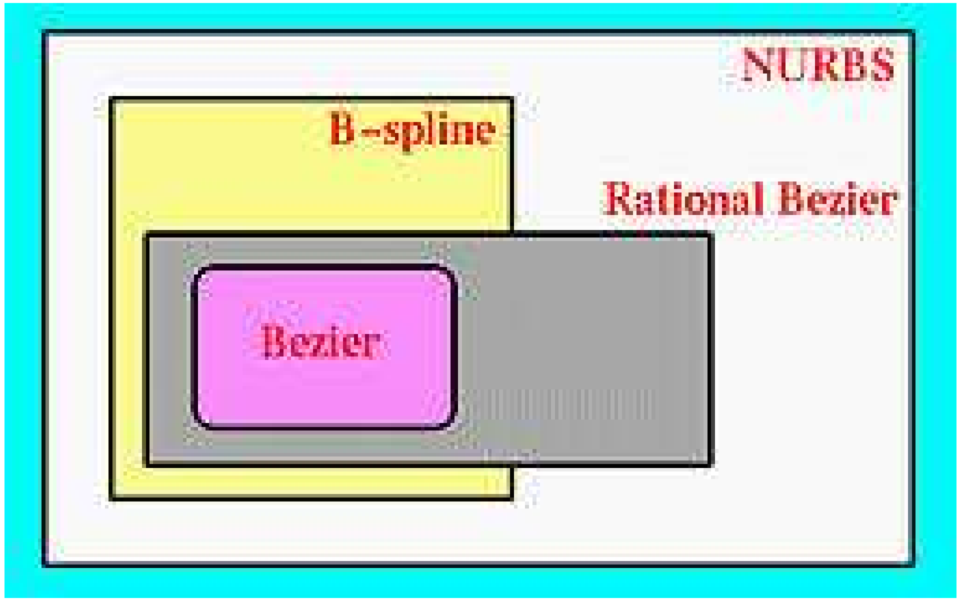
CURVAS PARAMÉTRICAS

Tratamiento unificado para diversos tipos de curvas



(intuitivas, flexibles, afín-invariantes, rápidas de computar, estables numéricamente)

Ver introducción a las Bezier en:

http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES

Dr. C.-K. Shene, Dept. Computer Science, Michigan Technological University

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.

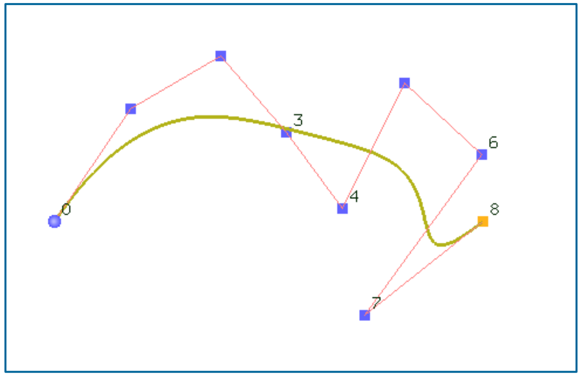
Curvas de Bezier

•Dados los puntos de control $p_0,\ldots, p_n$, la curva de Bezier se define utilizando la base de los polinomios de Bernstein:

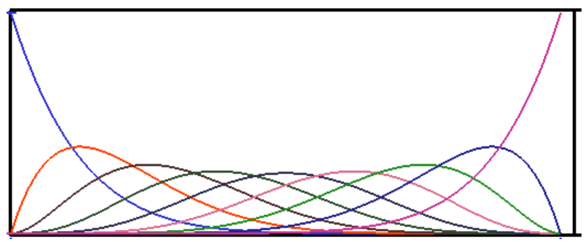
$$\boxed{B(t)=\sum_{i=0}^{n} p_i\, B_i^n(t), \qquad t \in [0, 1]}$$

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

Ejemplo:



curva de Bezier de orden 8

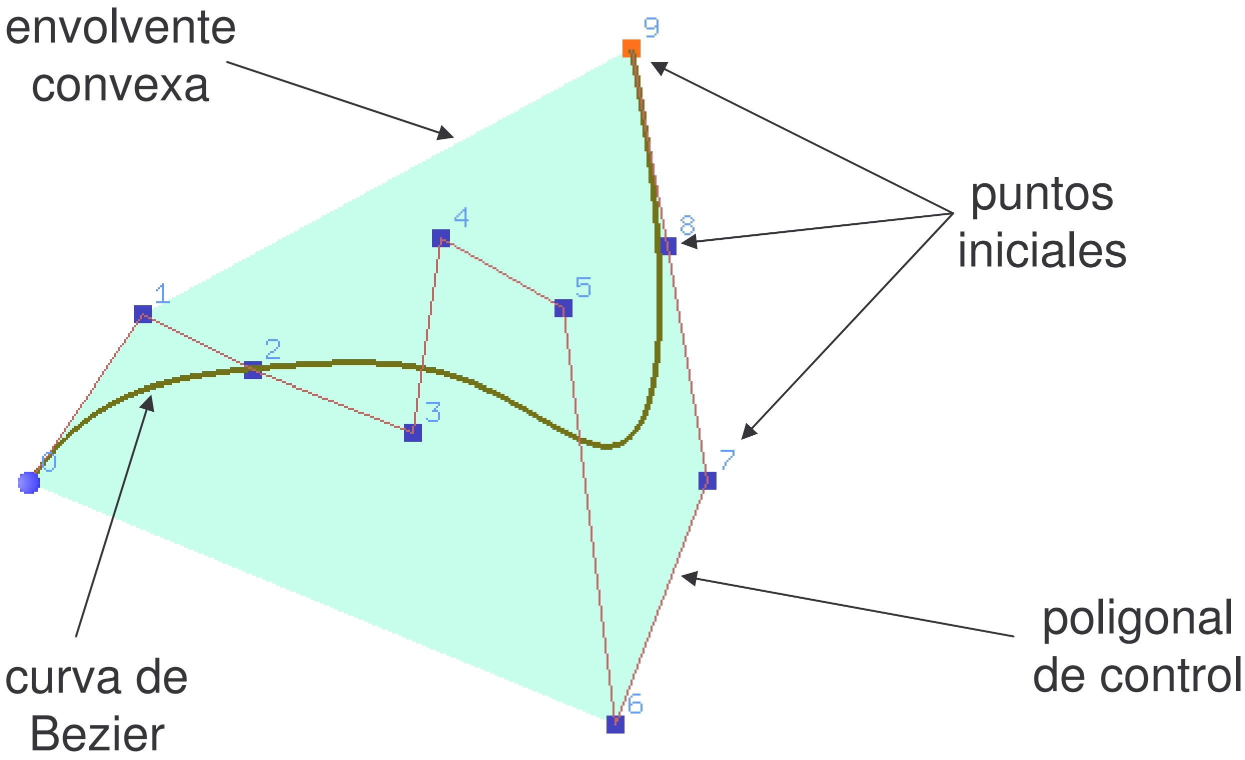


polinomios de Bernstein

Propiedades de las curvas de Bezier

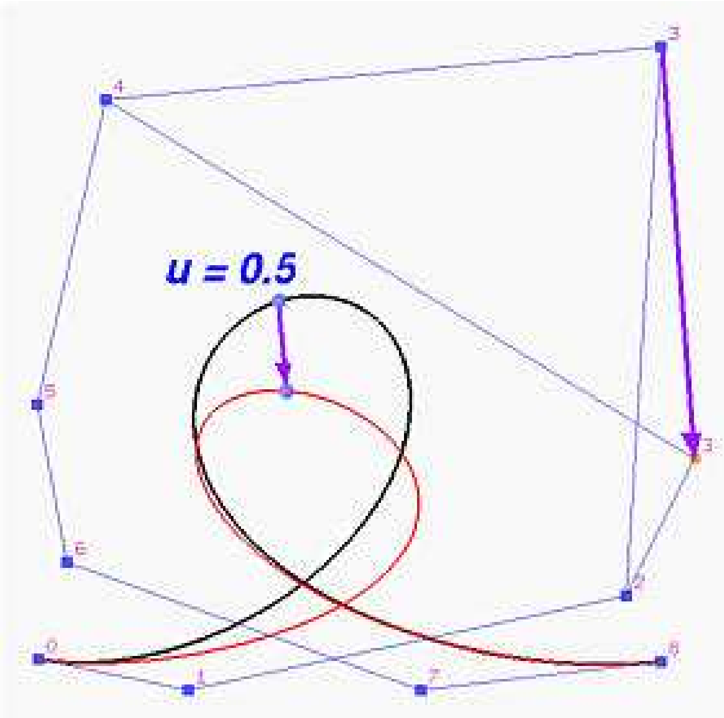
- Con n+1 puntos de control tiene grado n.
- Combinación convexa de sus puntos de control.
- Principio de invarianza afín.
- Contenida en la envolvente convexa.
- Interpolación en los extremos.
- Tangente en los extremos al polígono de control.
- No más compleja que la poligonal de control.

