Short guide to the function Bsplines

from the

”Boosted Partial Least-Squares” package

Jean-François Durand

jf.durand001@orange.fr

www.jf-durand-pls.com

2023/11/21

Abstract

This guide presents both mathematical and graphical properties of the function $\mathrm{Bsplines}$ from the free open-source package ”Boosted Partial Least-Squares” written in R (4) and downloadable from *www.jf-durand-pls.com*. Available since the release 23.00, this function replaces the old $\mathrm{Bspline}$ function.

$\mathrm{Bsplines}$ allows to play with splines by handling their tuning parameters (the degree and the knots) and to familarize himself with the similar inputs of PLSS, the Partial Least-Squares Splines function.

Moreover some new facilities are now present like the use of particular spline coefficients β called the ”nodal” weights leading to the spline identity. Actually, these nodal weights are well known in the context of modeling curves and surfaces in Computer Aided Design (CAD) as the Greville control points computed by the mean of successive knots. Here, managing online some variations around the spline identity through additive perturbations of the nodal coefficients, allows the user to change globally as well locally the values of the observed variable. At last, the nodal values are used to approximate a smooth function of class C^2 and the $\mathrm{Bsplines}$ function works in that context to display the evolution of the approximation splines with different degrees and knots.

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 spline identity.	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 is magnified.	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 witch 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 payed 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 (5).

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

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

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

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

where $\{B_j^m(.)\}_{j=1,...,m+K}$ 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 $\boldsymbol{\beta}$ is to be estimated by a regression method or directly chosen by the user. Notice that the price of the flexibility payed by the B-splines is the expansion of the dimension from m to $m + K$.

3. Splines are polynomials on $[a, b]$ when $K = 0$

Splines become usual polynomials on $[a, b]$ when no knots are used ($K = 0$). In fact, on $[0, 1]$ the B-splines are the Bernstein polynomials. 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.

