



Diseño de curvas por ordenador: el algoritmo de De Casteljaou

Apellidos, nombre	Benítez López, Julio (jbenitez@mat.upv.es)
Departamento	Departamento de Matemática Aplicada
Centro	Universitat Politècnica de València

1 Resumen de las ideas clave

En este artículo se explicará el algoritmo de De Casteljau y su importancia en el diseño gráfico. Veremos algunas propiedades de este algoritmo útiles en las aplicaciones prácticas.

2 Objetivos

Cuando se hayan asimilado los contenidos de este documento, el alumno debe poder

- Explicar el algoritmo de De Casteljau y usar este algoritmo para diseñar curvas.
- Aplicar las propiedades de este algoritmo para optimizar los cálculos.

3 El algoritmo de De Casteljau

3.1 Parábolas

Comencemos con el siguiente algoritmo: Sean $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ tres puntos en \mathbb{R}^3 (el algoritmo también es válido si se cambia \mathbb{R}^3 por \mathbb{R}^2) y $t \in [0, 1]$. Construimos los siguientes dos puntos:

$$\mathbf{b}_0^1(t) = (1-t)\mathbf{p}_0 + t\mathbf{p}_1, \quad \mathbf{b}_1^1(t) = (1-t)\mathbf{p}_1 + t\mathbf{p}_2.$$

A continuación construimos un último punto más:

$$\mathbf{b}_0^2(t) = (1-t)\mathbf{b}_0^1(t) + t\mathbf{b}_1^1(t).$$

Mira la parte izquierda de la figura 1. A medida que t varía entre 0 y 1, el punto $\mathbf{b}_0^2(t)$ describe una curva (mira la parte derecha de la figura 1). La curva $\mathbf{b}_0^2(t)$ se llama **curva de Bézier** asociada a los puntos $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$. Estos puntos se llaman **puntos de control**.

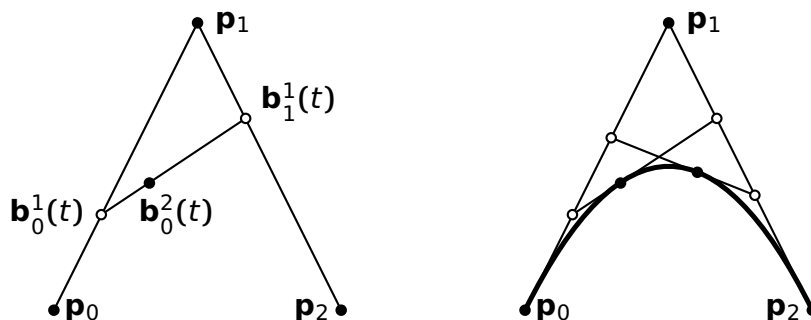


Figura 1: Se muestra a la izquierda el algoritmo de De Casteljau. A la derecha se muestra la curva de Bézier que resulta tras aplicar el algoritmo.

Este algoritmo se puede escribir de forma matricial: sean $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ tres puntos de \mathbb{R}^2 o \mathbb{R}^3 (los consideraremos columnas). Como

$$\begin{bmatrix} \mathbf{b}_0^1(t) & \mathbf{b}_1^1(t) \end{bmatrix} = \begin{bmatrix} (1-t)\mathbf{p}_0 + t\mathbf{p}_1 & (1-t)\mathbf{p}_1 + t\mathbf{p}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_2 \end{bmatrix} \begin{bmatrix} 1-t & 0 \\ t & 1-t \\ 0 & t \end{bmatrix}$$

y

$$\mathbf{b}_0^2(t) = \begin{bmatrix} \mathbf{b}_0^1(t) & \mathbf{b}_1^1(t) \end{bmatrix} \begin{bmatrix} 1-t \\ t \end{bmatrix},$$

entonces

$$\mathbf{b}_0^2(t) = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_2 \end{bmatrix} \begin{bmatrix} 1-t & 0 \\ t & 1-t \\ 0 & t \end{bmatrix} \begin{bmatrix} 1-t \\ t \end{bmatrix} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_2 \end{bmatrix}, \quad (\text{ecuación 1})$$

lo que proporciona

$$\mathbf{b}_0^2(t) = (1-t)^2 \mathbf{p}_0 + 2t(1-t) \mathbf{p}_1 + t^2 \mathbf{p}_2. \quad (\text{ecuación 2})$$

Por tanto, la curva de Bézier asociada a los puntos $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ es una parábola. Observa también que se cumple $\mathbf{b}_0^2(0) = \mathbf{p}_0$ y $\mathbf{b}_0^2(1) = \mathbf{p}_2$. Mira la figura 1.

