

Short guide to the function Bsplines
from the
”Boosted Partial Least-Squares” package

Revision of the 2024/03/01 version

Jean-François Durand

jf.durand001@orange.fr

www.jf-durand-pls.com

2026/09/29

Abstract

This guide presents both mathematical and graphical properties of the B -splines by the `Bsplines` function available in the free open-source package "Boosted Partial Least-Squares" written in R (4) and downloadable from www.jf-durand-pls.com. Since the release 23.00, this function replaces the old `Bspline` function.

`Bsplines` allows to play with splines by handling their tuning parameters (the degree and the knots) in different contexts. First, the statistical domain: nonlinear Data Analysis and functional regression. In particular this function allows the user to familiarize himself with the similar inputs of `PLSS`, see(5), the `Partial Least-Squares Splines` function available in the package. Second, the domain of the approximation by splines: approximation of a smooth function and in CAID, the approximation of some control points by smooth curves. These two domains are addressed at the section 8 of this guide through the `Bsplines` function. Notice that this function must not be in competition with the dedicated packages but just offers a glimpse of what statisticians inexperienced in CAID could desire to know.

In the new release, some facilities are now present in `Bsplines` like the use of particular spline coefficients β called the "nodal" weights leading to the identity spline. Actually, these nodal weights are well known in the context of approximation by splines as the Greville points computed by the mean of successive knots. Here, managing online some variations around the identity spline through additive alterations of the nodal coefficients, allows the user to change globally as well locally the values of a statistical sample and specifically that of a response variable in the regression context.

Contents

- 1. Inputs and Outputs 2
- 2. Short introduction to regression splines 3
- 3. *B*-Splines with no knots are Bernstein polynomials on $[a, b]$ 4
- 4. Definitions 6
- 5. Properties 7
- 6. A visit to the *B*-spline families 8
 - 6.1 degree = 0, knots = 2, equiknots = F 9
 - 6.2 degree = 1, knotslocation = c(86,93,93) 10
 - 6.3 degree = 2, knotslocation = c(86,86,93,93,93) 11
- 7. Variations around the identity spline 12
 - 7.1 Definition of the nodal weights 12
 - 7.2 Some departures from the identity spline 12
- 8. Approximation Splines 15
 - 8.1 The nodal values to approximate a C^2 function 15
 - 8.2 Splines in CAID, from *B*-splines to NURBS 17

1. Inputs and Outputs

Inputs	Type	Description	Default
X	numerical	data to be transformed by B-splines. If matrix, select a column.	missing
degree	integer	the degree of the splines.	1
knots	integer	the number of knots.	0
equiknots	logical	works with "knots": T, equally spaced knots; F, at quantiles.	T
knotslocation	numerical	location of the knots; if NULL, "knots" and "equiknots" work.	NULL
beta	numerical	the β weights; allows to build and display the spline function.	missing
nodal	logical	if T, β_{nodal} weights are computed to display the identity spline.	F
delta	numerical	additive perturbations of the β_{nodal} weights.	0
graph	logical	if T, figures and curves are displayed.	T
newplot	logical	if T, one B-spline per plot, F, all B-splines on a unique plot.	T
titlepar	logical	if T, the title "Bi" for each ith new plot. F, no title.	T
askpar	logical	the user is asked for input, before a new figure is drawn.	T
matrow	integer	number of rows for a matrix of plots.	1
matcol	integer	number of columns for a matrix of plots.	1
data	logical	if T, add tickmarks on the x-axis and transformed X data.	T
colorpar	logical	if F, curves in black, T, coloured curves.	T
nbpoints	integer	the number of points for discretized curves.	300
colknotspar	integer	the colour for marking the knots (location and multiplicity).	9
cexpar	numerical	the amount by which plotting text and symbols are magnified.	0.9
cex.labpar	numerical	the magnification to be used for x and y axis labels.	0.9
bgpar	text	background colour for the figures.	"white"

The blue inputs control the type of the splines, while the red coloured are those for the spline functions.

Green is dedicated to the figures. Two inputs default to "missing", X and beta. First, X is mandatory since it is used as the data to be transformed by splines. Note that X must be at least of two distinct values in order to build the interval on which the B -splines family will act. Second, when beta is missing, the program asks the user to enter the β weights. The answer "y" or "n" allows or not to add the plot of the spline function to the B -splines plots.

Outputs	Type	Description
B	numerical	the coding matrix of X by the B-splines family
degree	integer	the input "degree"
knots	integer	the input "knots"
equiknots	logical	the input "equiknots"
knotslocation	numerical	the location of the effective knots
beta	numerical	the eventual β weights
delta	numerical	the delta perturbations of nodal weights
X	numerical	the matrix-column of X data
Xt	numerical	eventually, the matrix-column of the transformed data

2. Short introduction to regression splines

Polynomials are a classical tool to capture nonlinearities in statistical modeling notwithstanding their well known rigidity to fit the data. To remedy that drawback, piecewise polynomials or splines present the advantage of a convenient flexibility that is to be paid by a local domain of utility say $[a, b]$, spline functions being null outside $[a, b]$, and the risk of overfitting the data.

To transform a continuous variable x whose values range within $[a, b]$, a spline function s is made of adjacent polynomials of same degree d , or of order $m = d + 1$, that join end to end at points called "the knots", with continuity conditions for the derivatives, see for splines the books of De Boor (1) and Shumaker (8).

Among other types of splines, regression splines present some interesting properties:

- Few knots τ_j whose number K and their locations joined to the degree (generally no larger than 3) are the tuning parameters.
- They belong to a functional linear space $S_{[a,b]}$ of dimension $m + K$ whose most popular basis functions are called the B -splines.

First of all, one has to explain how the knots work to control the continuity of the derivatives at the junction points. In fact, the K knots decided by the user are those inside $]a, b[$ and also called the "interior knots". The role of the m multiple knots that merge at a and b are explained a bit later when detailing the properties of the B -splines. So, the ordered sequence of knots can be written

$$\tau_1 = \dots = \tau_m = a < \tau_{m+1} \leq \dots \leq \tau_{m+K} < b = \tau_{m+K+1} = \dots = \tau_{2m+K}.$$

By analogy to polynomials of order m , $p(x, \beta) = \beta_1 + \beta_2 x + \dots + \beta_m x^{m-1}$, a spline $s \in S_{[a,b]}$ is

$$s(x, \beta) = \sum_{j=1}^{m+K} \beta_j B_j^m(x) \quad (1)$$

where $\{B_j^m(\cdot)\}_{j=1, \dots, m+K}$ is the family of B -splines basis functions. Sometimes, for simplicity, a spline is denoted $s(x)$ and in the case of equally spaced knots it is said as uniform. The vector of weights β is to be statistically estimated by a regression method or directly chosen by the user in other domains. Notice that the price of the flexibility paid by the B -splines is the expansion of the dimension from m to $m+K$.

3. B -Splines with no knots are Bernstein polynomials on $[a, b]$

Splines become Bernstein polynomials on $[a, b]$ when no knots are used ($K = 0$). Figure 1 displays 4 plots from the function `Bsplines` showing B -splines families of degree 0, 1, 2 and 3 with no knots on $[1, 10]$. One can see that $0 \leq B(x) \leq 1$. Note that in the R instruction of the Figure 1 caption, the input `data=F` leads to no observed values displayed, neither on the x -horizontal axis nor on the curves for the transformed values. Input `newplot=F` means that all the B -splines of the family are displayed on the same plot. On the contrary, `newplot=T` means one curve per plot.

