Apéndice

El algoritmo BFEA se implementó cuando se comercializaba la versión 4.0 de *Mathematica*. Esta versión no incluía el comando `Piecewise`. Actualmente se comercializa la versión 5.1 de *Mathematica* que incluye el comando `Piecewise`. Para las personas que saben programar en *Mathematica*, pueden pensar que el algoritmo BFEA es innecesario dado que, en esta nueva versión de *Mathematica*, es posible obtener las expresiones analíticas de los B-splines sin el algoritmo BFEA.

Esta fue también una primera observación que tuvimos los autores, pero hemos optado por seguir utilizando este algoritmo simplemente porque al codificarlo obtenemos un programa que es más veloz que otro en el que no se haga uso del mismo.

En este Apéndice hemos decidido presentar al lector las dos opciones y los tiempos que demanda la ejecución de ambas. De este modo podrá apreciarse la ventaja en el ahorro de tiempo cuando se usa el algoritmo BFEA.

A continuación mostramos el código de un programa para obtener las expresiones analíticas de los B-splines sin hacer uso del algoritmo BFEA.

Programa para obtener las expresiones analíticas de los B-splines sin usar el algoritmo BFEA.

```
$Divide[n_,d_]:=If[d===0,0,n/d]


Bs[t_,i_,k_,x_]:=
  PiecewiseExpand[
    $Divide[t-x[[i]],x[[i+k-1]]-x[[i]]]Bs[t,i,k-1,x]+
      $Divide[x[[i+k]]-t,x[[i+k]]-x[[i+1]]]Bs[t,i+1,k-1,x]
    ]

Bs[t_,i_,1,x_]:=Piecewise[{{0,t<x[[i]]},{1,t<x[[i+1]]}},0]
```

Como puede apreciarse el código es bastante corto y se hace uso del comando incorporado en *Mathematica* `PiecewiseExpand` para implementar el comando `Bs`.

Para que el lector note la diferencia en el tiempo de ejecución entre el comando `Bs` y el comando `Bsplines`, implementado en el paquete `BsplinesCurves` (que se

basa el algoritmo BFEA), usaremos los siguientes ejemplos. Hacemos la observación que no se muestran todos lo resultados por ser extensos; mostrándose solo el tiempo de ejecución que es lo que más interesa en este caso.

Tiempo empleado con el comando Bs.

```
k=5;

Bs[t,#,k,Range[0,15]]&/@Range[11]//Timing

{0.156 Second,...}
```

Tiempo empleado con el comando Bspline.

```
<<BsplinesCurves`

k=5;

Bspline[k,

Knots->#]&/@Partition[Range[0,15],{k+1},{1}]//Timing

{0.062 Second,...}
```

Autores:

Robert Ipanaqué Chero
robertchero@hotmail.com
http://www.unp.edu.pe/pers/ripanaque

Rubén Teodoro Urbina Guzmán
teourg@hotmail.com

Segundo Basilio Correa Erazo
sebacoer22@hotmail.com

DEPARTAMENTO ACADEMICO DE MATEMATICA
UNIVERSIDAD NACIONAL DE PIURA, PERU

Bibliografía

Escuela Universitaria Politécnica. *Capítulo V: Curvas B-splines*. Curso 2003/04. Web Page: http://www.pdipas.us.es/e/esplebrue/cap5a.pdf

Farin-Gerald. *A History of Curves and Surfaces in CAGD*. Computer Science and Engineering. Arizona State University.

Rogers-David y Adams-Alan. *Mathematical elements for computer graphics*. New York. Mc Graw-Hill, 1990.

Urbina-Rubén e Ipanaqué-Robert. *Aplicación de los polinomios de Bézier y B-Spline en el diseño de curvas en el plano y el espacio usando la computadora*. Tesis de Licenciatura. Universidad Nacional de Piura, 1998.

Wolfram Research. *The Mathematica Book.*

Capítulo 1

B-splines expresados analíticamente

En este capítulo presentamos el algoritmo implementado (algoritmo al que hemos decidido llamar algoritmo BFEA[1]) para generar las expresiones analíticas de las funciones B-splines con algún sistema de cálculo simbólico. Partimos de la definición recursiva, planteada por Cox-deBoor, de las funciones B-splines y a partir de ella elaboramos una definición alternativa que constituye por sí misma el algoritmo BFEA.

1.1 Definición recursiva de los B-splines

Sean $k \in \mathbf{N}$ y una sucesión no decreciente de números reales $x = \{x_i\}$ de longitud por lo menos $k + 1$, llamada vector de nodos o sucesión de nodos. El i-ésimo B-spline de orden k respecto al vector de nodos x se define por

$$N_{i,k}(t) = \frac{t - x_i}{x_{i+k-1} - x_i} N_{i,k-1}(t) + \frac{x_{i+k} - t}{x_{i+k} - x_{i+1}} N_{i+1,k-1}(t) \qquad \ldots(1)$$

para todo número real t, con

$$N_{i,1}(t) = \begin{cases} 0, & t < x_i \\ 1, & t < x_{i+1} \\ 0, & t \geq x_{i+1} \end{cases} \qquad \ldots(2)$$

Aquí se adopta la convención $\frac{0}{0} = 0$.

1.2 Fundamentos del algoritmo BFEA

En (2) las funciones $N_{i,1}$ ya tienen una expresión analítica clara. Veamos que sucede para los valores $k > 2$. Por ejemplo y de acuerdo a (1), la expresión analítica de las funciones $N_{i,2}$ se obtiene así

[1] B-splines Functions Expressed Analytically (Funciones B-splines Expresadas Analíticamente).

$$N_{i,2}(t) = \frac{t - x_i}{x_{i+k-1} - x_i} N_{i,1}(t) + \frac{x_{i+k} - t}{x_{i+k} - x_{i+1}} N_{i+1,1}(t) .$$

Al reemplazar las funciones $N_{i,1}$ y $N_{i+1,1}$ por sus respectivas expresiones analíticas obtenemos

$$N_{i,2}(t) = \frac{t - x_i}{x_{i+k-1} - x_i} \begin{cases} 0, & t < x_i \\ 1, & t < x_{i+1} \\ 0, & t \geq x_{i+1} \end{cases} + \frac{x_{i+k} - t}{x_{i+k} - x_{i+1}} \begin{cases} 0, & t < x_{i+1} \\ 1, & t < x_{i+2} \\ 0, & t \geq x_{i+2} \end{cases} . \qquad \text{... (3)}$$

La dificultad que encontramos es que no se obtiene directamente un valor 1 ó 0, para $N_{i,1}$ y $N_{i+1,1}$, sino que aparecen las "condicionales"

$$\begin{cases} 0, & t < x_i \\ 1, & t < x_{i+1} \\ 0, & t \geq x_{i+1} \end{cases} \quad \text{y} \quad \begin{cases} 0, & t < x_{i+1} \\ 1, & t < x_{i+2} \\ 0, & t \geq x_{i+2} \end{cases} .$$

Por supuesto que en forma manual la expresión (3) puede operarse hasta obtener el siguiente resultado

$$N_{i,2}(t) = \begin{cases} 0, & t < x_i \\ \frac{t - x_i}{x_{i+k-1} - x_i}, & t < x_{i+1} \\ 0, & t \geq x_{i+1} \end{cases} + \begin{cases} 0, & t < x_{i+1} \\ \frac{x_{i+k} - t}{x_{i+k} - x_{i+1}}, & t < x_{i+2} \\ 0, & t \geq x_{i+2} \end{cases}$$

más aun, esta expresión puede simplificarse aplicando las reglas de suma de funciones de tal manera que se obtiene

$$N_{i,2}(t) = \begin{cases} 0, & t < x_i \\ \frac{t - x_i}{x_{i+k-1} - x_i}, & t < x_{i+1} \\ \frac{x_{i+k} - t}{x_{i+k} - x_{i+1}}, & t < x_{i+2} \\ 0, & t \geq x_{i+2} \end{cases} . \qquad \text{... (4)}$$

Pero los lenguajes de programación simbólica no piensan ni toman decisiones para efectuar estas operaciones.

Ante semejante situación debemos buscar una forma de llegar al mismo resultado, efectuando operaciones matemáticas válidas, pero siguiendo un "camino" diferente.

Planteemos entonces la posibilidad de operar por intervalos. Ilustremos lo que se plantea trabajando con la misma función, $N_{i,2}$. Para esta función en intervalo $[x_i , x_{i+1}\rangle$ se tiene

$$N_{i,2}(t) = \frac{t - x_i}{x_{i+k-1} - x_i},$$

puesto que en este intervalo el valor de las funciones $N_{i,1}$ es 1 y el de las funciones $N_{i+1,1}$ es 0; y en el intervalo $[x_{i+1}, x_{i+2}\rangle$ las mismas funciones se expresan

$$N_{i,2}(t) = \frac{x_{i+k} - t}{x_{i+k} - x_{i+1}},$$

ya que en este otro intervalo el valor de las funciones $N_{i,1}$ es 0 y el de las funciones $N_{i+1,1}$ es 1. Teniendo en cuenta que fuera de estos intervalos el valor de las funciones $N_{i,2}$ es cero, podemos escribir las expresiones analíticas de las funciones $N_{i,2}$ como en (3).

De acuerdo a lo hasta aquí planteado vemos que surge la necesidad de conseguir, de alguna manera, que las funciones puedan trabajarse por intervalos para luego anexarle los respectivos dominios. ¿Y cómo lograr semejante resultado? Una primera idea para responder a esta interrogante es la que se explica a continuación.

