Regensburger
DISKUSSIONSBEITRÄGE
zur Wirtschaftswissenschaft

University of Regensburg Working Papers in Business,
Economics and Management Information Systems

A short introduction to splines in least squares regression analysis

Kathrin Kagerer*

March 28, 2013

Nr. 472

JEL Classification: C14, C51

Key Words: B-spline, truncated power basis, derivative, monotonicity, penalty, smoothing spline, R.

* University of Regensburg, Faculty of Business, Economics and Management Information Systems, Department of Economics and Econometrics, Chari of Econometrics, 93040 Regensburg, Germany
E-mail: Kathrin.Kagerer[at]wiwi.uni-regensburg.de

A short introduction to splines in least squares regression analysis

Kathrin Kagerer*

March 28, 2013

Summary. Splines are an attractive way of flexibly modeling a regression curve since their basis functions can be included like ordinary covariates in regression settings. An overview of least squares regression using splines is presented including many graphical illustrations and comprehensive examples. Starting from two bases that are widely used for constructing splines, three different variants of splines are discussed: simple regression splines, penalized splines and smoothing splines. Further, restrictions such as monotonicity constraints are considered. The presented spline variants are illustrated and compared in a bivariate and a multivariate example with well-known data sets. A brief computational guide for practitioners using the open-source software `R` is given.

Key words. B-spline, truncated power basis, derivative, monotonicity, penalty, smoothing spline, `R`.

*University of Regensburg, Faculty of Business, Economics and Management Information Systems, Department of Economics and Econometrics, Chair of Econometrics, Universitätsstr. 31, 93053 Regensburg, Germany. Email: Kathrin.Kagerer@wiwi.uni-regensburg.de.

1. Introduction

> *Systema [...] maxime probabile valorum incognitarum [...] id erit, in quo quadrata differentiarum inter [...] valores observatos et computatos summam minimam efficiunt.*
>
> [...] that will be the most probable system of values of the unknown quantities [...] in which the sum of the squares of the differences between the observed and computed values [...] is a minimum.

This insight still is of crucial interest more than 200 years after Carl Friedrich Gauss stated it in 1809 (Gauss, 1809, p. 245, translation from Davis, 1857, p. 260). The method of least squares is historically used to describe the course of planets by Gauss (1809) and Legendre (1805) who independently suggested the same method. Today there are many more fields that apply the method of least squares in regression analysis. Among these are geography, biology/medicine and economics.

Classical linear least squares regression can be applied to quantify the change of the expected outcome of the response y given x_1 and potential other factors $x_2, \ldots, x_q$ when x_1 varies by some amount while the other covariates $x_2, \ldots, x_q$ are held fixed. Accordingly, the expected value of y given the covariates $x_1, \ldots, x_q$, $\mathrm{E}(y|x_1, \ldots, x_q)$, is a function of the covariates, that is, it can be expressed as

$$\mathrm{E}(y|x_1, \ldots, x_q) = f(x_1, \ldots, x_q), \tag{1}$$

where f is a (unknown) function that describes the relationship between $\mathrm{E}(y|x_1, \ldots, x_q)$ and the covariates $x_1, \ldots, x_q$. Hence, the relationship between y and $f(x_1, \ldots, x_q)$ is given by

$$y = f(x_1, \ldots, x_q) + u, \tag{2}$$

where $\mathrm{E}(u|x_1, \ldots, x_q) = 0$.

An often applied choice when estimating the function f, is to assume a linear relationship between $\mathrm{E}(y|x_1, \ldots, x_q)$ and the covariates $x_1, \ldots, x_q$. That is, the functional form f is a linear combination of the covariates,

$$f(x_1, \ldots, x_q) = \beta_0 + \beta_1 x_1 + \ldots + \beta_q x_q, \tag{3}$$

where $\beta_0, \ldots, \beta_q$ are unknown parameters that need to be estimated. For a given sample $i = 1, \ldots, n$ (with $n \geq q + 1$) the parameters may be estimated by solving

$$\min_{\tilde{\beta}_0, \ldots, \tilde{\beta}_q \in \mathbb{R}} \sum_{i=1}^{n} \Big(y_i - (\tilde{\beta}_0 + \tilde{\beta}_1 \, x_{i1} + \ldots + \tilde{\beta}_q \, x_{iq}) \Big)^2 .$$

Hence, the estimates for the parameters $\beta_0, \ldots, \beta_q$ are those that minimize the sum of squared differences between the observed values y_i and the computed values $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 \, x_{i1} + \ldots + \hat{\beta}_q \, x_{iq}$. That is, the parameters are estimated by applying Gauss' method of least squares.

To allow for other functional forms, $\mathrm{E}(y|x_1, \ldots, x_q)$ could, for example, be expressed as a linear combination of higher-order polynomials or other transformations of the covariates. This implies a possibly extensive specification search for f. To avoid this, non- or semiparametric specifications for f can be applied. Spline regression is one of such methods. A great advantage of spline regression compared to other non- and semiparametric methods is that it fits within the class of linear regressions, i.e. the functional form of f is linear in the parameters, and hence the computational costs for estimating the corresponding regression curve are not high since a closed form solution exists. Haupt et al. (2011) compare the performance of linear, spline and kernel estimations.

To keep the following explanations on spline regression simple, consider a bivariate relationship between one covariate x and the response y (unless stated differently). Then, Equation (1) simplifies to

$$\mathrm{E}(y|x) = f(x). \tag{4}$$

If f is incorrectly specified for the estimation, several conclusions (e.g. interpretations of marginal effects and hypothesis tests) that build on a correctly specified model are invalid. Hence, a correct specification of f is crucial.

