

# Matrices de subdivisión para curvas Beta-spline cúbicas

**Liliana Castro**

Dto. de Matemática, Univ. Nac. del Sur  
Bahía Blanca, B8000CPB, Argentina  
lcastro@uns.edu.ar

**Silvia Castro**

Dto. de Cs. e Ing. de la Computación, Univ. Nac. del Sur  
Bahía Blanca, B8000CPB, Argentina  
smc@cs.uns.edu.ar

**Susana Kahnert**

Dto. de Cs. e Ing. de la Computación, Univ. Nac. del Sur  
Bahía Blanca, B8000CPB, Argentina  
sak@cs.uns.edu.ar

**Diana Salgado**

Dto. de Matemática, Univ. Nac. del Sur  
Bahía Blanca, B8000CPB, Argentina  
dsalgado@uns.edu.ar

## Abstract

In this paper we use a subdivision technique to calculate the control points that subdivide polynomial curves. If  $\mathbf{P} = (P_0, \dots, P_n)$  is the control polygon and  $\mathbf{B}[\mathbf{P}]$  is the polynomial curve of degree  $n$  for which a subdivision algorithm is to be constructed, we use matrix operations to obtain the left polygon  $\mathbf{L}$  and the right polygon  $\mathbf{R}$  that approximate the curve  $\mathbf{B}[\mathbf{P}]$ . Each one of the polygons  $\mathbf{P}$ ,  $\mathbf{L}$ , and  $\mathbf{R}$  represent a set of points in the plane. In this work we have considered the case of cubic Beta-spline curves, with parameters  $\beta_1$  and  $\beta_2$ , and we obtained the subdivision curves for different values of these parameters. We explicitly detail the subdivision matrices that we have used for each case, and present the graphic representation of the subpolygons obtained in the different steps of the subdivision.

## Resumen

En este trabajo se emplea una técnica de subdivisión para calcular los puntos de control que subdividen a las curvas polinómicas.

Sea  $\mathbf{P} = (P_0, \dots, P_n)$  el polígono de control y  $\mathbf{B}[\mathbf{P}]$  la curva polinómica de grado  $n$  para la cual se construye el algoritmo de subdivisión. Mediante operaciones matriciales se obtienen  $\mathbf{L}$

y  $\mathbf{R}$ , polígonos a izquierda y a derecha, respectivamente, que aproximan a la curva  $\mathbf{B}[\mathbf{P}]$ . Cada uno de los polígonos  $\mathbf{P}$ ,  $\mathbf{L}$ , y  $\mathbf{R}$  representan un conjunto de puntos en el plano.

Consideramos el caso de las curvas Beta-spline cúbicas, con parámetros  $\beta_1$  y  $\beta_2$ , y realizamos la subdivisión para distintos valores de estos parámetros. Detallamos explícitamente las matrices de subdivisión utilizadas para cada caso, así como también la representación gráfica de los subpolígonos obtenidos en los distintos pasos de la subdivisión.

## Palabras claves

matrices de subdivisión, polígono de control, curvas Beta-spline

## 1. INTRODUCCION

Recordemos que un algoritmo de subdivisión es una técnica para encontrar polígonos de control  $\mathbf{L}$  (a izquierda) y  $\mathbf{R}$  (a derecha) tales que dividen a la curva  $\mathbf{B}[\mathbf{P}](t)$  en algún valor  $c \in (0, 1)$ . Si suponemos que  $\mathbf{B}[\mathbf{P}]([a, b])$  representa la curva cuando el parámetro varía sobre el intervalo  $[a, b]$ , entonces

$$\mathbf{B}[\mathbf{P}]([0, 1]) = \mathbf{B}[\mathbf{L}]([0, 1]) \cup \mathbf{B}[\mathbf{R}]([0, 1]).$$

Más precisamente,

$$\begin{aligned}\mathbf{B}[\mathbf{L}]([0, 1]) &= \mathbf{B}[\mathbf{P}]([0, c]) \\ \mathbf{B}[\mathbf{R}]([0, 1]) &= \mathbf{B}[\mathbf{P}](c, 1).\end{aligned}$$

Goldman Ronald y DeRose Tony [4] desarrollaron una forma de encontrar las matrices de subdivisión a izquierda y a derecha que se aplican a un determinado polígono de control para obtener subpolígonos a izquierda y a derecha, considerando para ello el caso de la subdivisión de curvas polinómicas.