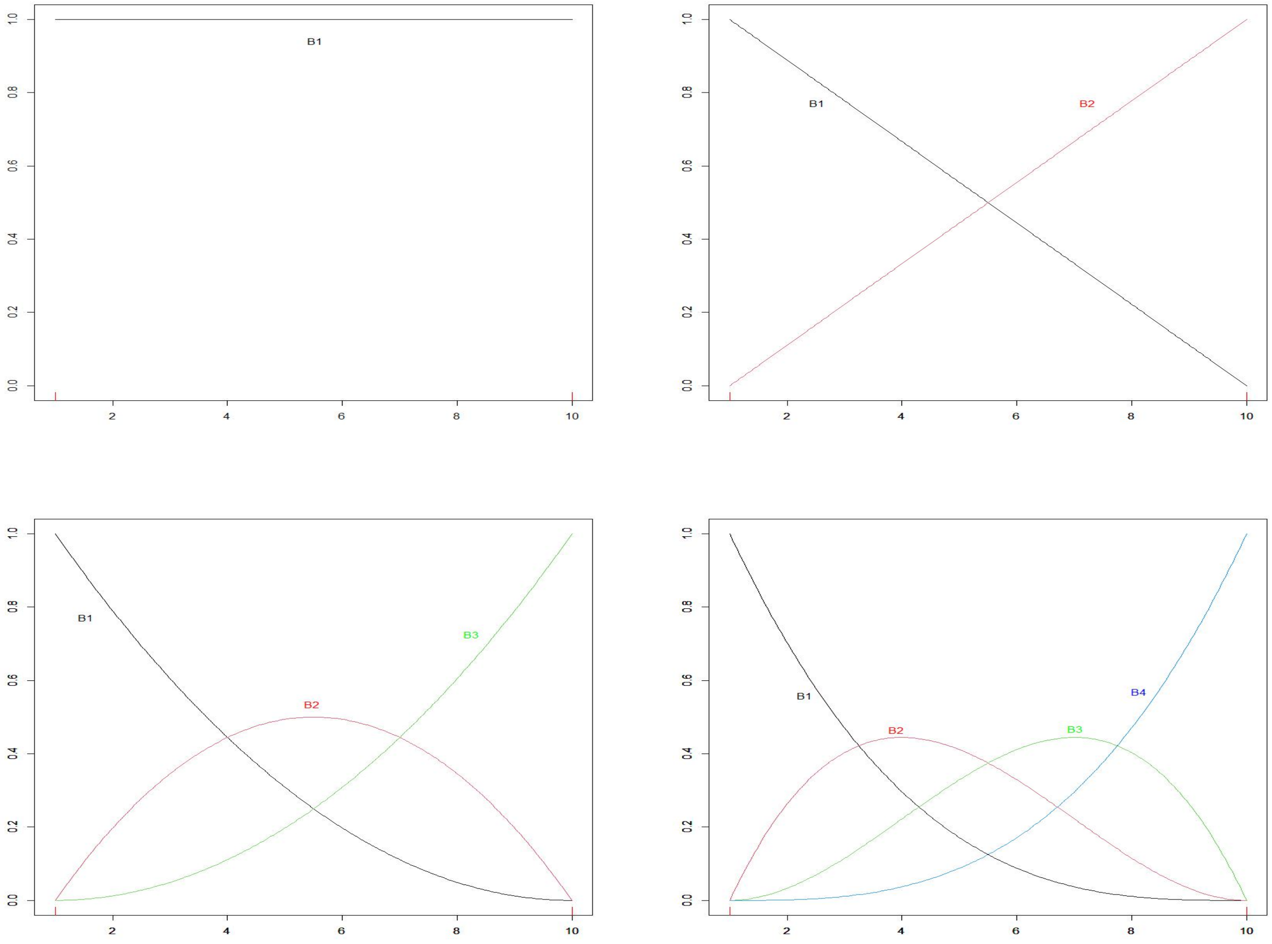

Figure 1: Bsplines(c(1,10),degree=i,knots=0,data=F,newplot=F,knotslocation=NULL), i=0,1,2,3.

4. Definition of the B-splines

- **Definition 1**

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

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

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

for $j = 1, ..., m + K$,

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

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

- **Definition 2**

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

$$\begin{cases} B_j^1(x) = 1 \text{ if } \tau_j \leq x < \tau_{j+1},\ 0 \text{ otherwise.} \\ \text{For } k = 2, ..., m \\ B_j^k(x) = \dfrac{x - \tau_j}{\tau_{j+k-1} - \tau_j} B_j^{k-1}(x) + \dfrac{\tau_{j+k} - x}{\tau_{j+k} - \tau_{j+1}} B_{j+1}^{k-1}(x), \end{cases}$$

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 $[1, 10]$.

$$B_1^2(x) = \frac{x - \tau_1}{\underbrace{\tau_2 - \tau_1}_{0}} B_1^1(x) + \frac{\tau_3 - x}{\tau_3 - \tau_1} B_2^1(x) = \frac{10 - x}{9}$$

$$B_2^2(x) = \frac{x - \tau_2}{\tau_3 - \tau_2} B_2^1(x) + \frac{\tau_4 - x}{\underbrace{\tau_4 - \tau_3}_{0}} B_3^1(x) = \frac{x - 1}{9}.$$

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 $\mathrm{bs()}$ is at disposal from $\mathrm{library(splines)}$. Here, the function $\mathrm{Bsplines}$ is based on the function $\mathrm{bs()}$. As easy to use than bs, $\mathrm{Bsplines}$ 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 $\boldsymbol{\beta}$ wheights allows to display the shapes of spline functions and particular variations around the spline identity.

5. Properties of the B-splines

- **P1: Local support**

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

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:**

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

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.

$$\boxed{\begin{array}{l}\text{Let } m_j \text{ be the multiplicity of } \tau_j, \quad 1 \leq m_j \leq m, \\ B\text{-splines of degee } d \geq m_j \text{ are } C^{d-m_j} \text{ at } \tau_j, \\ m_j = m \text{ leads to a discontinuity at } \tau_j \text{ (denoted } C^{-1}\text{).}\end{array}}$$

The table below details this property for the degrees generally used with regression splines.

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

- **P4: The fuzzy coding matrix $\boldsymbol{B}$**

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

  $$\boxed{\boldsymbol{B} = [B_1^m(\boldsymbol{x}), ..., B_{m+K}^m(\boldsymbol{x})],}$$

