

Geometría Analítica II
PRÁCTICA 2
CURVAS DE BEZIER A PARTIR DE UN POLÍGONO DE CONTROL

Ayudante: Guilmer González

Día 29 de abril, 2006

Curvas de Bezier

Hemos visto el Algoritmo de Casteljau para un polígono de control γG dado. Recordemos un poco en qué consiste cuando nuestro polígono es un triángulo ABC .

Considere el punto M que divide al segmento \overline{AC} en la razón t . El valor de t se encuentra entre cero y uno, ya que el punto M nos interesa dentro del segmento \overline{AC} .

De la misma forma, tomemos el punto N que divide a \overline{BC} en la misma razón t . Ahora bien, localicemos P sobre el segmento comprendido entre \overline{MN} que lo divida en la misma razón t .

Para cada valor de t entre 0 y 1, obtendremos un punto $P(t)$. El lugar geométrico de los puntos P es una parábola.

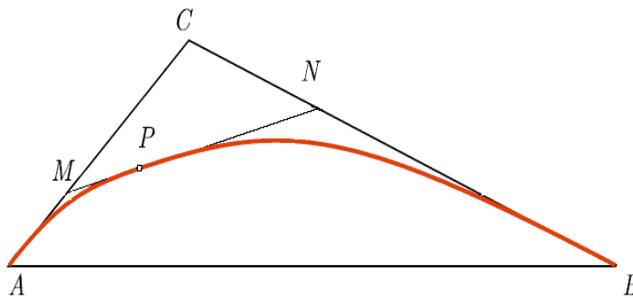


Figura 1: Polígono de control.

Consideremos un punto de origen, en alguna parte de la hoja, y usemos vectores. Según esto M divide a \overline{AC} en t . Esto significa que el punto M se

encuentra a una distancia t veces proporcional de A , y $1-t$ veces proporcional a C , esto es

$$\vec{OM} = (1-t)\vec{OA} + t\vec{OC}$$

Ahora bien, N divide a \overline{CB} en t . Esto significa que el punto N se encuentra a una distancia t veces proporcional de C , y $1-t$ veces proporcional a B , esto es

$$\vec{ON} = (1-t)\vec{OC} + t\vec{OB}$$

Siguiendo este argumento, el vector \vec{OP} lo escribimos como

$$\vec{OP} = (1-t)\vec{OM} + t\vec{ON}$$

Ahora bien, identifiquemos \vec{OP} en términos de los vértices del triángulo

$$\begin{aligned} \vec{OP} &= (1-t)\vec{OM} + t\vec{ON} \\ &= (1-t)[(1-t)\vec{OA} + t\vec{OC}] + t[(1-t)\vec{OC} + t\vec{OB}] \\ &= (1-t)^2\vec{OA} + 2t(1-t)\vec{OC} + t^2\vec{OB} \end{aligned}$$

Una representación compacta para esa colección de puntos P , en términos de el parámetro t .

Ahora generalicemos el procedimiento para trazar una curva de Bezier y programemos esto en `Matlab`.

Consideremos cuatro puntos en el plano $\mathbf{b}_0^0, \mathbf{b}_1^0, \mathbf{b}_2^0$, y \mathbf{b}_3^0 . Si aplicamos el anterior esquema para localizar un punto con la propiedad de ser punto de la curva de Bezier, se observa que cuando sobre cada segmento encontramos un punto que divida a este en la proporción t , nos conduce a otro polígono $\mathbf{b}_0^1, \mathbf{b}_1^1$, y \mathbf{b}_2^1 , al cual debemos aplicarle de nueva cuenta el algoritmo, llevándonos al segmento formado por los puntos \mathbf{b}_0^2 y \mathbf{b}_1^2 , al cual, al aplicarle el algoritmo nos conduce a un punto \mathbf{b}_0^3 el cual es un punto de la curva de Bezier.

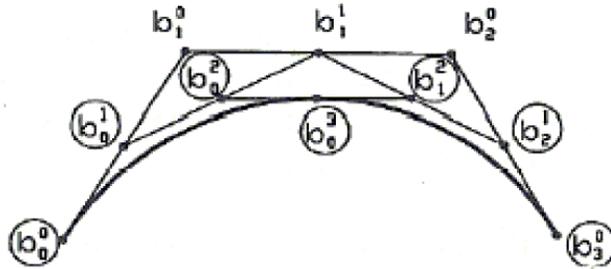


Figura 2: Construcción de una curva de Bezier.

En el script `bezier_capture.m` de Matlab, se tiene implementado este algoritmo para una colección de puntos que definan el polígono de control. Baje de la página del curso el script y observe el código.

El algoritmo es sencillo, partimos de un polígono representado matricialmente por P , matriz de 2 columnas, la primera $P(:, 1)$ contiene las coordenadas x del polígono y $P(:, 2)$ las coordenadas y , si m es el número de puntos, el algoritmo lo escribimos iterativamente como

```
for k=1:m-1,
    for i=1:m-k,
        P(i,1:2)=0.5.*P(i,1:2)+0.5.*P(i+1,1:2);
    end
end
```

Observe que recorremos cada segmento y en ese segmento calculamos un vértice del nuevo polígono de control. El procedimiento sigue hasta que finalizamos, hasta quedarnos con un segmento, y al aplicar el criterio obtenemos un punto sobre la curva de Bezier.

Pruebe el programa, juegue con distintas distribuciones de puntos, cambie el valor de la proporción sobre la cual buscar el punto

El programa finaliza mostrando una colección de puntos a valores de t . Observe que los puntos se encuentran uniformemente distribuidos en $[0, 1]$, pero no así a lo largo de la curva de Bezier.

Ejercicio: Para cuatro puntos, describa en forma paramétrica la curva de Bezier para el polígono de control que forma.