Nuestro interés es mostrar explícitamente las matrices de subdivisión para curvas Beta-splines cúbicas, para distintos valores de los parámetros de sesgo y de tensión. Estas matrices nos brindarán los puntos de control correspondientes a  $\mathbf{L}$  y  $\mathbf{R}$ .

Asimismo queremos mostrar, gráficamente, la influencia de estos parámetros en la subdivisión.

## 2. MATRICES DE SUBDIVISION

En esta sección se desarrolla una técnica para calcular los puntos de control que subdividen a las curvas o superficies polinómicas.

Sea  $B(t) = (B_0(t), \dots, B_n(t))$ , con  $B_0(t), \dots, B_n(t)$  funciones que forman una base para la curva polinómica  $\mathbf{B}[\mathbf{P}](t)$  de grado  $n$ , para la cual será construido un algoritmo de subdivisión, siendo  $\mathbf{P} = (P_0, \dots, P_n)$  el polígono de control y  $P_0, \dots, P_n$  los puntos de control. Es decir, consideremos

$$\mathbf{B}[\mathbf{P}](t) = \sum_{k=0}^n B_k(t)P_k = B(t) * \mathbf{P}^t,$$

siendo  $\mathbf{P}^t$  la traspuesta de  $\mathbf{P}$  y  $*$  representa la multiplicación de matrices. Las funciones  $B_0(t), \dots, B_n(t)$  son continuas, suman uno y, dado que forman una base, son linealmente independientes sobre el intervalo  $[0,1]$ . Estas tres propiedades nos aseguran que, si un algoritmo de subdivisión es aplicado a la curva  $\mathbf{B}[\mathbf{P}](t)$ , entonces los polígonos de control convergen a la curva [4].

Como  $B(t)$  es una base para el espacio de los polinomios de grado  $n$ , existen matrices  $S_L$  y  $S_R$  [4] que dependen de  $c$ ,  $c \in (0, 1)$  tales que:

$$\begin{cases} B(ct) = B(t) * S_L \\ B(c + (1 - c)t) = B(t) * S_R, \end{cases} \quad (1)$$

donde  $S_L$  y  $S_R$  son matrices que reparametrizan las funciones  $B_0(t), \dots, B_n(t)$ . En el caso de la subdivisión en el punto medio, consideramos  $c = \frac{1}{2}$ .

Sea  $\mathbf{P} = (P_0, \dots, P_n)$  el polígono de control; buscamos expresiones para  $\mathbf{L}$  y  $\mathbf{R}$  tales que:

$$\begin{cases} \mathbf{B}[\mathbf{L}](t) = \mathbf{B}[\mathbf{P}](ct) \\ \mathbf{B}[\mathbf{R}](t) = \mathbf{B}[\mathbf{P}](c + (1 - c)t), \end{cases} \quad (2)$$

donde  $\mathbf{L}$  y  $\mathbf{R}$  son los subpolígonos a izquierda y a derecha respectivamente, en los cuales queda dividido el polígono original  $\mathbf{P}$ .

En forma matricial, esto se puede escribir de la siguiente manera:

$$B(t) * \mathbf{L}^t = B(ct) * \mathbf{P}^t$$

$$(B_0(t) \quad \dots \quad B_n(t)) * \begin{pmatrix} L_0 \\ \vdots \\ L_n \end{pmatrix} = (B_0(ct) \quad \dots \quad B_n(ct)) * \begin{pmatrix} P_0 \\ \vdots \\ P_n \end{pmatrix}.$$

Sustituyendo por la primera ecuación de (1) obtenemos que

$$B(t) * \mathbf{L}^t = B(t) * S_L * \mathbf{P}^t.$$

Como  $B_0(t), \dots, B_n(t)$  son linealmente independientes, se concluye que

$$\mathbf{L}^t = S_L * \mathbf{P}^t \quad (3)$$

$$\begin{pmatrix} L_0 \\ \vdots \\ L_n \end{pmatrix} = S_L * \begin{pmatrix} P_0 \\ \vdots \\ P_n \end{pmatrix}.$$

Similarmente obtenemos que:

$$\mathbf{R}^t = S_R * \mathbf{P}^t.$$

Por lo tanto, las matrices que reparametrizan las funciones de la combinación lineal sirven, además, para subdividir los polígonos de control. El próximo lema nos da expresiones explícitas para las matrices.