The remainder of the paper is organized as follows: Section 2 describes how splines can be built from basis functions without referring to regression analysis. The latter is done in Section 3 where the specification details for least squares regressions are discussed. In Section 4 computational aspects concerning the open-source software `R` are addressed. Bivariate and multivariate examples are presented in Section 5 and the different variants of spline regression from Section 3 are illustrated and empirically compared. Finally, Section 6 summarizes and concludes.

2. Basis functions for Splines

Truncated power basis and splines Piecewise polynomial functions constitute an easy approach to adopt a flexible functional form without being demanding to implement. These functions can be generated in different ways. An intuitive way to understand the main principle is to consider the truncated power basis (see for example Ruppert et al., 2003, ch. 3, Dierckx, 1993, ch. 1.1, de Boor, 2001, ch. VIII as general references). Consider a straight line on some interval $[\kappa_0, \kappa_{m+1}]$ (e.g. $[\kappa_0, \kappa_{m+1}] = [\min_i(x_i), \max_i(x_i)]$ for a sample $i = 1, \ldots, n$) with a kink at some position κ_1 where $\kappa_0 < \kappa_1 < \kappa_{m+1}$. It can be described by a weighted sum (i.e. a linear combination) of the basis functions 1, x and $(x - \kappa_1)_+$, where the truncation function

$$(x - \kappa_1)_+ = \begin{cases} x - \kappa_1 & \text{for } x \geq \kappa_1 \\ 0 & \text{else} \end{cases}$$

gives the positive part of $x - \kappa_1$. The reasoning for the basis functions 1, x and $(x - \kappa_1)_+$ is as follows: To obtain a straight line f that is folded at κ_1 but continuous there, this function f can be written as $\beta_0 + \beta_1 x$ for $x < \kappa_1$ and as $\beta_0' + (\beta_1 + \alpha_1) x$ for $x \geq \kappa_1$. That means, the slope is β_1 until $x = \kappa_1$ and from $x = \kappa_1$ on the slope is changed by α_1. As f is constrained to be continuous at κ_1, this requires $\beta_0 + \beta_1 \kappa_1 = \beta_0' + (\beta_1 + \alpha_1) \kappa_1$ to hold or equivalently $\beta_0' = \beta_0 - \alpha_1 \kappa_1$. Overall, f then is given by

$$\begin{aligned} f(x) &= (\beta_0 + \beta_1 x) \cdot I_{\{x<\kappa_1\}} + \left(\beta_0' + (\beta_1 + \alpha_1) x\right) \cdot I_{\{x \geq \kappa_1\}} \\ &= (\beta_0 + \beta_1 x) \cdot I_{\{x<\kappa_1\}} + (\beta_0 + \beta_1 x + \alpha_1 (x - \kappa_1)) \cdot I_{\{x \geq \kappa_1\}} \\ &= \beta_0 + \beta_1 x + \alpha_1 (x - \kappa_1) I_{\{x \geq \kappa_1\}} \\ &= \beta_0 + \beta_1 x + \alpha_1 (x - \kappa_1)_+ \end{aligned}$$

where $I_{\{A\}}$ is the indicator function which is 1 if A holds and 0 else. That is, f can be written as a linear combination of the basis functions 1, x and $(x - \kappa_1)_+$.

Note that linear least squares regression with a constant and one covariate x has the two basis functions 1 and x and that for example an additional quadratic term leads to the additional basis function x^2.

Analogously to the line folded only at κ_1, a line folded m times at the positions $\kappa_1, \ldots, \kappa_m$ can be written as the weighted sum of the basis functions 1, x, $(x-\kappa_1)_+, \ldots, (x-$

$\kappa_m)_+$, where the κ_j's are called knots. With a proper choice of the knots κ_j, the functions generated from this basis can approximate other functions rather well. However, it yields a curve with sharp kinks that is not differentiable at the knots. These sharp kinks as well as the lacking differentiability are often undesirable. Smoother curves can be obtained by using higher powers of the basis functions. The respective basis of degree $p \geq 0$ consists of the functions $1, x, \ldots, x^p$, $(x-\kappa_1)_+^p, \ldots, (x-\kappa_m)_+^p$, where $(x-\kappa_j)_+^p := ((x-\kappa_j)_+)^p$ and $0^0 := 0$, and is called truncated power basis of degree p. The truncated power function $(x-\kappa_j)_+^p$ is $(p-1)$-times continuously differentiable at κ_j. Hence, also the linear combinations (called splines, Ruppert et al., 2003, p. 62) of the truncated power basis functions are $(p-1)$-times continuously differentiable at the knots κ_j, $j = 1, \ldots, m$.