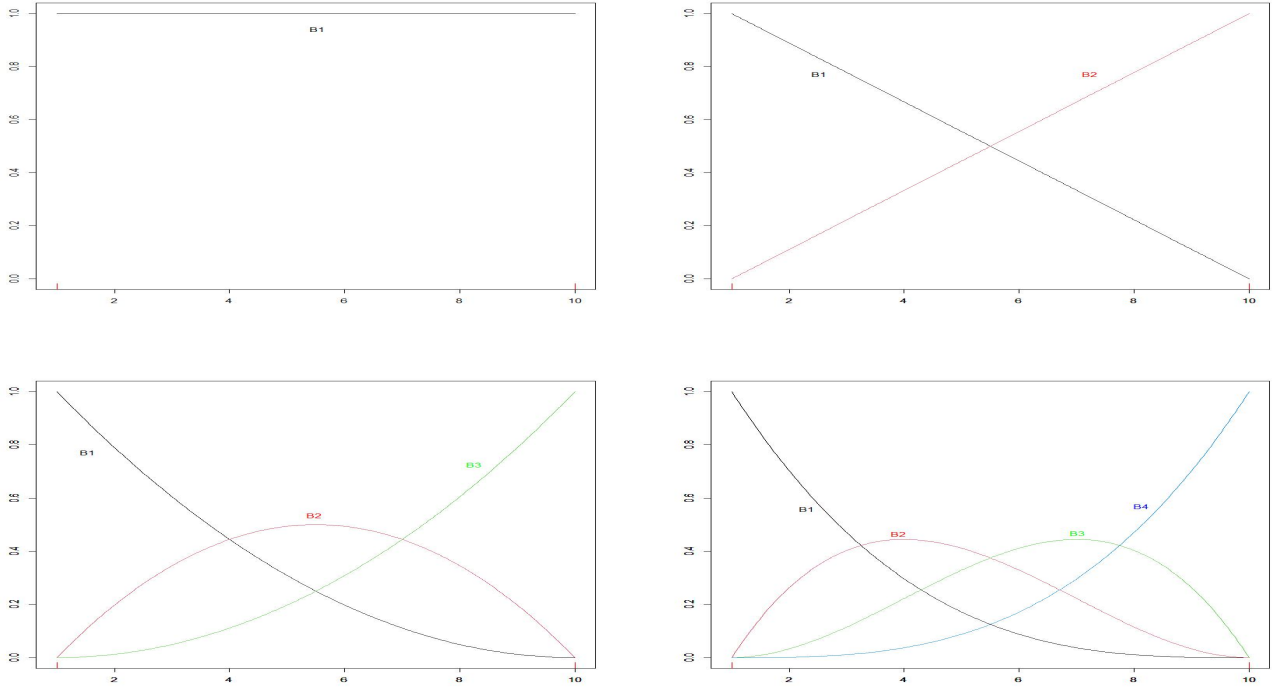


Figure 1: Bsplines(c(1,10),degree=**i**,knots=0,data=F,newplot=F,knotslocation=NULL), **i=0,1,2,3**.

$d = 1$: Bernstein basis = $\left(B_1^2(x) = \frac{b-x}{b-a}, B_2^2(x) = \frac{x-a}{b-a} \right)$.

(a_0, a_1) being the weights in the canonical basis $(1, x)$,

the matrix basis change:
$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \frac{1}{b-a} \begin{bmatrix} b & -a \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}.$$

$d = 2$:

$$\left(B_1^3(x) = \frac{(b-x)^2}{(b-a)^2}, B_2^3(x) = \frac{2(x-a)(b-x)}{(b-a)^2}, B_3^3(x) = \frac{(x-a)^2}{(b-a)^2} \right).$$

(a_0, a_1, a_2) : the weights in the canonical basis $(1, x, x^2)$,

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \frac{1}{(b-a)^2} \begin{bmatrix} b^2 & -2ab & a^2 \\ -2b & 2(a+b) & -2a \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}.$$

$d \geq 3$: Extend the $d = 1$ and $d = 2$ cases through the Newton's binomial terms...

4. Definitions

• Definition 1

Recall the notation of the complete set of knots when K knots are chosen on $]a, b[$

$$\tau_1 = \dots = \tau_m = a < \tau_{m+1} \leq \dots \leq \tau_{m+K} < b = \tau_{m+K+1} = \dots = \tau_{2m+K}.$$

The B -splines of degree d (order $m = d + 1$), are defined by

$$\text{for } j = 1, \dots, m + K,$$

$$B_j^m(x) = (-1)^m (\tau_{j+m} - \tau_j) [\tau_j, \dots, \tau_{j+m}](x - \tau)_+^d,$$

where $[\tau_j, \dots, \tau_{j+m}](x - \tau)_+^d$ is the divided difference of order m computed at $\tau_j, \dots, \tau_{j+m}$ for the truncated power function $\tau \rightarrow (x - \tau)_+^d$.

• Definition 2

According to the Casteljau algorithm, starting from B -splines of order 1 ($d = 0$), B -splines of successive orders k are recursively computed on each interval of distinct knots (τ_j, τ_{j+1}) :

$$\left\{ \begin{array}{l} B_j^1(x) = 1 \text{ if } \tau_j \leq x < \tau_{j+1}, 0 \text{ otherwise.} \\ \text{For } k = 2, \dots, m \\ B_j^k(x) = \frac{x - \tau_j}{\tau_{j+k-1} - \tau_j} B_j^{k-1}(x) + \frac{\tau_{j+k} - x}{\tau_{j+k} - \tau_{j+1}} B_{j+1}^{k-1}(x), \end{array} \right.$$

with the convention: a fraction with a null denominator is to be considered as 0.

To illustrate that recursion with the very simple case of no knots in $[a, b]$, let us construct $\{B_j^2(x)\}_{j=1}^2$ from $B^1(x)$ whose graphs are displayed on the plots at the top of Figure 1. Here the spline space is of dimension 2 ($k = m = 2$ and $K = 0$) and the set of knots is $\{\tau_1 = \tau_2 = 1, \tau_3 = \tau_4 = 10\}$. The two concerned intervals are $[\tau_1, \tau_3]$ for $B_1^2(x)$ and $[\tau_2, \tau_4]$ for $B_2^2(x)$. Note that, here, $B_1^1(x) = B_2^1(x) = B_3^1(x) = B^1(x) = 1$ on $[1, 10]$.

$$B_1^2(x) = \underbrace{\frac{x - \tau_1}{\tau_2 - \tau_1}}_0 B_1^1(x) + \frac{\tau_3 - x}{\tau_3 - \tau_1} B_2^1(x) = \frac{10 - x}{9}$$

$$B_2^2(x) = \frac{x - \tau_2}{\tau_3 - \tau_2} B_2^1(x) + \underbrace{\frac{\tau_4 - x}{\tau_4 - \tau_3}}_0 B_3^1(x) = \frac{x - 1}{9}.$$

Many statistical packages implement that recursion to compute, given an observation x , the $m + K$ values of the B -splines. For example, in the R-package, the native function `bs()` is at disposal from `library(splines)`. Here, the function `Bsplines` is based on the function `bs()`. As easy to use than `bs`, `Bsplines` adds many other graphical and statistical capabilities. It is possible to use B -splines of degree 0 that are not present in the R function although they are a basic element for some standard statistical methods using the coding (0,1) as Correspondance Analysis (simple and multiple), Supervised Classification, etc... Handling online β wheights allows to display the shapes of spline functions and particular variations around the identity spline.

5. Properties

- **P1: Local support**

$$B_j^m(x) = 0, \quad \forall x \notin [\tau_j, \tau_{j+m}].$$

Due to m multiple knots at the extreme values a and b , there are m non null basis functions that share the same support $[\tau_j, \tau_{j+m}]$. As a consequence, one observation x_i has a local influence on $s(x_i)$ that depends only on the m basis functions whose supports encompass this data. This property makes regression splines robust against the influence of outlying data. In return, $s(x) = 0$ outside $[a, b]$.

- **P2: Fuzzy coding functions:**

$$0 \leq B_j^m(x) \leq 1, \quad \text{and} \quad \sum_{j=1}^{m+K} B_j^m(x) = 1.$$

So $B_j^m(x_i)$ may be interpreted as a degree of membership of x_i to $[\tau_j, \tau_{j+m}]$.