**Lema 1** Sea  $T = (t_0, \dots, t_n)$  una sucesión de constantes tales que  $0 \leq t_k \leq 1$  (ver lema 2.1 de [4]), y sea  $0 \leq c \leq 1$  una constante. Entonces  $S_L$  y  $S_R$  están dadas explícitamente por

$$\begin{cases} S_L = B(T)^{-1} * B(cT) \\ S_R = B(T)^{-1} * B(c + (1 - c)T) \end{cases} \quad (4)$$

donde  $cT = (ct_0, \dots, ct_n)$ ,  $c + (1 - c)T = (c + (1 - c)t_0, \dots, c + (1 - c)t_n)$  y

$$B(T) = \begin{pmatrix} B_0(t_0) & \cdots & B_n(t_0) \\ \vdots & & \vdots \\ B_0(t_n) & \cdots & B_n(t_n) \end{pmatrix}$$

*Demostración:* De (1),  $B(ct_i) = B(t_i) * S_L$ , con  $i = 0, \dots, n$ , y ésto en forma matricial es  $B(cT) = B(t) * S_L$

$$\begin{pmatrix} B_0(ct_0) & B_1(ct_0) & \cdots & B_n(ct_0) \\ B_0(ct_1) & B_1(ct_1) & \cdots & B_n(ct_1) \\ \vdots & \vdots & & \vdots \\ B_0(ct_n) & B_1(ct_n) & \cdots & B_n(ct_n) \end{pmatrix} = \begin{pmatrix} B_0(t_0) & B_1(t_0) & \cdots & B_n(t_0) \\ B_0(t_1) & B_1(t_1) & \cdots & B_n(t_1) \\ \vdots & \vdots & & \vdots \\ B_0(t_n) & B_1(t_n) & \cdots & B_n(t_n) \end{pmatrix} * S_L$$

Así,  $S_L = B(T)^{-1} * B(cT)$ .

En forma similar se encuentra la expresión de  $S_R$ . □

Estas matrices de subdivisión pueden determinarse a partir de las matrices de subdivisión de Bézier  $S_L^B$  y  $S_R^B$ , como en el caso de cualquier técnica polinomial. Dadas las funciones de Bézier de grado  $n$  (los polinomios de Bernstein) denotadas por  $b_0^n(t), \dots, b_n^n(t)$ , es decir

$$b_i^n(t) = \binom{n}{i} t^i (1 - t)^{n-i}, i = 0, \dots, n,$$

las matrices  $S_L^B$  y  $S_R^B$  se obtienen a partir de un algoritmo desarrollado en [2] y en [3], y están dadas por

$$S_L^B = \begin{pmatrix} b_0^0(c) & 0 & 0 & \cdots & 0 \\ b_0^1(c) & b_1^1(c) & 0 & \cdots & 0 \\ \vdots & \vdots & & & \vdots \\ b_0^n(c) & b_1^n(c) & b_2^n(c) & \cdots & b_n^n(c) \end{pmatrix} \tag{5}$$

$$S_R^B = \begin{pmatrix} b_0^n(c) & b_1^n(c) & \cdots & b_n^n(c) \\ 0 & b_0^{n-1}(c) & \cdots & b_{n-1}^{n-1}(c) \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & b_0^0(c) \end{pmatrix}. \tag{6}$$

Sea  $C$  la matriz cuyas componentes son los coeficientes de las combinaciones lineales que surgen de expresar cada función de la base  $B(t) = (B_0(t), \dots, B_n(t))$  en términos de las funciones de la base de Bernstein  $b(t) = (b_0^n(t), \dots, b_n^n(t))$ , esto es  $C$  es tal que:

$$\begin{aligned} B(t) &= b(t) * C \\ b(t) &= B(t) * C^{-1}. \end{aligned}$$

Entonces por (1),

$$B(t) * S_L = B(ct) = b(ct) * C = b(t) * S_L^B * C = B(t) * C^{-1} * S_L^B * C.$$

Por la linealidad de  $B(t)$  se verifica que

$$S_L = C^{-1} * S_L^B * C. \tag{7}$$

En forma similar, se demuestra que

$$S_R = C^{-1} * S_R^B * C. \tag{8}$$

De (7) y (8) resulta que  $S_L$  es similar a  $S_L^B$  y que  $S_R$  es similar a  $S_R^B$ .

Las matrices de los ejemplos que daremos en este trabajo fueron construidas usando las ecuaciones (7) y (8).