Propiedades de las curvas de Bezier



Variación de 1 punto

•Con sólo modificar un punto de control, se cambia la forma completa de la curva y sólo se mantiene la interpolación en los extremos. Si se cambia p_i por p_i+v…



$$\boxed{C(t) = B(t) + v\, B_i^n(t)}$$

nueva curva | curva original

la curva sufre su mayor variación en t=i/n, (máximo del polinomio de Bernstein $B_i^n(t)$)

Algoritmo de de Casteljau

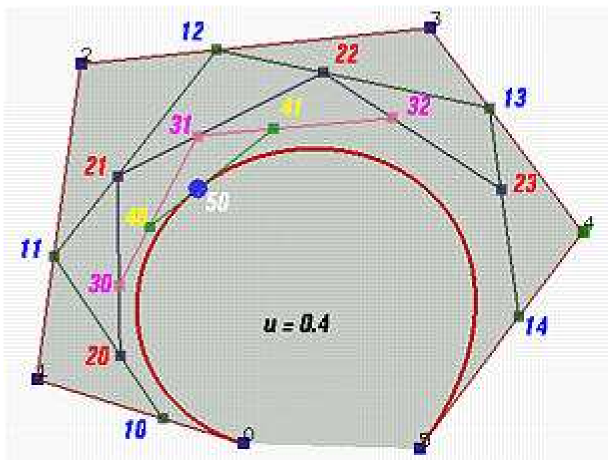
Para calcular puntos concretos sobre la curva sin calcular sus ecuaciones:

$$
\left(
\begin{array}{lllllll}
b_0^0 & \rightarrow & b_0^1(t) & \rightarrow & b_0^2(t) & \dots & \rightarrow \; b_0^n(t) \\
 & \nearrow & & \nearrow & & & \nearrow \\
b_1^0 & \rightarrow & b_1^1(t) & \rightarrow & b_1^2(t) & & \\
 & \nearrow & & \nearrow & & & \\
b_2^0 & \rightarrow & b_2^1(t) & & & & \\
 & \nearrow & & & & & \\
\vdots & & \vdots & & & & \\
b_{n-1}^0 & \rightarrow & b_{n-1}^1(t) & & & & \\
 & \nearrow & & & & & \\
b_n^0 & & & & & &
\end{array}
\right)
$$

$\uparrow$

puntos de control

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

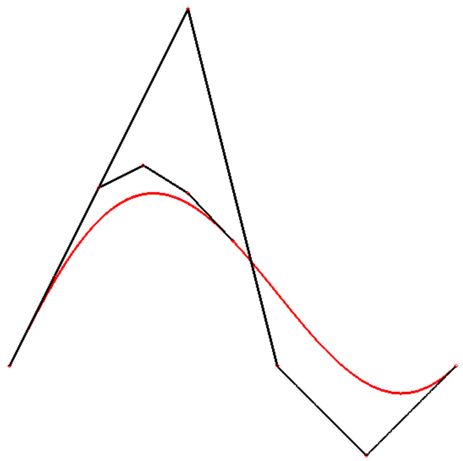


¡¡¡FÁCIL DE PROGRAMAR!!!
¡¡¡RÁPIDO!!!

Subdivisión

•<u>Subdivisión</u>: 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 $t \in [c,1]$, $c \leq 1$.