Sea la secuencia de nodos $x = \{x_1, x_2, \ldots, x_{n+1}\}$, con $n \in \mathbf{N}$. Con respecto a x se pueden definir n funciones $N_{i,1} : \mathbf{R} \to \mathbf{R}$, a saber:

$$N_{1,1}(t) = \begin{cases} 0, & t < x_1 \\ 1, & t < x_2 \\ 0, & t \geq x_2 \end{cases}, \quad N_{2,1}(t) = \begin{cases} 0, & t < x_2 \\ 1, & t < x_3 \\ 0, & t \geq x_3 \end{cases}, \ \ldots, \ N_{n,1}(t) = \begin{cases} 0, & t < x_n \\ 1, & t < x_{n+1} \\ 0, & t \geq x_{n+1} \end{cases}.$$

Nótese que la secuencia $\{x_1, x_2, \ldots, x_{n+1}\}$ se particiona en n intervalos, los cuales son: $[x_1, x_2\rangle$, $[x_2, x_3\rangle$, …, $[x_n, x_{n+1}\rangle$.

A continuación vamos a construir una matriz cuadrada de orden n. En esta matriz las columnas indicaran los n intervalos que se han generado con los nodos $\{x_1, x_2, \ldots, x_{n+1}\}$ y las filas indicaran el valor de cada una de las n funciones $N_{i,1} : \mathbf{R} \to \mathbf{R}$, definidas con respecto a $\{x_1, x_2, \ldots, x_{n+1}\}$, en los intervalos indicados por las columnas. Esta matriz, entonces tiene la siguiente forma:

	$[x_1, x_2\rangle$	$[x_2, x_3\rangle$	$\cdots$	$[x_i, x_{i+1}\rangle$	$\cdots$	$[x_n, x_{n+1}\rangle$
$N_{1,1}(t)$	1	0	$\cdots$	0	$\cdots$	0
$N_{2,1}(t)$	0	1		0	$\cdots$	0
$\vdots$	$\vdots$		$\ddots$			$\vdots$
$N_{i,1}(t)$	0	0		1		0
$\vdots$	$\vdots$	$\vdots$			$\ddots$	
$N_{n,1}(t)$	0	0	$\cdots$	0		1

... (5)

Claramente se aprecia que la matriz así formada es una *matriz identidad*. Lo que resta es aprovechar esta construcción para nuestro propósito. Con esta finalidad anotaremos, a continuación, dos ideas clave:

i. Primeramente ha surgido, con esta construcción, una forma de obtener el valor de cualquiera de las funciones indicadas en las columnas en cualquiera de los intervalos indicados en las filas de la matriz obtenida.

ii. En segundo lugar destaquemos que es factible utilizar el algoritmo para construir una matriz identidad, el cual nos dice lo siguiente:

$$\mathbf{I}_n = (\mathbf{i}_{ij})_{n\times n} \,/\, \mathbf{i}_{ij} = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases} = \delta_i^j ,$$

donde δ es la función *delta de Kronecker*.

Para lograr nuestro propósito definamos ahora a $[x_j , x_{j+1}\rangle$ como el j -ésimo intervalo formado a partir de los nodos x_j , x_{j+1}. A continuación añadiremos el subíndice j a $N_{i,1}$ para indicar que el valor que se obtenga es el valor de $N_{i,1}$ en el intervalo $[x_j , x_{j+1}\rangle$, en forma breve podemos anotar $N_{i,1,j}(t) = \delta_i^j$.

Como podrá verse esta nueva notación nos permitirá obtener el valor de cualquier función $N_{i,1}$ en algún j -ésimo intervalo. Por ejemplo, $N_{1,1,1}(t)$ devuelve el valor de $N_{1,1}(t)$ en el primer intervalo (esto es, en $[x_1, x_2\rangle$), así tenemos $N_{1,1,1}(t) = 1$. De la misma manera $N_{1,1,2}(t)$ devuelve el valor de $N_{1,1}(t)$ en el segundo intervalo (esto es, en $[x_2 , x_3\rangle$), por lo cual $N_{1,1,2}(t) = 0$; etc.

Este resultado no sería útil de no ser porque es factible añadir este mismo subíndice a las funciones $N_{i,k}$ y de esta manera puede operarse por intervalos para obtener las funciones $N_{i,k}$ ($k > 1$) expresadas analíticamente. Estas operaciones se asemejan a las que se realizaron cuando se obtuvo (4) y (5).

Por ejemplo, para obtener las expresiones analíticas de las funciones $N_{i,2}$ procederemos como sigue. En el intervalo $j = i$ (esto es $[x_i , x_{i+1}\rangle$) de acuerdo a (1) obtenemos

$$N_{i,2,i}(t) = \frac{t - x_i}{x_{i+k-1} - x_i}$$

y en el intervalo $j = i + 1$(esto es $[x_{i+1}, x_{i+2}\rangle$) también de acuerdo a (1) obtenemos

$$N_{i,2,i+1}(t) = \frac{x_{i+k} - t}{x_{i+k} - x_{i+1}}.$$

Y puesto que fuera de estos intervalos el valor de las funciones $N_{i,2}$ es cero, obtenemos nuevamente la expresión (3).

De manera similar podemos trabajar por intervalos para valores de k mayores que 2 y así partiendo de las ideas (*i*) y (*ii*) hemos llegado a implementar un algoritmo que puede emplearse para obtener las expresiones analíticas de las funciones B-splines con asistencia de un lenguaje de programación simbólico. Esto gracias a que al operar por intervalos nos independizamos de "condicionales" como las que aparecieron en (2) y trabajamos directamente con el valor 1 o con el valor 0.

1.3 Algoritmo BFEA para generar las expresiones analíticas de los B-splines

En resumen, las expresiones analíticas de las funciones B-splines de orden k definidas respecto al vector de nodos $x = \{x_i\}$, pueden obtenerse mediante la siguiente fórmula recursiva

$$N_{i,k,j}(t) = \frac{t - x_i}{x_{i+k-1} - x_i} N_{i,k-1,j}(t) + \frac{x_{i+k} - t}{x_{i+k} - x_{i+1}} N_{i+1,k-1,j}(t),$$

con $N_{i,1,j}(t) = \delta_i{}^j$; donde j representa el j -ésimo intervalo $[x_j, x_{j+1}\rangle$ y δ es la función delta de Kronecker. Incluyendo la convención $\frac{0}{0} = 0$,

1.4 El algoritmo BFEA codificado en *Mathematica*

La codificación del algoritmo BFEA en *Mathematica* es bastante sencilla. En este texto damos la idea general de la misma, siendo ésta la misma que se ha utilizado en la codificación del paquete **`BsplinesCurves`** y cuya funcionalidad se explica en detalle en el capítulo 3.

A continuación definimos dos comandos: **$Divide** y **NN**. El comando **$Divide** permite adoptar la convención $\frac{0}{0}=0$ y el comando **NN** es la definición recursiva de las funciones B-splines.

Definición de los comandos $Divide y NN.

```
$Divide[nn_, dd_] := If[dd === 0, 0, nn/dd];

NN[t_, i_, k_, j_, x_] :=
    Plus[$Divide [t - x[[i]], x[[i + k - 1]] - x[[i]]]*
        NN[t, i, k - 1, j, x],
      $Divide [x[[i + k]] - t, x[[i + k]] - x[[i + 1]]]*
        NN[t, i + 1, k - 1, j, x]
         ]

NN[t_, i_, 1, j_, x_] := If[i == j, 1, 0]
```

Con estos dos comandos ya es posible generar las expresiones analíticas de los B-splines. Primero daremos una idea explicativa de la obtención de las mismas por intervalos, como lo mencionamos en el artículo anterior, donde explicábamos la obtención del algoritmo.

Supongamos que para la secuencia de nodos $x=\{0,1,2,3,4,5,6\}$ se desea encontrar las ecuaciones matemáticas de la funciones B-splines de primer orden (k = 1) en el primer intervalo, esto es en el intervalo $[0,1\rangle$. Teniendo en cuenta que se generan seis funciones, digitamos

Funciones B-splines $N_{i,1}$ con respecto a la secuencia de nodos X = {0,1,2,3,4,5,6} en el intervalo $[0,1\rangle$.

```
Table[NN[t,i,1,1,{0,1,2,3,4,5,6}],{i,6}]
{1, 0, 0, 0, 0, 0}
```

Obtengamos, para la misma secuencia de nodos, las ecuaciones matemáticas de las funciones B-splines de primer orden (k = 1) en el segundo intervalo, esto es en el intervalo $[1, 2\rangle$.

Funciones B-splines $N_{i,1}$ con respecto a la secuencia de nodos X = {0,1,2,3,4,5,6} en el intervalo $[1, 2\rangle$.

```
Table[NN[t,i,1,2,{0,1,2,3,4,5,6}],{i,6}]
{0, 1, 0, 0, 0, 0}
```

Más aún podemos obtener las expresiones analíticas de las funciones B-splines de primer orden en todos los intervalos que se forman para la secuencia de nodos $X = \{0,1,2,3,4,5,6\}$.

