

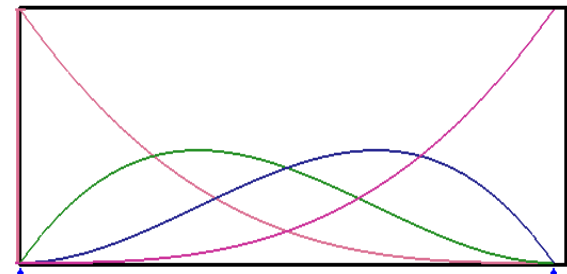
- Cada  $B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i$

- Propiedades de los polinomios de Bernstein

$$B_i^n(t) = (1-t) B_i^{n-1}(t) + t B_{i-1}^{n-1}(t).$$

$$\sum_{i=0}^n B_i^n(t) = 1.$$

$$\forall t \in [0, 1], 0 \leq B_i^n(t) \leq 1.$$



Ej. de grado 3

# La curva de Bezier

---

- Dados los puntos de control  $b_0, \dots, b_n$ , la curva de Bezier se puede calcular utilizando los polinomios de Bernstein.

$$B[b_0, \dots, b_n; t] = \sum_{i=0}^n b_i B_i^n(t)$$

(justificación de esta fórmula por inducción)

- Por las propiedades de los polinomios de Bernstein:

$B[b_0, \dots, b_n; t]$  es combinación convexa de  $b_0 \dots b_n$

1. Invarianza afín
2. Envolverte convexa

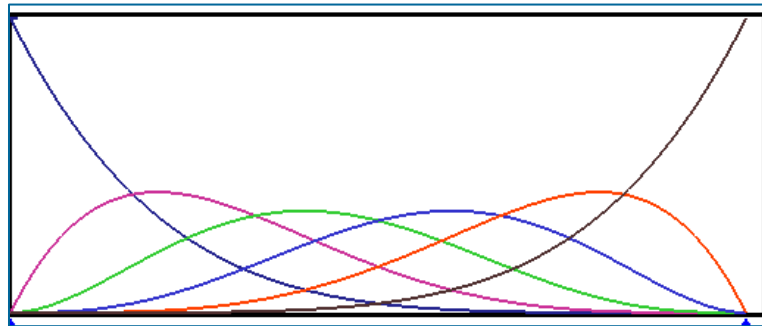
# Otras propiedades

---

- Precisión lineal (consecuencia de la envolvente convexa)

si  $b_0, \dots, b_n$  están alineados, entonces  $B[b_0, \dots, b_n; t]$  es el segmento entre  $b_0$  y  $b_n$

- Cada polinomio de Bernstein tiene un único máximo, que se alcanza en  $t=i/n$ . Así que si se modifica el valor de  $b_i$ , la curva de Bezier sufre su mayor variación en  $t=i/n$ .