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

6. The attractive B-splines family for coding data

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
```

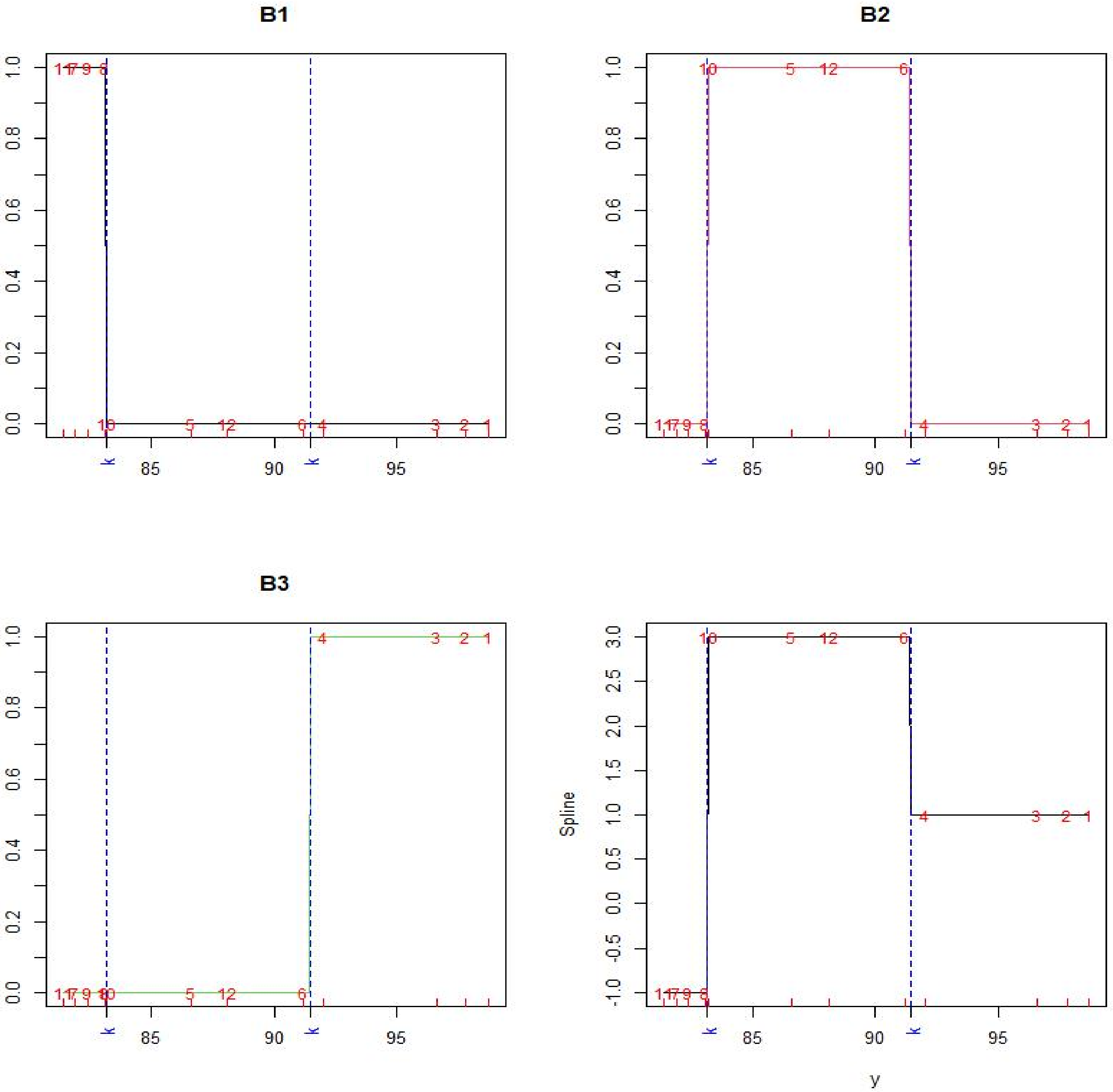

Figure 2: Bsplines of degree 0 with two knots at quantiles

```
> 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)
```

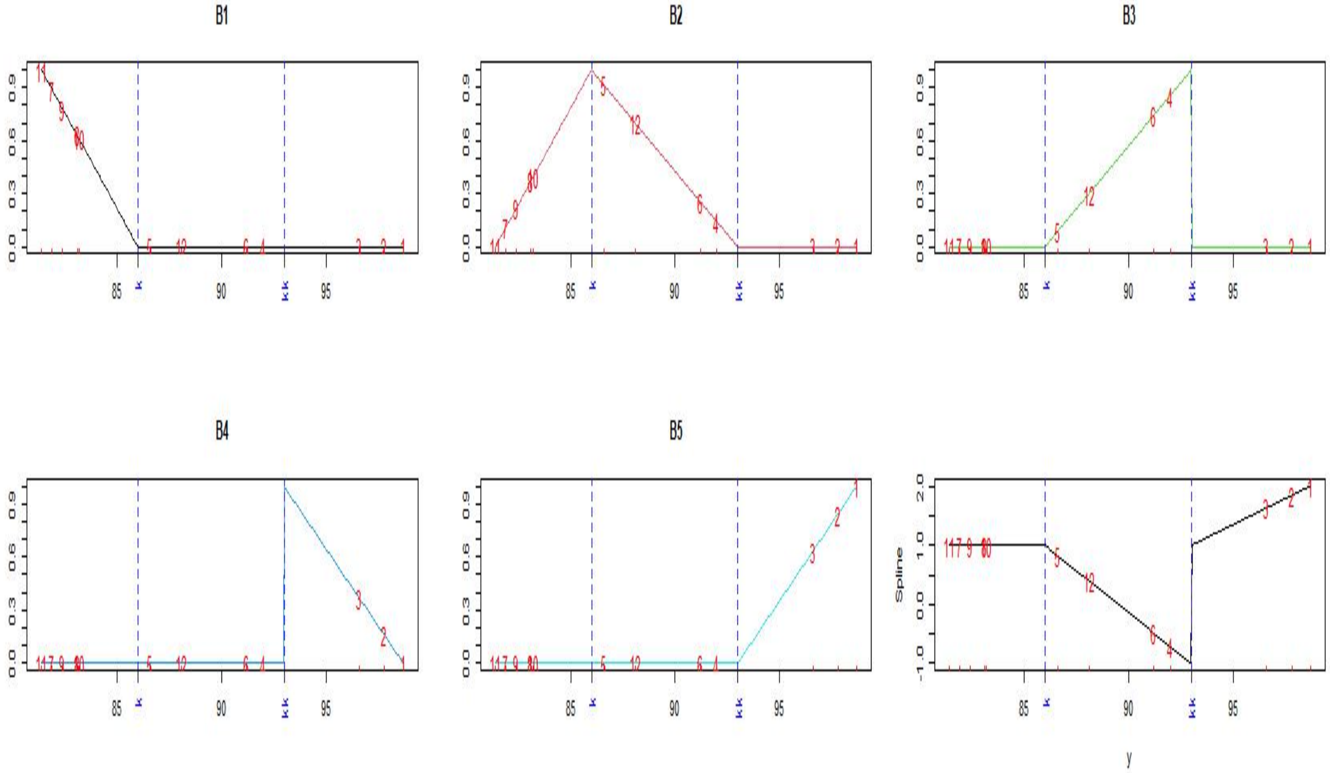

Figure 3: Bsplines of degree 1 with two knots wich one of multiplicity 2, discontinuity at 93.

Due to the property **P2**, no matter the degree, m successive equal wheights 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))
```