Additionally including lower powers ($< p$) of $(x-\kappa_j)_+$ in the basis, changes the differentiability properties. That is, if the basis consists for example of the functions $1, x, \ldots, x^p$, $(x-\kappa_1)_+^p, \ldots, (x-\kappa_1)_+^{p-m_1+1}$, $(x-\kappa_2)_+^p, \ldots, (x-\kappa_m)_+^p$, the resulting linear combinations are $(p-m_1)$-times continuously differentiable at κ_1 and $(p-1)$-times continuously differentiable at $\kappa_2, \ldots, \kappa_m$, where m_1 can be regarded as multiplicity of the knot κ_1 (e.g. Eubank, 1984, p. 447f.). For $m_j = p+1$, functions constructed as linear combination from the truncated power basis functions of degree p have a discontinuity/jump at κ_j. If $m_j > 1$, the respective knots and basis functions are denoted as $(x-\kappa_j)_+^p, \ldots, (x-\kappa_{j+m_j-1})_+^{p-m_1+1}$, where $\kappa_j = \ldots = \kappa_{j+m_j-1}$. That is, the knot κ_j has multiplicity m_j (and so have $\kappa_{j+1}, \ldots, \kappa_{j+m_j-1}$ where $m_j = \ldots = m_{j+m_j-1}$). This notation is consistent with the notation for the equivalent B-spline basis which is described later on. Further, the truncated power basis of degree p thus always consists of $p+1+m$ basis functions that are identified once the knot sequence is given.

The panels in the upper row of Figure 1 show the functions of the above described bases on $[0;1]$: the basis for linear functions (1, x), the basis for cubic functions (1, x, x^2, x^3), the truncated power basis of degree 1 with one knot at 0.4 (1, x, $(x-0.4)_+$), the truncated power basis of degree 1 with four knots at 0.2, ..., 0.8 (1, x, $(x-0.2)_+$, ..., $(x-0.8)_+$) and the truncated power basis of degree 3 with four knots at 0.2, ..., 0.8 (1, x, x^2, x^3, $(x-0.2)_+^3$, ..., $(x-0.8)_+^3$). Additionally, the lower row shows an arbitrary example for a linear combination of the basis functions for each of the illustrated bases.

A disadvantage of the truncated power basis is that the basis functions are correlated and hence estimation results are often numerically instable (Ruppert et al., 2003, p. 70).

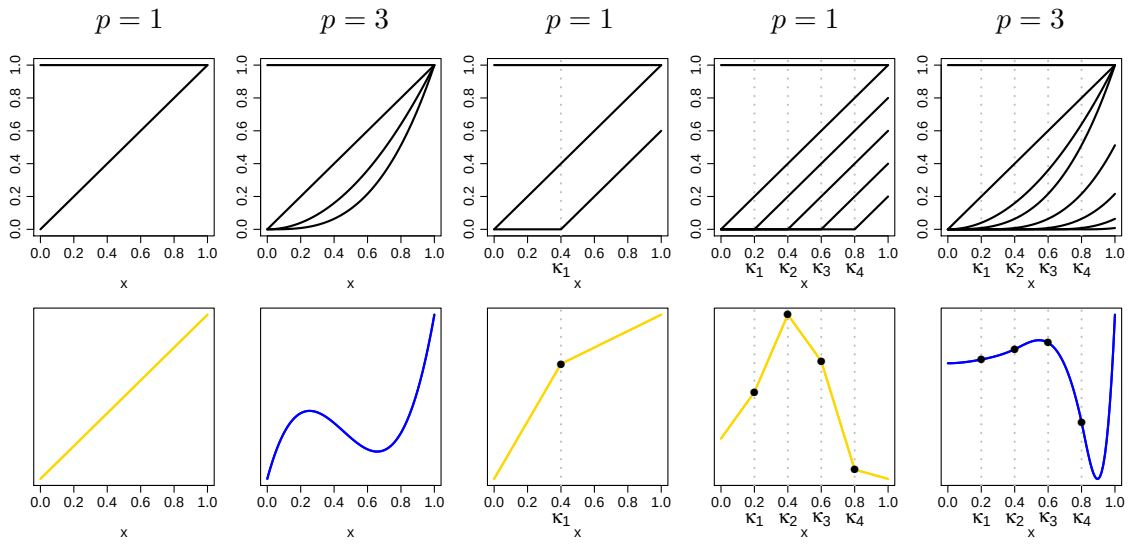

Figure 1: **Top**: different bases (from the left): basis functions for straight line, for cubic polynomial, for straight line with one kink, for straight line with four kinks, for cubic polynomial with four kinks. **Bottom**: arbitrary examples for linear combinations of the basis functions from the corresponding upper panel.

An equivalent basis that does not have this problem (Dierckx, 1993, p. 5, Ruppert et al., 2003, p. 70, de Boor, 2001, p. 85f.) and leads (apart from computational accuracy) to the same fit on $[\kappa_0, \kappa_{m+1}]$ (Ruppert et al., 2003, p. 70) is the B-spline basis of order $k = p + 1 \geq 1$ where p is the degree of the truncated power basis. In the following, B-splines are introduced.

B-spline basis and splines In a nutshell, the functions from the B-spline basis are piecewise polynomial functions of order k that are connected at the knots and have only small support. Then, the spline of order k, which is a linear combination of the basis functions, is also a piecewise polynomial function of order k. On the interval $[\kappa_0, \kappa_{m+1}]$, it exhibits the same properties as the respective linear combination of the functions from the truncated power basis of degree $p = k - 1$ with the same knots $\kappa_1, \ldots, \kappa_m$. A short review concerning B-splines can be found e.g. in the work of Eilers & Marx (1996) who summarize the definition and properties of B-splines while de Boor (2001), Dierckx (1993) and Ruppert et al. (2003) provide a more extensive discussion of splines.