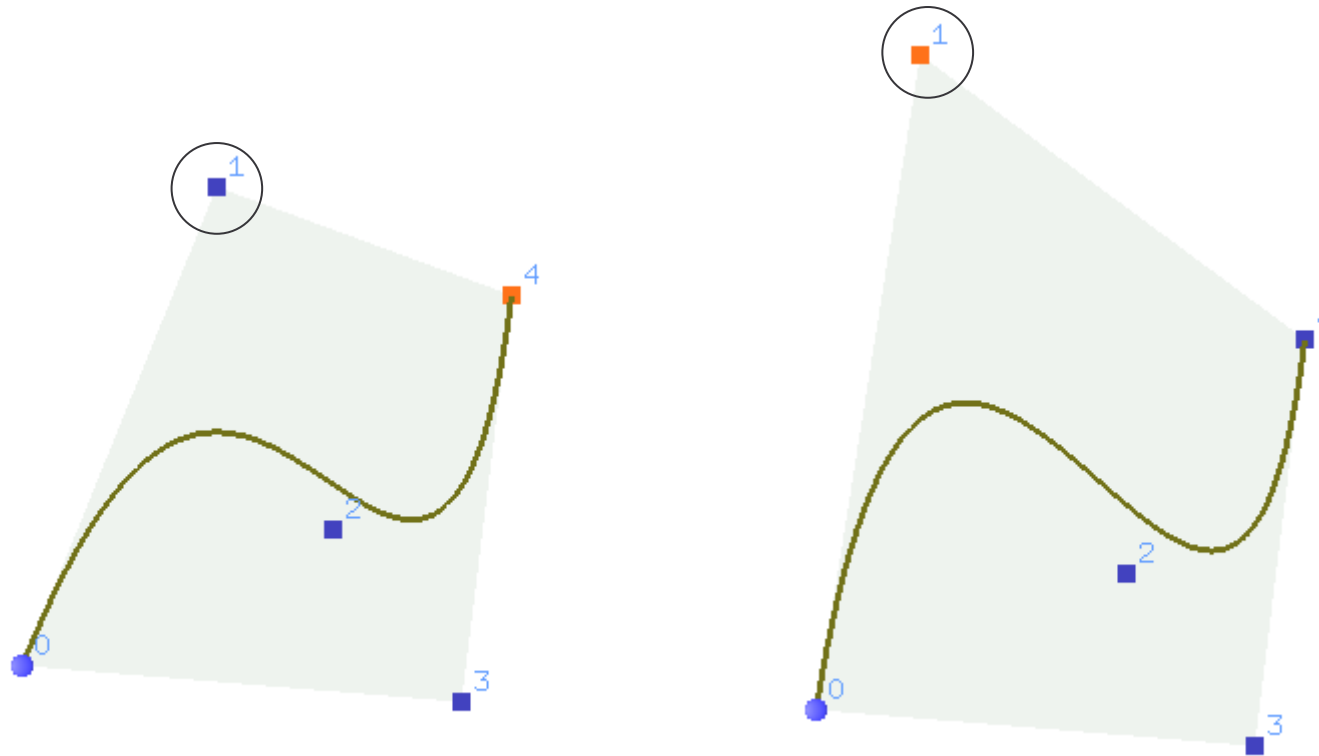


Ej: polinomios de Bernstein de grado 6

# Ejemplo 1: variación de 1 punto

---

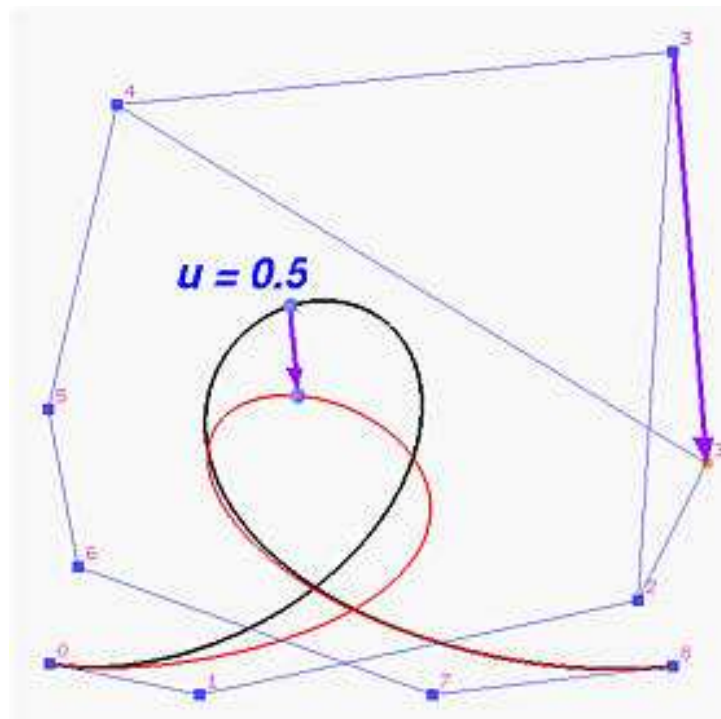
El mayor cambio de la curva se produce en la zona cercana al punto que hemos modificado



## Ejemplo 2: variación de 1 punto

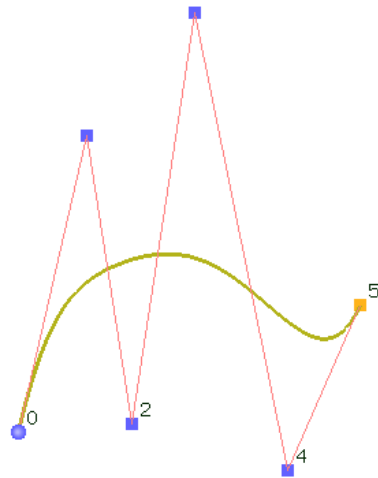
---

- Notar que con sólo modificar un punto de control, se cambia la forma completa de la curva y sólo se mantiene la interpolación en los extremos.