¿Hay alguna relación entre los segmentos $\overline{\mathbf{p}_0\mathbf{p}_1}$, $\overline{\mathbf{p}_1\mathbf{p}_2}$ y la parábola? Parece que ambos segmentos son tangentes a la parábola. Vamos a comprobar este hecho. Para ello, usaremos que si $\mathbf{r} : [a, b] \rightarrow \mathbb{R}^n$ es una curva, entonces un vector tangente a la curva en el punto $\mathbf{r}(t_0)$ es el vector $\mathbf{r}'(t_0)$. Si usamos la ecuación 2, entonces como $\mathbf{b}_0^2(0) = \mathbf{p}_0$ y

$$\frac{d\mathbf{b}_0^2}{dt} = 2(t-1)\mathbf{p}_0 + (2-4t)\mathbf{p}_1 + 2t\mathbf{p}_2 \quad \Rightarrow \quad \left. \frac{d\mathbf{b}_0^2}{dt} \right|_{t=0} = -2\mathbf{p}_0 + 2\mathbf{p}_1 = 2\overline{\mathbf{p}_0\mathbf{p}_1},$$

el vector $\overline{\mathbf{p}_0\mathbf{p}_1}$ es tangente a la curva $\mathbf{b}_0^2(t)$ en \mathbf{p}_0 . De la misma manera se prueba que el segmento $\mathbf{p}_1\mathbf{p}_2$ es tangente a la curva en \mathbf{p}_2 .

Ejemplo: Vamos a dibujar (aproximadamente) el primer cuarto de la circunferencia $x^2 + y^2 = 1$ mediante una parábola de Bézier.

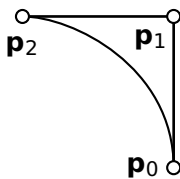


Figura 2: Un cuarto de circunferencia.

Como los puntos inicial y final del primer cuadrante de la circunferencia son $(1, 0)$ y $(0, 1)$, entonces definimos $\mathbf{p}_0 = (1, 0)$ y $\mathbf{p}_2 = (0, 1)$. Ahora falta determinar \mathbf{p}_1 . Como el segmento $\overline{\mathbf{p}_0\mathbf{p}_1}$ es tangente al arco de circunferencia que queremos aproximar, entonces \mathbf{p}_1 está en la recta $x = 1$. Y como el segmento $\overline{\mathbf{p}_1\mathbf{p}_2}$ es tangente al arco de circunferencia, entonces \mathbf{p}_1 está en la recta $y = 1$. Por tanto, $\mathbf{p}_1 = (1, 1)$. En la figura 1 puedes ver estos tres puntos de control junto la parábola.

3.2 Algoritmo general de De Casteljaou

Si se quiere una mayor flexibilidad, hay que aumentar el número de puntos de control. Esto se logra modificando el algoritmo descrito anteriormente.

Dados los $n + 1$ puntos $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$ y $t \in [0, 1]$, en primer lugar se calculan n puntos

$$\mathbf{b}_i^1(t) = (1-t)\mathbf{p}_i + t\mathbf{p}_{i+1}, \quad i = 0, \dots, n-1.$$

A continuación se calculan $n - 1$ puntos

$$\mathbf{b}_i^2(t) = (1-t)\mathbf{b}_i^1(t) + t\mathbf{b}_{i+1}^1(t), \quad i = 0, \dots, n-2.$$

Y así progresivamente hasta calcular

$$\mathbf{b}_0^n(t) = (1-t)\mathbf{b}_0^{n-1}(t) + t\mathbf{b}_1^{n-1}(t).$$

Este algoritmo se ve mejor si se pone en forma triangular, como en la tabla siguiente con cuatro puntos iniciales.

$$\begin{array}{l}
 \mathbf{p}_0 \\
 \swarrow \\
 \mathbf{p}_1 \rightarrow \mathbf{b}_0^1 = (1-t)\mathbf{p}_0 + t\mathbf{p}_1 \\
 \swarrow \\
 \mathbf{p}_2 \rightarrow \mathbf{b}_1^1 = (1-t)\mathbf{p}_1 + t\mathbf{p}_2 \rightarrow \mathbf{b}_0^2 = (1-t)\mathbf{b}_0^1 + t\mathbf{b}_1^1 \\
 \swarrow \\
 \mathbf{p}_3 \rightarrow \mathbf{b}_2^1 = (1-t)\mathbf{p}_2 + t\mathbf{p}_3 \rightarrow \mathbf{b}_1^2 = (1-t)\mathbf{b}_1^1 + t\mathbf{b}_2^1 \rightarrow \mathbf{b}_0^3 = (1-t)\mathbf{b}_0^2 + t\mathbf{b}_1^2
 \end{array}$$

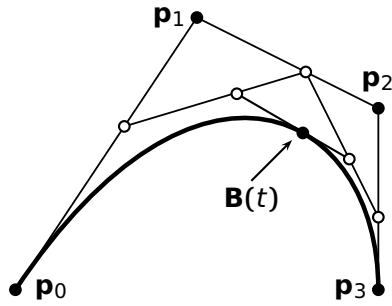


Figura 3: El algoritmo de De Casteljau en acción para dibujar una cúbica de Bézier (cuatro puntos de control).

Los puntos $\mathbf{p}_0, \dots, \mathbf{p}_n$ se llaman **puntos de control** y la curva final se llama **curva de Bézier** asociada a los puntos $\mathbf{p}_0, \dots, \mathbf{p}_n$, que será denotada en lo sucesivo por $\mathbf{B}[\mathbf{p}_0, \dots, \mathbf{p}_n](t)$ o bien por $\mathbf{B}(t)$ cuando no haya confusión. Mira la figura 3.