Funciones B-splines $N_{i,1}$ con respecto a la secuencia de nodos X = {0,1,2,3,4,5,6}.

```
Table[
  NN[t,i,1,j,{0,1,2,3,4,5,6}],
  {i,6},{j,6}]
{{1,0,0,0,0,0},{0,1,0,0,0,0},{0,0,1,0,0,0},{0,0,0,1,0,0},
 {0,0,0,0,1,0},{0,0,0,0,0,1}}

MatrixForm[%]
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

La última salida obtenida es un caso particular de la matriz identidad (5) presentada en la sección 1.3.

Obtengamos, ahora, las expresiones analíticas de las funciones B-splines de segundo orden (k = 2), definidas con respecto a la misma secuencia de nodos.

Funciones B-splines $N_{i,2}$ con respecto a la secuencia de nodos X = {0,1,2,3,4,5,6} .

```
Table[
  NN[t,i,2,j,{0,1,2,3,4,5,6}],
  {i,5},{j,6}]
{{t,2-t,0,0,0,0},{0,-1+t,3-t,0,0,0},
 {0,0,-2+t,4-t,0,0},{0,0,0,-3+t,5-t,0},{0,0,0,0,-4+t,6-t}}
```

```
MatrixForm[%]
```

$$\begin{pmatrix} t & 2-t & 0 & 0 & 0 & 0 \\ 0 & -1+t & 3-t & 0 & 0 & 0 \\ 0 & 0 & -2+t & 4-t & 0 & 0 \\ 0 & 0 & 0 & -3+t & 5-t & 0 \\ 0 & 0 & 0 & 0 & -4+t & 6-t \end{pmatrix}$$

Y luego las expresiones analíticas de las funciones B-splines de tercer orden (k = 3), definidas con respecto a la misma secuencia de nodos.

Funciones B-splines $N_{i,3}$ con respecto a la secuencia de nodos X = {0,1,2,3,4,5,6} .

```
Table[
  NN[t,i,3,j,{0,1,2,3,4,5,6}],
  {i,5},{j,6}]
```

$$\left\{\left\{\frac{t^2}{2}, \frac{1}{2}(3-t)(-1+t)+\frac{1}{2}(2-t)t, \frac{1}{2}(3-t)^2, 0, 0, 0\right\},\right.$$
$$\left\{0, \frac{1}{2}(-1+t)^2, \frac{1}{2}(4-t)(-2+t)+\frac{1}{2}(3-t)(-1+t), \frac{1}{2}(4-t)^2, 0, 0\right\},$$
$$\left\{0, 0, \frac{1}{2}(-2+t)^2, \frac{1}{2}(5-t)(-3+t)+\frac{1}{2}(4-t)(-2+t), \frac{1}{2}(5-t)^2, 0\right\},$$
$$\left.\left\{0, 0, 0, \frac{1}{2}(-3+t)^2, \frac{1}{2}(6-t)(-4+t)+\frac{1}{2}(5-t)(-3+t), \frac{1}{2}(6-t)^2\right\}\right\}$$

```
MatrixForm[%//Simplify]
```

$$\begin{pmatrix} \frac{t^2}{2} & -\frac{3}{2}+3t-t^2 & \frac{1}{2}(-3+t)^2 & 0 & 0 & 0 \\ 0 & \frac{1}{2}(-1+t)^2 & -\frac{11}{2}+5t-t^2 & \frac{1}{2}(-4+t)^2 & 0 & 0 \\ 0 & 0 & \frac{1}{2}(-2+t)^2 & -\frac{23}{2}+7t-t^2 & \frac{1}{2}(-5+t)^2 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(-3+t)^2 & -\frac{39}{2}+9t-t^2 & \frac{1}{2}(-6+t)^2 \end{pmatrix}$$

Capítulo 2

Expresiones Analíticas de las curvas de B-splines

En este capítulo utilizamos el algoritmo BFEA para obtener las expresiones analíticas de las curvas de B-splines. Este mismo algoritmo resulta útil para obtener las expresiones analíticas de las curvas de B-splines puesto que éstas se definen a partir de las funciones B-splines.

2.1 Definición recursiva de las curvas de B-splines

Sean $k \in \mathbf{N}$, la secuencia de nodos $x = \{x_i\}$ con $xi \le x_{i+k} \quad \forall i \in \mathbf{N}$. La restricción de

$$\alpha(t) = \sum_{i=1}^{n+1} p_i N_{i,k}(t), \quad p_i \in \mathbf{R}^n,$$

(incluyendo la convención $\frac{0}{0} = 0$) al intervalo $[a, b] \subset \mathbf{R}$ es llamada **curva de B-splines de orden *k*** respecto a la secuencia de nodos $x = \{x_i\}$ y asociada al conjunto de puntos p_i. Los coeficientes p_i son llamados *puntos de control* o *puntos de de Boor*. Al polígono determinado por el conjunto $P = \{p_i\}_{i=1}^{n+1}$ se le denomina polígono de control.

2.2 Definición de polígono de control (cápsula convexa)

Más adelante cuando se mencionen las propiedades de las curvas de B-splines se mencionará la llamada cápsula convexa. Por ello daremos, a continuación, la definición de la misma.

Definición.

(1) Un conjunto C de puntos es convexo si para todo $p, q \in C$ el segmento de recta $\overline{pq}$ está contenido en C.

(2) La cápsula convexa de un conjunto *S* de puntos es el conjunto convexo más pequeño *C* que contiene a *S*.

Observación.

La cápsula convexa de un conjunto finito de puntos *S* es un polígono convexo cuyos vértices (puntos de esquina) son elementos de *S*.

Mathematica incluye el paquete **ComputationalGeometry** que incorpora el comando **ConvexHull**, con el cual es posible obtener la cápsula convexa de un conjunto finito de puntos del plano (ver [2]).

A continuación mostramos un ejemplo de cómo funciona este comando, para ello consideramos el siguiente conjunto de puntos del plano

(4.4,14), (6.7,15.25), (6.9,12.8), (2.1,11.1), (9.5,14.9), (13.2,11.9), (10.3,12.3), (6.8,9.5), (3.3,7.7), (0.6,5.1), (5.3,2.4), (8.45,4.7), (11.5,9.6), (13.8,7.3), (12.9,3.1), (11,1.1)

asignados a la variable **data2D**.

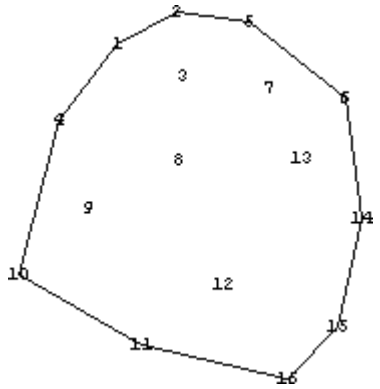
Cargando el paquete ComputationalGeometry y obteniendo la cápsula convexa del conjunto de puntos dado.

```
<<DiscreteMath`ComputationalGeometry`

data2D = {{4.4, 14}, {6.7, 15.25}, {6.9, 12.8}, {2.1, 11.1},
{9.5, 14.9}, {13.2, 11.9}, {10.3, 12.3}, {6.8, 9.5},
{3.3, 7.7}, {0.6, 5.1}, {5.3, 2.4}, {8.45, 4.7},
{11.5, 9.6}, {13.8, 7.3}, {12.9, 3.1}, {11, 1.1}};

cápsulaconvexa = ConvexHull[data2D]

{14,6,5,2,1,4,10,11,16,15}
```

Esta salida nos indica que la cápsula convexa esta conformada por la unión de los puntos que están en las posiciones 14, 6, 5, 2, 1, 4, 10, 11, 16, 15, 14, en ese orden. Note que esta salida ha sido asignada a la variable **capsulaconvexa**.

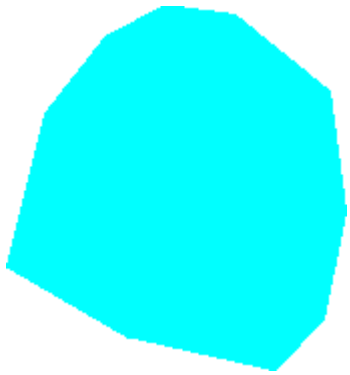
Gráfica de la cápsula convexa del conjunto de puntos data2D.

```
PlanarGraphPlot[data2D, cápsulaconvexa]
```



Otra forma de graficar la cápsula convexa del conjunto de puntos data2D.

```
Show[
  Graphics[
    {RGBColor[0,1,1],
     Polygon[Map[data2D[[#]]&,cápsulaconvexa]]}
    ],AspectRatio->Automatic
  ]
```



2.3 El algoritmo BFEA en la obtención de las expresiones analíticas de las curvas de B-splines

En la definición de las curvas de B-splines aparecen las funciones B-splines. Por esta razón el algoritmo BFEA también es de utilidad para obtener las expresiones analíticas de las curvas de B-splines.

Podemos escribir, entonces, que las expresiones analíticas de las curvas de B-splines de orden k definidas respecto a la secuencia de nodos $x = \{x_i\}$, con $x_i < x_{i+k} \quad \forall i \in \mathbf{N}$, y asociada al conjunto de puntos $P = \{p_i\}_{i=1}^{n+1} \subset \mathbf{R}^n$ pueden obtenerse mediante

$$\alpha(t) = \sum_{i=1}^{n+1} p_i N_{i,k,j}(t), \quad p_i \in \mathbf{R}^n$$

incluyendo la convención $\frac{0}{0} = 0$, donde $j \in \mathbf{Z}$ representa el j-ésimo intervalo, $[x_j, x_{j+1}\rangle$.

Lo que estamos obteniendo con esta definición alternativa son las ecuaciones matemáticas de la curva de B-splines por intervalos, es decir por trozos.

2.4 Curvas de B-splines obtenidas con el algoritmo BFEA en *Mathematica*

En el artículo anterior se da una fórmula, basada en el algoritmo BFEA, para obtener las expresiones analíticas de las curvas de B-splines por intervalos. En el presente artículo implementaremos esta fórmula para probar su efectividad.

Primero codificamos las funciones B-splines, tal como se hizo en la sección 1.4.

Definición de los comandos Convention y NN.