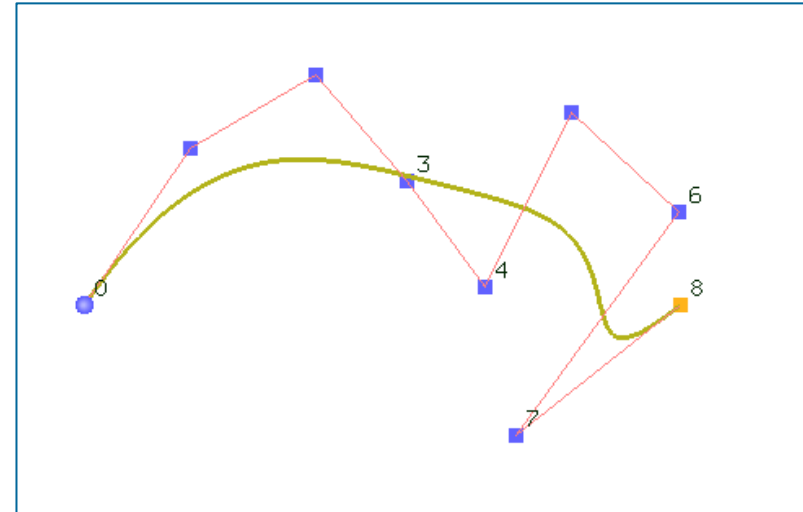


# Ejemplos de curvas y polinomios

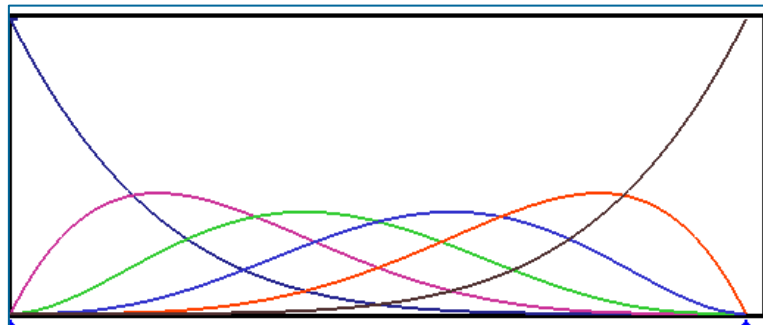
---



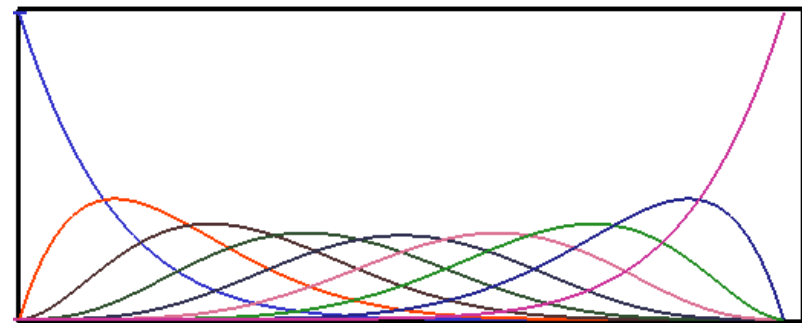
curva de Bezier de orden 5



curva de Bezier de orden 8



polinomios de Bernstein



polinomios de Bernstein



# Derivadas de curvas de Bezier

---

- Derivada de  $B[b_0, \dots, b_n; t]$  respecto de  $t$ :

$$\frac{d}{dt} B[b_0, \dots, b_n; t] = \sum_{i=0}^n b_i \underbrace{\frac{d}{dt} B_i^n(t)}$$

derivada del polinomio  
de Bernstein

- Derivando cada  $B_i^n(t)$  y agrupando términos:

$$\frac{d}{dt} B[b_0, \dots, b_n; t] = n \underbrace{\sum_{i=0}^{n-1} (\Delta b_i) B_i^{n-1}(t)}$$