En la figura 4 se han dibujado varias curvas de Bézier con cuatro puntos de control. Podemos observar que la curva de Bézier es una especie de versión suavizada de la poligonal que une los puntos de control.

La representación matricial es análoga a la ecuación 1. Con cuatro puntos de control esta representación es

$$\begin{aligned}
 \mathbf{B}(t) &= \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 \end{bmatrix} \begin{bmatrix} 1-t & 0 & 0 \\ t & 1-t & 0 \\ 0 & t & 1-t \\ 0 & 0 & t \end{bmatrix} \begin{bmatrix} 1-t & 0 \\ t & 1-t \\ 0 & t \end{bmatrix} \begin{bmatrix} 1-t \\ t \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 \end{bmatrix} = \begin{bmatrix} (1-t)^3 \\ 3t(1-t)^2 \\ 3t^2(1-t) \\ t^3 \end{bmatrix},
 \end{aligned}$$

por lo que

$$\mathbf{B}[\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3](t) = (1-t)^3\mathbf{p}_0 + 3t(1-t)^2\mathbf{p}_1 + 3t^2(1-t)\mathbf{p}_2 + t^3\mathbf{p}_3. \quad (\text{ecuación 3})$$

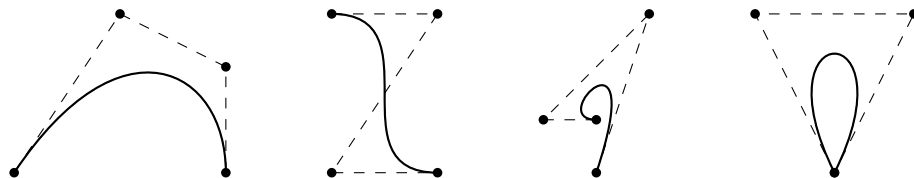


Figura 4: Varias cúbicas de Bézier.

Observa que una curva de Bézier asociada a tres puntos es un polinomio de grado 2 y una curva de Bézier asociada a cuatro puntos es un polinomio de grado 3. En general, una curva de Bézier asociada a $n + 1$ puntos es un polinomio de grado n .

Ejemplo: Vamos a dibujar la zona sombreada de la parte izquierda de la figura 5. Para

ello, la ubicamos en un sistema de coordenadas sencillo y sean

$$\mathbf{a} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

los puntos que definen los ejes (mira la parte derecha de la figura 5).

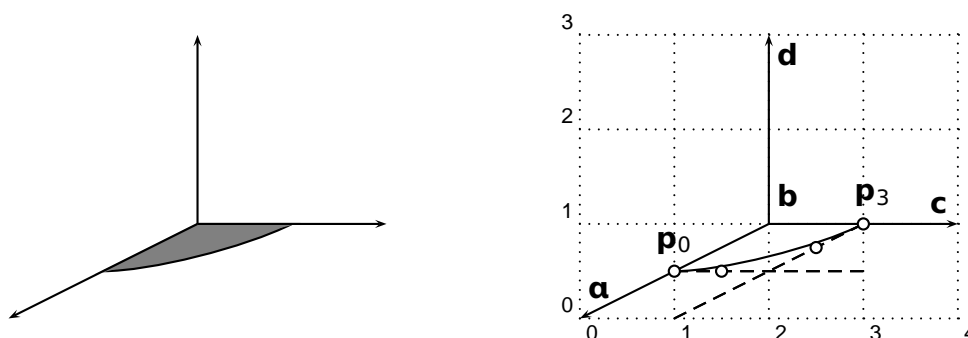


Figura 5: Un uso de las cúbicas de Bézier.

Cojamos un punto del segmento \overline{ab} y otro del segmento \overline{bc} ; por ejemplo $(1, 0.5)$ y $(3, 1)$: estos puntos son el primero y el último de los puntos de control. Ahora falta ubicar \mathbf{p}_1 y \mathbf{p}_2 . Si queremos que la tangente a la curva en \mathbf{p}_0 sea paralela al segmento \mathbf{bc} , entonces \mathbf{p}_1 tiene que estar en la recta que pasa por \mathbf{p}_0 y cuyo vector director es $\overline{bc} = \mathbf{c} - \mathbf{b} = (2, 0)$. Es decir, \mathbf{p}_1 tiene que ser de la forma $\mathbf{p}_1 = \mathbf{p}_0 + \lambda \overline{bc} = (1 + 2\lambda, 0.5)$ siendo $\lambda \in \mathbb{R}$. Similarmente, \mathbf{p}_2 tiene que estar en la recta que pasa por \mathbf{p}_3 y con vector director $\overline{ab} = \mathbf{b} - \mathbf{a} = (2, 1)$, es decir, $\mathbf{p}_2 = \mathbf{p}_3 + \mu \overline{ab} = (3 + 2\mu, 1 + \mu)$, siendo $\mu \in \mathbb{R}$. Ahora basta *jugar* con los números λ y μ para elegir los más estéticos. En la figura 5 se han elegido $\mathbf{p}_1 = (1.5, 0.5)$ y $\mathbf{p}_2 = (2.5, 0.75)$. ■

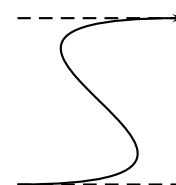


Figura 6: Figura para el problema 3.