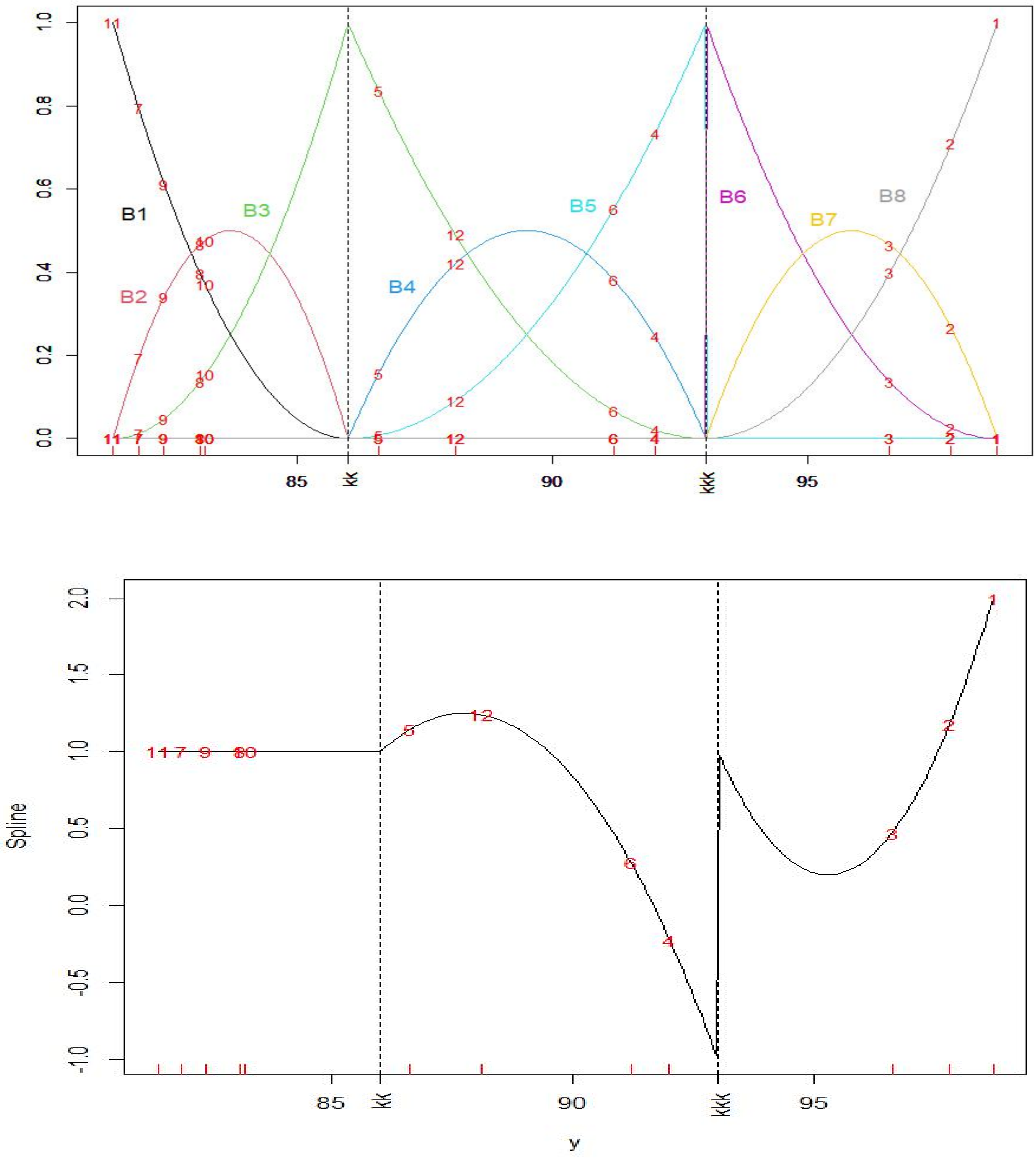

Figure 4: Degree 2 with two knots of multiplicity 2 and 3, C^0 at 86 and discontinuity at 93.