Concluimos esta sección con algunos resultados que nos permiten asegurar con precisión cuándo los polígonos de subdivisión permanecen en la cápsula convexa del polígono de control original.

**Lema 2** Sea  $T = (t_0, \dots, t_n)$  una sucesión de números reales tales que  $0 \leq t_k \leq 1$ . Consideremos

$$B(T) = \begin{pmatrix} B_0(t_0) & \cdots & B_n(t_0) \\ \vdots & & \vdots \\ B_0(t_n) & \cdots & B_n(t_n) \end{pmatrix}$$

y sea  $B(T)^{-1} = \{b_{ij}(T)\}$ . Entonces  $\sum_j b_{ij}(T) = 1, i = 0, \dots, n$ . Esto es, las filas de  $B(T)^{-1}$  suman uno.

*Demostración:* Sea  $\mathbf{u} = (1, \dots, 1)$  un vector fila de  $n$  componentes. Sabemos que  $\sum_{j=0}^n B_j(t_i) = 1, i = 0, \dots, n$ , o lo que es equivalente

$$B(T) * \mathbf{u}^t = \mathbf{u}^t.$$

Multiplicamos ambos miembros por  $B(T)^{-1}$ , obtenemos

$$B(T)^{-1} * \mathbf{u}^t = \mathbf{u}^t,$$

es decir

$$\sum_j b_{ij}(T) = 1, i = 0, \dots, n. \square$$

**Lema 3** Las filas de  $S_L = (l_{ij})$  suman uno. Esto es  $\sum_j l_{ij} = 1$ . Análogamente, las filas de  $S_R = (r_{ij})$  suman uno.

*Demostración:* Recordemos que  $B_0(t), \dots, B_n(t)$  suman uno  $\forall t \in [0, 1]$  y como cada elemento de  $cT$  está en  $[0, 1]$ , las filas de  $B(cT)$  deben sumar uno. Esto es, si  $\mathbf{u} = (1, 1, \dots, 1)$ , entonces

$$B(cT) * \mathbf{u}^t = \mathbf{u}^t.$$

Multiplicando a ambos lados por  $B(T)^{-1}$  y aplicando el Lema 2 y el Lema 1, obtenemos

$$S_L * \mathbf{u}^t = B(T)^{-1} * B(cT) * \mathbf{u}^t = B(T)^{-1} * \mathbf{u}^t = \mathbf{u}^t.$$

Entonces las filas de  $S_L$  suman uno. Un análisis similar muestra que las filas de  $S_R$  suman uno.  $\square$

**Lema 4** Se garantiza que los polígonos de subdivisión  $\mathbf{L}$  ( $\mathbf{R}$ ) permanecen por completo en la cápsula convexa de  $\mathbf{P}$  si y sólo si  $S_L$  ( $S_R$ ) consisten solamente de elementos no negativos.

*Demostración:* Sea  $\mathbf{L} = (L_0, \dots, L_n)$ ,  $S_L = (l_{ij})$ . Entonces, de (3) sigue que

$$L_i = \sum_j l_{ij} P_j.$$

Así,  $L_i$  permanece en la cápsula convexa de  $P_j$  si y sólo si  $l_{i0}, \dots, l_{in}$  son todos no negativos y

$$\sum_j l_{ij} = 1. \quad (9)$$

Esto es, las filas de  $S_L$  deben sumar uno y sus entradas deben ser no negativas. Pero, por el Lema 3, se sabe que las filas suman uno, entonces, todo lo que se requiere es que las entradas sean no negativas.

Un argumento similar se usa para probar el resultado para los puntos de control de  $\mathbf{R}$ .  $\square$

### 3. EJEMPLO: CURVA BETA-SPLINE CUBICA

La Beta-spline cúbica es una generalización de la B-spline cúbica uniforme, en la cual la propiedad de continuidad paramétrica de segundo grado ( $C^2$ ) es reemplazada por continuidad geométrica ( $G^2$ ). A continuación repasaremos estos dos conceptos.

Dada una curva paramétrica cúbica a trozos de ecuación  $Q(t)$ , con  $t \in [t_0, t_m]$ , el *vector tangente unitario* a  $Q(t)$  está definido por :

$$T(t) = \frac{Q^{(1)}(t)}{|Q^{(1)}(t)|};$$

y el *vector curvatura* por:

$$K(t) = k(t)N(t) = k(t) \frac{T^{(1)}(t)}{|T^{(1)}(t)|};$$

donde  $k(t)$  es la curvatura de  $Q(t)$  en  $t$  y  $N(t)$  es el vector normal unitario. Los vectores  $T(t)$  y  $K(t)$  representan la dirección del movimiento y la curvatura en un punto de la curva.