Problema 1 Diseña una curva como la de la figura 6 por medio de una cúbica de Bézier. Las líneas discontinuas son paralelas y horizontales.

La siguiente función de Octave dibuja la curva de Bézier asociada a n puntos de \mathbb{R}^2 .

```
function castel(P)
[m,n] = size(P);
x = zeros(1,21); y = zeros(1,21); i = 1;
for t = 0:0.05:1
    B = P;
    for k = n-1:-1:1
        ceros = zeros(1,k);
        C = [(1-t)*eye(k);ceros] + [ceros;t*eye(k)];
        B = B*C;
    end
    x(i) = B(1); y(i) = B(2); i = i+1;
end
plot(x,y,'linewidth',4); hold on
plot(P(1,:),P(2,:),'.r','markersize',20)
plot(P(1,:),P(2,:),'g','linewidth',2)
```

Y ahora viene la explicación. Si queremos introducir la curva de Bézier asociada a los puntos $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{n-1}$ tenemos que introducir `castel(P)`, donde \mathbf{P} es una matriz con dos filas y n columnas, siendo la columna i -ésima el punto \mathbf{p}_i . Por ejemplo, si se quiere dibujar $\mathbf{B}[\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$, donde $\mathbf{p}_0 = (1, 0)$, $\mathbf{p}_1 = (0, 2)$, $\mathbf{p}_2 = (3, 3)$, $\mathbf{p}_3 = (2, 0)$, entonces se debe introducir `P = [1 0 3 2; 0 2 3 0]; cast(P)`.

El número de filas de \mathbf{P} es m (si el usuario ha introducido bien la matriz, entonces $m=2$) y el número de columnas es n (el número de puntos). La curva $\mathbf{B}[\mathbf{p}_0, \dots, \mathbf{p}_{n-1}](t)$, al ser de \mathbb{R}^2 , es de la forma $(x(t), y(t))$. Se van a dibujar los 21 puntos $(x(0), y(0))$, $(x(0.05), y(0.05))$, \dots , $(x(1), y(1))$. Por eso se inicializan los vectores \mathbf{x} e \mathbf{y} ; y en cada vuelta del bucle externo (el de t) se calcula una nueva componente de los vectores \mathbf{x} e \mathbf{y} . Para entender el resto, fijemos un t concreto y un valor sencillo de n , por ejemplo $n = 4$. Si

$$C_3 = \begin{bmatrix} 1-t & 0 & 0 \\ t & 1-t & 0 \\ 0 & t & 1-t \\ 0 & 0 & t \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1-t & 0 \\ t & 1-t \\ 0 & t \end{bmatrix}, \quad C_1 = \begin{bmatrix} 1-t \\ t \end{bmatrix},$$

entonces $\mathbf{B}[\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3](t) = [\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]C_3C_2C_1$, que se calcula recursivamente por medio de

$$[\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]C_3 \rightarrow ([\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]C_3)C_2 \rightarrow ([\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]C_3C_2)C_1.$$

Este es el objeto de la línea $\mathbf{B} = \mathbf{B} * \mathbf{C}$. En cada iteración del bucle de k se halla la matriz \mathbf{C} , que corresponde a las matrices C_3, C_2 y C_1 escritas antes. Para entender la línea

$$\mathbf{C} = [(1-t) * \text{eye}(k); \text{ceros}] + [\text{ceros}; t * \text{eye}(k)];$$

observemos que para C_3 se tiene

$$C_3 = (1-t) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} + t \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Fíjate que antes se ha definido el vector `ceros` como el vector nulo fila de \mathbb{R}^k .

Los siguientes enlaces de Geogebra muestran el algoritmo de De Cateljau para dibujar parábolas y cúbicas.

<https://www.geogebra.org/m/qv5udf77>

<https://www.geogebra.org/m/zhbzhwwg>

4 Curvas de Bézier y polinomios de Bernstein

Es conveniente tener una forma explícita para estudiar las propiedades de estas curvas. Sea $t \in [0, 1]$. Se definen las siguientes matrices: $\mathbf{P} = [\mathbf{p}_0 \ \mathbf{p}_1 \ \dots \ \mathbf{p}_{n-1} \ \mathbf{p}_n]$ y

$$C(t) = \begin{bmatrix} 1-t & 0 & 0 & \dots & 0 & 0 \\ t & 1-t & 0 & \dots & 0 & 0 \\ 0 & t & 1-t & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1-t & 0 \\ 0 & 0 & 0 & \dots & t & 1-t \end{bmatrix} = tL + (1-t)I_{n+1}, \quad (\text{ecuación 4})$$

en donde L es la matriz cuadrada de orden $n + 1$ con unos en la diagonal inferior a la principal y el resto de sus entradas nulas. Como

$$PC(t) = \begin{bmatrix} (1-t)\mathbf{p}_0 + t\mathbf{p}_1 & (1-t)\mathbf{p}_1 + t\mathbf{p}_2 & \cdots & (1-t)\mathbf{p}_{n-1} + t\mathbf{p}_n & (1-t)\mathbf{p}_n \end{bmatrix},$$

las n primeras columnas de $PC(t)$ producen los n puntos tras la primera etapa en el algoritmo de De Casteljaou. Las $n - 1$ primeras columnas de $(PC(t))C(t) = PC(t)^2$ producen los $n - 1$ puntos tras dos etapas del algoritmo. Y así sucesivamente. Por tanto

$$\mathbf{B}[\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n](t) = \text{la primera columna de } PC^n(t). \quad (\text{ecuación 5})$$