- **P3: The multiplicity of knots controls the smoothness**

The multiplicity of a knot is the number of knots that merge at the same point. The multiplicity may

vary from 1 (a simple knot) to m , the order of the spline, and controls the number of continuous derivatives at that point.

Let m_j be the multiplicity of τ_j , $1 \leq m_j \leq m$,
 B -splines of degree $d \geq m_j$ are C^{d-m_j} at τ_j ,
 $m_j = m$ leads to a discontinuity at τ_j (denoted C^{-1}).

The table below details the degrees generally used in regression splines:

degree	0	1		2			3			
multiplicity m_j	1	1	2	1	2	3	1	2	3	4
smoothness at τ_j	C^{-1}	C^0	C^{-1}	C^1	C^0	C^{-1}	C^2	C^1	C^0	C^{-1}

- **P4: The fuzzy coding matrix B**

Let $\mathbf{x} = (x_1, \dots, x_n)'$ be a n -sample (column-vector) of the variable x ,

$$\mathbf{B} = [B_1^m(\mathbf{x}), \dots, B_{m+K}^m(\mathbf{x})],$$

is the $n \times (m + K)$ coding matrix of $\mathbf{x}$. Notice that $d = 0$ provides a binary coding matrix $\mathbf{B}$. The property **P2** implies that the rows of $\mathbf{B}$ sum up to 1. As a consequence, when the columns of $\mathbf{B}$ are centered, $\text{rank}(\mathbf{B}) \leq \min(n - 1, d + K)$.

Sometimes, two variables, x_1 and x_2 , with the same number of observations n , are supposed to act together. This is the case in domains such as, in statistic, the multivariate additive regression by Partial Least-Squares Splines, see (6), or in CAID, NURBS to design 3D curves and surfaces, see section 8.2. Then, to make that x_2 interacts with x_1 , one needs the tensor product of the two B -splines families coded by B^{m_2} and B^{m_1} .

6. A visit to the B -spline families

The response y of the downloadable "cornell" data set will be used to illustrate how Bsplines works:

```
cornell
      x1    x2    x3    x4    x5    x6    x7    y
1  0.00  0.23  0.00  0.00  0.00  0.74  0.03  98.7
2  0.00  0.10  0.00  0.00  0.12  0.74  0.04  97.8
3  0.00  0.00  0.00  0.10  0.12  0.74  0.04  96.6
4  0.00  0.49  0.00  0.00  0.12  0.37  0.02  92.0
5  0.00  0.00  0.00  0.62  0.12  0.18  0.08  86.6
6  0.00  0.62  0.00  0.00  0.00  0.37  0.01  91.2
7  0.17  0.27  0.10  0.38  0.00  0.00  0.08  81.9
8  0.17  0.19  0.10  0.38  0.02  0.06  0.08  83.1
9  0.17  0.21  0.10  0.38  0.00  0.06  0.08  82.4
10 0.17  0.15  0.10  0.38  0.02  0.10  0.08  83.2
11 0.21  0.36  0.12  0.25  0.00  0.00  0.06  81.4
12 0.00  0.00  0.00  0.55  0.00  0.37  0.08  88.1
```

6.1 degree = 0, knots = 2, equiknots = F

```
try=Bsplines(cornell,degree=0,knots=2,equiknots=F,data=T,newplot=T,
matrow=2,matcol=2,titlepar=T,colknots="blue")
```

The variable column number (≤ 8) 1: 8

To plot the regression spline, enter the 3 beta weights,
if not enter RETURN

```
1: -1
2: 3
3: 1
```

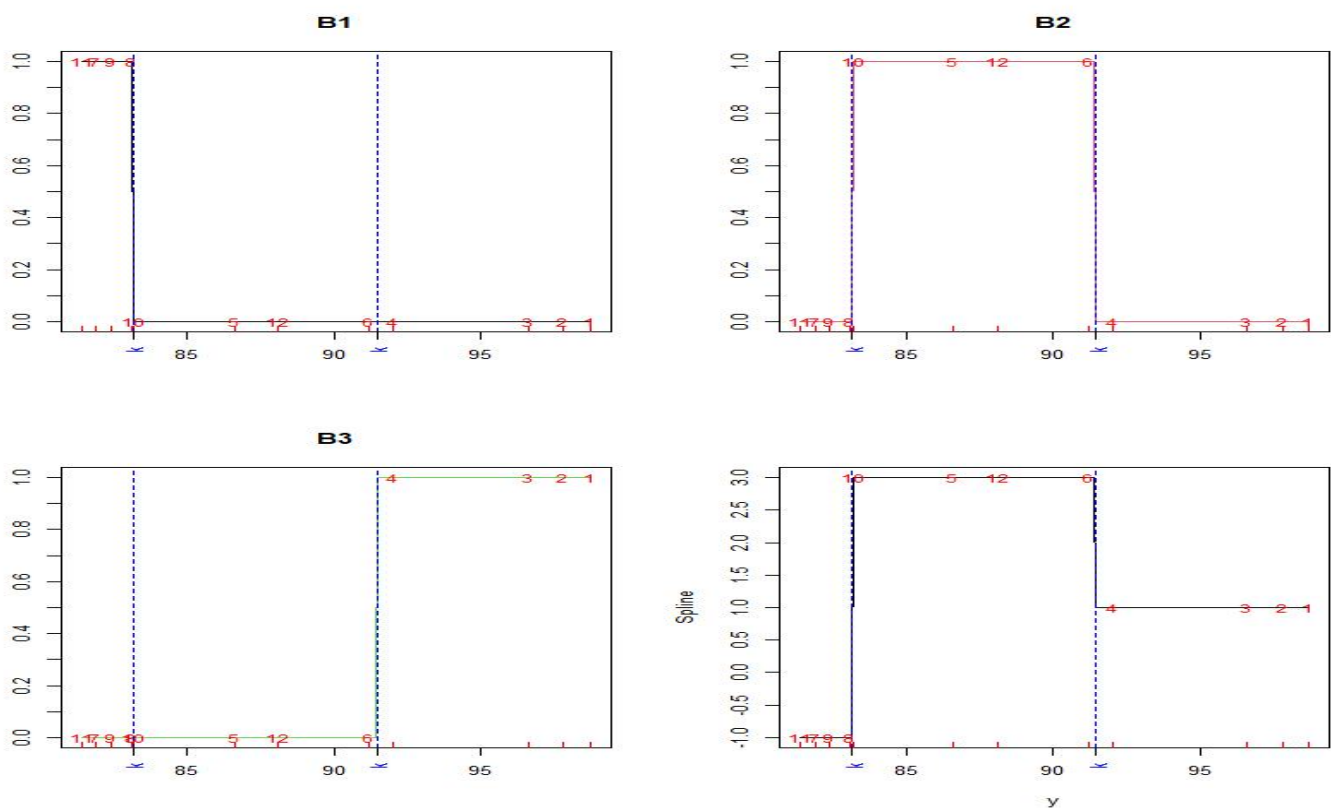


Figure 2: Bsplines of degree 0 with two knots at quantiles

```
cbind(try$X,try$B,try$Xt)
```

	y	y1	y2	y3	y
1	98.7	0	0	1	1
2	97.8	0	0	1	1
3	96.6	0	0	1	1
4	92.0	0	0	1	1
5	86.6	0	1	0	3
6	91.2	0	1	0	3
7	81.9	1	0	0	-1
8	83.1	1	0	0	-1
9	82.4	1	0	0	-1
10	83.2	0	1	0	3
11	81.4	1	0	0	-1
12	88.1	0	1	0	3

6.2 degree = 1, knotslocation = c(86,93,93)

```
try=Bsplines(cornell[,8,drop=F],degree=1,knotslocation=c(86,93,93),
newplot=T,beta=c(1.1,-1.1,2),matrow=2,matcol=3,colknots="blue",cexpar=1.2)
```