Notaremos  $Q^{(n)}(t_i^+)$  y  $Q^{(n)}(t_i^-)$ ,  $n = 0, 1, 2$ , los límites a derecha y a izquierda de  $Q^{(n)}$  en  $t_i$ , respectivamente, siendo  $Q^{(n)}$  la derivada  $n$ -ésima de  $Q(t)$ . Si deseamos que  $Q(t)$ ,  $T(t)$  y  $K(t)$  sean continuas en algún  $t_i$ , es decir, que la curva tenga *continuidad geométrica de segundo grado* (indica que sea  $G^2$ ), se deben verificar las siguientes condiciones para cada  $t_i$  y para cada  $\beta 1_i$  y  $\beta 2_i$ :

- i)  $Q(t_i^-) = Q(t_i^+)$
- ii)  $\beta 1_i Q^{(1)}(t_i^-) = Q^{(1)}(t_i^+)$
- iii)  $\beta 1_i^2 Q^{(2)}(t_i^-) + \beta 2_i Q^{(1)}(t_i^-) = Q^{(2)}(t_i^+)$ .

La primera condición requiere la continuidad del vector posición. La segunda, se refiere a que las derivadas laterales primeras con respecto al parámetro en  $t_i$  sean colineales, pero admite que difieran en una constante; esto significa que puede existir un cambio en la velocidad pero no en la dirección. La última condición, está relacionada con la continuidad del vector curvatura.

Estas condiciones son menos restrictivas que las que verifican las curvas B-splines cúbicas uniformes ( $\beta 1_i = 1$  y  $\beta 2_i = 0$ ), ya que éstas tienen *continuidad paramétrica de segundo grado*, es decir, verifican:

- i)  $Q(t_i^-) = Q(t_i^+)$
- ii)  $Q^{(1)}(t_i^-) = Q^{(1)}(t_i^+)$
- iii)  $Q^{(2)}(t_i^-) = Q^{(2)}(t_i^+)$ .

En el ejemplo que estamos considerando, un trozo de la curva Beta-spline cúbica está caracterizado por las funciones  $B_0(t), \dots, B_3(t)$  definidas por:

$$\begin{cases} B_0(t) = \frac{1}{\gamma}[2\beta_1^3 - 6\beta_1^3t + 6\beta_1^3t^2 - 2\beta_1^3t^3] \\ B_1(t) = \frac{1}{\gamma}[(\beta_2 + 4\beta_1^2 + 4\beta_1) + (6\beta_1^3 - 6\beta_1)t - (3\beta_2 + 6\beta_1^3 + 6\beta_1^2)t^2 \\ \quad + (2\beta_2 + 2\beta_1^3 + 2\beta_1^2 + 2\beta_1)t^3] \\ B_2(t) = \frac{1}{\gamma}[2 + 6\beta_1t + (3\beta_2 + 6\beta_1^2 - (2\beta_2 + 2\beta_1^2 + 2\beta_1 + 2)t^3)] \\ B_3(t) = \frac{1}{\gamma}2t^3, \end{cases}$$

donde  $\gamma = \frac{1}{\beta_2+12}$ ,  $\beta_1$  es el parámetro de sesgo y  $\beta_2$  es el parámetro de tensión [1].

Si se desea que la curva posea la propiedad de la cápsula convexa, basta con asignarle valores a los parámetros, de tal manera que las funciones  $B_0(t), \dots, B_3(t)$  sean no negativas.

La matriz de subdivisión  $S_L = [l_{ij}]$  se construye a partir de la ecuación (7)  $S_L = C^{-1} * S_L^B * C$ . La matriz  $S_R = [r_{ij}]$  es tal que  $r_{ij} = l_{5-i,5-j}$  con  $i, j = 1, \dots, 4$ .

Para construir la matriz  $C$  realizamos el siguiente procedimiento:

- Expresamos cada  $B_i(t)$ ,  $i = 0, \dots, 3$ , como una combinación lineal de las funciones  $b_0^3(t)$ ,  $b_1^3(t)$ ,  $b_2^3(t)$  y  $b_3^3(t)$ .
- Cada uno de los coeficientes de esta combinación lineal formarán la  $(i + 1)$ -ésima columna de  $C$ , con  $i = 0, \dots, 3$ .