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 eaqual to 1 makes the spline function constatnt 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 spline identity

The Bsplines function offers the possibility of computing the weights associated to a particular spline function called the "spline identity", see Shumaker (5) for the definition of its $\boldsymbol{\beta}$ 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 spline identity in nonlinear multiple regression as the first iteration in a sequence of spline transformations of the predictors.

7.1 Definition of the nodal weights of the spline identity

Whatever $d > 0$, there exists some weights call the nodal weights and denoted $\boldsymbol{\beta}_{\text{nodal}}$, such that

$$s(x, \boldsymbol{\beta}_{\text{nodal}}) = x, \qquad \forall d > 0, \quad \forall x \in [a, b],$$

and defined by the mean of d successive knots

$$\boxed{\text{for } j \in 1, ..., m + K, \qquad [\boldsymbol{\beta}_{\text{nodal}}]_j = \frac{1}{d} \sum_{k=1}^{d} t_{j+k}.}$$

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

7.2 Some departures from the spline identity

The input vector delta (defaulting to 0 that leads to the spline identity) proposes additive perturbations of the $\boldsymbol{\beta}_{nodal}$ weights that allows to experiment online some changes in the x observations. All the following examples involve splines of the same degree 2.

```
> try=Bsplines(cornell[,8,drop=F],degree=2,knotslocation=c(86,93),
nodal=T,newplot=T,matrow=2,matcol=3,cexpar=1.2)
******************************************
The spline identity 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
```

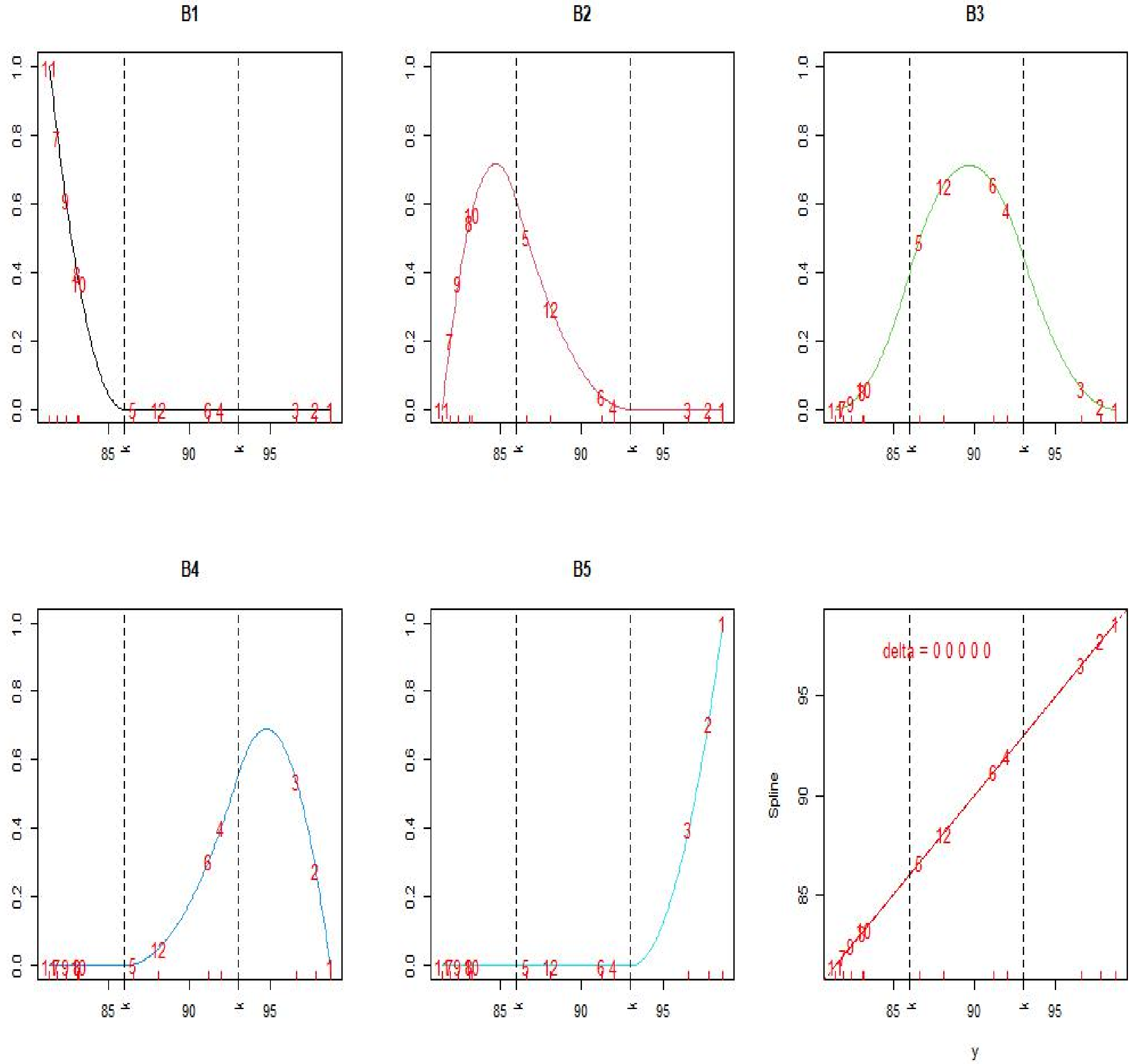

Figure 5: Bsplines ($d = 2$, two knots) and the spline identity (delta=0).

```
>  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 spline identity 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
```

