Geometría Analítica II

Práctica 3

Ayudante: Guilmer González Día 7 de abril, 2006

El día de hoy veremos:

0) Comentarios sobre los trabajos últimos.

1) Sobre el cambio de coordenadas y las curvas de Bezier.

1 Cambio de coordenadas

En la Lectura 3 del semestre, discutimos cambio de coordenadas entre dos sistemas locales generados a partir de dos vectores en el plano. La clase pasado observamos cómo construir curvas de Bezier e incluso graficamos estas. Ahora transformemos una curvas de Bezier en pantalla a un sistema coordenado dado. Revise la Lectura 3.

Consideremos los puntos $P_0(0,-1)$, $P_1(2,0)$, $P_2(-1,1)$ en el plano. Para la colección $\{P_0, P_1, P_2\}$ tenemos una representación para P. Si contamos con otra colección $\{Q_0, Q_1, Q_2\}$ obtendremos otra representación para el mismo punto. La idea es observar una representación de un conjunto en otro, es decir, hacer un cambio de coordenadas.

Consideremos los tres puntos $P_0(0,-1)$, $P_1(2,0)$ y $P_2(-1,1)$. Usemos vectores para obtener una representación del plano mediante esos puntos. Fijemos un punto O en plano como punto de referencia, usaremos el origen. La idea es representar $\vec{P_0P}$ en términos de $\vec{P_0P_1}$ y $\vec{P_0P_2}$.

Para lograrlo esto, observemos a $\vec{P_0P}$ como suma de vectores

$$\begin{aligned} \vec{P_0P} &= \vec{P_0O} + \vec{OP} \\ &= (0,1) + (x,y) \\ &= (x, y+1) \end{aligned}$$

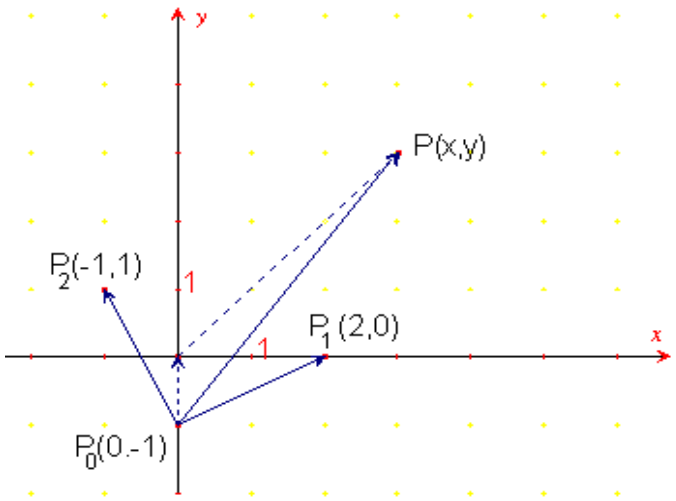

Figura 1: Un sistema de referencia para P.

el cual podemos expresarlo como combinación lineal entre $P_0\vec{P}_1$ y $P_0\vec{P}_2$, estos generan el espacio

$$\begin{aligned}(x, y+1) &= \alpha P_0\vec{P}_1 + \beta P_0\vec{P}_2 \\ &= \alpha(2,1) + \beta(-1,2) \\ &= (2\alpha - \beta, \alpha + 2\beta)\end{aligned}$$

con esto, logramos el sistema

$$\begin{aligned}x &= 2\alpha - \beta \\ y &= \alpha + 2\beta - 1\end{aligned}$$

Resolviendo para α y β tenemos

$$\begin{aligned}\alpha &= 2/5x + 1/5y + 1/5 \\ \beta &= -1/5x + 2/5y + 2/5\end{aligned}$$

esto nos permite pasar de un sistema de coordenadas a otro de manera adecuada. Es decir, si tenemos la posición de $P(x, y)$, tenemos los valores correspondientes en el sistema $\{\vec{P_0P_1}, \vec{P_0P_2}\}$, y si naturalmente contamos con los valores de α y β, contamos con un punto en ese sistema.