Luego hay que calcular $C(t)^n = (tL + (1-t)I_{n+1})^n$. Se puede probar que si A y B son matrices tales que $AB = BA$, entonces $(A + B)^n = \sum_{k=0}^n \binom{n}{k} A^k B^{n-k}$ (esta es la fórmula del binomio de Newton). Por lo que si denotamos $B_k^n(t) = \binom{n}{k} t^k (1-t)^{n-k}$, entonces

$$PC(t)^n = P(tL + (1-t)I_{n+1})^n = P \sum_{k=0}^n \binom{n}{k} t^k L^k (1-t)^{n-k} = \sum_{k=0}^n B_k^n(t) PL^k.$$

Y por tanto,

$$\mathbf{B}[\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n](t) = \sum_{k=0}^n B_k^n(t) (\text{la primera columna de } PL^k).$$

Es evidente que la primera columna de PL^0 es \mathbf{p}_0 puesto que $L^0 = I_{n+1}$. Es fácil comprobar que la primera fila de PL^k es \mathbf{p}_k para $k \in \{0, \dots, n\}$. Esto permite probar que si se define

$$B_k^n(t) = \binom{n}{k} t^k (1-t)^{n-k}, \quad k = 0, 1, \dots, n,$$

entonces

$$\mathbf{B}[\mathbf{p}_0, \dots, \mathbf{p}_n](t) = \sum_{k=0}^n B_k^n(t) \mathbf{p}_k. \quad (\text{ecuación 6})$$

Problema 2 Comprueba, por medio de la ecuación 6, que

$$\mathbf{B}[\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2](t) = (1-t)^2 \mathbf{p}_0 + 2(1-t)t \mathbf{p}_1 + t^2 \mathbf{p}_2$$

y

$$\mathbf{B}[\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3](t) = (1-t)^3 \mathbf{p}_0 + 3(1-t)^2 t \mathbf{p}_1 + 3(1-t)t^2 \mathbf{p}_2 + t^3 \mathbf{p}_3.$$

Los polinomios $B_k^n(t)$ que han aparecido han sido muy estudiados y se llaman **polinomios de Bernstein**.

5 Propiedades de las curvas de Bézier

5.1 Invarianza afín

Las aplicaciones afines juegan un papel importante en el diseño de objetos, pues a menudo, éstos deben ser trasladados, girados, escalados, ...

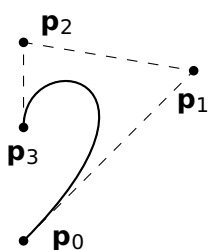
Supongamos que hemos dibujado la curva de Bézier $\mathbf{B}[\mathbf{p}_0, \dots, \mathbf{p}_n]$. A continuación queremos dibujar la imagen de esta curva mediante una aplicación afín $T : \mathbb{R}^m \rightarrow \mathbb{R}^m$. La invarianza afín permite resolver este problema de dos modos:

1. Calcular la imagen por T de los puntos de la curva ya dibujada.
2. Primero calcular $T(\mathbf{p}_0), \dots, T(\mathbf{p}_n)$ y luego dibujar la curva de Bézier asociada a los puntos de control $T(\mathbf{p}_0), \dots, T(\mathbf{p}_n)$.

Normalmente es más cómodo en la práctica la segunda opción. Veamos un ejemplo de la utilidad de esta propiedad: Vamos a dibujar mediante curvas de Bézier un corazón como el que se muestra a la derecha: dibujaremos primero una parte y luego su simétrica. Optamos por una cúbica (de hecho, las cúbicas de Bézier han demostrado ser lo suficientemente flexibles para modelar bien la mayor parte de las curvas). Tras unos pocos ensayos tomamos como puntos de control

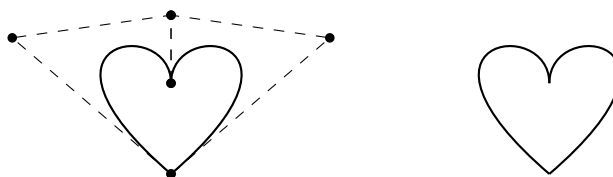


$$\mathbf{p}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{p}_1 = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}, \quad \mathbf{p}_2 = \begin{bmatrix} 0 \\ 1.75 \end{bmatrix}, \quad \mathbf{p}_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$



produciendo la curva que se muestra a la izquierda.

