Symbolic Manipulation of Bspline Basis Functions with *Mathematica*

A. Iglesias[1], R. Ipanaqué[2], and R.T. Urbina[2]

[1] Department of Applied Mathematics and Computational Sciences, University of Cantabria, Avda. de los Castros, s/n, E-39005, Santander, Spain
iglesias@unican.es

[2] Department of Mathematics, National University of Piura, Urb. Miraflores s/n, Castilla, Piura, Perú

Abstract. Bspline curves and surfaces are the most common and most important geometric entities in many fields, such as computer design and manufacturing (CAD/CAM) and computer graphics. However, up to our knowledge no computer algebra package includes especialized symbolic routines for dealing with Bsplines so far. In this paper, we describe a new *Mathematica* program to compute the Bspline basis functions symbolically. The performance of the code along with the description of the main commands are discussed by using some illustrative examples.

1 Introduction

Bspline curves and surfaces are the most common and most important geometric entities in many fields, such as computer design and manufacturing (CAD/CAM) and computer graphics. In fact, they become the standard for computer representation, design and data exchange of geometric information in the automotive, aerospace and ship-building industries [1]. In addition, they are very intuitive, easy to modify and manipulate - thus allowing the designers to modify the shape interactively. Moreover, the algorithms involved are quite fast and numerically stable and, therefore, well suited for real-time applications in a variety of fields, such as CAD/CAM [1,7], computer graphics and animation, geometric processing [5], artificial intelligence [2,3] and many others.

Although there is a wealth of powerful algorithms for Bsplines (see, for instance, [6]), they usually perform in a numerical way. Surpringly, although there is a large collection of very powerful general-purpose computer algebra systems, none of them includes specific commands or specialized routines for dealing with Bsplines symbolically. The present work is aimed at bridging this gap. This paper describes a new *Mathematica* program for computing Bspline basis functions in a fully symbolic way. Because these basis functions are at the core of almost any algorithm for Bspline curves and surfaces, their efficient manipulation is a critical step we have accomplished in this paper. The program is also able to deal with Bspline curves and surfaces. However, this paper focuses on the computation of Bspline basis functions because of limitations of space. The program has been

Y. Shi et al. (Eds.): ICCS 2007, Part II, LNCS 4488, pp. 194–202, 2007.

implemented in *Mathematica* v4.2 [8] although later releases are also supported. The program provides the user with a highly intuitive, mathematical-looking output consistent with *Mathematica*'s notation and syntax [4].

The structure of this paper is as follows: Section 2 provides some mathematical background on Bspline basis functions. Then, Section 3 introduces the new *Mathematica* program for computing them and describes the main commands implemented within. The performance of the code is also discussed in this section by using some illustrative examples.

2 Mathematical Preliminaries

Let $\mathcal{T} = \{u_0, u_1, u_2, \ldots, u_{r-1}, u_r\}$ be a nondecreasing sequence of real numbers called *knots*. $\mathcal{T}$ is called the *knot vector*. The *ith Bspline basis function* $N_{i,k}(t)$ *of order* k (or equivalently, degree $k-1$) is defined by the recurrence relations

$$N_{i,1}(t) = \begin{cases} 1 & \text{if } u_i \le t < u_{i+1} \\ 0 & \text{otherwise} \end{cases}, \qquad i = 0, 1, 2, \ldots, r-1 \tag{1}$$

and

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

for $k > 1$. Note that i-th Bspline basis function of order 1, $N_{i,1}(t)$, is a piecewise constant function with value 1 on the interval $[u_i, u_{i+1})$, called the *support* of $N_{i,1}(t)$, and zero elsewhere. This support can be either an interval or reduce to a point, as knots u_i and u_{i+1} must not necessarily be different. If necessary, the convention $\frac{0}{0} = 0$ in eq. (2) is applied. The number of times a knot appears in the knot vector is called the *multiplicity* of the knot and has an important effect on the shape and properties of the associated basis functions. Any basis function of order $k > 1$, $N_{i,k}(t)$, is a linear combination of two consecutive functions of order $k-1$, where the coefficients are linear polinomials in t, such that its order (and hence its degree) increases by 1. Simultaneously, its support is the union of the (partially overlapping) supports of the former basis functions of order $k-1$ and, consequently, it usually enlarges.

3 Symbolic Computation of Bspline Basis Functions

This section describes the *Mathematica* program we developed to compute the Bspline basis functions in a fully symbolic way. For the sake of clarity, the program will be explained through some illustrative examples.