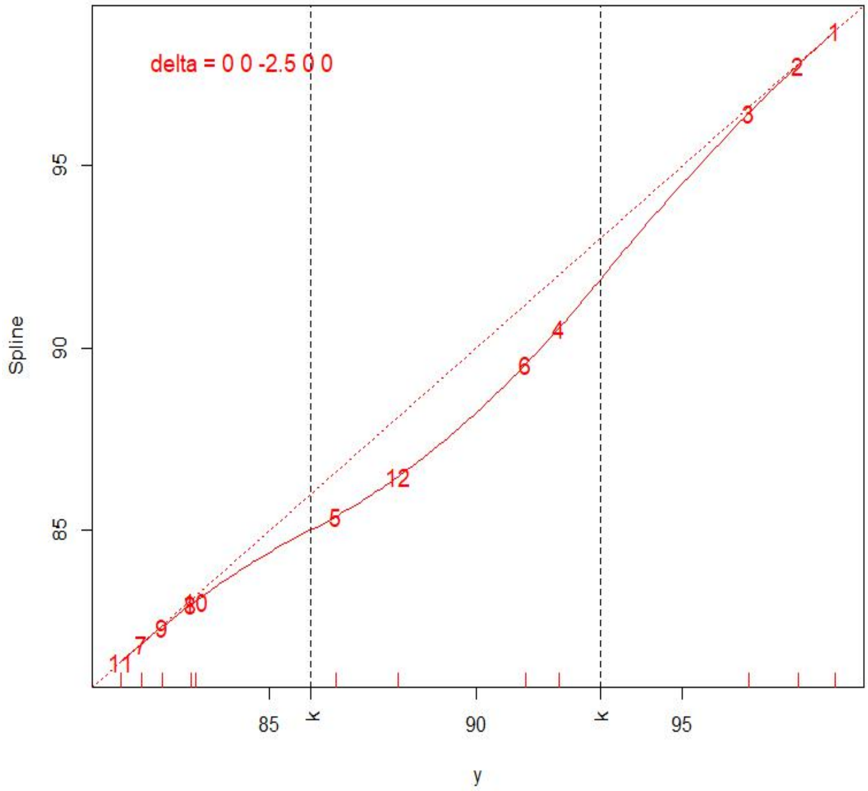

Figure 6: Degree 2, the spline identity (red dotted line) and its delta pertubation .

```
> 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
```

Figure 7 illustrates a particular case when the spline identity is perturbated by two opposit constants, $c = 3$ and $c = -3$. As a consequence of the property **P2**, the two graphs are straight lines that are symetrical with respect to the spline identity. So, in that case,

$$s(x, \boldsymbol{\beta}_{nodal} + \boldsymbol{c}) = x + c.$$

$$s^{-1}(x, \boldsymbol{\beta}_{\text{nodal}} + \boldsymbol{c}) = s(x, \boldsymbol{\beta}_{\text{nodal}} - \boldsymbol{c}) = x - c.$$