Regresemos al polígono de control representado por los puntos $\mathbf{b}_0, \mathbf{b}_1$, $\mathbf{b}_1$ y $\mathbf{b}_3$, como se muestra en la figura

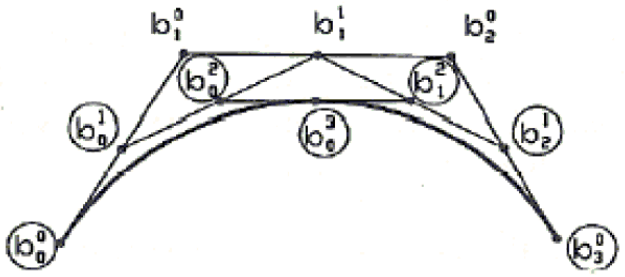

Figura 2: Construcción de una curva de Bezier.

En el `script bezier_capture.m` de `Matlab`, se tiene implementado este algoritmo para una colección de puntos que definan el poígono 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
```

En el caso que nos compete $m = 4$, para el cual, es muy sencio observar una dependencia de los coeficientes

Veamos esto, por una parte, siguiento el algoritmo de Casteljau, en la primera iteración obtenemos

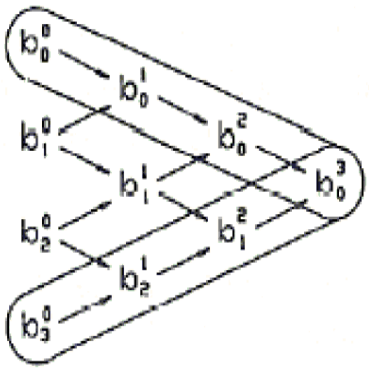

Figura 3: Dependencia de los puntos de control al formar la curva de Bezier.

$$\begin{aligned}
\mathbf{b}_0^1 &= (1-t)\mathbf{b}_0^0 + t\mathbf{b}_1^0 \\
\mathbf{b}_1^1 &= (1-t)\mathbf{b}_1^0 + t\mathbf{b}_2^0 \\
\mathbf{b}_2^1 &= (1-t)\mathbf{b}_2^0 + t\mathbf{b}_3^0
\end{aligned}$$

y en la iteración 2

$$\begin{aligned}
\mathbf{b}_0^2 &= (1-t)\mathbf{b}_0^1 + t\mathbf{b}_1^1 \\
\mathbf{b}_1^2 &= (1-t)\mathbf{b}_1^1 + t\mathbf{b}_2^1
\end{aligned}$$

para entonces lograr un punto sobre la curva de la forma

$$\mathbf{b}_0^3 = (1-t)\mathbf{b}_0^2 + t\mathbf{b}_1^2$$

Ese es el algoritmo de Casteljau, ahora veremos la dependencia de ese punto de la curva en relación con los puntos de control.

$$
\begin{aligned}
\mathbf{b}_0^3 &= (1-t)\mathbf{b}_0^2 + t\mathbf{b}_1^2 \\
&= (1-t)[(1-t)\mathbf{b}_0^1 + t\mathbf{b}_1^1] + t[(1-t)\mathbf{b}_1^1 + t\mathbf{b}_2^1] \\
&= (1-t)^2\mathbf{b}_0^1 + 2(1-t)t\mathbf{b}_1^1 + t^2\mathbf{b}_2^1 \\
&= (1-t)^2[(1-t)\mathbf{b}_0^0 + t\mathbf{b}_1^0] + 2(1-t)t[(1-t)\mathbf{b}_1^0 + t\mathbf{b}_2^0] \\
&+ t^2[(1-t)\mathbf{b}_2^0 + t\mathbf{b}_3^0] \\
&= (1-t)^3\mathbf{b}_0^1 + 32(1-t)^2t\mathbf{b}_1^0 + 3(1-t)t^2\mathbf{b}_2^0 + t^3\mathbf{b}_3^0
\end{aligned}
$$

lo cual puede escribirse en forma vectorial como

$$
\mathbf{P}(t) = \begin{bmatrix} (1-t)^3 & 3(1-t)^2t & 3(1-t)t^2 & t^3 \end{bmatrix} \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}
$$

lo que en la base canónica

$$
\mathbf{P}(t) = \begin{bmatrix} 1 & t & 3t^2 & t^3 \end{bmatrix} M \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}
$$

donde la matriz M se escribe como

$$
M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}
$$

Práctica: Construya un `m-file` donde, dado el sistema coordenado local, transforme un punto del sistema lógico al físico. Luego use el `m-file` de la página `bezier_4pts.m` y modifíquelo para que en la segunda ventana aparezca la curva de bezier transformada.

`Hacer algunos comentarios`