To derive the B-spline basis of order k, some definitions are necessary. Let $\boldsymbol{\kappa} = (\kappa_{-(k-1)}, \ldots, \kappa_{m+k})$ be a non-decreasing sequence of knots (i.e. $\kappa_{-(k-1)} \leq \ldots \leq \kappa_{m+k}$),

where at most k adjacent knots coincide (i.e. $\kappa_j \neq \kappa_{j+k}$). The two boundary knots κ_0 and κ_{m+1} define the interval of interest and the m knots $\kappa_1, \ldots, \kappa_m$ are called inner knots. The remaining $2\,(k-1)$ exterior knots $\kappa_{-(k-1)}, \ldots, \kappa_{-1}$ and $\kappa_{m+2}, \ldots, \kappa_{m+k}$ are required to ensure regular behavior on the interval $[\kappa_0, \kappa_{m+1}]$. The B-spline basis functions (also called B-splines) are denoted as $B_j^{\boldsymbol{\kappa},k}$, $j = -(k-1), \ldots, m$.

B-splines can be motivated in different ways. One of them is to derive them by using divided differences (for a definition of divided differences and the corresponding representation of B-splines see for example de Boor, 2001, p. 3, 87). Another way that does not involve the definition of divided differences and can be shown to lead to the same result (cf. de Boor, 2001, p. 88) is to recursively calculate the B-splines of order $k > 1$ from the B-splines of lower order using the recurrence relation from de Boor (2001, p. 90)

$$B_j^{\boldsymbol{\kappa},k}(x) = \frac{x - \kappa_j}{\kappa_{j+k-1} - \kappa_j} B_j^{\boldsymbol{\kappa},k-1}(x) \; - \; \frac{x - \kappa_{j+k}}{\kappa_{j+k} - \kappa_{j+1}} B_{j+1}^{\boldsymbol{\kappa},k-1}(x),$$

where

$$B_j^{\boldsymbol{\kappa},1}(x) = (\kappa_{j+1} - x)_+^0 - (\kappa_j - x)_+^0 = \begin{cases} 1 & \text{for } \kappa_j \leq x < \kappa_{j+1} \\ 0 & \text{else} \end{cases}$$

is the B-spline of order 1 (de Boor, 2001, p. 89 and respective erratum) and the index j runs from $-(k-1)$ to m. To obtain the properties of B-splines on the complete interval $[\kappa_0, \kappa_{m+1}]$ but not only on $[\kappa_0, \kappa_{m+1})$, the definition of $B_m^{\boldsymbol{\kappa},1}$ and $B_{m+1}^{\boldsymbol{\kappa},1}$ is modified such that $B_m^{\boldsymbol{\kappa},1}(x) = 1$ for $x = \kappa_{m+1}$ and $B_{m+1}^{\boldsymbol{\kappa},1}(x) = 0$ for $x = \kappa_{m+1}$ (cf. de Boor, 2001, p. 94).

For equidistant knots, that is $\Delta\kappa_j = \kappa_j - \kappa_{j-1} =: h$ for $j = -(k-1)+1, \ldots, m+k$, it can be shown (based on an extension of Problem 2 from de Boor, 2001, p. 106) that the function $B_j^{\boldsymbol{\kappa},k}$ can also be written as

$$B_j^{\boldsymbol{\kappa},k}(x) = \frac{1}{h^{k-1}(k-1)!}\, \Delta^k (\kappa_{j+k} - x)_+^{k-1}$$

where $\Delta^k := \Delta(\Delta^{k-1})$.

Figure 2 gives examples for B-spline bases of different orders k for an equidistant knot sequence $\boldsymbol{\kappa}$ with $m = 3$ inner knots. In the first row, the basis functions $B_j^{\boldsymbol{\kappa},k}$, $j = -(k-1), \ldots, m$, and their sum are shown. Further details of Figure 2 are given in the following when the respective features are explained.