```
Convention[nn_, dd_] := If[dd === 0, 0, nn/dd];

NN[var_, i_, k_, j_, x_] :=
  If[k == 1,
    If[i == j, 1, 0],
    Plus[Convention[var - x[[i]], x[[i + k - 1]] - x[[i]]]*
        NN[var, i, k - 1, j, x],
      Convention[x[[i + k]] - var, x[[i + k]] - x[[i + 1]]]*
        NN[var, i + 1, k - 1, j, x]]
    ]
```

Luego codificamos las curvas de B-splines como sigue.

Definición del comando Curva.

```
Curva[var_,k_,j_,x_,pts_]:=
  Sum[
    pts[[i]]*NN[t,i,k,j,x],
    {i,Length[pts]}]
```

Una vez hecha esta codificación la utilizamos para encontrar las ecuaciones matemáticas de la curva de B-splines de tercer orden (k = 3) definida para el siguiente conjunto de puntos: (1,1), (2,3), (3,1), (5,4), (7,2) y la secuencia de nodos {0,1,2,3,4,5,6,7}; en cada uno de los intervalos que forman a partir de la secuencia de nodos dada.

Curva de B-plines de tercer orden definida para el conjunto de puntos (1,1), (2,3), (3,1), (5,4), (7,2), la secuencia de nodos {0,1,2,3,4,5,6,7}, en el intervalo $[0,1\rangle$.

```
Curva[t,3,1,{0,1,2,3,4,5,6,7},
{{1,1},{2,3},{3,1},{5,4},{7,2}}]
```

$$\left\{\frac{t^2}{2}, \frac{t^2}{2}\right\}$$

Par abreviar utilizaremos el comando **`Table`**.

Curva de B-plines de tercer orden definida para el conjunto de puntos (1,1), (2,3), (3,1), (5,4), (7,2), la secuencia de nodos {0,1,2,3,4,5,6,7}.

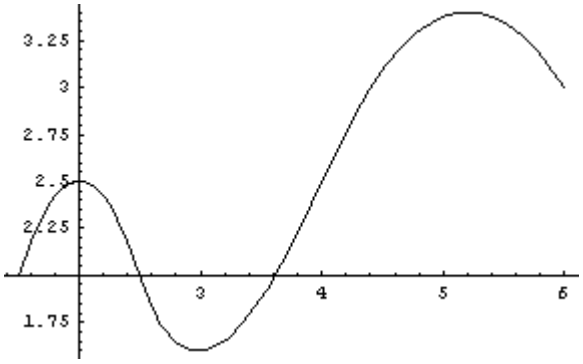
```
cc=Table[
      Curva[t,3,j,{0,1,2,3,4,5,6,7},
        {{1,1},{2,3},{3,1},{5,4},{7,2}}],{j,7}]//Simplify
```

$$\left\{\left\{\frac{t^2}{2}, \frac{t^2}{2}\right\}, \left\{-\frac{1}{2}+t, \frac{t^2}{2}\right\}, \left\{-\frac{1}{2}+t, -2\,(5-5\,t+t^2)\right\},\right.$$
$$\left\{\frac{1}{2}\,(8-4\,t+t^2), \frac{1}{2}\,(61-34\,t+5\,t^2)\right\},$$
$$\left\{2\,(-2+t), \frac{1}{2}\,(-99+46\,t-5\,t^2)\right\},$$
$$\left.\left\{\frac{1}{2}\,(-233+94\,t-9\,t^2), 13-2\,t\right\}, \left\{\frac{7}{2}\,(-7+t)^2, (-7+t)^2\right\}\right\}$$

Hemos asignado la salida en la variable **cc**, con el propósito de visualizar la gráfica de la curva, para ello seguimos el siguiente proceso.

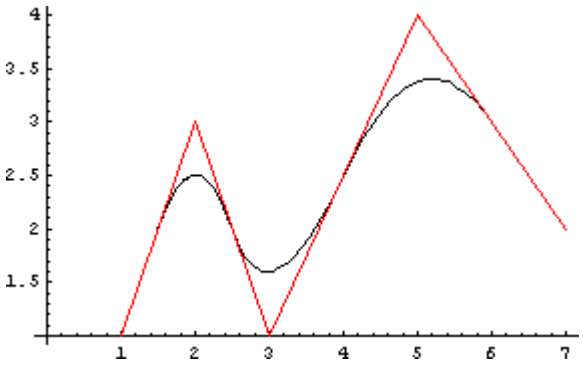
Gráfica de la curva de B-plines de tercer orden definida para el conjunto de puntos (1,1), (2,3), (3,1), (5,4), (7,2), la secuencia de nodos {0,1,2,3,4,5,6,7}.

```
graf=ParametricPlot[
  Which[t<1,cc[[1]],t<2,cc[[2]],t<3,
    cc[[3]],t<4,cc[[4]],t<5,cc[[5]],t<6,cc[[6]],t<7,cc[[7]]]
  ,{t,2,5}]
```



Gráfica de la curva de B-plines de tercer orden definida para el conjunto de puntos (1,1), (2,3), (3,1), (5,4), (7,2), la secuencia de nodos {0,1,2,3,4,5,6,7} y su polígono de control.

```
Show[graf,Graphics[{RGBColor[1,0,0],Line[{{1,1},{2,3},{3,1},
{5,4},{7,2}}]}]]
```



De la misma manera se puede obtener las expresiones analíticas de una curva de B-splines definida para un conjunto de puntos de $\mathbf{R}^3$. Esto se hace de la siguiente manera.

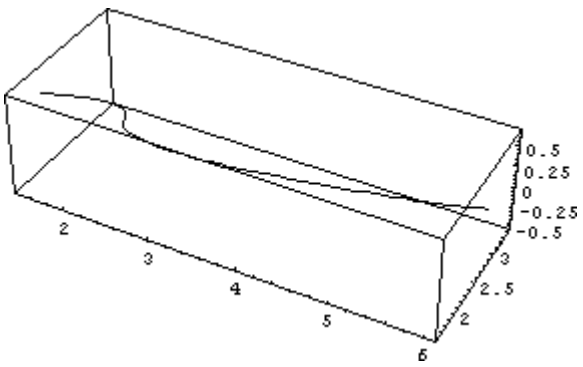
Curva de B-plines de tercer orden definida para el conjunto de puntos (1,1,1), (2,3,0), (3,1,1), (5,4,-1), (7,2,1), la secuencia de nodos {0,1,2,3,4,5,6,7}.

```
cc=Table[
      Curva[t,3,j,{0,1,2,3,4,5,6,7},
        {{1,1,1},{2,3,0},{3,1,1},{5,4,-1},{7,2,1}}],
      {j,7}]//Simplify
```

$$\left\{\left\{\frac{t^2}{2}, \frac{t^2}{2}, \frac{t^2}{2}\right\}, \left\{-\frac{1}{2}+t, \frac{t^2}{2}, -\frac{3}{2}+3t-t^2\right\},\right.$$
$$\left\{-\frac{1}{2}+t, -2\,(5-5t+t^2), \frac{13}{2}-5t+t^2\right\},$$
$$\left\{\frac{1}{2}\,(8-4t+t^2), \frac{1}{2}\,(61-34t+5t^2), -16+10t-\frac{3t^2}{2}\right\},$$
$$\left\{2\,(-2+t), \frac{1}{2}\,(-99+46t-5t^2), 2\,(20-9t+t^2)\right\},$$
$$\left\{\frac{1}{2}\,(-233+94t-9t^2), 13-2t, \frac{1}{2}\,(-95+34t-3t^2)\right\},$$
$$\left.\left\{\frac{7}{2}\,(-7+t)^2, (-7+t)^2, \frac{1}{2}\,(-7+t)^2\right\}\right\}$$

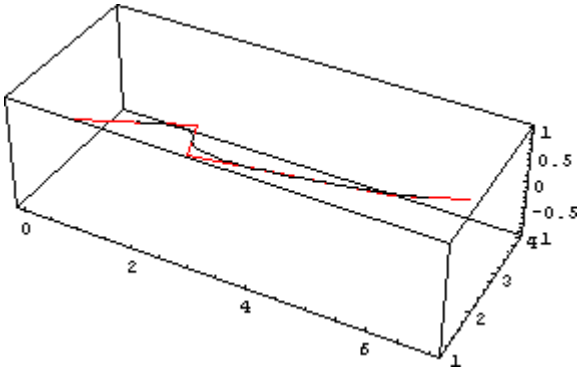
Gráfica de la curva de B-plines de tercer orden definida para el conjunto de puntos (1,1,1), (2,3,0), (3,1,1), (5,4,-1), (7,2,1), la secuencia de nodos {0,1,2,3,4,5,6,7}.