A continuación, simetrizamos la curva respecto el eje $x = 0$. Es trivial observar que esta simetría es $T(x, y) = (-x, y)$. Por la propiedad de la invarianza afín, basta simetrizar los puntos de control: $T(\mathbf{p}_0) = (0, 0)$, $T(\mathbf{p}_1) = (-1.5, 1.5)$, $T(\mathbf{p}_2) = (0, 1.75)$ y $T(\mathbf{p}_3) = (0, 1)$ y después, dibujar la curva de Bézier cuyos puntos de control son $T(\mathbf{p}_0), T(\mathbf{p}_1), T(\mathbf{p}_2), T(\mathbf{p}_3)$. En la figura siguiente (izquierda) se han dibujado las dos curvas juntas con los puntos de control y a la derecha sin los puntos de control.



Problema 3 ¿Cuáles son los puntos de control de las curvas de la figura 7 si los puntos inferiores de los tres corazones son $(0, 0)$, $(1.25, 0.5)$, $(2.5, 1)$? ¿Qué es lo que se tiene que hacer para dibujar un corazón en miniatura?

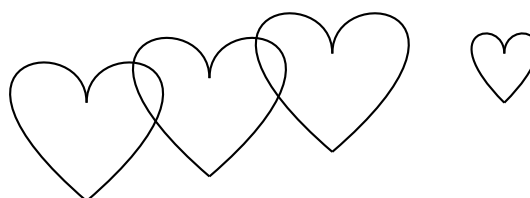


Figura 7: Un corazón, varios trasladados suyos y una versión escalada.

5.2 Interpolación inicial y final

La curva de Bézier pasa por el primer y último punto de control.

Problema 4 Prueba esto, es decir, que $\mathbf{B}[\mathbf{p}_0, \dots, \mathbf{p}_n](0) = \mathbf{p}_0$ y $\mathbf{B}[\mathbf{p}_0, \dots, \mathbf{p}_n](1) = \mathbf{p}_n$.

5.3 Vectores tangentes

En el diseño gráfico es importante saber calcular tangentes a las curvas de Bézier, es decir tenemos que saber simplificar¹

$$\frac{d}{dt}\mathbf{B}[\mathbf{p}_0, \dots, \mathbf{p}_n](t).$$

Si nos fijamos en la figura 8, observamos que el segmento que une los penúltimos puntos calculados en el algoritmo de De Casteljau es tangente a la curva de Bézier, luego el algoritmo de De Casteljau calcula la tangente sin coste extra. Observa que esto generaliza al hecho de que los segmentos $\mathbf{p}_0\mathbf{p}_1$ y $\mathbf{p}_{n-1}\mathbf{p}_n$ son tangentes a la curva $\mathbf{B}[\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n](t)$ en \mathbf{p}_0 y \mathbf{p}_n , respectivamente.

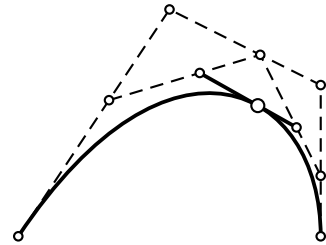


Figura 8: El algoritmo de De Casteljau también calcula tangentes.

Si $P = [\mathbf{p}_0 \ \mathbf{p}_1 \ \dots \ \mathbf{p}_n]$ y $C(t)$ está definida en la ecuación 4, como $\mathbf{B}[\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n](t)$ es la primera columna de $PC(t)^n$, entonces si \mathbf{e}_1 es el primer vector de la base canónica de \mathbb{R}^{n+1} , se cumple $\mathbf{B}[\mathbf{p}_0, \dots, \mathbf{p}_n](t) = PC^n(t)\mathbf{e}_1$. Usamos la fórmula

$$(A^n)' = \sum_{k=1}^n A^{k-1}A'A^{n-k}$$

y en particular, si A y A' conmutan, entonces $(A^n)' = nA^{n-1}A'$. Como $C(t) = tL + (1-t)I_{n+1}$, entonces $C(t)$ conmuta con $C' = L - I_{n+1}$. Por lo que

$$\frac{d}{dt}\mathbf{B}[\mathbf{p}_0, \dots, \mathbf{p}_n] = P(nC^{n-1}C')\mathbf{e}_1 = nPC^{n-1}(L - I_{n+1})\mathbf{e}_1 = n[PC^{n-1}L\mathbf{e}_1 - PC^{n-1}\mathbf{e}_1].$$

Observa que $L\mathbf{e}_1 = \mathbf{e}_2$ y por tanto, $PC^{n-1}L\mathbf{e}_1 = PC^{n-1}\mathbf{e}_2$ es la segunda columna de la matriz PC^{n-1} . También debes observar que, así como la primera columna de PC^n produce el último punto del algoritmo de De Casteljau, las dos primeras columnas de PC^{n-1} producen los penúltimos puntos del algoritmo de De Casteljau. Es decir, $PC^{n-1}L\mathbf{e}_1 - PC^{n-1}\mathbf{e}_1 = PC^{n-1}\mathbf{e}_2 - PC^{n-1}\mathbf{e}_1$ es el vector que une los dos penúltimos puntos calculados en el algoritmo de De Casteljau.

Ejemplo: Vamos a dibujar aproximadamente una circunferencia usando cúbicas de Bézier.