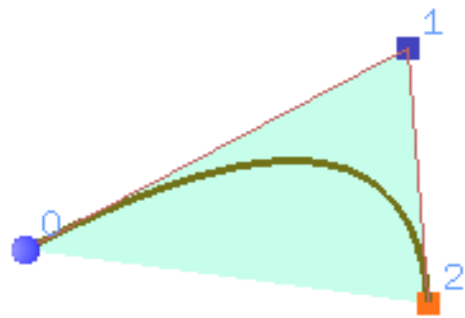
donde

$$\Delta b_i = b_{i+1} - b_i$$

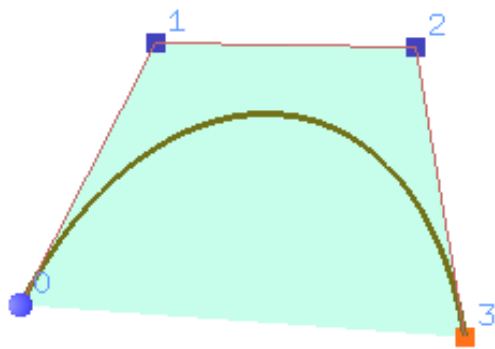
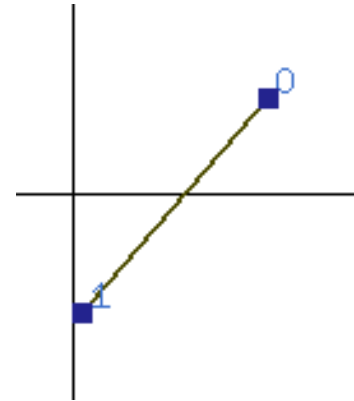
es de nuevo una  
curva de Bezier de  
un grado menos

# Derivadas. Ejemplos

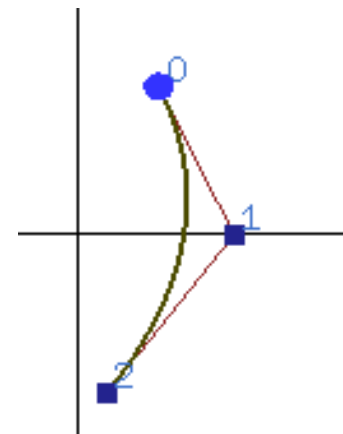
---



derivada



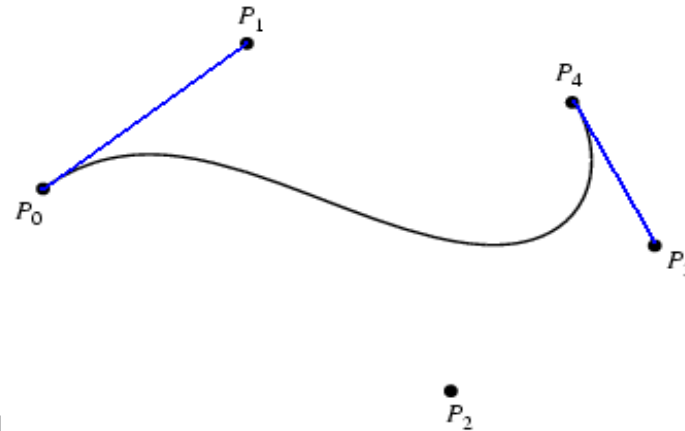
derivada



# Derivadas. Propiedades

Derivadas en  $t=0$  y  $t=1$ :

- en  $t=0$  vale  $n(b_1 - b_0)$
- en  $t=1$  vale  $n(b_n - b_{n-1})$



¡¡consecuencias en el gráfico de la curva!!