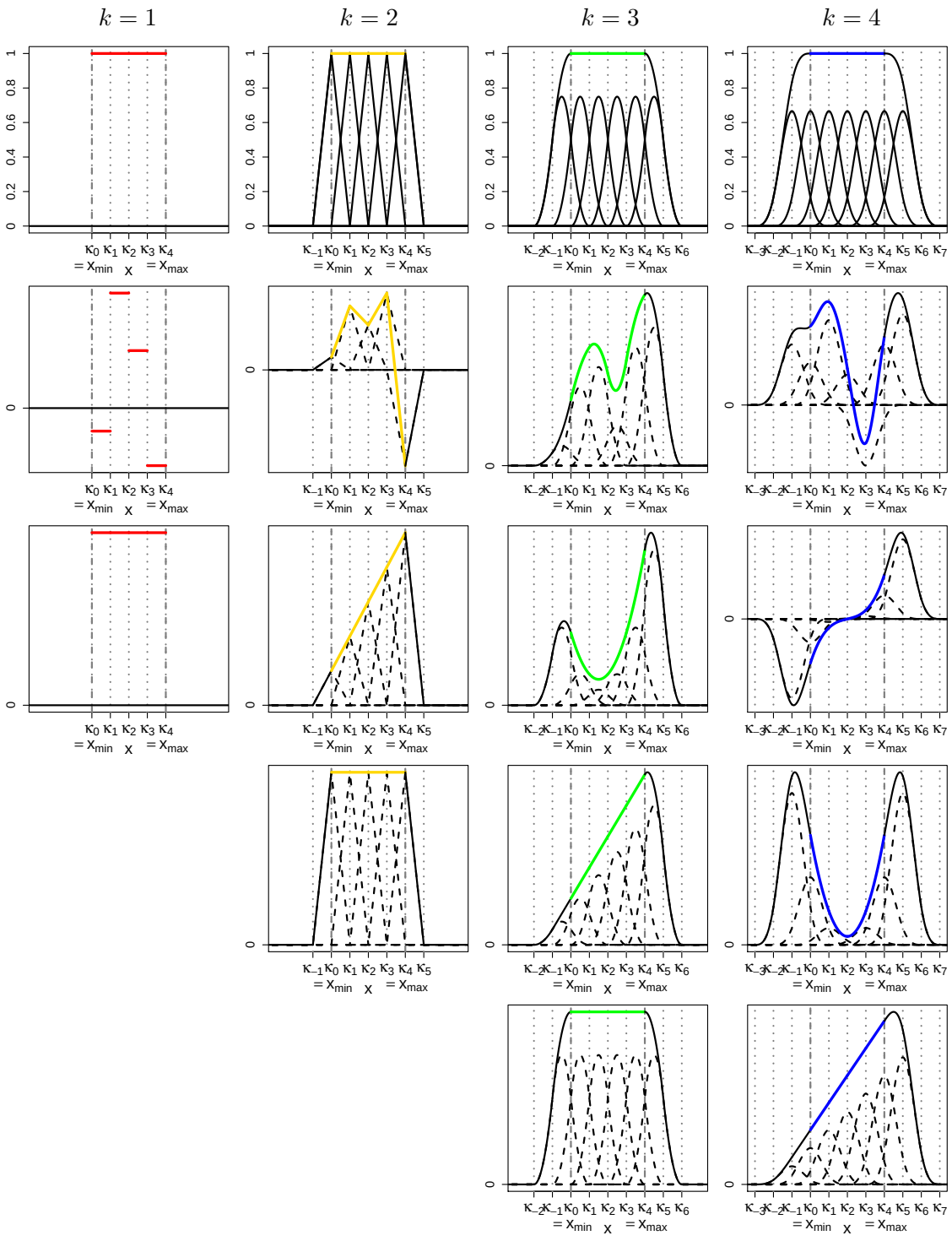

Figure 2: B-spline bases, sums and linear combinations for different orders k with equidistant knots and $m = 3$. **First row**: B-spline basis functions of orders 1, 2, 3 and 4, and their sum, which is 1 on $[\kappa_0, \kappa_{m+1}]$. **Second row**: arbitrarily weighted B-spline basis functions of orders 1, 2, 3 and 4, and their sum. **Third row**: B-spline basis functions of order 1, 2, 3 and 4, weighted such that their sum is a polynomial of degree $k-1$ on $[\kappa_0, \kappa_{m+1}]$. **Fourth row**: B-spline basis functions of orders 2, 3 and 4, weighted such that their sum is a polynomial of degree $k-2$ on $[\kappa_0, \kappa_{m+1}]$. **Fifth row**: B-spline basis functions of orders 3 and 4, weighted such that their sum is a polynomial of degree $k-3$ on $[\kappa_0, \kappa_{m+1}]$.

B-splines have several useful properties. Firstly, they form a partition of unity on the interval $[\kappa_0, \kappa_{m+1}]$, that is, on $[\kappa_0, \kappa_{m+1}]$ it holds that

$$\sum_{j=-(k-1)}^{m} B_j^{\boldsymbol{\kappa},k}(x) = 1 \tag{5}$$

(de Boor, 2001, p. 96). This can be observed in the first row of Figure 2. Moreover, each of the basis functions has only small support. More precisely, the support of the function $B_j^{\boldsymbol{\kappa},k}$ is the interval (κ_j, κ_{j+k}) (de Boor, 2001, p. 91), hence $B_j^{\boldsymbol{\kappa},k}(x) \cdot B_{j+d}^{\boldsymbol{\kappa},k}(x)$ is zero for $|d| \geq k$, which is the reason for the numerical stability mentioned on page 6. Further, the B-splines are up to $(k - m_j - 1)$-times continuously differentiable at the knot κ_j and the $(k - m_j)$th derivative has a jump at κ_j where m_j is the multiplicity of the knot κ_j (i.e. $\kappa_j = \ldots, \kappa_{j+m_j-1}$) (Dierckx, 1993, p. 9, de Boor, 2001, p. 99). This property carries over to linear combinations of the basis functions (called spline, de Boor, 2001, p. 93), that is, the functions

$$B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}(x) = \sum_{j=-(k-1)}^{m} \alpha_j B_j^{\boldsymbol{\kappa},k}(x), \tag{6}$$

where $\boldsymbol{\alpha} = \begin{pmatrix} \alpha_{-(k-1)} & \ldots & \alpha_m \end{pmatrix}'$, are also $(k - m_j - 1)$-times continuously differentiable at the knot κ_j. The second row of Figure 2 shows examples for linear combinations of the basis functions from the first row with arbitrarily chosen α_j.

The linear combinations of the B-spline basis functions of order k can generate all polynomial functions (in contrast to piecewise polynomial functions) of degree smaller than k on the interval $[\kappa_0, \kappa_{m+1}]$. Note that this is also a spline/polynomial of order k. This justifies the use of the notation order instead of degree since all polynomials of degree $< k$ are polynomials of order k. In the third (fourth, fifth) row of Figure 2, examples for polynomials of degree $k - 1$ ($k - 2$, $k - 3$) are given for $k \geq 1$ ($k \geq 2$, $k \geq 3$).