The main command, $\mathtt{N}_{i,k}$`[knots,var]`, returns the i-th Bspline basis function of order k in the variable `var` associated with an arbitrary knot vector `knots`, as defined by eqs. (1)-(2). For instance, eq. (1) can be obtained as:

`In[1]:=`$\mathtt{N}_{0,1}$`[{`$\mathtt{u}_i$`,`$\mathtt{u}_{i+1}$`},t]`

$Out[1] := Which[t < u_i, 0, u_i \le t < u_{i+1}, 1, t \ge u_{i+1}, 0]$

where the output consists of several couples (*condition,value*) that reproduce the structure of the right-hand side of eq. (1). The command `Which` evaluates those conditions and returns the value associated with the first condition yielding `True`. Our command `PiecewiseForm` displays the same output with a more similar appearance to eq. (1):

`In[2]:=PiecewiseForm[%]`

$$Out[2] := \left\{\begin{array}{ll} 0 & t < u_i \\ 1 & u_i \leq t < u_{i+1} \\ 0 & t \geq u_{i+1} \end{array}\right\}$$

This output shows the good performance of these commands to handle fully symbolic input. Let us now consider a symbolic knot vector of length 4 such as:

`In[3]:=Array[x,4]`

$Out[3] := \{x(1), x(2), x(3), x(4)\}$

Now, we compute the basis functions up to order 3 for this knot vector as follows:

`In[4]:Table[Table[`$N_{i,k}$`[%,t] // PiecewiseForm,{i,0,3-k}],{k,1,3}]`

$Out[4] :=$

$$\left\{\begin{array}{l}
\left\{\left\{\begin{array}{ll} 0 & t < x(1) \\ 1 & x(1) \leq t < x(2) \\ 0 & x(2) \leq t < x(3) \\ 0 & x(3) \leq t < x(4) \\ 0 & t \geq x(4) \end{array}\right\}, \left\{\begin{array}{ll} 0 & t < x(1) \\ 0 & x(1) \leq t < x(2) \\ 1 & x(2) \leq t < x(3) \\ 0 & x(3) \leq t < x(4) \\ 0 & t \geq x(4) \end{array}\right\}, \left\{\begin{array}{ll} 0 & t < x(1) \\ 0 & x(1) \leq t < x(2) \\ 0 & x(2) \leq t < x(3) \\ 1 & x(3) \leq t < x(4) \\ 0 & t \geq x(4) \end{array}\right\}\right\}, \\
\left\{\left\{\begin{array}{cc} 0 & t < x(1) \\ \frac{t-x(1)}{-x(1)+x(2)} & x(1) \leq t < x(2) \\ \frac{t-x(3)}{x(2)-x(3)} & x(2) \leq t < x(3) \\ 0 & x(3) \leq t < x(4) \\ 0 & t \geq x(4) \end{array}\right\}, \left\{\begin{array}{cc} 0 & t < x(1) \\ 0 & x(1) \leq t < x(2) \\ \frac{t-x(2)}{-x(2)+x(3)} & x(2) \leq t < x(3) \\ \frac{t-x(4)}{x(3)-x(4)} & x(3) \leq t < x(4) \\ 0 & t \geq x(4) \end{array}\right\}\right\}, \\
\left\{\left\{\begin{array}{cc} 0 & t < x(1) \\ \frac{(t-x(1))^2}{(x(1)-x(2))\,(x(1)-x(3))} & x(1) \leq t < x(2) \\ \frac{(t-x(1))\,(t-x(3))}{(x(1)-x(3))\,(-x(2)+x(3))} + \frac{(t-x(2))\,(-t+x(4))}{(x(2)-x(3))\,(x(2)-x(4))} & x(2) \leq t < x(3) \\ \frac{(t-x(4))^2}{(-x(2)+x(4))\,(-x(3)+x(4))} & x(3) \leq t < x(4) \\ 0 & t \geq x(4) \end{array}\right\}\right\}
\end{array}\right\}$$

Note that, according to eq. (2), the i-th basis function of order k is obtained from the i-th and $(i+1)$-th basis functions of order $k-1$. This means that the number of basis functions decreases as the order increases and conversely. Therefore, for the set of basis functions up to order 3 we compute the $N_{i,k}$, with $i = 0, \ldots, 3-k$ for $k = 1, 2, 3$. The whole set exhibits a triangular structure of embedded lists in *Out[4]* for each hierarchical level (i.e. for each order value).