Derivadas de órdenes superiores:

$$\frac{d}{dt^r} B[b_0, \dots, b_n; t] = \frac{n!}{(n-r)!} \sum_{i=0}^{n-r} (\Delta^r b_i) B_i^{n-r}(t)$$

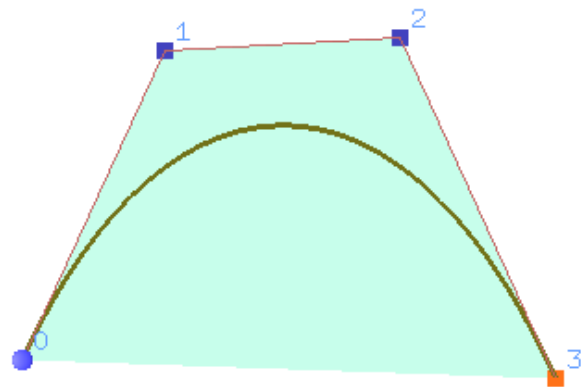
donde

$$\Delta^r b_i = \Delta^{r-1} b_{i+1} - \Delta^{r-1} b_i$$

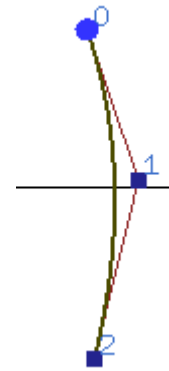
con  $\Delta^0 b_i = b_i$

es de nuevo una curva de Bezier

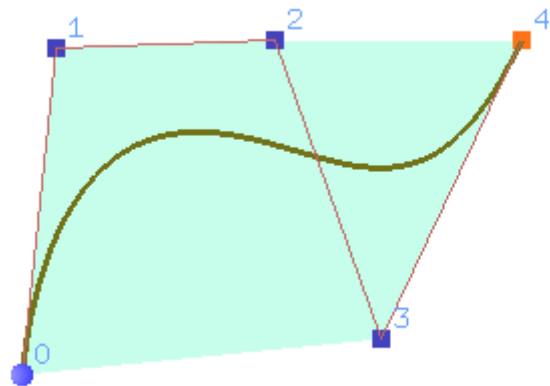
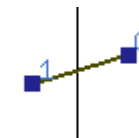
# Derivadas. Ejemplos



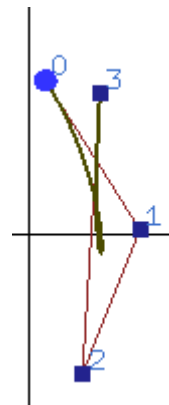
derivada  
primera



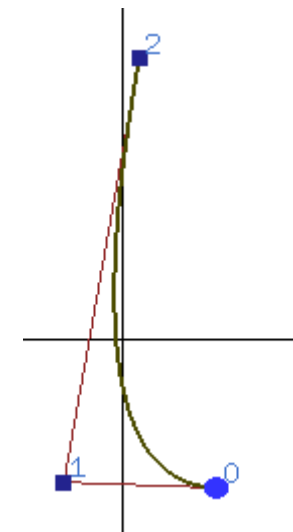
derivada  
segunda



derivada  
primera



derivada  
segunda

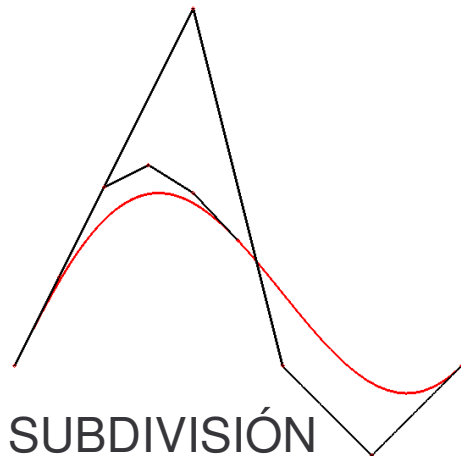


# El proceso de subdivisión

---

Supongamos que tenemos una curva de Bezier  
 $B[b_0, \dots, b_n; t], t \in [0, 1]$

- Subdivisión: considerar  $t \in [0, c], c \leq 1$ , y buscar nuevos vértices de control  $c_i$  para una nueva curva de Bezier que coincida con la original en  $t \in [0, c]$ .  
Análogamente, considerar  $t \in [c, 1], c \leq 1$ .