Supongamos que la circunferencia está centrada en el origen y tiene radio 1 (luego veremos qué hacer en el caso general). Debido a la simetría de la circunferencia, basta dibujar un cuarto de la circunferencia, el cual lo supondremos en el primer cuadrante.

El objetivo es hallar los puntos $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ tales que $\mathbf{r} = \mathbf{B}[\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$ es la cúbica buscada (mira la figura 9). Ya que el inicio del cuarto de circunferencia es $(1, 0)$ y el final es $(0, 1)$, exigiremos $\mathbf{p}_0 = (1, 0)$ y $\mathbf{p}_3 = (0, 1)$.

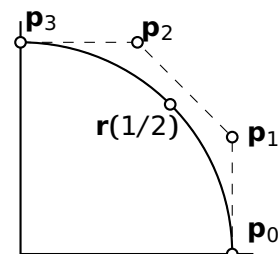


Figura 9: Aproximación de un cuarto de circunferencia.

¹Usaremos indistintamente la notación df/dt o f' para la derivada.

Como la tangente en $(1, 0)$ es vertical, exigimos que \mathbf{p}_1 está en la recta vertical que pasa por \mathbf{p}_0 , es decir, $\mathbf{p}_1 = (1, \lambda)$ para algún número λ . Por simetría, se exige también $\mathbf{p}_2 = (\lambda, 1)$. Por tanto, solo hace falta determinar λ . Forzamos que el punto que está en la mitad de la curva de Bézier pase por la mitad del cuarto de circunferencia, es decir $\mathbf{r}(1/2) = (\sqrt{2}/2, \sqrt{2}/2)$. Si usamos la ecuación 3, $\mathbf{r}(1/2) = ((4 + 3\lambda)/8, (4 + 3\lambda)/8)$, luego $\sqrt{2}/2 = (4 + 3\lambda)/8$. Y tras despejar λ , obtenemos $\lambda = 4(\sqrt{2} - 1)/3 \approx 0.55228$.

Ahora falta simetrizar la figura y para ello aplicamos la invarianza afín. Los cuadrantes que faltan por dibujar son el segundo, el tercero y el cuarto que se obtienen respectivamente a partir de las simetrías $S_1(x, y) = (-x, y)$, $S_2(x, y) = (-x, -y)$, $S_3(x, y) = (x, -y)$. Por tanto, simetrizamos los puntos de control que hemos usado para dibujar el primer cuarto de circunferencia.

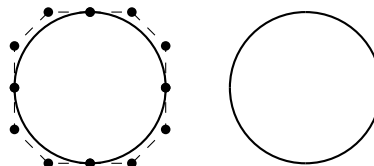


Figura 10: Aproximación de una circunferencia por una cúbica de Bézier.

Luego hay que dibujar las tres siguientes curvas de Bézier: $\mathbf{B}[(-1, 0), (-1, \lambda), (-\lambda, 1), (0, 1)]$, $\mathbf{B}[(-1, 0), (-1, -\lambda), (-\lambda, -1), (0, -1)]$ y $\mathbf{B}[(-1, 0), (-1, \lambda), (-\lambda, 1), (0, 1)]$, respectivamente.

El resultado puede verse en la figura 10. A la izquierda se han dibujado las cuatro cúbicas con sus puntos de control y a la derecha la misma curva sin los puntos de control. ■

Problema 5 El resultado del ejemplo anterior es prácticamente indistinguible de una circunferencia. Pero vamos a comprobarlo numéricamente. Sea $\mathbf{B}[\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3](t) = (x(t), y(t))$. Calcula (con Octave) $x(t)^2 + y(t)^2 - 1$ para $t = 0, 0.05, 0.1, \dots, 0.95, 1$. El valor

$$\text{máx}\{|x(t)^2 + y(t)^2 - 1| : t = 0, 0.05, 0.1, \dots, 0.95, 1\},$$

que indica el error máximo debe salirte 0.00054259, un valor realmente pequeño.

Ejemplo: Veamos cómo dibujar la circunferencia C_r centrada en el origen y de radio r , sabiendo que los puntos de control de C_1 ya están hallados. Como C_r se obtiene aplicando la homotecia $H(\mathbf{x}) = r\mathbf{x}$ a C_1 , entonces para dibujar C_r basta aplicar H a los puntos de control que permiten dibujar C_1 . Concretamente, para dibujar el primer cuadrante de C_r , los puntos de control son $(r, 0)$, $(r, r\lambda)$, $(r\lambda, r)$ y $(0, r)$. El resto de los cuadrantes son análogos.