```
graf=ParametricPlot[
  Which[t<1,cc[[1]],t<2,cc[[2]],t<3,
    cc[[3]],t<4,cc[[4]],t<5,cc[[5]],t<6,cc[[6]],t<7,cc[[7]]]
  ,{t,2,5}]
```



Gráfica de la curva de B-plines de tercer orden definida para el conjunto de puntos (1,1,1), (2,3,0), (3,1,1), (5,4,-1), (7,2,1), la secuencia de nodos {0,1,2,3,4,5,6,7} y su polígono de control.

```
Show[graf,Graphics3D[{RGBColor[1,0,0],Line[{{1,1,1},{2,3,0},
{3,1,1},{5,4,-1},{7,2,1}}]}]]
```



Como se aprecia con el algoritmo BFEA hemos sido capaces de obtener las expresiones analíticas de las curvas de B-splines, para dos conjuntos de puntos. Por supuesto, que con este mismo código pueden obtenerse las expresiones analíticas de las curvas de B-splines para un conjunto de puntos de $\mathbf{R}^n$. Esto nos muestra la utilidad y efectividad del algoritmo BFEA.

Capítulo 3

El paquete `BsplinesCurves`

En este capítulo se describe el paquete **`BsplinesCurves`**, señalando los comandos en él implementados.

3.1 Comandos implementados

El paquete **`BsplinesCurves`** incorpora los siguientes comandos: **`Uniform`**, **`Open`**, **`Bspline`**, **`Curve`**, cuyo uso se describe a continuación.

Comando	*Descripción*
`Uniform[`*n*+1, *k*`]`	genera un vector de nodos uniforme para *n*+1 puntos y orden *k*
`Open[`*n*+1, *k*`]`	genera un vector de nodos abierto para *n*+1 puntos y orden *k*
`Bspline[`*k*`]`	genera la expresión analítica de una función b-spline de orden *k* para un vector de nodos uniforme
`Bspline[`*var*, *k*`]`	genera la expresión analítica de una función b-spline de orden *k* para un vector de nodos uniforme en la variable *var*
`Bspline[`*val*, *k*`]`	genera el valor de la función b-spline de orden *k* para el vector de nodos uniforme evaluada en *val*
`Curve[`*pts*`]`	genera la expresión analítica de una curva de b-splines de orden 3 para un vector de nodos abierto en la variable *t*
`Curve[`*var*, *pts*`]`	genera la expresión analítica de una curva de b-splines de orden 3 para un vector de nodos abierto en la variable *var*
`Curve[`*val*, *pts*`]`	genera el valor de una curva de b-splines de orden 3 para un vector de nodos abierto evaluada en *val*

Adicionalmente los comandos **Bspline** y **Curves** incorporan las siguientes opciones:

Opción	*Valor por defecto*	*Descripción*
Knots	Uniform	es una opción de los comandos Bspline y Curve que permite ingresar el vector de nodos.
Order	3	es una opción del comando Curve que permite ingresar el orden de la curva.
Range	True	es una opción del comando Curve que permite obtener la curva con o sin el intervalo válido.

3.2 Guía de usuario

Para utilizar el paquete **BsplinesCurves** primero debe grabarse el archivo **BsplinesCurves.m** en la carpeta ExtraPackages de WolframResearch, específicamente en la dirección que se indica a continuación C:\Archivos de programa\Wolfram Research\Mathematica\5.1\AddOns\ExtraPackages

Después de realizar el proceso anterior podemos cargar el paquete mediante

```
<<BsplinesCurves`
```

Una vez ejecutado este comando, es decir, después de presionar en simultáneo las teclas Shift-Enter, es posible obtener los resultados que se detallan a continuación.

3.2.1 Mensajes de ayuda

Cada uno de los comandos implementados posee un mensaje de ayuda que puede ser invocado digitando **?Comando**, por ejemplo

Invocando los mensajes de ayuda para los comandos Uniform y Bsplines.

```
?Uniform
Uniform[nmas1, k] is a command that returns an uniform knots
vector of order 'k', for 'nmas1' points. Remember that 1 <=
k <= nmas1.
```

```
?Bspline
Bspline[(var,) k, x] is a command which admits a variable
'var', order 'k' and knots sequence 'x' and returns the B-
splines function of order 'k' defined regarding the knots
sequence 'x'. In the case of not entering the variable is
assumed that it is 't'.
```

3.2.2 Secuencias de nodos

Para obtener secuencias de nodos uniformes y abiertas lo hacemos como se indica a continuación.

Dos secuencias de nodos: uniforme y abierta, respectivamente.

```
Uniform[6,3]
{0,1,2,3,4,5,6,7,8}

Open[6,3]
{0,0,0,1,2,3,4,4,4}
```

El paquete sólo acepta valores entero positivos para n+1 y k ($1 \leq k \leq n$+1), en caso de ingresar otro tipo de datos devuelve la misma entrada.

Casos en los que se devuelve la misma entrada.

```
Uniform[m,3]
Uniform[m,3]


Open[y,x]
Open[y,x]
```

3.3.3 Funciones B-splines

Para obtener funciones B-splines de orden k (1, 2, 3, ...) definidas con repecto a la secuencia de nodos $x = \{x_1, x_2, x_3, ...\}$ lo hacemos como se indica a continuación.

Función B-spline de orden 3 (k=3) definida con respecto a una secuencia de nodos uniforme.

```
Bspline[3]
```

$$\begin{cases} 0 & t < 0 \\ \frac{t^2}{2} & t < 1 \\ \frac{1}{2}(-3+6t-2t^2) & t < 2 \\ \frac{1}{2}(-3+t)^2 & t < 3 \end{cases}$$

Función B-spline de orden 1 (k=1) definida con respecto a la secuencia de nodos X={x[1],x[2]}.

```
Bspline[1,Knots->Array[x,2]]
```

$$\begin{cases} 0 & t < x[1] \\ 1 & t < x[2] \end{cases}$$

Función B-spline de orden 2 (k=2) definida con respecto a la secuencia de nodos X={x[4],x[5],x[6]}.

```
Bspline[2,Knots->Array[x,3,4]]
```

$$\begin{cases} 0 & t < x[4] \\ \frac{-t+x[4]}{x[4]-x[5]} & t < x[5] \\ \frac{-t+x[6]}{-x[5]+x[6]} & t < x[6] \end{cases}$$

Los resultados mostrados anteriormente son para casos generales. En estos casos no es posible obtener una expresión gráfica de las ecuaciones matemáticas de las funciones obtenidas.

Caso en que nos es posible graficar la función B-spline.

```
Bspline[1,Knots->Array[x,2]]
```

$$\begin{cases} 0 & t < x[1] \\ 1 & t < x[2] \end{cases}$$

```
Plot[%,{t,x[1],x[2]}]
Plot::plln :
 Limiting value x[1] in {t, x[1], x[2]} is not a machine-size real number. More…
Plot[%,{t,x[1],x[2]}]
```

En casos particulares, en los cuales se obtienen funciones que pueden ser graficadas por *Mathematica*, tenemos

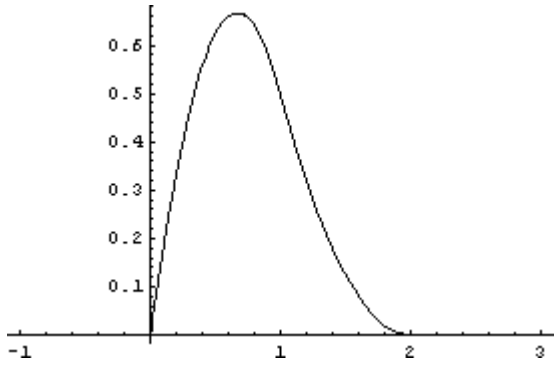
Función B-spline de orden 3 (k=3) definida con respecto a la secuencia de nodos X={0,0,1,2}.

```
f=Bspline[3,Knots->{0,0,1,2}]
```

$$\begin{cases} 0 & t < 0 \\ \frac{1}{2}\,(4t - 3t^2) & t < 1 \\ \frac{1}{2}\,(-2+t)^2 & t < 2 \end{cases}$$

Gráfica de la función f, anteriormente definida

```
Plot[f,{t,-1,3},PlotRange->All]
```



Note que hasta aquí las funciones obtenidas dependen de la variable *t*, puesto que ésta es asumida por defecto. Sin embargo, también es posible obtener funciones B-splines dependientes de cualquier variable (que se ajuste a cierta necesidad del usuario).

En el siguiente ejemplo se muestra un caso en el que la función B-spline depende de la variable θ.

Función B-spline de orden 5 (k=5) definida con respecto a la secuencia de nodos X={-2,-1,0,1,3,6} que depende de la variable θ.

```
f=Bspline[θ,5]
```

$$\begin{cases} 0 & \theta < -2 \\ \frac{1}{30}\,(2+\theta)^4 & \theta < -1 \\ \frac{1}{210}\,(82 + 104\theta - 12\theta^2 - 64\theta^3 - 23\theta^4) & \theta < 0 \\ \frac{1}{630}\,(246 + 312\theta - 36\theta^2 - 192\theta^3 + 71\theta^4) & \theta < 1 \\ \frac{1}{630}\,(162 + 648\theta - 540\theta^2 + 144\theta^3 - 13\theta^4) & \theta < 3 \\ \frac{1}{630}\,(-6+\theta)^4 & \theta < 6 \end{cases}$$

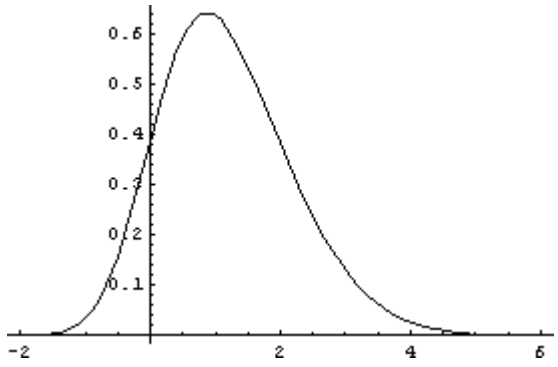
Función B-spline de orden 5 (k=5) definida con respecto a la secuencia de nodos X={-2,-1,0,1,3,6} que depende de la variable θ.