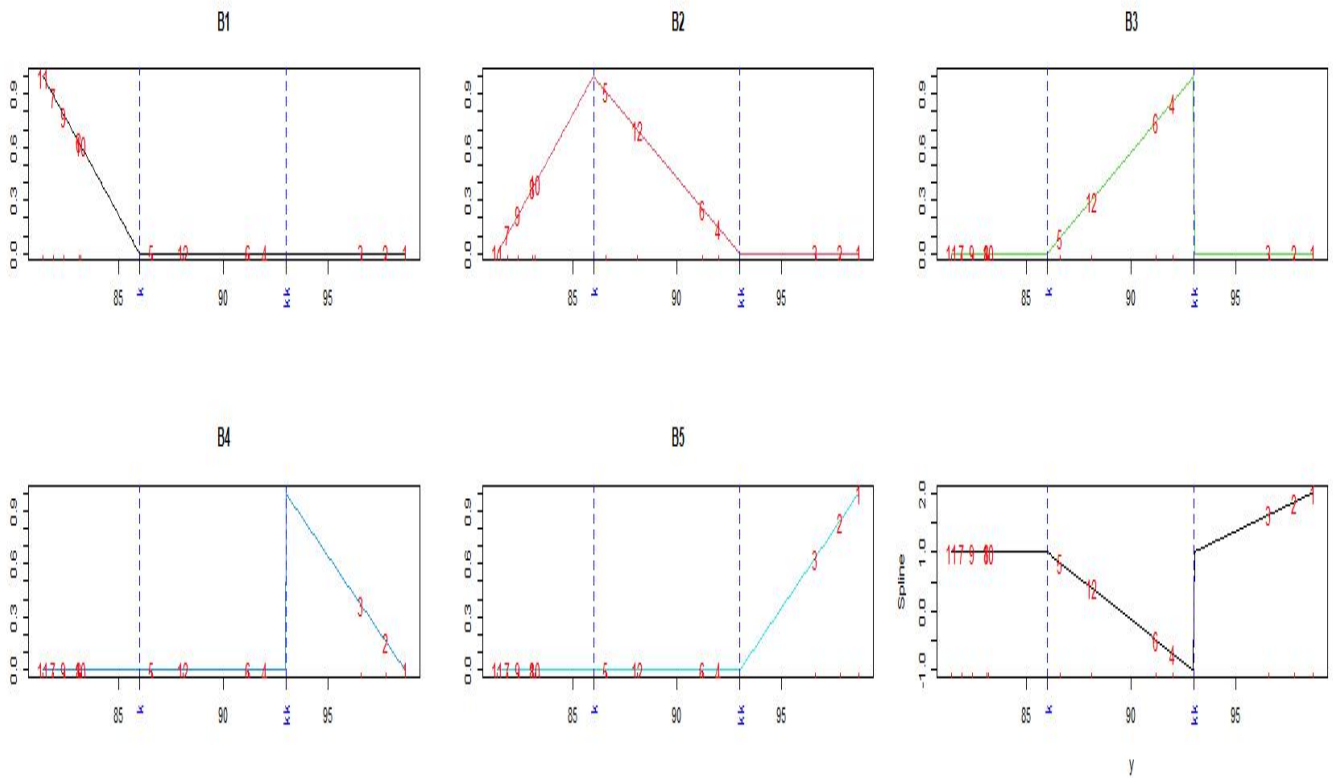


Figure 3: B-splines of degree 1 with two knots with one of multiplicity 2, discontinuity at 93.

Due to the property **P2**, no matter the degree, m successive equal weights on the concerned interval of knots lead to a spline function constant on that interval, here $\beta_1 = \beta_2 = 1$ gives $s(x) = 1$ on $[81.4, 86]$.

6.3 degree = 2, knotslocation = c(86,86,93,93,93)

```
Bsplines(cornell[,8,drop=F],degree=2,knotslocation=c(86,86,93,93,93),
newplot=F,beta=c(1,1,1,-1,1,-1,2,-3))
```

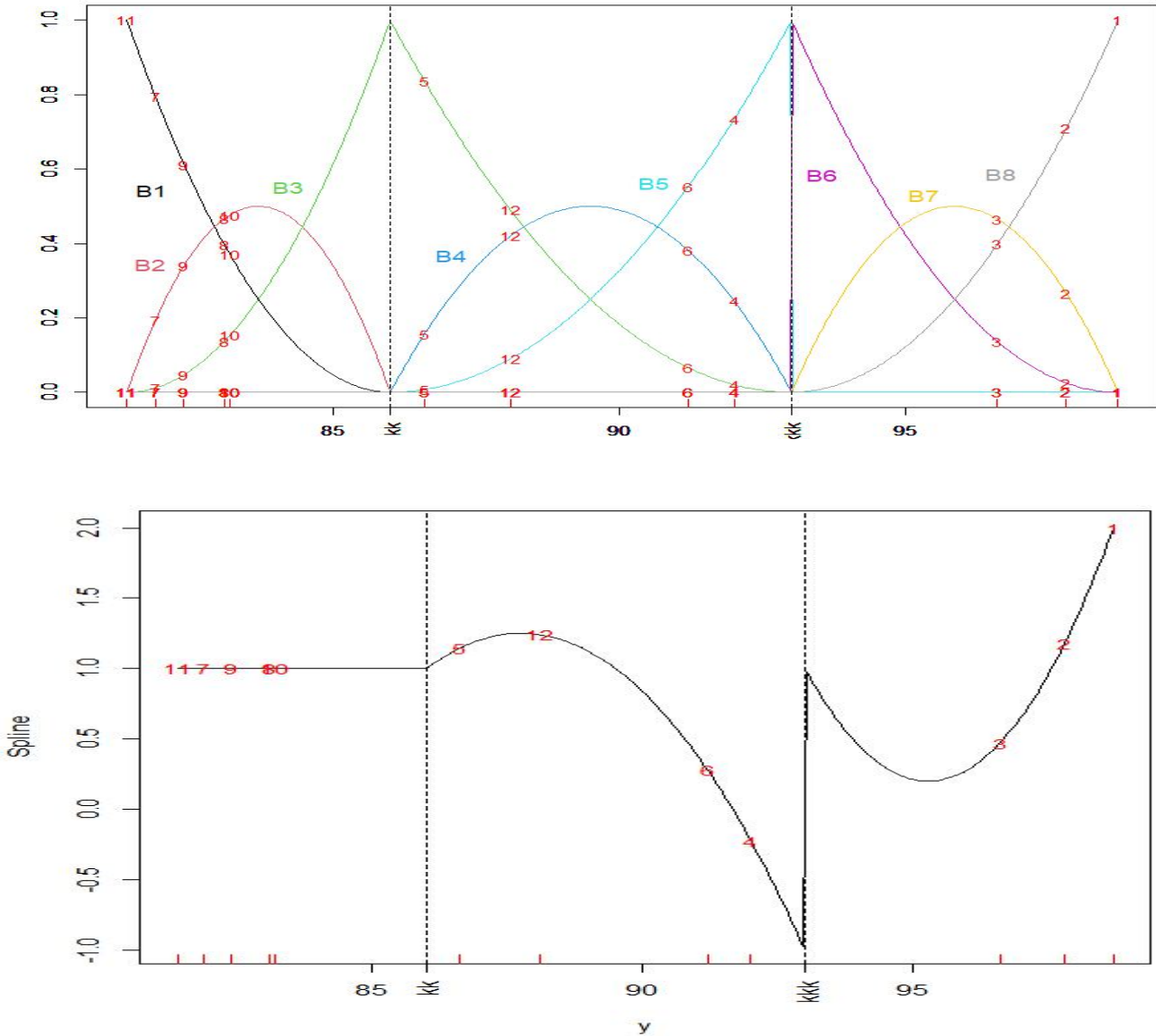


Figure 4: Degree 2 with two knots of multiplicity 2 and 3, C^0 at 86 and discontinuity at 93.

The plots of Figure 4 enlight properties P1, P2 and P3 of the B -splines. First, there are $m = 3$ non null basis functions on each interval of successive knots. Second, m weights equal to 1 makes the spline function constant on the first interval of knots. Third, the multiplicity of knots, here 2 and 3, marked by overturned "k"s on the horizontal axis, controls the smoothness of the splines at the junction points.