Hallar los nuevos vértices de control usando  $t^a$  de Blossoms:

$$c_i = b[0^{<n-i>}, c^{<i>}] = b_0^i(c)$$

# El proceso de subdivisión

---

Utilidad principal:

Aproximación de una curva de Bezier por segmentos

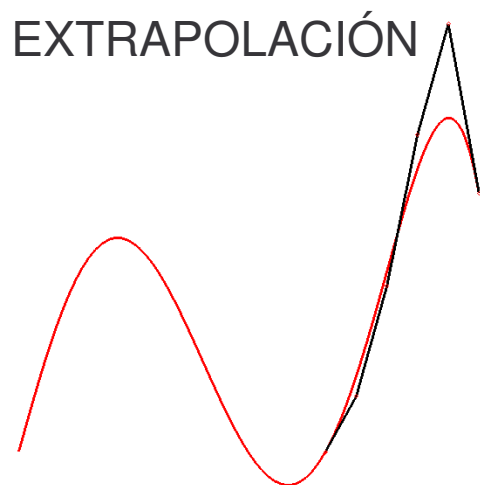
- subdividir la curva sucesivas veces hasta que la aproximación por segmentos de la curva no diste mucho de la curva original
- habitualmente se suele subdividir a la mitad, y de nuevo a la mitad...

# El proceso de extrapolación

---

Supongamos que tenemos una curva de Bezier  
 $B[b_0, \dots, b_n; t], t \in [0, 1]$

- Extrapolación: considerar  $t \in [1, d], d \geq 1$ , y buscar nuevos vértices de control  $d_i$  para una nueva curva de Bezier que coincida con la original en  $t \in [1, d]$ .



Hallar los nuevos vértices de control usando  $t^a$  de Blossoms:

$$d_i = b[1^{<n-i>}, d^i] = b_{n-i}^i(d)$$

# Elevación de grado

---

- Si tenemos varias curvas de Bezier calculadas de modo independiente, podemos querer tratarlas al mismo tiempo y nos puede interesar que todas tengan el mismo grado.
- Elevación de grado: buscamos una curva de Bezier idéntica a  $B[b_0, \dots, b_n; t]$  que esté definida con más vértices de control ( $\Rightarrow$  se eleva el grado).

Si queremos que la curva tenga grado  $n+1$ :