```
f=Bspline[θ,5,Knots->{-2,-1,0,1,3,6}]
```

$$\begin{cases} 0 & \theta < -2 \\ \frac{1}{30}(2+\theta)^4 & \theta < -1 \\ \frac{1}{210}(82+104\theta-12\theta^2-64\theta^3-23\theta^4) & \theta < 0 \\ \frac{1}{630}(246+312\theta-36\theta^2-192\theta^3+71\theta^4) & \theta < 1 \\ \frac{1}{630}(162+648\theta-540\theta^2+144\theta^3-13\theta^4) & \theta < 3 \\ \frac{1}{630}(-6+\theta)^4 & \theta < 6 \end{cases}$$

Gráfica de la función f, anteriormente definida.

```
Plot[f,{θ,-2,6},PlotRange->All]
```



También es posible obtener el valor de la función en cualquier número real del dominio.

Función B-spline de orden 7 (k=7) definida con respecto a la secuencia de nodos X={0,0,1,2,6,7,8,9}; evaluada en –10, 0.5 y $\sqrt{3}$, respectivamente.

```
Bspline[-10,7,Knots->{0,0,1,2,6,7,8,9}]
```

0

```
Bspline[0.5,7,Knots->{0,0,1,2,6,7,8,9}]
```

0.000231454

```
Bspline[Sqrt[3],7,Knots->{0,0,1,2,6,7,8,9}]
```

$$\frac{64733 - 35280\sqrt{3}}{62720}$$

Utilizaremos, a continuación, el comando **`Partition`** (ver [2]) como una forma práctica de obtener las funciones B-splines de orden 3 ($k = 3$) definidas con respecto a la secuencia de nodos {0,1,1,3,4,6,6,6}.

Aplicando Partition a la secuencia de nodos X={0,1,1,3,4,6,6,6}.

```
k=3;
kv=Partition[{0,1,1,3,4,6,6,6},{k+1},{1}]
n=Length[kv];
{{0,1,1,3},{1,1,3,4},{1,3,4,6},{3,4,6,6},{4,6,6,6}}
```

Ahora hacemos uso del comando **`Map`**, que también puede escribirse **`/@`** (ver [2]), para obtener funciones B-splines requeridas.

Funciones B-splines de orden 3 (k = 3) definidas con respecto a la secuencia de nodos {0,1,1,3,4,6,6,6}.

```
ff=Bspline[k,Knots->#]&/@kv
```

$$\left\{\begin{cases} 0 & t < 0 \\ t^2 & t < 1 \\ \frac{1}{4}(-3+t)^2 & t < 3 \end{cases},\right.$$

$$\begin{cases} 0 & t < 1 \\ \frac{1}{12}(-17+22\,t-5\,t^2) & t < 3 \\ \frac{1}{3}(-4+t)^2 & t < 4 \end{cases},$$

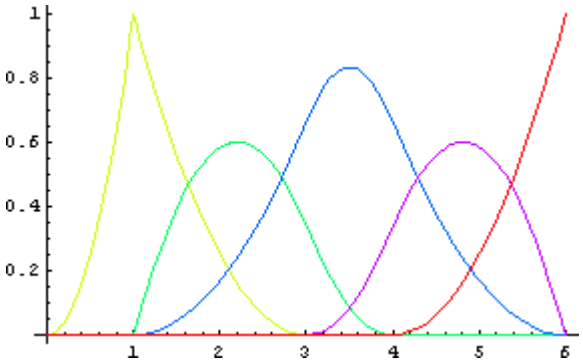
$$\begin{cases} 0 & t < 1 \\ \frac{1}{6}(-1+t)^2 & t < 3 \\ -\frac{2}{3}(11-7\,t+t^2) & t < 4 \\ \frac{1}{6}(-6+t)^2 & t < 6 \end{cases},$$

$$\begin{cases} 0 & t < 3 \\ \frac{1}{3}(-3+t)^2 & t < 4 \\ \frac{1}{12}(-108+48\,t-5\,t^2) & t < 6 \end{cases},$$

$$\left.\begin{cases} 0 & t < 4 \\ \frac{1}{4}(-4+t)^2 & t < 6 \end{cases}\right\}$$

Gráfica de las funciones B-splines de orden 3 definidas con respecto a la secuencia de nodos {0,1,1,3,4,6,6,6}.

```
Plot[
  ff//Evaluate,{t,0,6},PlotStyle->Table[Hue[i/n],{i,n}]
  ]
```



Cabe mencionar que un caso particular de las fucniones B-splines son las llamadas funciones de Bézier (ver [1]), las cuales se obtienen cuando el orden (k) es igual al número de puntos (n+1). El siguiente ejemplo nos presenta un caso particular en el que se obtienen funciones de Bézier. En este caso el orden elegido es 6 y la secuencia de nodos, es {0,0,0,0,0,0,1,1,1,1,1,1}.

Funciones B-splines de orden 6 (k = 6) definidas con respecto a la secuencia de nodos {0,0,0,0,0,0,1,1,1,1,1,1}.

```
k=6;
kv=Partition[Open[k,k],{k+1},{1}]
n=Length[kv];
{{0,0,0,0,0,0,1},{0,0,0,0,0,1,1},{0,0,0,0,1,1,1},
{0,0,0,1,1,1,1},{0,0,1,1,1,1,1},{0,1,1,1,1,1,1}}
ff=Bspline[k,Knots->#]&/@kv
```

$$\left\{\left\{\begin{cases} 0 & t<0 \\ -(-1+t)^5 & t<1 \end{cases}, \begin{cases} 0 & t<0 \\ 5(-1+t)^4 t & t<1 \end{cases}\right.\right.$$

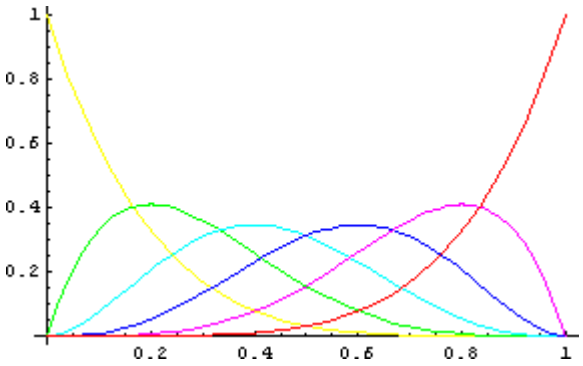
$$\begin{cases} 0 & t<0 \\ -10(-1+t)^3 t^2 & t<1 \end{cases}$$

$$\begin{cases} 0 & t<0 \\ 10(-1+t)^2 t^3 & t<1 \end{cases}$$

$$\left.\left.\begin{cases} 0 & t<0 \\ -5(-1+t) t^4 & t<1 \end{cases}, \begin{cases} 0 & t<0 \\ t^5 & t<1 \end{cases}\right\}\right\}$$

Gráfica de las funciones B-splines de orden 6 definidas con respecto a la secuencia de nodos {0,0,0,0,0,0,1,1,1,1,1,1}.

```
Plot[
  ff//Evaluate,{t,0,1},PlotStyle->Table[Hue[i/n],{i,n}]
  ]
```



3.3.4 Curvas de B-splines

Para obtener las ecuaciones matemáticas de las curvas de B-splines, como puede leerse en el artículo 2, se ha implementado el comando **`Curve`**. Con este comando podemos obtener curvas definidas de **R** en **R**, de **R** en $\mathbf{R}^2$, de **R** en $\mathbf{R}^3$ y en general de **R** en $\mathbf{R}^n$. Por supuesto que las gráficas sólo pueden obtenerse para los tres primeros casos.

En los ejemplos dados a continuación se presentan algunos de estos casos.

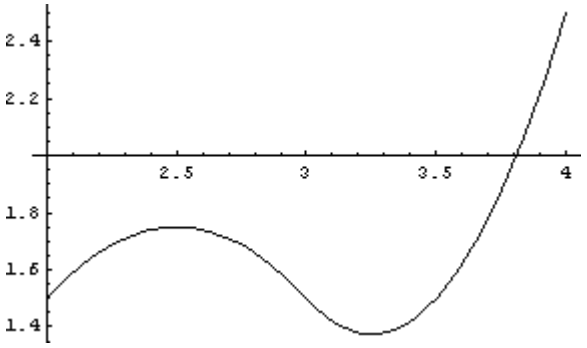
Curva de B-splines de orden 3 definida para un vector de nodos abierto y los puntos de control (1),(2),(1),(4).

```
pts={1,2,1,4};
c=Curve[pts]
```

$$\text{Sequence}\left[\begin{cases} \{0\} & t < 0 \\ \left\{\frac{t^2}{2}\right\} & t < 1 \\ \left\{\frac{1}{2}(-1+2t)\right\} & t < 2 \\ \left\{\frac{1}{2}(-9+10t-2t^2)\right\} & t < 3 \\ \left\{\frac{1}{2}(45-26t+4t^2)\right\} & t < 4 \\ \left\{\frac{1}{2}(-131+62t-7t^2)\right\} & t < 5 \\ \{2(-6+t)^2\} & t < 6 \\ \{0\} & t \geq 6 \end{cases}, \{t, 2, 4.\}\right]$$

Gráfica de la Curva de B-splines definida anteriormente.