The knot vectors can be classified into three groups. The first one is the *uniform knot vector*; in it, each knot appears only once and the distance between

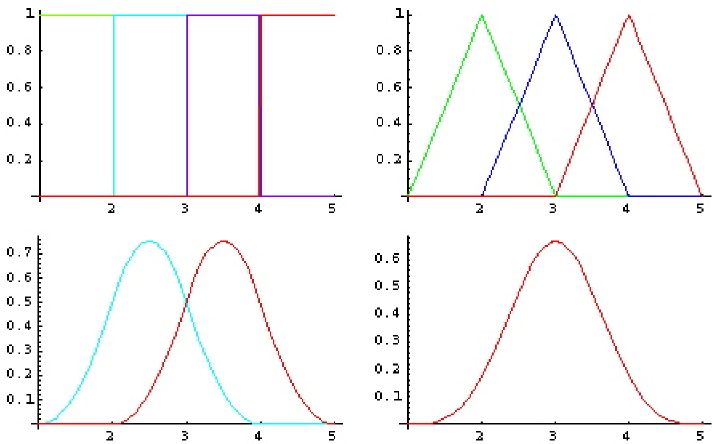

Fig. 1. (top-bottom, left-right) Bspline basis functions for the uniform knot vector $\{1, 2, 3, 4, 5\}$ and orders 1, 2, 3 and 4 respectively

consecutive knots is always the same. As a consequence, each basis function is similar to the previous one but shifted to the right according to such a distance. To illustrate this idea, let us proceed with a numerical knot vector so that the corresponding basis functions can be displayed graphically. We compute the basis functions of order 1 for the uniform knot vector $\{1, 2, 3, 4, 5\}$:

```
In[5]:=Table[N_{i,1}[{1,2,3,4,5},t] //PiecewiseForm,{i,0,3}]
```

$Out[5] :=$

$$\left\{\left\{\begin{array}{ll} 0 & t<1 \\ 1 & 1\le t<2 \\ 0 & 2\le t<3 \\ 0 & 3\le t<4 \\ 0 & 4\le t<5 \\ 0 & t\ge 5 \end{array}\right\}, \left\{\begin{array}{ll} 0 & t<1 \\ 0 & 1\le t<2 \\ 1 & 2\le t<3 \\ 0 & 3\le t<4 \\ 0 & 4\le t<5 \\ 0 & t\ge 5 \end{array}\right\}, \left\{\begin{array}{ll} 0 & t<1 \\ 0 & 1\le t<2 \\ 0 & 2\le t<3 \\ 1 & 3\le t<4 \\ 0 & 4\le t<5 \\ 0 & t\ge 5 \end{array}\right\}, \left\{\begin{array}{ll} 0 & t<1 \\ 0 & 1\le t<2 \\ 0 & 2\le t<3 \\ 0 & 3\le t<4 \\ 1 & 4\le t<5 \\ 0 & t\ge 5 \end{array}\right\}\right\}$$

From (2) we can see that the basis functions of order 2 are linear combinations of these step functions of order 1 (shown in Figure 1(top-left)). The coefficients of such a linear combination are linear polynomials as well, so the resulting basis functions are actually piecewise linear functions (see Fig. 1(top-right)):

```
In[6]:=Table[N_{i,2}[{1,2,3,4,5},t] //PiecewiseForm,{i,0,2}]
```

$$Out[6] := \left\{\left\{\begin{array}{cc} 0 & t<1 \\ -1+t & 1\le t<2 \\ 3-t & 2\le t<3 \\ 0 & 3\le t<4 \\ 0 & 4\le t<5 \\ 0 & t\ge 5 \end{array}\right\}, \left\{\begin{array}{cc} 0 & t<1 \\ 0 & 1\le t<2 \\ -2+t & 2\le t<3 \\ 4-t & 3\le t<4 \\ 0 & 4\le t<5 \\ 0 & t\ge 5 \end{array}\right\}, \left\{\begin{array}{cc} 0 & t<1 \\ 0 & 1\le t<2 \\ 0 & 2\le t<3 \\ -3+t & 3\le t<4 \\ 5-t & 4\le t<5 \\ 0 & t\ge 5 \end{array}\right\}\right\}$$