The first two rows of Figure 3 show B-spline bases with $k = 2, 3, 4$ where two knots of the knot sequence $\boldsymbol{\kappa}$ coincide. In the first row, $\alpha_j = 1$ for all j, $j = -(k-1), \ldots, m$, and in the second row, the α_j are chosen arbitrarily. For $k = 2$ the resulting spline now has a jump where the double knot is placed. For $k = 3$ the spline is still continuous, but its derivative is not, and for $k = 4$ the first derivative is also continuous, but the second derivative is not. For $k = 1$ no graphic is presented since twofold knots with $\kappa_j = \kappa_{j+1}$ do not make sense in this case, because $B_j^{\boldsymbol{\kappa},1}$ would be zero (de Boor, 2001, p. 89) and could

be excluded from the basis. The third row of Figure 3 shows weighted B-spline bases and the respective resulting splines for $k = 3$ and $k = 4$ where $\kappa_j = \kappa_{j+1} = \kappa_{j+2}$ for some j (i.e. $m_j = 3$). Then the spline of order 3 has a jump at κ_j and the spline of order 4 is continuous. For $k = 2$ one of the threefold knots is meaningless (as discussed for $k = 1$ and twofold knots). In the last line of Figure 3, a spline of order 4 is shown that has a jump at a fourfold knot position ($m_j = 4$ there).

To contrast equidistant knot sequences to those with non-equidistant knots, Figure 4 exemplary shows $B_j^{\boldsymbol{\kappa},k}$, $j = -(k-1), \ldots, m$, and $\sum_{j=-(k-1)}^{m} B_j^{\boldsymbol{\kappa},k}(x)$ for a non-equidistant knot sequence. This is analogous to the first row of Figure 2 with the only difference that the knots are not equidistant and hence the basis functions $B_j^{\boldsymbol{\kappa},k}$ look different. But still they sum up to unity on $[\kappa_0, \kappa_{m+1}]$.

Derivative and monotonicity The first derivative of the B-spline functions is

$$\frac{\partial B_j^{\boldsymbol{\kappa},k}(x)}{\partial x} = \frac{k-1}{\kappa_{j+k-1} - \kappa_j} B_j^{\boldsymbol{\kappa},k-1}(x) - \frac{k-1}{\kappa_{j+k} - \kappa_{j+1}} B_{j+1}^{\boldsymbol{\kappa},k-1}(x) \tag{7}$$

for $k > 1$ (de Boor, 2001, p. 115). For $k = 1$ it is defined to be 0 according to the argumentation in de Boor (2001, p. 117). Hence, the first derivative of a B-spline of order k is a spline of order $k-1$ since it is a linear combination of B-splines of order $k-1$. From Equation (7) it can be shown that the first derivative of a spline as linear combination of the B-spline basis functions is given by

$$\frac{\partial B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}(x)}{\partial x} = \frac{\partial}{\partial x} \sum_{j=-(k-1)}^{m} \alpha_j B_j^{\boldsymbol{\kappa},k}(x) = (k-1) \sum_{j=-(k-1)}^{m+1} \frac{\alpha_j - \alpha_{j-1}}{\kappa_{j+k-1} - \kappa_j} B_j^{\boldsymbol{\kappa},k-1}(x) \tag{8}$$

where $\alpha_{-(k-1)-1} := 0 =: \alpha_{m+1}$ (de Boor, 2001, p. 116). On the interval $[\kappa_0, \kappa_{m+1}]$ it holds that $B_{-(k-1)}^{\boldsymbol{\kappa},k-1}(x) = B_{m+1}^{\boldsymbol{\kappa},k-1}(x) = 0$ and hence the summation reduces to

$$\frac{\partial B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}(x)}{\partial x} = (k-1) \sum_{j=-(k-1)+1}^{m} \frac{\alpha_j - \alpha_{j-1}}{\kappa_{j+k-1} - \kappa_j} B_j^{\boldsymbol{\kappa},k-1}(x) \tag{9}$$

on $[\kappa_0, \kappa_{m+1}]$. For equidistant knot sequences, Equations (7) and (8)/(9) simplify to

$$\frac{\partial B_j^{\boldsymbol{\kappa},k}(x)}{\partial x} = \frac{1}{h} B_j^{\boldsymbol{\kappa},k-1}(x) - \frac{1}{h} B_{j+1}^{\boldsymbol{\kappa},k-1}(x)$$

and

$$\frac{\partial B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}(x)}{\partial x} = \frac{1}{h} \sum_{j=-(k-1)}^{m+1} (\alpha_j - \alpha_{j-1}) B_j^{\boldsymbol{\kappa},k-1}(x) = \frac{1}{h} \sum_{j=-(k-1)+1}^{m} (\alpha_j - \alpha_{j-1}) B_j^{\boldsymbol{\kappa},k-1}(x), \tag{10}$$