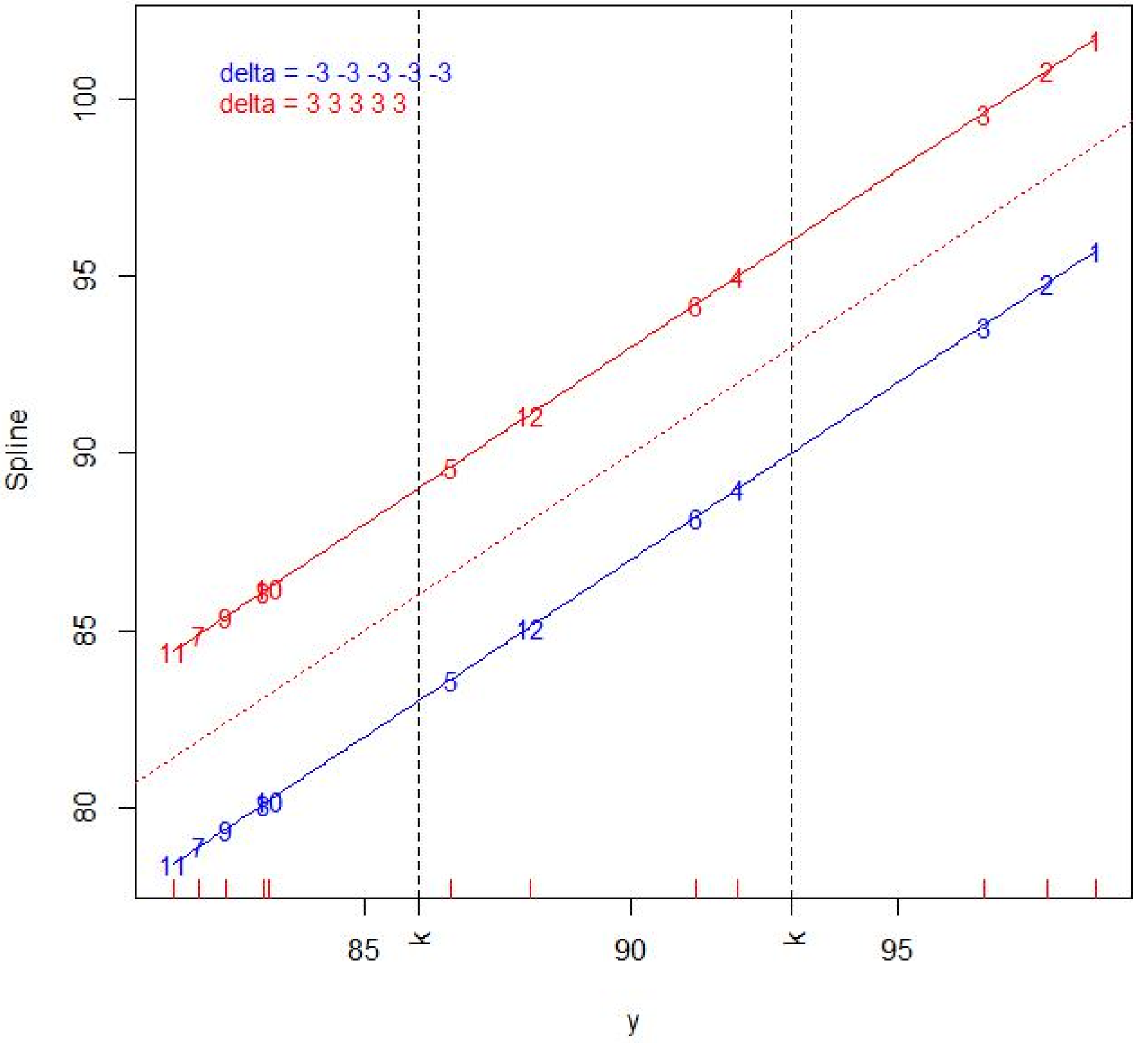

Figure 7: Degree 2, pertubations of the spline identity by two opposite constants 3 and -3.

Suggestion: Using the function Bsplines, write the R-instuctions that display the plots of Figure 7 and Figure 8.

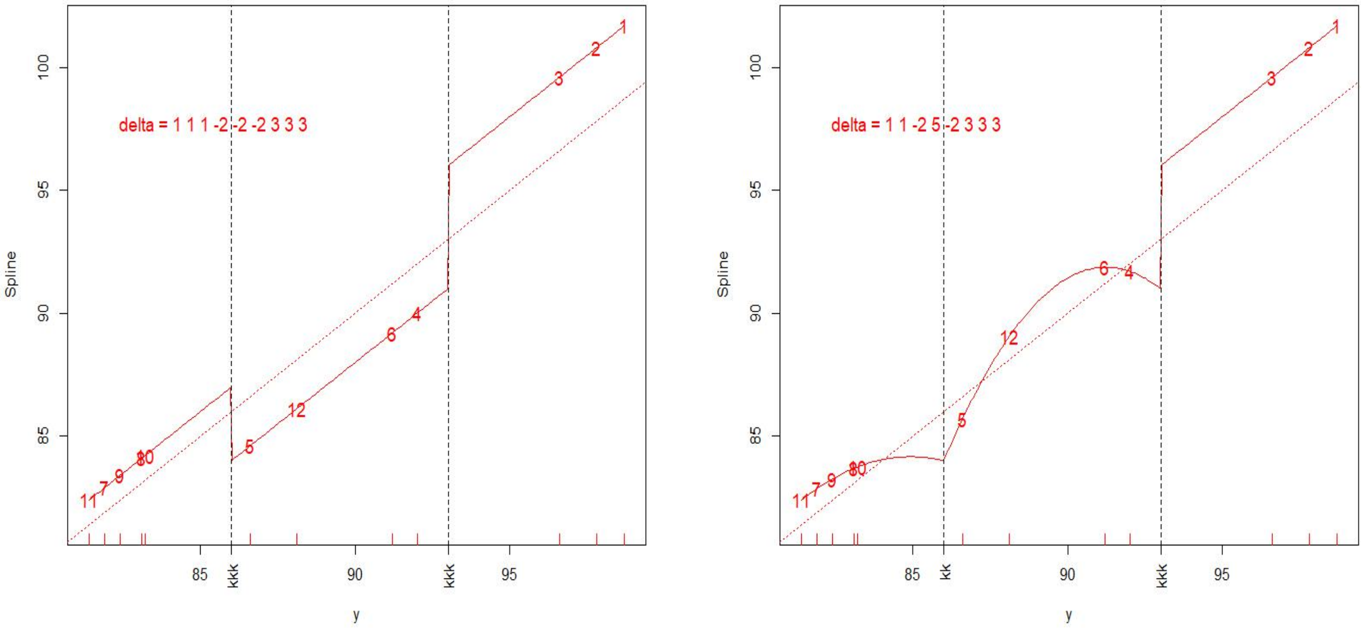

Figure 8: Degree 2, pertubations of the spline identity with multiple knots.