La matriz  $C$  presenta la siguiente forma general:

$$\begin{pmatrix} 2\gamma\beta_1^3 & \gamma(\beta_2 + 4\beta_1^2 + 4\beta_1) & 2\gamma & 0 \\ 0 & \gamma(2\beta_1^3 + 2\beta_1 + \beta_2 + 4\beta_1^2) & \gamma(2\beta_1 + 2) & 0 \\ 0 & \gamma(2\beta_1^3 + 2\beta_1^2) & \gamma(\beta_2 + 2\beta_1^2 + 4\beta_1 + 2) & 0 \\ 0 & 2\gamma\beta_1^3 & \gamma(\beta_2 + 4\beta_1 + 4\beta_1^2) & 2\gamma \end{pmatrix}$$

Debido a que estamos considerando  $n = 3$ , la matriz  $S_L^B$  será de orden cuatro, (ver (5)), y cuatro serán también los puntos de control del polígono original  $\mathbf{P}$ .

Una vez obtenidas las matrices  $C$  y  $S_L^B$  estamos en condiciones de hallar  $S_L$ . Luego, a partir de ésta, se calcula  $S_R$ . Si multiplicamos  $S_L$  y  $S_R$  por el vector columna  $\mathbf{P}^t$ , obtenemos los vectores columnas  $\mathbf{L}^t$  y  $\mathbf{R}^t$ , respectivamente.

Utilizando el software Mathematica desarrollamos un algoritmo que nos permite hallar  $S_L$  y  $S_R$  para distintos valores de  $\beta_1$  y  $\beta_2$  y para un  $c$  cualquiera (ver Apéndice A).

En lo que respecta a la representación gráfica, hemos elaborado un algoritmo, utilizando el software MATLAB, que genera los puntos de control a izquierda y a derecha, en los distintos pasos de la subdivisión y grafica los polígonos que estos puntos determinan, respectivamente.

En todos los ejemplos hemos considerado  $\beta_1 = 1$ ,  $c = 1/2$  y comenzamos la subdivisión a partir de los cuatro puntos de control originales y su polígono asociado.

En primer lugar, sea  $\beta_2 = 0$  y  $c = \frac{1}{2}$ , es decir realizamos la subdivisión en el punto medio. Luego de realizar tres pasos en la subdivisión, la unión de las gráficas de los subpolígonos

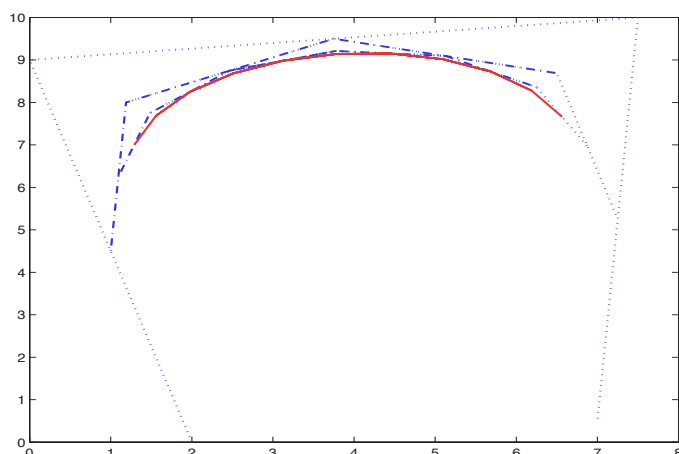


Figura 1: Tres pasos de la subdivisión en el punto medio para  $\beta_1 = 1$  y  $\beta_2 = 0$

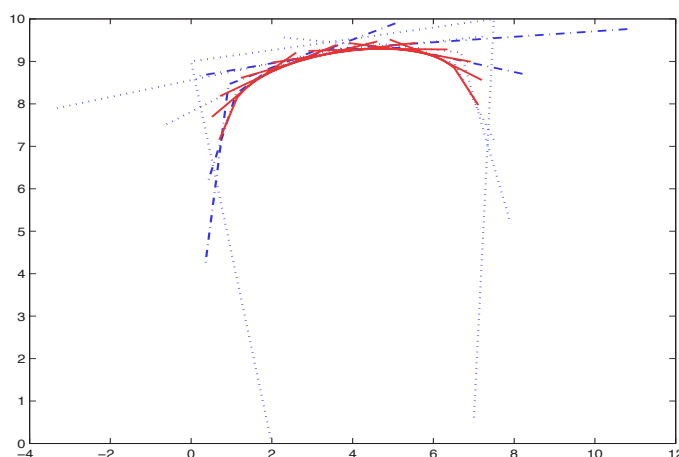


Figura 2: Tres pasos de la subdivisión en el punto medio para  $\beta_1 = 1$  y  $\beta_2 = 5$

obtenidos generan un polígono final que converge a la curva Beta-spline cúbica y que, además, está contenido en la cápsula convexa del polígono original, como puede observarse en la Figura 1.