Similarly, the basis functions of order 3 are linear combinations of the basis functions of order 2 in *Out[6]* according to (2):

```
In[7]:=Table[N_{i,3}[{1,2,3,4,5},t] //PiecewiseForm,{i,0,1}]
```

$$Out[7] := \left\{ \begin{cases} 0 & t<1 \\ \frac{(-1+t)^2}{2} & 1\le t<2 \\ -\frac{11}{2}+5\,t-t^2 & 2\le t<3 \\ \frac{(4-t)^2}{2} & 3\le t<4 \\ 0 & 4\le t<5 \\ 0 & t\ge 5 \end{cases}, \begin{cases} 0 & t<1 \\ 0 & 1\le t<2 \\ \frac{(-2+t)^2}{2} & 2\le t<3 \\ -\frac{23}{2}+7\,t-t^2 & 3\le t<4 \\ \frac{(5-t)^2}{2} & 4\le t<5 \\ 0 & t\ge 5 \end{cases} \right\}$$

Note that we obtain two piecewise polynomial functions of degree 2 (i.e. order 3), displayed in Fig. 1(bottom-left), both having a similar shape but shifted by length 1 with respect to each other. Finally, there is only one basis function of order 4 for the given knot vector (the piecewise polynomial function of degree 3 in Fig. 1(bottom-right)):

```
In[8]:=N_{i,4}[{1,2,3,4,5},t] //PiecewiseForm
```

$$Out[8] := \begin{cases} 0 & t<1 \\ \frac{(-1+t)^3}{6} & 1\le t<2 \\ \frac{31-45\,t+21\,t^2-3\,t^3}{6} & 2\le t<3 \\ \frac{-131+117\,t-33\,t^2+3\,t^3}{6} & 3\le t<4 \\ \frac{-(-5+t)^3}{6} & 4\le t<5 \\ 0 & t\ge 5 \end{cases}$$

One of the most exciting features of modern computer algebra packages is their ability to integrate symbolic, numerical and graphical capabilities within a unified framework. For example, we can easily display the basis functions of *Out*[5]-*Out*[8] on the interval $(1,5)$:

```
In[9]:=Plot[Table[N_{i,#}[{1,2,3,4,5},t],{i,0,4-#}]
      //Evaluate,{t,1,5},PlotStyle->Table[Hue[(i+1)/(5-#)],
      {i,0,4-#}],DisplayFunction->Identity]& /@ Range[4];
In[10]:=Show[GraphicsArray[Partition[%,2],
                        DisplayFunction->$DisplayFunction]]
```

Out[10] := *See Figure* 1

A qualitatively different behavior is obtained when any of the knots appears more than once (this case is usually referred to as *non-uniform knot vector*). An example is given by the knot vector $\{0,0,1,1,2,2,2\}$. In this case, the basis functions of order 1 are given by:

```
In[11]:=Table[N_{i,1}[{0,0,1,1,2,2,2},t] // PiecewiseForm,{i,0,5}]
```

$$Out[11] := \left\{ \begin{array}{l} \begin{cases} 0 & t<0 \\ 0 & 0\le t<1 \\ 0 & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \begin{cases} 0 & t<0 \\ 1 & 0\le t<1 \\ 0 & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \begin{cases} 0 & t<0 \\ 0 & 0\le t<1 \\ 0 & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \\ \begin{cases} 0 & t<0 \\ 0 & 0\le t<1 \\ 1 & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \begin{cases} 0 & t<0 \\ 0 & 0\le t<1 \\ 0 & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \begin{cases} 0 & t<0 \\ 0 & 0\le t<1 \\ 0 & 1\le t<2 \\ 0 & t\ge 2 \end{cases} \end{array} \right\}$$

Note that the knot spans involving the same knot ($t = 0$, $t = 1$ or $t = 2$) at both ends reduce to a single point. This causes some basis functions (N_{01}, N_{21}, N_{41} and N_{51} in *Out[11]*) to be zero. This behavior continues until the order reaches the multiplicity value of the multiple knot minus 2. For instance, there is an identically null basis function of order 2, namely N_{42}:

```
In[12]:=Table[N_{i,2}[{0,0,1,1,2,2,2},t] // PiecewiseForm,{i,0,4}]
```

$$Out[12] := \left\{ \begin{cases} 0 & t<0 \\ 1-t & 0\le t<1 \\ 0 & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \begin{cases} 0 & t<0 \\ t & 0\le t<1 \\ 0 & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \begin{cases} 0 & t<0 \\ 0 & 0\le t<1 \\ 2-t & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \begin{cases} 0 & t<0 \\ 0 & 0\le t<1 \\ -1+t & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \begin{cases} 0 & t<0 \\ 0 & 0\le t<1 \\ 0 & 1\le t<2 \\ 0 & t\ge 2 \end{cases} \right\}$$