8. The nodal values to approximate a C^2 smooth 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

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

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 $h = sup_j(t_{j+1} - t_j)$.

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

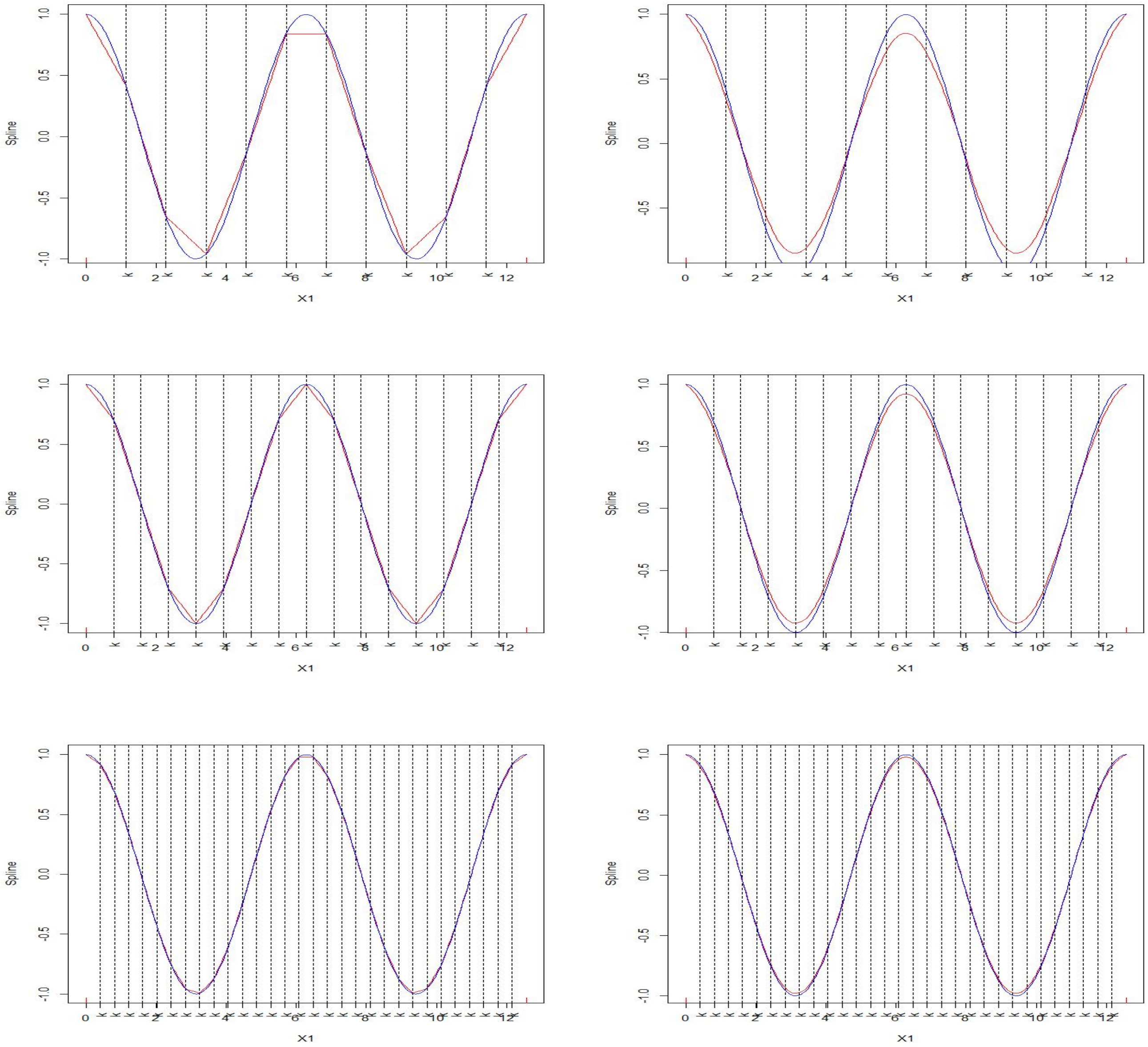

Figure 9: In blue, $f(x)=cos(x)$. In red, the approximation splines with 10, 15 and 30 knots, d=1 and 2.

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). Generalysed principal component analysis with respect to instrumental variables via univariate spline transformations. *Computational Statistics* & *Dtata 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] Shumaker, L. L. (1981). *Spline Functions: Basic Theory.* John Wiley & Sons: New York, Chichester, Brisbane, Toronto.