$c=1/2$



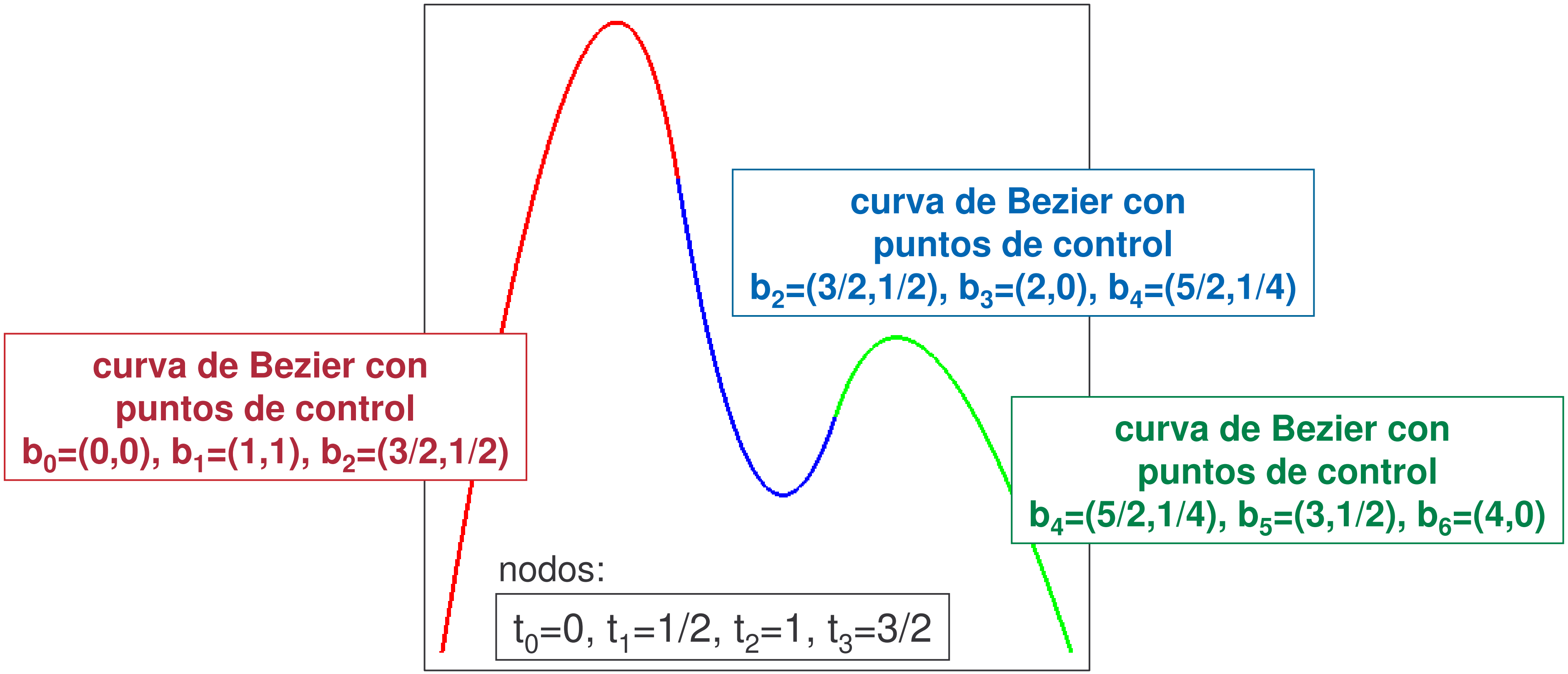
los nuevos puntos de control se calculan mediante el algoritmo de de Casteljau

utilidad: aproximación por segmentos

- •subdividir la curva sucesivas veces hasta que la aproximación por segmentos de la curva no diste mucho de la curva original
- •representar las poligonales de control

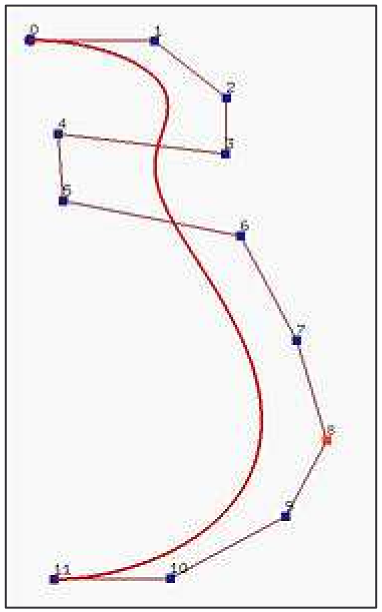
Curvas de Bezier a trozos

•Para modelar curvas complejas con curvas de Bezier, se necesitan grados muy altos.
•<u>Alternativa</u>: construir curvas de Bezier <u>a trozos</u> (de menor grado) y se ***unen*** conservando la regularidad.
•En CAGD suele ser suficiente que las curvas sean C^1/G^1 ó C^2/G^2-continuas.

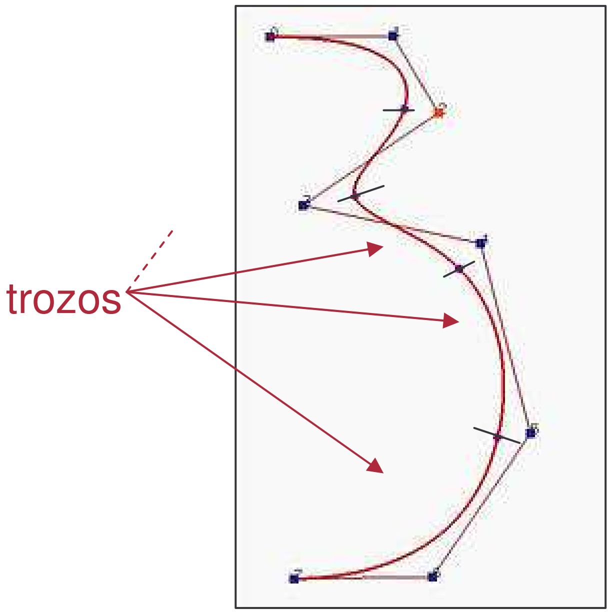


El proceso de ir pegando trozos de curvas puede ser tedioso…

- Los <u>B-splines</u> generalizan las curvas de Bezier a trozos.
- El grado de las funciones polinomiales a trozos se impone a priori.
- Se usan “nodos” que vienen a ser los tiempos en los que se subdivide la curva



Curva de Bezier



B-spline