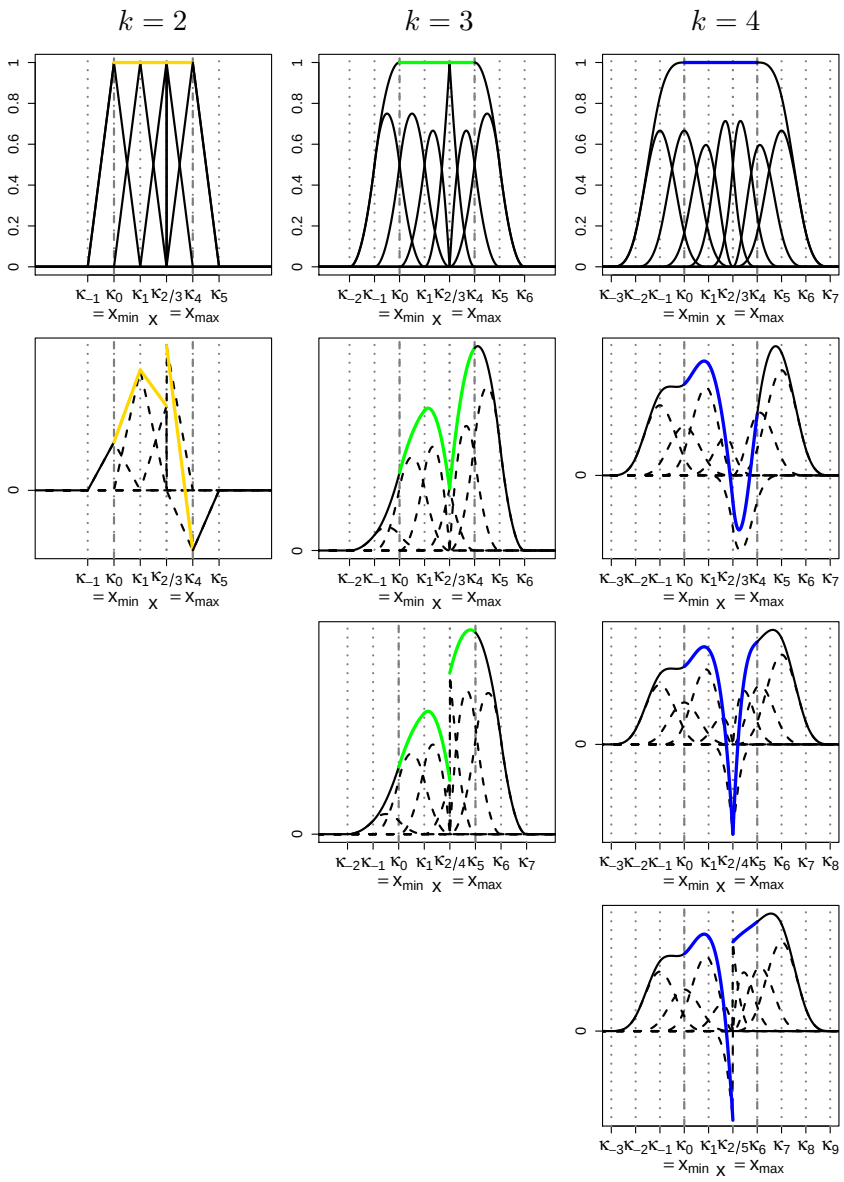

Figure 3: B-spline bases, sums and linear combinations for different orders k with equidistant knots where one knot position is multiply occupied. **First row**: B-spline basis functions of orders 2, 3 and 4, where $\kappa_2 = \kappa_3$, and their sum which is 1 on $[\kappa_0, \kappa_{m+1}]$. **Second row**: arbitrarily weighted B-spline basis functions of orders 2, 3 and 4, where $\kappa_2 = \kappa_3$, and their sum. **Third row**: arbitrarily weighted B-spline basis functions of orders 3 and 4, where $\kappa_2 = \kappa_3 = \kappa_4$, and their sum. **Fourth row**: arbitrarily weighted B-spline basis functions of order 4, where $\kappa_2 = \kappa_3 = \kappa_4 = \kappa_5$, and their sum.

respectively, on $[\kappa_0, \kappa_{m+1}]$. Higher order derivatives can also be calculated from Equations (7) or (8).

Since all of the terms $k-1$, $\kappa_{j+k-1} - \kappa_j$ and $B_j^{\boldsymbol{\kappa},k-1}(x)$, $j = -(k-1)+1, \ldots, m$ are

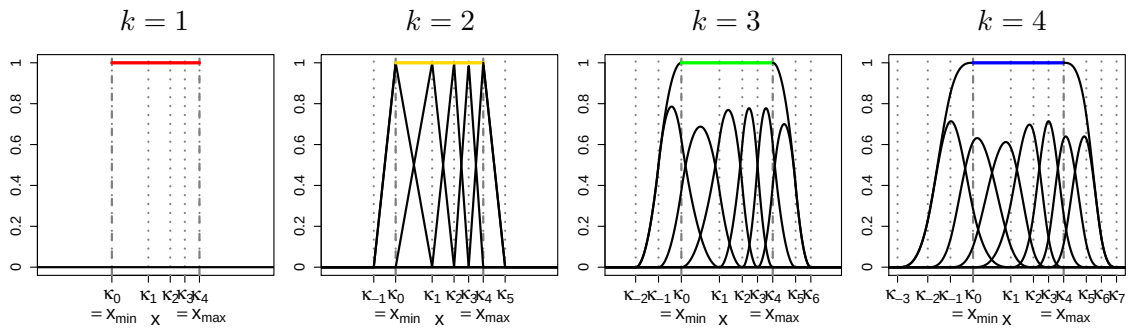

Figure 4: B-spline basis functions of orders 1, 2, 3 and 4 with non-equidistant knots ($m = 3$), and their sum which is 1 on $[\kappa_0, \kappa_{m+1}]$.