The basis functions of order 3 become:

```
In[13]:=Table[N_{i,3}[{0,0,1,1,2,2,2},t] // PiecewiseForm,{i,0,3}]
```

$$Out[13] := \left\{ \begin{cases} 0 & t<0 \\ 2t-2t^2 & 0\le t<1 \\ 0 & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \begin{cases} 0 & t<0 \\ t^2 & 0\le t<1 \\ (-2+t)^2 & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \begin{cases} 0 & t<0 \\ 0 & 0\le t<1 \\ -4+6t-2t^2 & 1\le t<2 \\ 0 & t\ge 2 \end{cases}, \begin{cases} 0 & t<0 \\ 0 & 0\le t<1 \\ (-1+t)^2 & 1\le t<2 \\ 0 & t\ge 2 \end{cases} \right\}$$

Multiple knots do influence the shape and properties of basis functions; for instance, each time a knot is repeated, the continuity of the basis functions whose support includes this multiple knot decreases. In particular, the continuity of $N_{i,k}$ at an interior knot is C^{k-m-1} [6], m being the multiplicity of the knot. To illustrate this fact, we compute the unique basis function of order 6:

```
In[14]:=(f6=N_{0,6}[{0,0,1,1,2,2,2},t]) // PiecewiseForm
```

$$Out[14] := \left\{ \begin{cases} 0 & t<0 \\ \frac{1}{8}(10-7t)t^4 & 0\le t<1 \\ -\frac{1}{8}(t-2)^3\left(23t^2-32t+12\right) & 1\le t<2 \\ 0 & t\ge 2 \end{cases} \right\}$$

As we can see, $m = 2$ for the knot $t = 1$ and hence $N_{0,6}$ is C^3-continuous at this point. This implies that its third derivative, given by:

```
In[15]:=(f63=D[f6,{t,3}])//Simplify //PiecewiseForm
```

$$Out[15] := \left\{ \begin{cases} \frac{15}{2}(4-7t)t & 0\le t<1 \\ -\frac{15}{2}\left(23t^2-68t+48\right) & 1\le t<2 \end{cases} \right\}$$

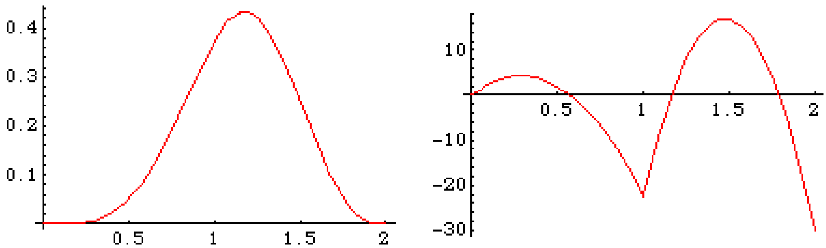

Fig. 2. (left) *6th*-order basis function; (right) its third derivative

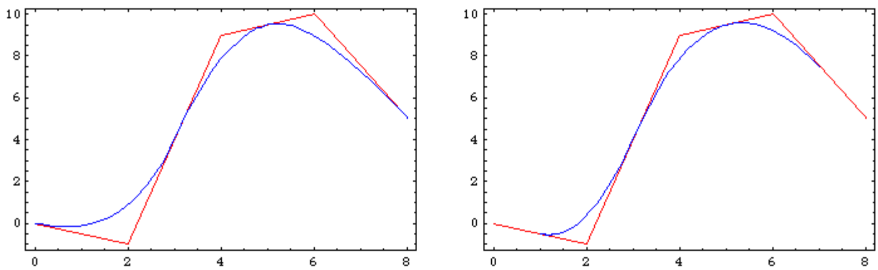

Fig. 3. Bspline curve and its control polygon (the set of segments connecting the control points) for: (left) a non-periodic knot vector; (right) a uniform knot vector

is still continuous but no longer smooth (the continuity of tangent vectors is lost at this point). Figure 2 displays both the basis function of order 6 (on the left) and its third derivative (on the right):

```
In[16]:=Plot[#,{t,0,2},PlotStyle->{RGBColor[1,0,0]},
                         PlotRange->All]& /@ {f6,f63}
```