Si suponemos  $\beta_2 = 5$ , la primera y la última fila de  $S_L$  contienen elementos negativos, por lo tanto, los primeros y los últimos puntos de los polígonos de subdivisión no quedan incluidos en la cápsula convexa del polígono original. Esto ocurre siempre que  $\beta_2$  sea positivo.

El polígono formado por la unión de las gráficas de los polígonos de subdivisión  $\mathbf{L}$  y  $\mathbf{R}$  no describen una curva continua, como se puede observar en la Figura 2. Sin embargo, luego de sucesivos pasos de subdivisión, la unión de las gráficas de los polígonos de subdivisión converge a la curva original.

Por último, consideramos  $\beta_2 = -10$ , y observamos, en este caso, un comportamiento similar al caso anterior, (ver la Figura 3). Es decir, si  $\beta_2 < 0$ , cuanto menor es el valor de este parámetro, la curva no pertenece a la cápsula convexa.

## 4. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo hemos mostrado algunos ejemplos de subdivisión de curvas Beta-spline cúbicas y sus correspondientes matrices de subdivisión.



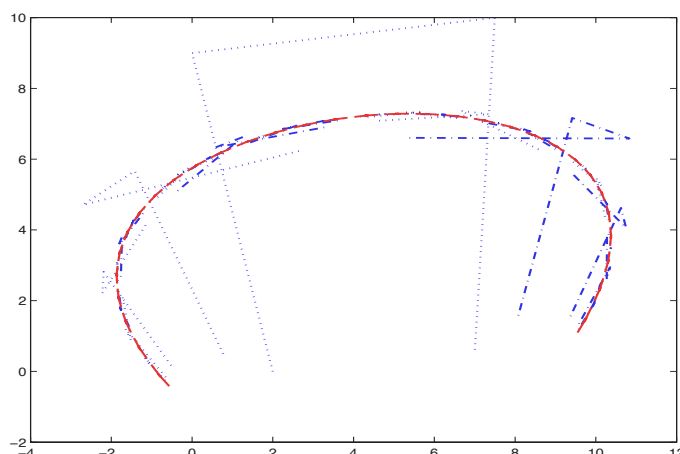


Figura 3: Cinco pasos de la subdivisión en el punto medio para  $\beta_1 = 1$  y  $\beta_2 = -10$

Se construyó un algoritmo que genera estas matrices para distintos valores de los parámetros de sesgo y tensión y para la subdivisión tanto en el punto medio como en otros casos.

En todos los ejemplos mostrados se trabajó con funciones de la base que son continuas, forman una partición de la unidad y son linealmente independientes, lo cual asegura la convergencia uniforme de los polígonos de subdivisión a la curva original. No se hizo uso de la propiedad de la cápsula convexa, ya que esta propiedad no es necesaria para la convergencia de un algoritmo de subdivisión [4].

Está claro que, a partir de cuatro puntos de control, se genera un solo trozo de la curva. Una extensión a un conjunto con mayor número de puntos de control forma parte de un trabajo futuro, así como la extensión al caso de la subdivisión de superficies Beta-spline cúbicas.

## Referencias

- [1] Barsky B.A. and Beatty J.C. Local control of bias and tension in Beta-splines. *ACM Transactions on Graphics*, 2(2):109–134, 1983.
- [2] Goldman R.N. Using degenerate Bézier triangles and tetrahedra to subdivide Bézier curves. *Computer Aided Geometric Design* 14, 6:307–311.
- [3] Goldman R.N. Markov chains and computer-aided geometric design, part II: Examples and subdivision matrices. *ACM Transactions on Graphics*, 4(1):12–40, 1985.
- [4] Goldman R.N. and DeRose T.D. Recursive subdivision without the convex hull property. *Computer Aided Geometric Design* 3, 4:247–265, December 1986.