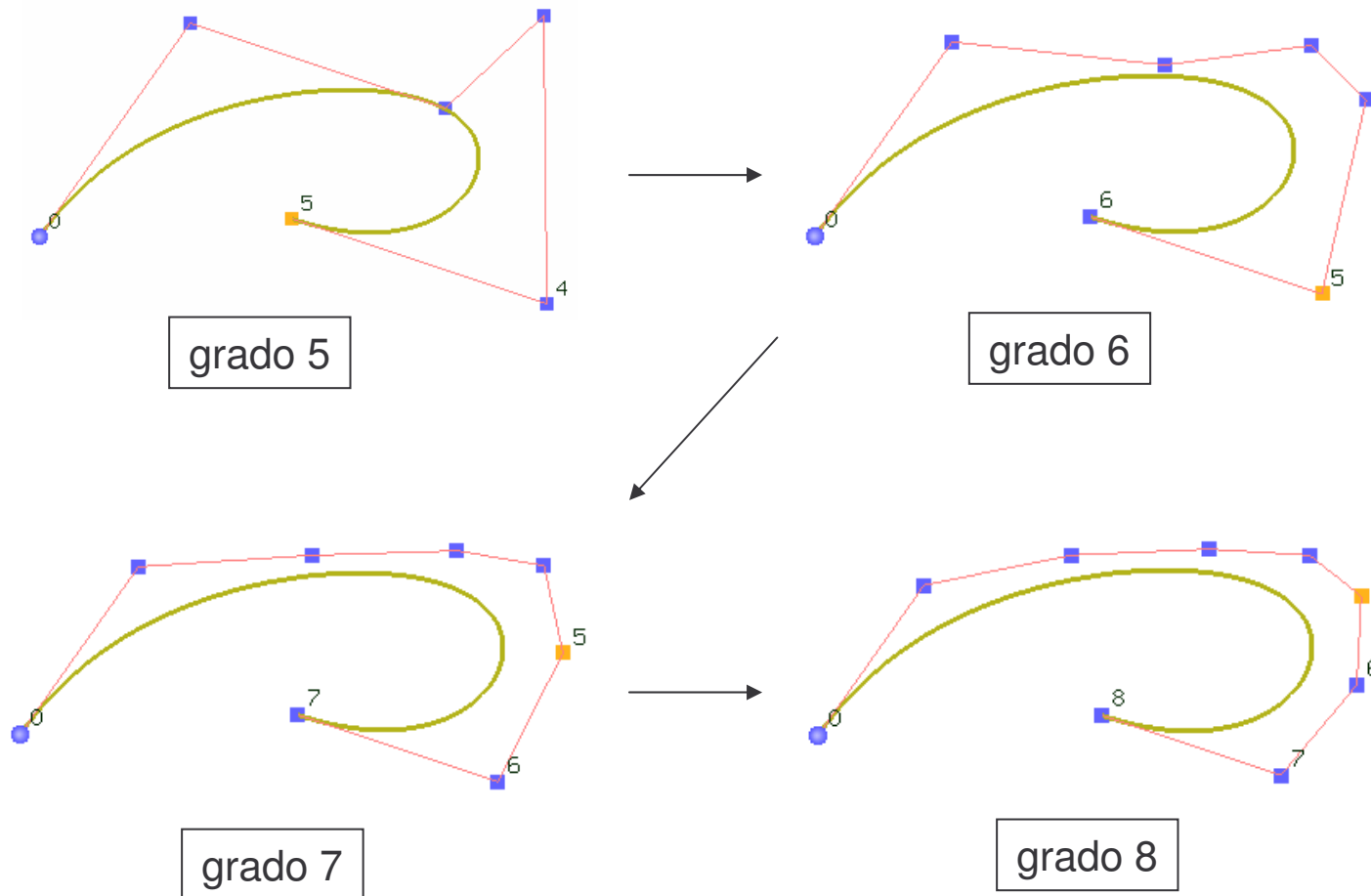
- Nuevos vértices de control:  $b_0^{(1)}, \dots, b_{n+1}^{(1)}$ .

Se impone que ambas curvas sean iguales y se obtiene:

$$b_i^{(1)} = \frac{i}{n+1} b_{i-1} + \left(1 - \frac{i}{n+1}\right) b_i, \quad i=0 \dots n+1.$$

# Ejemplos de elevación de grado

Ejemplos de la misma curva, tras aplicar el proceso de elevación de grado.



# Reducción de grado

---

• Reducción de grado: buscamos una curva de Bezier que aproxime a  $B[b_0, \dots, b_n; t]$  y esté definida usando menos vértices de control ( $\Rightarrow$  se reduce el grado).

Si queremos que la curva tenga grado  $n-1$ :

• Nuevos vértices de control:  $b_0^{(0)}, \dots, b_{n-1}^{(0)}$ .

Se supone que la curva  $B[b_0, \dots, b_n; t]$  se ha obtenido elevando el grado a  $B[b_0^{(0)}, \dots, b_{n-1}^{(0)}; t]$  y se obtiene:

$$b_i^{(0)} = \frac{nb_{i-1} b_{i-1}^{(0)}}{n-i}$$

$$i=0, 1 \dots n-1$$

$$b_{i-1}^{(0)} = \frac{nb_i - (n-i)b_i^{(0)}}{i}$$

$$i=n, n-1 \dots 1$$

Dos fórmulas para hallar los  $b_i$ . Se pueden combinar ambas.

# Temas de ampliación

---

En el capítulo 5 del libro de Farin también podemos encontrar información acerca de:

- Curvas de Bezier no paramétricas
- Proyecciones (cross plots)
- Integrales de curvas de Bezier
- Forma baricéntrica de una curva de Bezier

# Ejercicios

---

Trabajo obligatorio para entregar:

Ejercicio 1: Lee la sección VII.3 del libro de Buss sobre el proceso de subdivisión en curvas de Bezier de grado 3 y sus aplicaciones, y haz un resumen.

Ejercicio 2: utiliza el programa Design Mentor para dibujar las iniciales de tu nombre y tus apellidos mediante trozos de curvas de Bezier.

La entrega se realizará como máximo el 8 de mayo de 2009.