7. Variations around the identity spline

The Bsplines function offers the possibility of computing the weights associated to a particular spline function called the "identity spline", see Shumaker (8) for the definition of its β coefficients called the "nodal" weights also called the Greville points in the context of approximation splines, see section 8. See also Durand (3), and (2) for the use of the identity spline in nonlinear multiple regression as the first iteration in a sequence of spline transformations of the predictors. In the same statistical regression context, the aim is here, Durand et al. (7), to give the user through variations of the identity spline, the online opportunity to experience responses that are more desirable than the one observed.

7.1 Definition of the nodal weights

Whatever $d > 0$ and whatever the knots, there exists some weights call the nodal weights denoted β_{nodal} , such that $s(x, \beta_{nodal}) = x$ and defined by the mean of d successive knots

$$\text{for } j \in 1, \dots, m + K, \quad [\beta_{nodal}]_j = \frac{1}{d} \sum_{k=1}^d \tau_{j+k}.$$

The logical input `nodal = T` leads to call the sub-function `nodal` that computes these weights to display the identity spline (if `graph = T`) and to experiment some variations around that function.

7.2 Some departures from the identity spline

A variation of the identity spline function is based on the vector δ of length $m + K$, used for additive alterations of the nodal weights, $\beta_{nodal} + \delta$. Splines belonging to a functional linear space, a δ -scenario for the variable x , denoted $x(\delta)$, leads to a δ -spline shift of the variable:

$$x(\delta) = s(x, \beta_{nodal} + \delta) = s(x, \beta_{nodal}) + s(x, \delta) = x + s(x, \delta).$$

A δ -scenario basically depends on the decided spline family (degree and knots). Notice that $x(0) = x$.

When `nodal=T` in the R-function `Bsplines`, the input `delta` allows online changes in the observations. All the following examples involve splines of the same degree 2. Figure 5 displays the nodal wheights and the identity spline.

```
try=Bsplines(cornell[,8,drop=F],degree=2,knotslocation=c(86,93),
nodal=T,newplot=T,matrow=2,matcol=3,cexpar=1.2)
*****
The identity spline with nodal beta weights is computed if nodal=T.
Additive perturbations are controlled by delta
*****
Nodal beta = 81.4 83.7 89.5 95.85 98.7
delta = 0 0 0 0 0
```

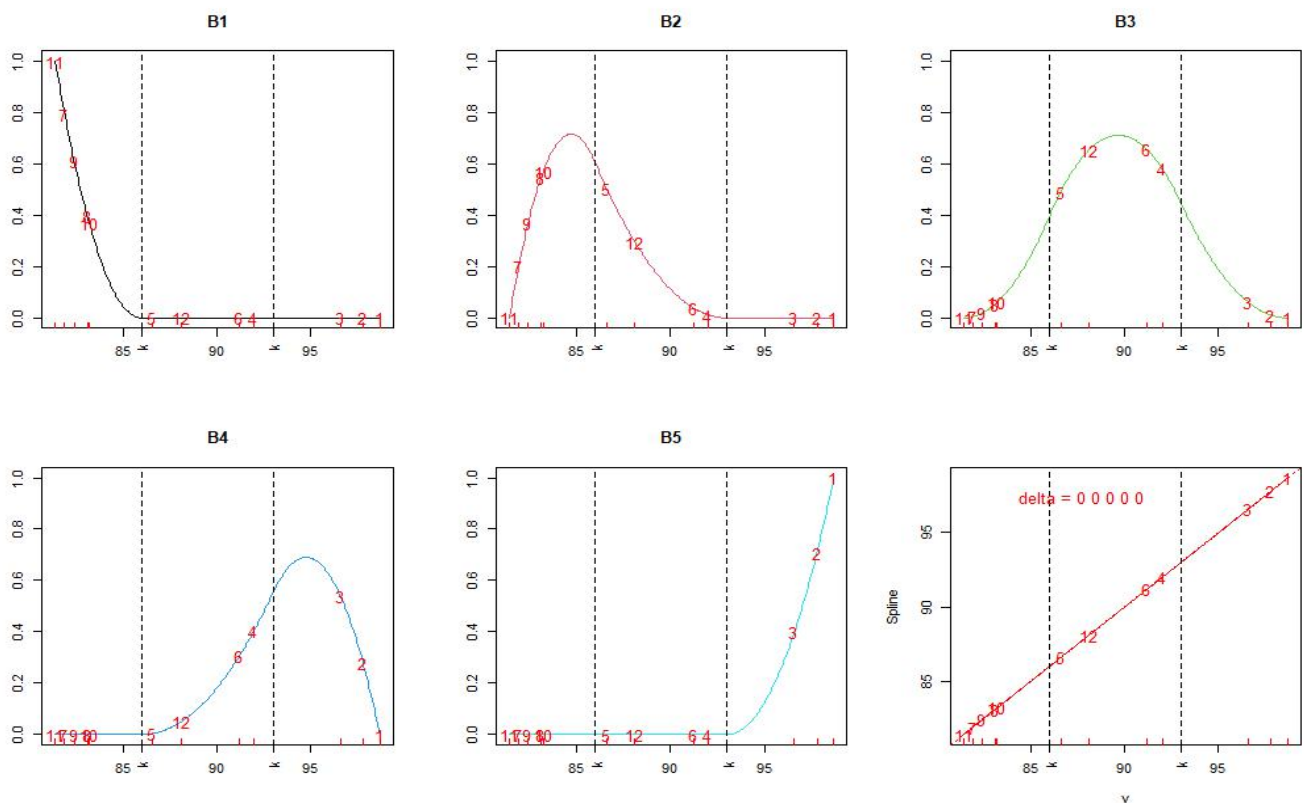


Figure 5: B-splines ($d = 2$, two knots) and the identity spline function ($\delta = 0$).

```
try=Bsplines(cornell[,8,drop=F],degree=2,knotslocation=c(86,93),
newplot=F,nodal=T,delta=c(0,0,-2.5,0,0),matrow=1,matcol=1,cexpar=1.2)
*****
The identity spline with nodal beta weights is computed if nodal=T.
Additive perturbations are controlled by delta
*****
Nodal beta = 81.4 83.7 89.5 95.85 98.7
delta = 0 0 -2.5 0 0
Other additive pertubations on nodal beta? Enter y.
Then, enter delta (5 values).
1: n
```

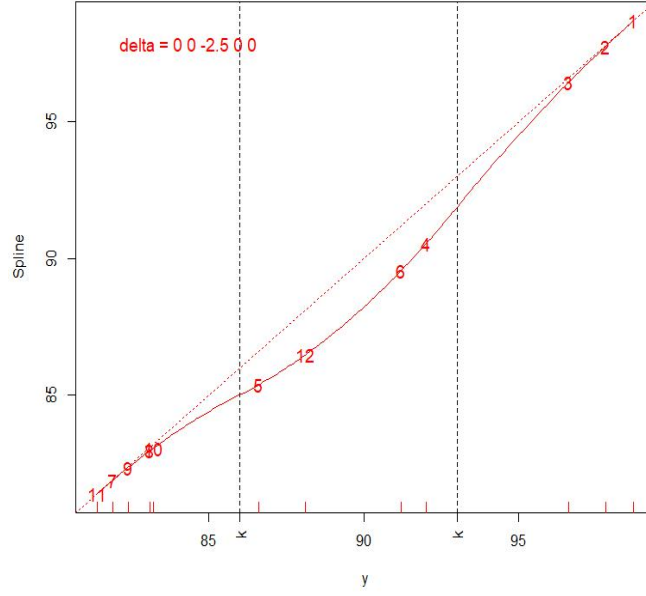


Figure 6: Degree 2, the identity spline (red dotted line) and the δ -scenario .

```
cbind(try$X, try$B, try$Xt)
```