```
Plot[c//Evaluate]
```



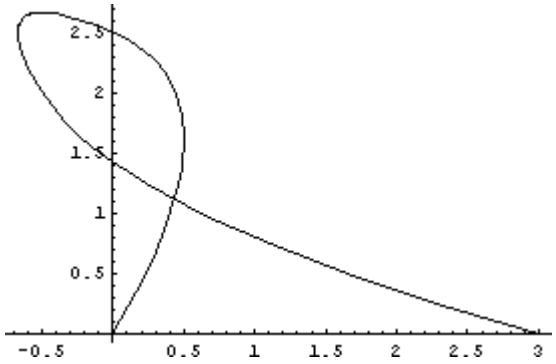
Curva de B-splines de orden 3 definida para un vector de nodos abierto y los puntos de control (0,0),(1,2), (-1,3),(0,1),(3,0).

```
pts = {{0,0},{1,2},{-1,3},{0,1},{3,0}};
c=Curve[pts]
```

$$\text{Sequence}\left[\begin{cases} \{0, 0\} & t < 0 \,||\, t < 1 \\ \left\{\frac{1}{2}(-1+t)^2, (-1+t)^2\right\} & t < 2 \\ \left\{\frac{1}{2}(-15+14t-3t^2), \frac{1}{2}(-10+8t-t^2)\right\} & t < 3 \\ \left\{\frac{1}{2}(39-22t+3t^2), \frac{1}{2}(-28+20t-3t^2)\right\} & t < 4 \\ \left\{\frac{1}{2}(23-14t+2t^2), \frac{1}{2}(36-12t+t^2)\right\} & t < 5 \\ \left\{-\frac{3}{2}(59-22t+2t^2), \frac{1}{2}(-6+t)^2\right\} & t < 6 \\ \left\{\frac{3}{2}(-7+t)^2, 0\right\} & t < 7 \\ \{0, 0\} & t \geq 7 \end{cases}, \{t, 2, 5.\}\right]$$

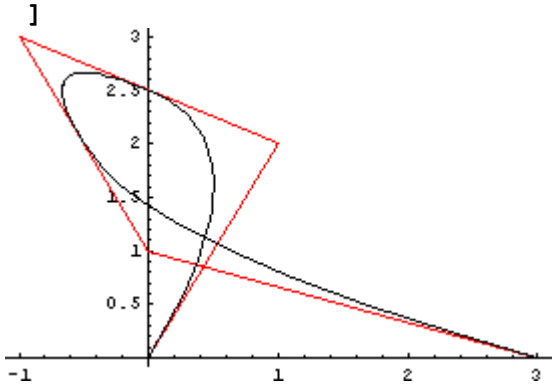
Gráfica de la Curva de B-splines definida anteriormente.

```
cc=ParametricPlot[c//Evaluate]
```



Curva de B-splines de orden 3 definida para un vector de nodos abierto y los puntos de control (0,0),(1,2), (-1,3),(0,1),(3,0) y la poligonal que une estos puntos.

```
Show[
  Graphics[{RGBColor[1, 0, 0],Line[pts]}],
  cc,PlotRange->All,Axes->True
  ]
```



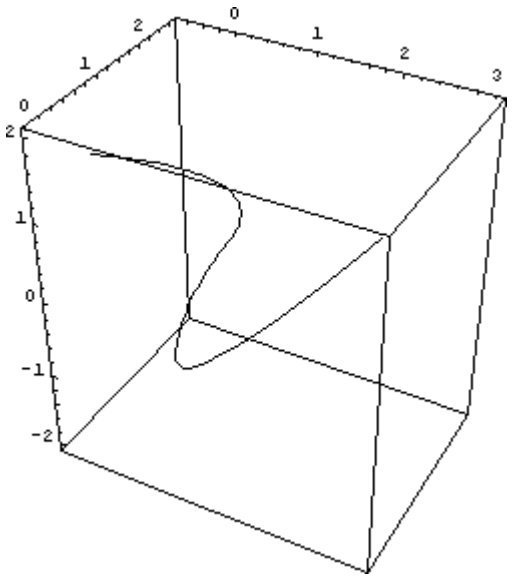
Curva de B-splines de orden 3 definida para un vector de nodos abierto y los puntos de control (0,0,2),(1,2,1), (-1,3,-2),(0,1,-2),(3,0,2).

```
pts={{0,0,2},{1,2,1},{-1,3,-2},{0,1,-2},{3,0,2}};
c=Curve[pts]
```

```
Sequence[

 {{0, 0, 0}
 {0, 0, t^2}
 {1/2 (-1 + t)^2, (-1 + t)^2, 1/2 (-5 + 10 t - 3 t^2)}
 {1/2 (-15 + 14 t - 3 t^2), 1/2 (-10 + 8 t - t^2), 1/2 (-1 + 6 t - 2 t^2)}
 {1/2 (39 - 22 t + 3 t^2), 1/2 (-28 + 20 t - 3 t^2), 1/2 (44 - 24 t + 3 t^2)}
 {1/2 (23 - 14 t + 2 t^2), 1/2 (36 - 12 t + t^2), 2 (15 - 8 t + t^2)}
 {-3/2 (59 - 22 t + 2 t^2), 1/2 (-6 + t)^2, -95 + 34 t - 3 t^2}
 {3/2 (-7 + t)^2, 0, (-7 + t)^2}
 {0, 0, 0}

 , {t, 2, 5.}]
```

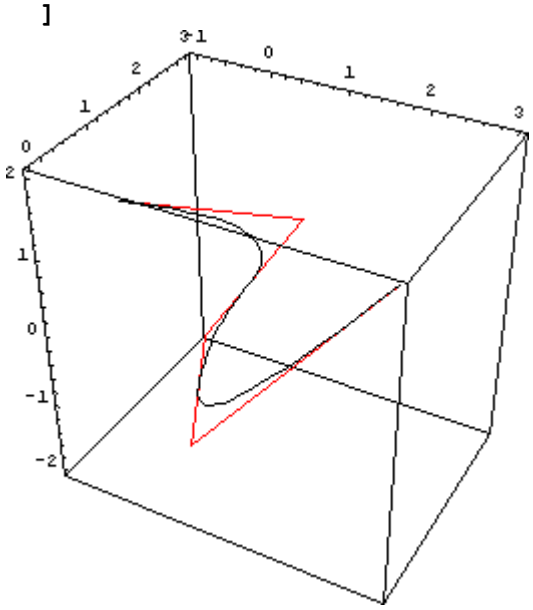
Gráfica de la Curva de B-splines definida anteriormente.

```
cc=ParametricPlot3D[c//Evaluate]
```



Curva de B-splines de orden 3 definida para un vector de nodos abierto y los puntos de control (0,0,2),(1,2,1), (-1,3,-2),(0,1,-2),(3,0,2) y la poligonal que une estos puntos.

```
Show[
  Graphics3D[{RGBColor[1, 0, 0],Line[pts]}],
  cc, Axes->True
  ]
```



Curva de B-splines de orden 3 definida para un vector de nodos abierto y los puntos de control (0,0,2,-1),(1,2,1,0),(-1,3,-2,4),(0,1,-2,3),(3,0,2,1).

```
pts={{0,0,2,-1},{1,2,1,0},{-1,3,-2,4},{0,1,-2,3},{3,0,2,1}};
c=Curve[pts]
```

$$\text{Sequence}\left[\begin{cases} \{0, 0, 0, 0\} & t < 0 \\ \left\{0, 0, t^2, -\frac{t^2}{2}\right\} & t < 1 \\ \left\{\frac{1}{2}(-1+t)^2, (-1+t)^2, \frac{1}{2}(-5+10t-3t^2), \frac{1}{2}(3-6t+2t^2)\right\} & t < 2 \\ \left\{\frac{1}{2}(-15+14t-3t^2), \frac{1}{2}(-10+8t-t^2), \frac{1}{2}(7-2t), \frac{1}{2}(7-10t+3t^2)\right\} & t < 3 \\ \left\{\frac{1}{2}(39-22t+3t^2), \frac{1}{2}(-28+20t-3t^2), \frac{1}{2}(-4+t)^2, \frac{1}{2}(-65+38t-5t^2)\right\} & t < 4, \\ \left\{\frac{1}{2}(23-14t+2t^2), \frac{1}{2}(36-12t+t^2), 0, \frac{1}{2}(-1+6t-t^2)\right\} & t < 5 \\ \left\{-\frac{3}{2}(59-22t+2t^2), \frac{1}{2}(-6+t)^2, 0, \frac{1}{2}(49-14t+t^2)\right\} & t < 6 \\ \left\{\frac{3}{2}(-7+t)^2, 0, 0, \frac{1}{2}(-7+t)^2\right\} & t < 7 \\ \{0, 0, 0, 0\} & t \geq 7 \end{cases}, \{t, 2, 5.\}\right]$$

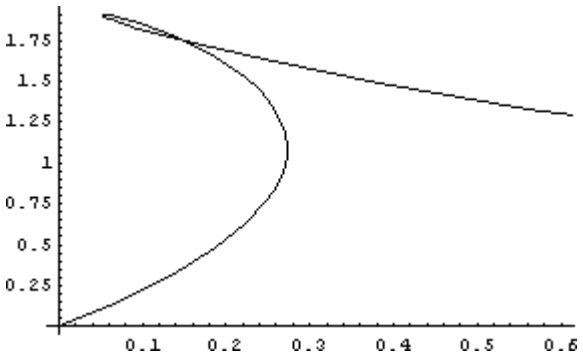
Para finalizar veamos algunos ejemplos en los que se cambian los valores por defecto de las opciones implementadas en el comando Curve.

Curva de B-splines de orden 5 definida para un vector de nodos abierto y los puntos de control (0,0),(1,2), (-1,3),(0,1),(3,0).