Y ahora veamos cómo dibujar la circunferencia centrada en $\mathbf{q} = (q_1, q_2)$ y de radio r (llamémosla $C_{\mathbf{q}r}$). Si T es la traslación $T(\mathbf{x}) = \mathbf{x} + \mathbf{q}$, entonces $T(C_r) = C_{\mathbf{q}r}$ (si trasladamos C_r , obtenemos $C_{\mathbf{q}r}$). Por lo que para dibujar $C_{\mathbf{q}r}$ basta trasladar los puntos de control que permiten dibujar C_r . Concretamente, para dibujar el primer cuadrante de $C_{\mathbf{q}r}$, los puntos de control son $(r + q_1, q_2)$, $(r + q_1, r\lambda + q_2)$, $(r\lambda + q_1, r + q_2)$ y $(q_1, r + q_2)$. ■

Ejemplo: Sean \mathbf{p}_0 , \mathbf{p}_1 , \mathbf{p}_2 y \mathbf{p}_3 los puntos de control usados para dibujar aproximadamente la parte de la circunferencia $x^2 + y^2 = 1$ contenida en el primer cuadrante (mira la figura 11).

Vamos a dibujar la elipse $x^2/a^2 + y^2/b^2 = 1$. Esta elipse es la imagen de la circunferencia $x^2 + y^2 = 1$ por medio de la transformación $A(x, y) = (ax, by)$. Por tanto, los puntos de control para dibujar la elipse de forma aproximada son $A(\mathbf{p}_0)$, $A(\mathbf{p}_1)$, $A(\mathbf{p}_2)$ y $A(\mathbf{p}_3)$. Mira de nuevo la figura 11. ■

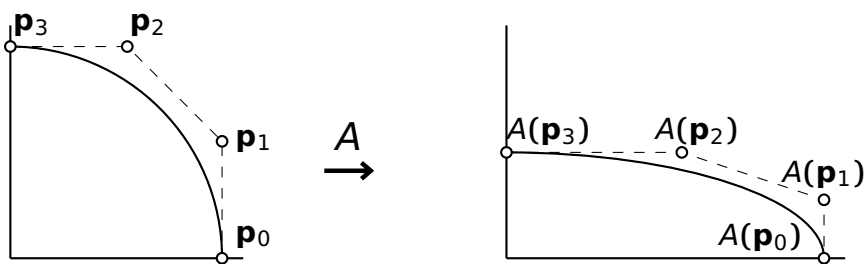


Figura 11: Una aplicación de la invarianza afín para dibujar elipses.

5.4 Pseudocontrol local

¿Qué ocurre si se mueve un punto de control? Sean las curvas

$$\mathbf{r}_1(t) = \mathbf{B}[\mathbf{p}_0, \dots, \mathbf{p}_{k-1}, \mathbf{p}, \mathbf{p}_{k+1}, \dots, \mathbf{p}_n](t), \quad \mathbf{r}_2(t) = \mathbf{B}[\mathbf{p}_0, \dots, \mathbf{p}_{k-1}, \mathbf{q}, \mathbf{p}_{k+1}, \dots, \mathbf{p}_n](t).$$

Observa que \mathbf{r}_2 se obtiene a partir de \mathbf{r}_1 moviendo el k -ésimo punto de control. Vamos a estudiar la expresión $\mathbf{r}_2(t) - \mathbf{r}_1(t)$, que mide la diferencia entre ambas curvas:

$$\begin{aligned} \mathbf{r}_2(t) - \mathbf{r}_1(t) &= \left(\sum_{j=0}^{k-1} B_j(t) \mathbf{p}_j + B_k^n(t) \mathbf{q} + \sum_{j=k}^n B_j(t) \mathbf{p}_j \right) - \left(\sum_{j=0}^{k-1} B_j(t) \mathbf{p}_j + B_k^n(t) \mathbf{p} + \sum_{j=k}^n B_j(t) \mathbf{p}_j \right) \\ &= B_k^n(t) (\mathbf{q} - \mathbf{p}) = B_k^n(t) \overrightarrow{\mathbf{p}\mathbf{q}}. \end{aligned}$$

¿Qué significa esta última expresión? Veamos un ejemplo en la figura 12.

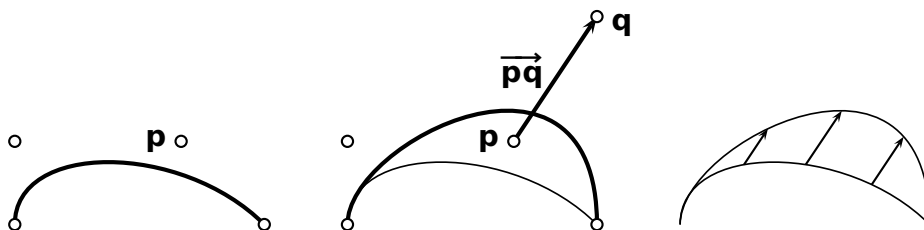


Figura 12: Cómo afecta el movimiento de un punto de control a una curva de Bézier.

Observa que $\mathbf{r}_2(t) - \mathbf{r}_1(t)$ es un múltiplo escalar de $\overrightarrow{\mathbf{p}\mathbf{q}}$ (mira la parte de la derecha de la figura 12). Pero además, $\|\mathbf{r}_2(t) - \mathbf{r}_1(t)\|$ (que mide la distancia entre las curvas punto a punto) es variable y se hace más grande alrededor del punto de control que se mueve (para un estudio más profundo, habría que calcular los extremos de la función $B_k^n(t)$ en el intervalo $t \in [0, 1]$, es decir, habría que estudiar $dB_k^n(t)/dt$).