	y	B*1	B*2	B*3	B*4	B*5	y
1	98.7	0.0000000	0.00000000	0.000000000	0.000000000	1.0000000	98.70000
2	97.8	0.0000000	0.00000000	0.011189391	0.279669335	0.7091413	97.77203
3	96.6	0.0000000	0.00000000	0.060920017	0.540188017	0.3988920	96.44770
4	92.0	0.0000000	0.01231527	0.582735348	0.404949381	0.0000000	90.54316
5	86.6	0.0000000	0.50443350	0.491517009	0.004049494	0.0000000	85.37121
6	91.2	0.0000000	0.03990148	0.655936542	0.304161980	0.0000000	89.56016
7	81.9	0.7944234	0.20089140	0.004685157	0.000000000	0.0000000	81.88829
8	83.1	0.3974480	0.54839157	0.054160420	0.000000000	0.0000000	82.96460
9	82.4	0.6124764	0.36878300	0.018740630	0.000000000	0.0000000	82.35315
10	83.2	0.3705104	0.56876996	0.060719640	0.000000000	0.0000000	83.04820
11	81.4	1.0000000	0.00000000	0.000000000	0.000000000	0.0000000	81.40000
12	88.1	0.0000000	0.29568966	0.654704046	0.049606299	0.0000000	86.46324

The first plot of Figure 7 illustrates the particular case of the identity spline altered by two opposite constants, $c = 3$ and $c = -3$. As a consequence of the property P2, whatever the degree and the knots, the two graphs are straight lines that are symmetrical with respect to the identity spline. So, in that case,

$$s(x, \beta_{nodal} + c) = x + c.$$

$$s^{-1}(x, \beta_{nodal} + c) = s(x, \beta_{nodal} - c) = x - c.$$

As a result of properties P1 and P2, one can see on the second and third plots of Figure 7 that

$$\{\delta_i = c \text{ for } i = j, \dots, j + m - 1\} \implies \{s(x, \beta_{nodal} + \delta) = x + c, \quad \forall x \in [\tau_j, \tau_{j+m}]\}.$$

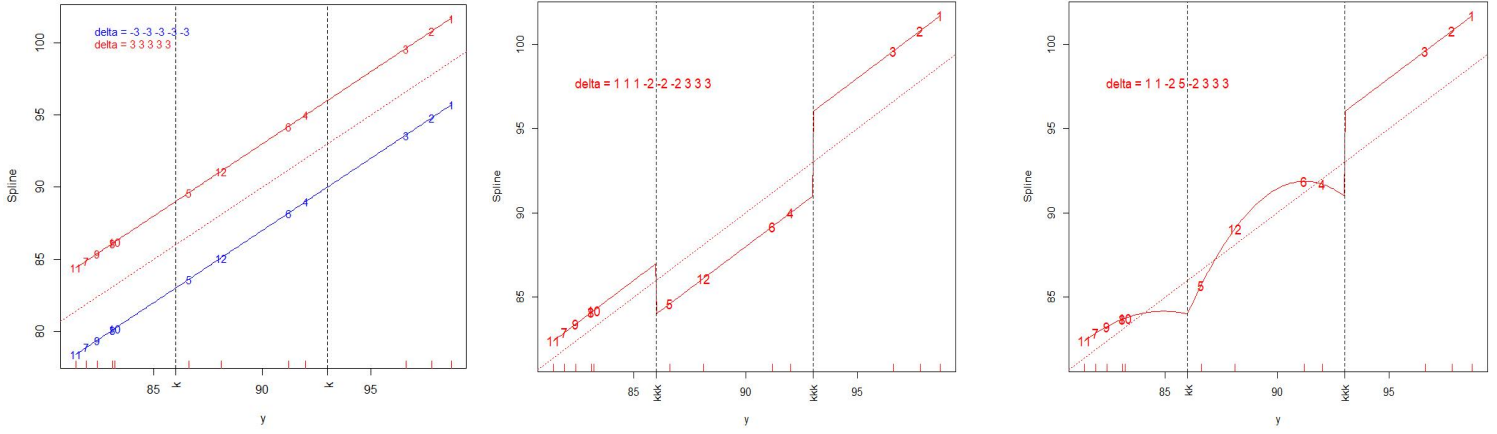


Figure 7: Degree 2, global, local, continuous and discontinuous variations of the identity spline.

Knots' multiplicity for discontinuities is marked by overturned k letters.

8. Approximation Splines

Here, the context is not that of a statistical regression but the approximation by splines of two different mathematical objects, smooth functions and points.

8.1 The nodal values to approximate a C^2 function

Let f be a C^2 function on $[a, b]$ and suppose that the K knots are equally spaced in a broad sense (may be with multiplicity > 1). The uniform spline

$$s_f(x) = \sum_{j=1}^{m+K} f([\beta_{\text{nodal}}]_j) B_j^m(x) \quad (2)$$

is a spline approximation of f on $[a, b]$ with the following upper bound error

$$\|f - s_f\|_{\infty} \leq \frac{h^2}{2d^2} \|f''\|_{\infty}$$

with $\|f\|_{\infty} = \sup_{x \in [a, b]} |f(x)|$ and $h = \sup_j (\tau_{j+1} - \tau_j)$.

In order to show the function `Bsplines` in that context one can use two successive calls to that function:

the first, to compute the nodal values, the second, to display the approximation spline.

```
#-----
# Example: f(x)=cos(x), [a=0,b=4*pi], degree=1, 10 equally spaced knots
#-----

try=Bsplines(c(0,4*pi),degree=1,knots=10,equiknots=T,nodal=T,graph=F)

# try$beta gives the nodal values with nodal=T and delta=0 by default

Bsplines(c(0,4*pi),degree=1,knots=10,equiknots=T,beta=cos(try$beta),

newplot=F,data=F)

# Now, the graph of f(x)= cos(x)

x=seq(0,4*pi,length=200)

lines(x,cos(x),col="blue",lwd=1.5)
```