```
pts = {{0,0},{1,2},{-1,3},{0,1},{3,0}};
c=Curve[pts,Order->5]
```

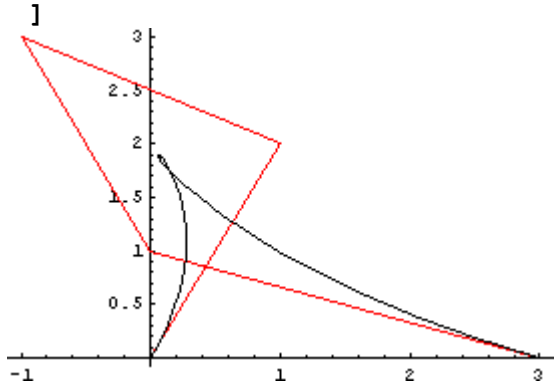
$$\text{Sequence}\left[\begin{cases} \{0, 0\} & t < 0 \\ \{4t - 18t^2 + 24t^3 - 7t^4, 2(4t - 3t^2 - 4t^3 + 3t^4)\} & t < 1, \{t, 0, 1.\} \\ \{0, 0\} & t \geq 1 \end{cases}\right]$$

```
cc=ParametricPlot[c//Evaluate]
```



Curva de B-splines de orden 5 definida para un vector de nodos abierto y los puntos de control (0,0),(1,2), (-1,3),(0,1),(3,0) y la poligonal que une estos puntos.

```
Show[
  Graphics[{RGBColor[1, 0, 0],Line[pts]}],
  cc,PlotRange->All,Axes->True
  ]
```



Curva de B-splines de orden 4 definida para el vector de nodos {-1,-1,2,3.5,4,6,7,7,7} y los puntos de control (0,0),(1,2), (-1,3),(0,1),(3,0).

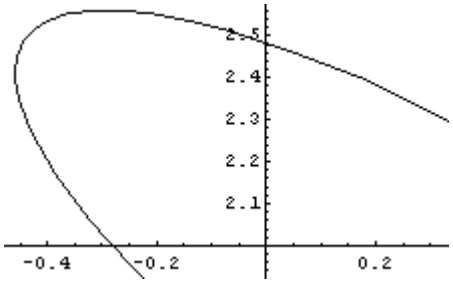
```
pts = {{0,0},{1,2},{-1,3},{0,1},{3,0}};
c=Curve[pts,Order->4,Knots->{-1,-1,2,7/2,4,6,7,7,7}]
```

Sequence[

$$\begin{cases} \{0, 0\} & t < -1 \\ \left\{\frac{2}{135}(1+t)^3, \frac{4}{135}(1+t)^3\right\} & t < 2 \\ \left\{\frac{1}{270}(604 - 888\,t + 462\,t^2 - 71\,t^3), \frac{1}{540}(616 - 852\,t + 498\,t^2 - 59\,t^3)\right\} & t < \frac{7}{2} \\ \left\{\frac{3}{70}(-1540 + 1288\,t - 350\,t^2 + 31\,t^3), \frac{1}{140}(2548 - 2268\,t + 714\,t^2 - 71\,t^3)\right\} & t < 4 \\ \left\{\frac{1}{105}(2254 - 1092\,t + 147\,t^2 - 4\,t^3), \frac{-34412+22596\,t-4326\,t^2+257\,t^3}{1260}\right\} & t < 6 \\ \left\{\frac{1}{21}(19502 - 9744\,t + 1617\,t^2 - 89\,t^3), \frac{1}{63}(-2401 + 1470\,t - 273\,t^2 + 16\,t^3)\right\} & t < 7 \\ \{0, 0\} & t \geq 7 \end{cases},$$

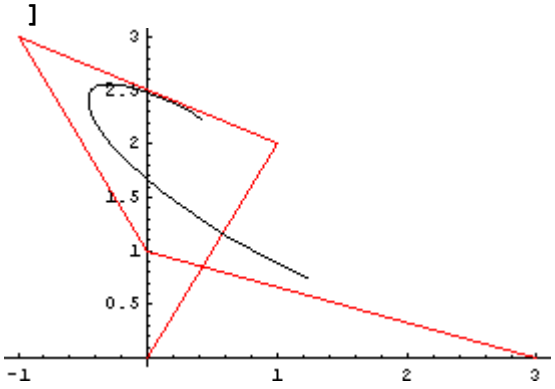
$$\left\{t, \frac{7}{2}, 6.\right\}\Big]$$

```
cc=ParametricPlot[c//Evaluate]
```



Curva de B-splines de orden 4 definida para el vector de nodos {-1,-1,2,3.5,4,6,7,7,7} y los puntos de control (0,0),(1,2), (-1,3),(0,1),(3,0) y la poligonal que une estos puntos.

```
Show[
  Graphics[{RGBColor[1, 0, 0],Line[pts]}],
  cc,PlotRange->All,Axes->True
  ]
```



B-SPLINES CON MATHEMATICA 5.1

ISBN: 84-689-3603-0

Indice

Introducción

La construcción de curvas de forma libre en un ambiente industrial se remonta a la época Romana, con la construcción naval. Las costillas de una nave producidas con tablones de madera que emanan de la quilla debían producirse con plantillas que pudieran usarse muchas veces. Esto llevó al empleo de técnicas que fueron perfeccionadas por los venecianos (siglos XIII al XVI). La forma de las costillas se definió en términos de arcos circulares tangentes continuos. El casco de la nave se obtuvo variando la forma de las costillas a lo largo de la quilla (una manifestación temprana de las superficies definidas mediante producto tensorial en la actualidad). Hasta esta época no existió ningún dibujo para definir el casco de una nave, estos se hicieron populares en Inglaterra allá por el año 1600. Y probablemente en aquella época se inventó el clásico "spline" definido como una tira flexible de madera o caucho usada para dibujar curvas lisas.

En general, las curvas fueron empleadas para diseñar durante siglos; la mayoría de estas curvas eran círculos, pero algunas fueron de "forma libre". El empleo de estas curvas parte desde el diseño de naves hasta llegar a la arquitectura. Cuando las curvas tuvieron que ser dibujadas exactamente, la herramienta comúnmente usada fue un juego de plantillas conocidas como "curvas francesas" que consisten en porciones de cónicas y espirales. Otra herramienta mecánica utilizada fue el llamado *spline*, tira de madera a la que se le daba cierta forma y para mantener esa forma se utilizaban pesas de metal conocidas como *ducks*. La contraparte matemática de un spline mecánico es una *curva spline* definida en forma paramétrica y que al igual que un spline mecánico se utilizó para diseñar.

Posteriormente, en los años cincuenta, dos matemáticos franceses (Paul de Faget de Casteljau y Pierre Bézier) trabajando en forma independiente llegaron a resultados similares descubriendo así las hoy conocidas curvas de Bézier. El descubrimiento de estas curvas fue de tal trascendencia que su uso en el diseño se adoptó a nivel mundial.

Posteriormente se mejoran los resultados obtenidos con las curvas de Bézier al descubrirse su generalización: las curvas de B-splines (nombre corto para Basis Splines).

Alrededor de los años sesenta, C. de Boor empezó a trabajar para los laboratorios de investigación de la General Motors usando en este trabajo los B-splines para efectuar representaciones geométricas. Más tarde se vuelve uno de los más arduos propulsores de los B-splines en la teoría de aproximación. La evaluación recursiva de las curvas B-splines se debe a él y en la actualidad se conoce como el algoritmo de de Boor. Esta evaluación recursiva fue descubierta en forma independiente por de Boor, L. Mansfield y M. Cox. Gracias a esta evaluación recursiva los B-splines se convierten en una herramienta viable en CAGD, ya que antes de su descubrimiento los B-splines se definieron usando un tedioso método de diferencias divididas que era muy inestable.

La programación de los B-splines en lenguajes tradicionales como Fortran, Pascal, C, etc. mediante la evaluación recursiva propuesta por de Boor está sumamente difundida. Sin embargo, a partir de la década de los ochenta con el nacimiento de los sistemas de cálculo simbólico (*Mathematica*, *Maple,* etc.) es posible avanzar un poco más y obtener las expresiones analíticas de los B-splines que resultarían útiles, en primera instancia, para fines académicos en cursos de Pre-Grado y Post-Grado[1]. Con la finalidad de aprovechar estas nuevas herramientas, específicamente *Mathematica*, es que hemos elaborado este texto en el que se plantea un algoritmo nuevo basado en el de de Boor que al codificarse en *Mathematica* 5.1 nos permite construir el paquete `BsplinesCurves` con el cual generamos las expresiones analíticas de los B-splines.

[1] *Maple* ya incluye un paquete (`CurveFitting`) en el que se han implementado comandos para obtener estas expresiones