Out[16] := *See Figure* 2

The most common case of non-uniform knot vectors consists of repeating the end knots as many times as the order while interior knots appear only once (such a knot vector is called *non-periodic knot vector*). In general, a Bspline curve does not interpolate any of the control points; interpolation only occurs for non-periodic knot vectors (the Bspline curve does interpolate the end control points) [6,7]. To illustrate this property, we consider the `BSplineCurve` command (whose input consists of the list of control points `pts`, the order `k`, the knot vector `knots` and the variable `var`), defined as:

```
In[17]:=BSplineCurve[pts_List,k_,knots_List,var_]:=
    Module[{bs,n=Length[pts]},bs=Table[N_{i,k}[knots,var],{i,0,n-1}];
    bs.pts // Simplify];
```

For instance, let us consider a set of 2D control points and two different knot vectors (a non-periodic vector `kv1` and a uniform knot vector `kv2`) and compute the Bspline curve of order 3:

```
In[18]:=cp={{0,0},{2,-1},{4,9},{6,10},{8,5}};
In[19]:={kv1,kv2}={{0,0,0,1,2,3,3,3},{1,2,3,4,5,6,7,8}};
In[20]:=BSplineCurve[cp,3,#,t]& /@ {kv1,kv2};
In[21]:=MapThread[Show[Graphics[{RGBColor[1,0,0],Line[pts]}],
       ParametricPlot[#1 //Evaluate,#2,PlotRange->All,
       PlotStyle->RGBColor[0,0,1],DisplayFunction->Identity],
       PlotRange->All,Frame->True,
       DisplayFunction->$DisplayFunction]&,{%,{{t,0,3},{t,3,6}}}];
In[22]:=Show[GraphicsArray[%]]
```

$Out[22] := See\ Figure\ 3$

The curve interpolates the end control points in the first case, while no control points are interpolated in the second case at all. For graphical purposes, the support of the Bspline curves restrict to the points such that $\sum_{i=0}^{r-k} N_{i,k}(t) = 1$. The next input computes the graphical support for the curves in Fig. 3:

$$\texttt{In[23]:=}\sum_{i=0}^{4} \texttt{N}_{i,3}\texttt{[\#,t]\& /@ \{kv1,kv2\} // PiecewiseForm}$$

$$Out[23] := \left\{ \left\{ \begin{array}{ll} 0 & t<0 \\ 1 & 0\le t<1 \\ 1 & 1\le t<2 \\ 1 & 2\le t<3 \\ 0 & t\ge 3 \end{array} \right\}, \left\{ \begin{array}{cc} 0 & t<1 \\ \frac{1}{2}(t-1)^2 & 1\le t<2 \\ \frac{1}{2}\left(-t^2+6t-7\right) & 2\le t<3 \\ 1 & 3\le t<4 \\ 1 & 4\le t<5 \\ 1 & 5\le t<6 \\ -\frac{t^2}{2}+6t-17 & 6\le t<7 \\ \frac{1}{2}(t-8)^2 & 7\le t<8 \end{array} \right\} \right\}$$

This result makes evident that the Bspline curves in Fig. 3 must be displayed on the intervals $(0,3)$ and $(3,6)$ respectively (see the last line of `In[21]`).

Acknowledgements. This research has been supported by the Spanish Ministry of Education and Science, Project Ref. #TIN2006-13615.

References

1. Choi, B.K., Jerard, R.B: Sculptured Surface Machining. Theory and Applications. Kluwer Academic Publishers, Dordrecht/Boston/London (1998)
2. Echevarría, G., Iglesias, A., Gálvez, A.: Extending neural networks for B-spline surface reconstruction. Lectures Notes in Computer Science, **2330** (2002) 305-314
3. Iglesias, A., Echevarría, G., Gálvez, A.: Functional networks for B-spline surface reconstruction. Future Generation Computer Systems, **20**(8), (2004) 1337-1353
4. Maeder, R.: Programming in Mathematica, Second Edition, Addison-Wesley, Redwood City, CA (1991)
5. Patrikalakis, N.M., Maekawa, T.: Shape Interrogation for Computer Aided Design and Manufacturing. Springer Verlag (2002)

6. Piegl, L., Tiller, W.: The NURBS Book (Second Edition). Springer Verlag, Berlin Heidelberg (1997)
7. Rogers, D.F.: An Introduction to NURBS. With Historical Perspective. Morgan Kaufmann, San Francisco (2001)
8. Wolfram, S.: The Mathematica Book, Fourth Edition, Wolfram Media, Champaign, IL & Cambridge University Press, Cambridge (1999)