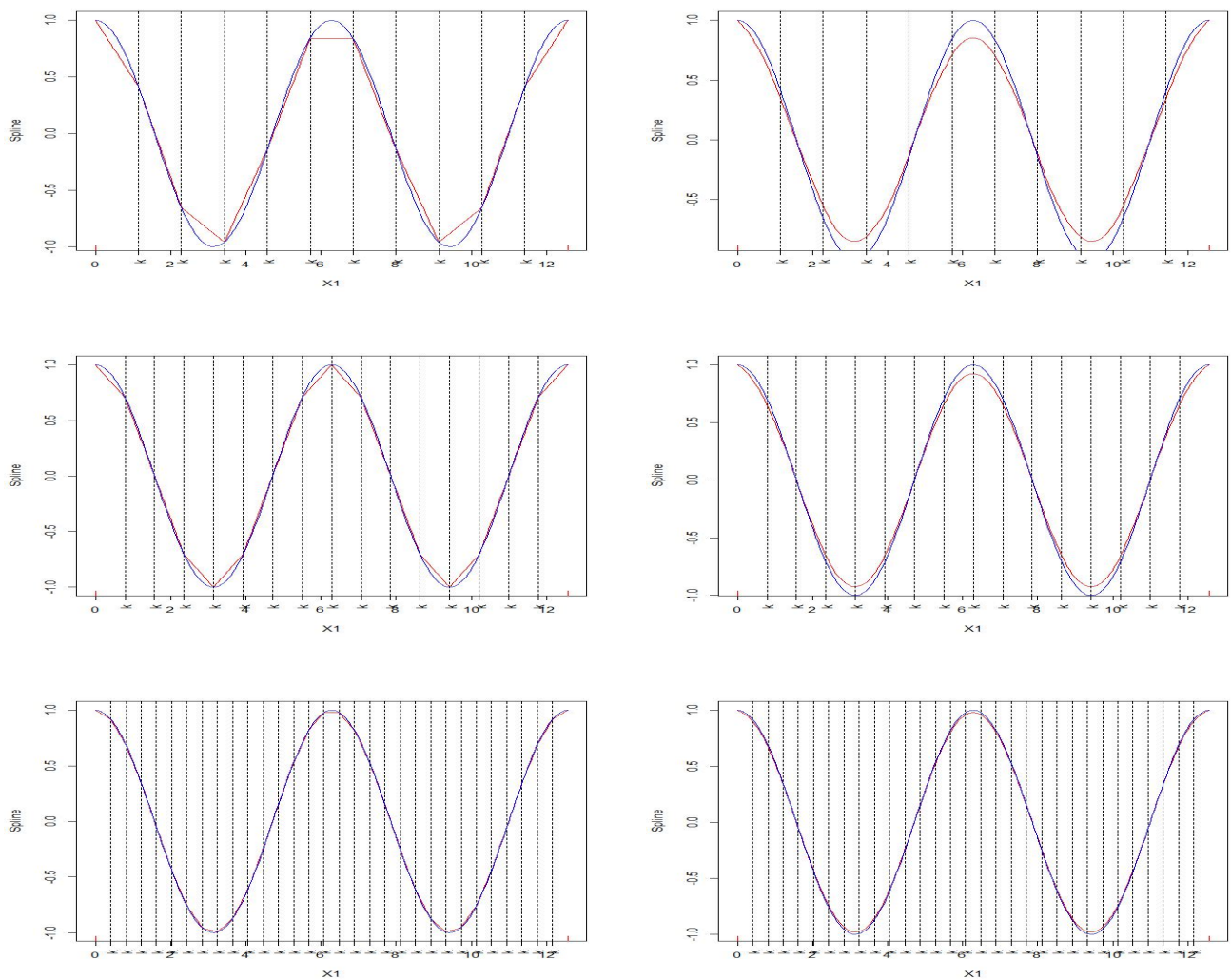


Figure 8: In blue, $f(x) = \cos(x)$. In red, approximation splines with 10, 15 and 30 knots, $d=1$ and 2.

8.2 Splines in CAID, from B -splines to NURBS

In Computer Aid Design (CAID) the computer is used to design smooth flexible curves close to some 2D control points (the 3D case can be considered as a straightforward extension). To that aim the main tool is the B -splines also used to design surfaces through their tensor product.

The B -spline curves

The β_j in the definition (1) are replaced by the control points $\mathbf{P}_j = (X_j, Y_j)'$, and smooth curves require to use degrees 2 or 3. A B -spline curve controlled by the polygon $\mathbf{P} = \{\mathbf{P}_j\}_{j=1,\dots,N}$ is defined by

$$\forall x \in [a, b], \quad \mathbf{S}(x, \mathbf{P}) = \sum_{j=1}^N \mathbf{P}_j B_j^m(x). \quad (3)$$

The number of knots, K , is imposed to fit with the definition (1) and depends on N and d

$$K = N - m. \quad (4)$$

So, $N \geq d + 1$. For example, the degree 2 requires at least 3 control points and $N = 3$ leads to $K = 0$ and a quadratic polynomial approximation of the 3 points.

A spline given by (3) and (4), also denoted $\mathbf{S}(x)$ for simplicity, is a parametric curve $(X(x), Y(x))'$ depending on the dumb variable x . So, $[a, b]$ can be replaced by $[0, 1]$ without loss of generality.

The R lines above detail how the R-function works to illustrate some properties of the curve approximating the control points.

```
#-----  
# Example: construction of P, the Nx2 matrix of control points  
#-----  
  
P=matrix(c(0, 2, 2, 5, 5, 1, 8, 4, 11, -1), ncol=2, byrow=T)  
  
dimnames(P)=list(paste("P", 1:nrow(P), sep=""), c("X", "Y"))
```

```

P
  X  Y
P1  0  2
P2  2  5
P3  5  1
P4  8  4
P5 11 -1

plot(P,type="n")

text(P,labels=dimnames(P)[[1]])

lines(P,lty=2)

# Uniform clamped quadratic curve in red

try1=Bsplines(seq(0,1,length=200),degree=2,knots=nrow(P)-3,
  equiknots=T,graph=F)

lines(try1$B%*%P,col="red")

# Suggestion: write the R-lines for the remaining curves of Figures 10 and 11.

#-----

```

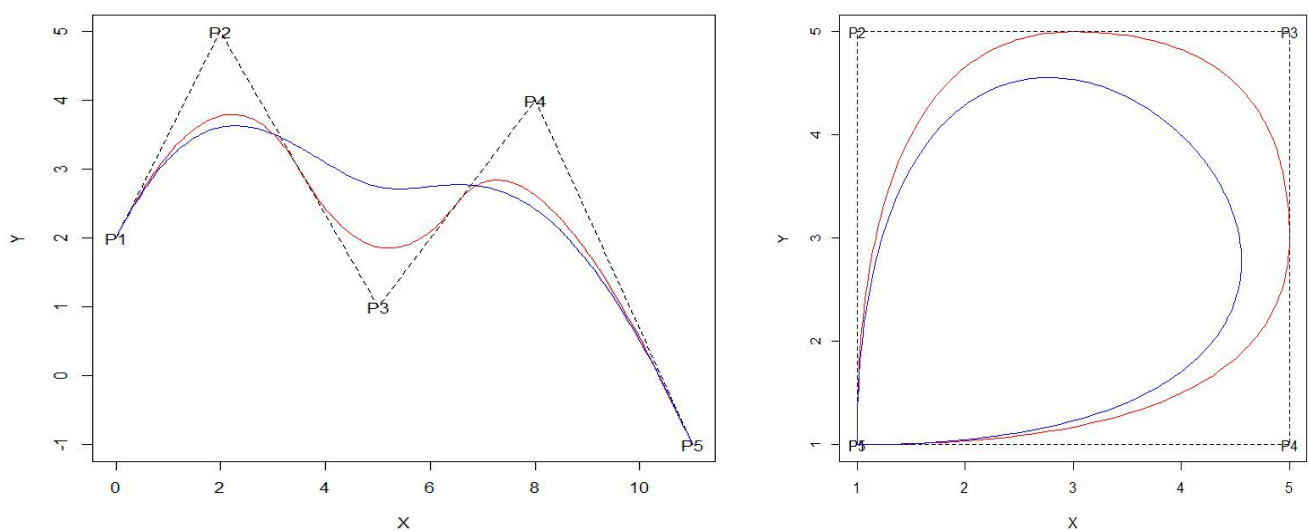


Figure 9: Open and closed uniform clamped curves, $d = 2$ (red) and $d = 3$ (blue) with 5 control points.

Notice that $m = d + 1$ multiple knots at 0 and 1 leads to a curve that passes through P_1 and P_N . One says that the curve is clamped to the extremities. The R-function of the package enforces that choice that is not imperative in general. However this allows closed curves when $P_1 = P_N$.