## A. Apéndice

El programa para hallar las matrices de subdivisión a derecha y a izquierda fue realizado utilizando el software Mathematica.

```

b := Function[{i, n, t},  $\frac{n!}{(n-i)! i!} t^i (1-t)^{n-i}$ ];
Matrices[ $\beta_1$ _,  $\beta_2$ _, t_] := Module[{SLB, SRB, C, SL, SR,  $\gamma$ },
  SLB =  $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ ;
  For[i = 1, i < 5, For[j = 1, j < i + 1, SLB[[i, j]] = b[j - 1, i - 1, t]; j++]; i++];
  SRB =  $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ ;
  For[i = 1, i < 5, For[j = i, j < 5, SRB[[i, j]] = b[j - i, 4 - i, t]; j++]; i++];
   $\gamma = \frac{1}{12 + \beta_2}$ ;
  C =  $\begin{pmatrix} 2 \gamma \beta_1^3 & \gamma (\beta_2 + 4 \beta_1^2 + 4 \beta_1) & 2 \gamma & 0 \\ 0 & \gamma (2 \beta_1^3 + 2 \beta_1 + \beta_2 + 4 \beta_1^2) & \gamma (2 \beta_1 + 2) & 0 \\ 0 & \gamma (2 \beta_1^3 + 2 \beta_1^2) & \gamma (\beta_2 + 2 \beta_1^2 + 4 \beta_1 + 2) & 0 \\ 0 & 2 \gamma \beta_1^3 & \gamma (\beta_2 + 4 \beta_1 + 4 \beta_1^2) & 2 \gamma \end{pmatrix}$ ;
  SL = Simplify[Inverse[C].SLB.C]; SR =  $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ ;
  For[i = 1, i < 5, For[j = 1, j < 5, SR[[i, j]] = SL[[5 - i, 5 - j]]; j++]; i++];
  {Print[MatrixForm[SL]], Print[MatrixForm[SR]]}

```

A continuación mostramos cómo se obtienen las matrices correspondientes a los ejemplos que hemos desarrollado en este trabajo. En primer lugar, consideramos  $\beta_1 = 1$ ,  $\beta_2 = 0$  y  $t = 1/2$  (la subdivisión se realiza en el punto medio), es decir calculamos **Matrices**[1,0,1/2] y obtenemos las matrices siguientes:

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \end{pmatrix} \quad \begin{pmatrix} \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

En el segundo ejemplo consideramos  $\beta_1 = 1$ ,  $\beta_2 = 5$  y  $t = 1/2$ , y luego de calcular Matrices[1,5,1/2] obtenemos:

$$\begin{pmatrix} \frac{79}{153} & \frac{39}{68} & -\frac{55}{612} & 0 \\ \frac{11}{153} & \frac{14}{17} & \frac{16}{153} & 0 \\ \frac{5}{306} & \frac{39}{68} & \frac{251}{612} & 0 \\ -\frac{65}{1224} & -\frac{29}{68} & \frac{829}{612} & \frac{1}{8} \end{pmatrix} \begin{pmatrix} \frac{1}{8} & \frac{829}{612} & -\frac{29}{68} & -\frac{65}{1224} \\ 0 & \frac{251}{612} & \frac{39}{68} & \frac{5}{306} \\ 0 & \frac{16}{153} & \frac{14}{17} & \frac{11}{153} \\ 0 & -\frac{55}{612} & \frac{39}{68} & \frac{79}{153} \end{pmatrix}$$

Por último, en el tercer ejemplo, si  $\beta_1 = 1$ ,  $\beta_2 = -10$  y  $t = 1/2$ , debemos calcular Matrices[1,-10,1/2] y obtenemos:

$$\begin{pmatrix} \frac{11}{12} & -\frac{3}{4} & \frac{5}{6} & 0 \\ \frac{1}{3} & -\frac{1}{2} & \frac{7}{6} & 0 \\ \frac{5}{12} & -\frac{3}{4} & \frac{4}{3} & 0 \\ \frac{5}{24} & \frac{1}{8} & \frac{13}{24} & \frac{1}{8} \end{pmatrix} \begin{pmatrix} \frac{1}{8} & \frac{13}{24} & \frac{1}{8} & \frac{5}{24} \\ 0 & \frac{4}{3} & -\frac{3}{4} & \frac{5}{12} \\ 0 & \frac{7}{6} & -\frac{1}{2} & \frac{1}{3} \\ 0 & \frac{5}{6} & -\frac{3}{4} & \frac{11}{12} \end{pmatrix}$$