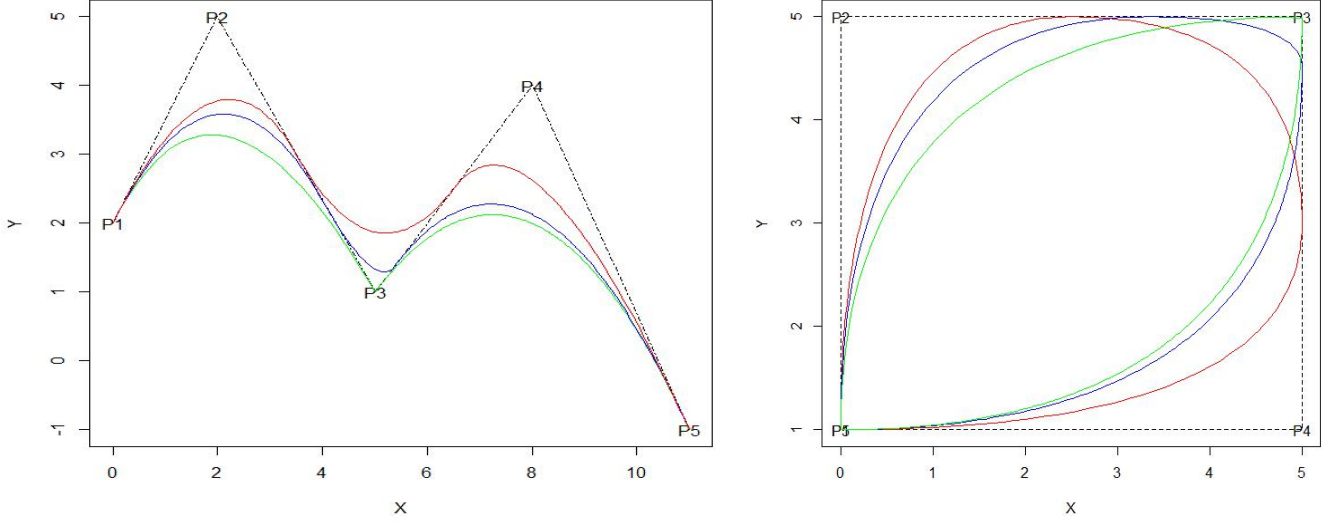


Figure 10: $d = 2$. Uniform curves in red, knots = $(1/3, 2/3)$; non uniform, in blue with knots = $(0.2, 0.3)$, in green with multiple knots = $(0.2, 0.2)$.

Figure 10 and Figure 11 illustrate some other properties of the spline curve:

- the curve lies inside the convex hull of the control points since (3) is a barycentric relationship with positive weights;
- a clamped curve is tangent to $P_1 P_2$ at P_1 and tangent to $P_{N-1} P_N$ at P_N .

From B -splines to NURBS curves

The main drawback of the B -spline curves, see the right plot of Figure 10 and Figure 11, is that one cannot design exactly conics like circles, ellipses, parabolas... To remedy that disadvantage, Non-Uniform Rational B-Splines, NURBS, is an extension to the B -spline curves (3) and (4) in which the control points

have homogeneous coordinates.

$$\forall x \in [a, b], \quad \mathbf{N}(x, \mathbf{P}) = \frac{\sum_{j=1}^N \omega_j \mathbf{P}_j B_j^m(x)}{\sum_{j=1}^N \omega_j B_j^m(x)}, \quad (5)$$

with the fitting condition, imposed by definition (1), between the number of knots, the order of the splines and the number of control points

$$K = N - m. \quad (6)$$

Due to the property **P2**, when $\omega_j = \omega$, NURBS become the usual B -spline curves.

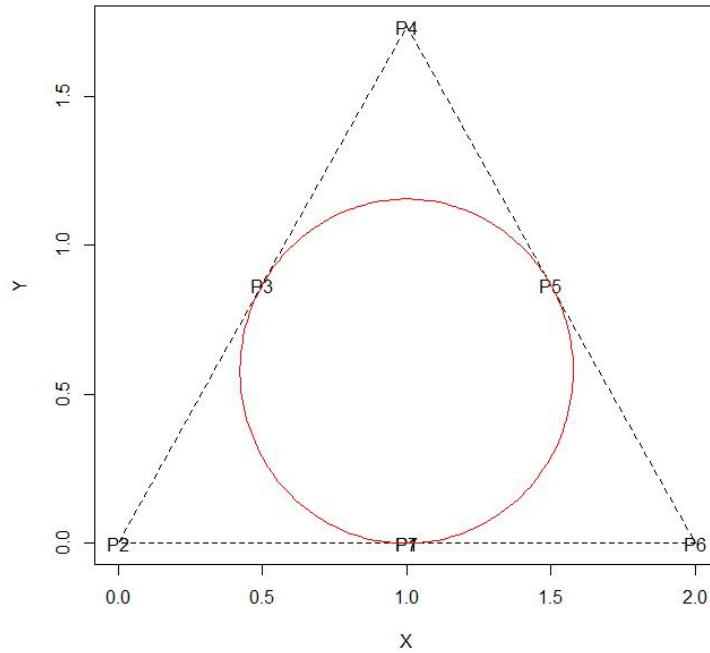


Figure 11: NURBS circle of degree 2, 7 control points , 2 multiple knots of multiplicity 2.

```
#-----
# Example: a NURBS circle of degree 2 with 7 control points
#-----
# P=(1,0) , (0,0) , (1/2,sqrt(3)/2) , (1,sqrt(3)) , (3/2,sqrt(3)/2) , (2,0) , (1,0)
```

```

# omega=1,0.5,1,0.5,1,0.5,1

# degree 2, [a=0,b=3], 2 multiple knots (1,1),(2,2)

#-----

P=matrix(c(1,0,0,0,1/2,sqrt(3)/2,1,sqrt(3),3/2,sqrt(3)/2,2,0,1,0),

ncol=2,byrow=T)

dimnames(P)=list(paste("P",1:nrow(P),sep=""),c("X","Y"))

omegaVec=c(1,0.5,1,0.5,1,0.5,1)

omega=cbind(omegaVec,omegaVec)

omegaP=omega*P

plot(P,type="n")

text(P,dimnames(P)[[1]])

lines(P,lty=2)

try=Bsplines(seq(0,3,length=200),degree=2,knotslocation=c(1,1,2,2),graph=F)

Numerator=try$B%*%omegaP

Denominator=try$B%*%omega

lines(Numerator/Denominator,col="red")

#-----

```

From B -splines to NURBS surfaces

Let $\mathbf{P} = [\mathbf{P}_{i,j}]$ the $N_2 \times N_1$ matrix of the 3D control points $\mathbf{P}_{i,j}$,

$$\forall u \in [a_1, b_1], \forall v \in [a_2, b_2],$$

$$\mathbf{S}(u, v, \mathbf{P}) = \sum_{i=1}^{N_2} \sum_{j=1}^{N_1} \mathbf{P}_{i,j} B_i^{m_2}(v) B_j^{m_1}(u) = \sum_{i=1}^{N_2} B_i^{m_2}(v) \sum_{j=1}^{N_1} \mathbf{P}_{i,j} B_j^{m_1}(u) \quad (7)$$

$$\mathbf{N}(u, v, \mathbf{P}) = \frac{\sum_{i=1}^{N_2} B_i^{m_2}(v) \sum_{j=1}^{N_1} \omega_{i,j} \mathbf{P}_{i,j} B_j^{m_1}(u)}{\sum_{i=1}^{N_2} B_i^{m_2}(v) \sum_{j=1}^{N_1} \omega_{i,j} B_j^{m_1}(u)}, \quad (8)$$

with the fitting conditions for the numbers of knots and control points

$$K_2 = N_2 - m_2 \quad \text{and} \quad K_1 = N_1 - m_1. \quad (9)$$

Notice that (7) is a particular (8) with $\omega_{i,j} = \omega$.

Bibliography

- [1] De Boor, C. (1978). *A Practical Guide to Splines*. Springer-Verlag, Berlin.
- [2] Durand, J.F., Sabatier, R. (1997). Additive splines for partial least squares regression. *Journal of the American Statistical Association*, **92**, 440–467.
- [3] Durand, J.F. (1993). Generalised principal component analysis with respect to instrumental variables via univariate spline transformations. *Computational Statistics & Data Analysis*, **16**, 423–440.
- [4] R Development Core Team (2006). *R : A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.
- [5] Durand, J.F. (2001). Local polynomial additive regression through PLS and splines: PLSS. *Chemometrics and Intelligent Laboratory Systems*, **58**, 235–246.
- [6] Lombardo, R., Durand, J. F. & De Veaux, R. (2009). Model building in multivariate additive partial least squares splines via the GCV criterion. *Journal of Chemometrics*, **23**, 605– 617.
- [7] Durand, JF., Lombardo R., Camminatiello, I., 2025. Identity spline variations in boosted Partial Least-Squares: a study on poverty. *Statistical Methods & Applications*, **34**, 1077–1093.
- [8] Shumaker, L. L. (1981). *Spline Functions: Basic Theory*. John Wiley & Sons: New York, Chichester, Brisbane, Toronto.