greater or equal to zero (de Boor, 2001, p. 91), the sign of $\frac{\partial B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}(x)}{\partial x}$ only depends on the differences $\alpha_j - \alpha_{j-1} =: \delta_j$, $j = -(k-1)+1, \ldots, m$. It can be seen from Equation (8) that

$$\alpha_j \geq \alpha_{j-1} \quad \text{(i.e. } \delta_j \geq 0), \quad j = -(k-1)+1, \ldots, m, \tag{11}$$

ensures a completely non-negative first derivative of $B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}$, and hence $B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}$ is monotonically increasing. Analogously, $B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}$ is monotonically decreasing if

$$\alpha_j \leq \alpha_{j-1} \quad \text{(i.e. } \delta_j \leq 0), \quad j = -(k-1)+1, \ldots, m. \tag{12}$$

If $\alpha_j \geq \alpha_{j-1}$, $j = -(k-1)+1, \ldots, m$, holds (where $\alpha_j \neq 0$ for at least one j), it follows that $\alpha_{-(k-1)} \not\geq \alpha_{-(k-1)-1}$ or $\alpha_{m+1} \not\geq \alpha_m$ since the auxiliary parameters $\alpha_{-(k-1)-1}$ and α_{m+1} from Equation (8) are zero. This implies that the spline is only monotonically increasing on the interval $[\kappa_0, \kappa_{m+1}]$ (for an example, see Figure 5). Hence, the derivative (9) where the sum is indexed from $j = -(k-1)+1$ to $j = m$ and not (8) with $j = -(k+1), \ldots, m+1$ has to be regarded. For $\alpha_j \leq \alpha_{j-1}$ these considerations apply analogously.

Equations (11) and (12) can also be written in matrix notation as $\mathbf{C}\,\boldsymbol{\alpha} \geq \mathbf{0}$ with

$$\mathbf{C} = \begin{pmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \\ & & & \ddots & \ddots \end{pmatrix} \qquad \text{and} \qquad \mathbf{C} = \begin{pmatrix} 1 & -1 & & \\ & 1 & -1 & \\ & & 1 & -1 \\ & & & \ddots & \ddots \end{pmatrix}, \tag{13}$$

respectively.

Trivially, for $k = 1$ the conditions (11) and (12) are each necessary and sufficient conditions for monotonicity. For $k > 1$ this issue is illustrated in Figures 5 and 6 for equidistant knot sequences but the same reasoning holds for non-equidistant knot sequences. The upper rows of both figures show splines $B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}$ and the underlying basis

functions $B_j^{\boldsymbol{\kappa},k}$ weighted by α_j while the lower rows picture the respective derivatives $\frac{1}{h}\sum_{j=-(k-1)}^{m+1}(\alpha_j - \alpha_{j-1})B_j^{\boldsymbol{\kappa},k-1}(x) = \frac{1}{h}\sum_{j=-(k-1)}^{m+1}\delta_j B_j^{\boldsymbol{\kappa},k-1}(x)$ and the underlying basis functions $B_j^{\boldsymbol{\kappa},k-1}$ weighted by δ_j. Figure 5 gives an example where $\alpha_j \geq \alpha_{j-1}$ (i.e. $\delta_j \geq 0$) holds for all $j = -(k-1)+1, \ldots, m$. Hence all splines in the upper row are monotonically increasing on the interval $[\kappa_0, \kappa_{m+1}]$. In contrast, in Figure 6 the condition $\delta_j \geq 0$ is hurt for some j. For $k = 2$ and $k = 3$ this implies a derivative that is negative within a certain interval and hence the spline is non-monotone on $[\kappa_0, \kappa_{m+1}]$. But for $k = 4$ (and also for $k \geq 5$ what is not illustrated here) some combinations of α_j exist where the respective spline $B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}$ is monotonically increasing on $[\kappa_0, \kappa_{m+1}]$ even if δ_j is negative for some j. That is, for $k \leq 3$ the conditions (11) and (12) are each necessary and sufficient for monotonicity and for $k \geq 4$ they are each sufficient but not necessary for monotonicity. A compendious discussion can also be found in Dierckx (1993, sec. 7.1).

3. Spline regression

Spline specification For a given order k and a knot sequence $\boldsymbol{\kappa}$, the regression function f for the estimation of $\mathrm{E}(y|x) = f(x)$ in (4) can be specified as

$$f(x) = \sum_{j=-(k-1)}^{m} \alpha_j B_j^{\boldsymbol{\kappa},k}(x), \tag{14}$$

where the parameters α_j, $j = -(k-1), \ldots, m$, have to be estimated for a given sample (y_i, x_i), $i = 1, \ldots, n$ (with $n \geq m + k$). Hence, the regression function is easy to estimate using least squares regression since it it linear in its parameters and it is flexible since it can generate all piecewise polynomial functions of order k with differentiability depending on the knot sequence. Piecewise polynomial functions can well approximate quite arbitrary functions. This can be justified by Weierstrass' approximation theorem (cf. Mackenzie et al., 2005, p. 396) applied to pieces of f defined by two neighboring knots.

The task of the researcher when applying splines for regression purposes is to specify the order k of the spline and the knot sequence $\boldsymbol{\kappa}$ (i.e. the number and position of the knots) and if necessary impose some restrictions and/or penalties on the parameters to estimate. Several approaches concerning these issues as well as how to estimate the parameters α_j, $j = -(k-1), \ldots, m$, are presented in this section.

The different approaches of splines estimation are regression splines, penalized splines and smoothing splines (e.g. Cao et al., 2010, p. 892, Eilers & Marx, 1996, p. 89). The term

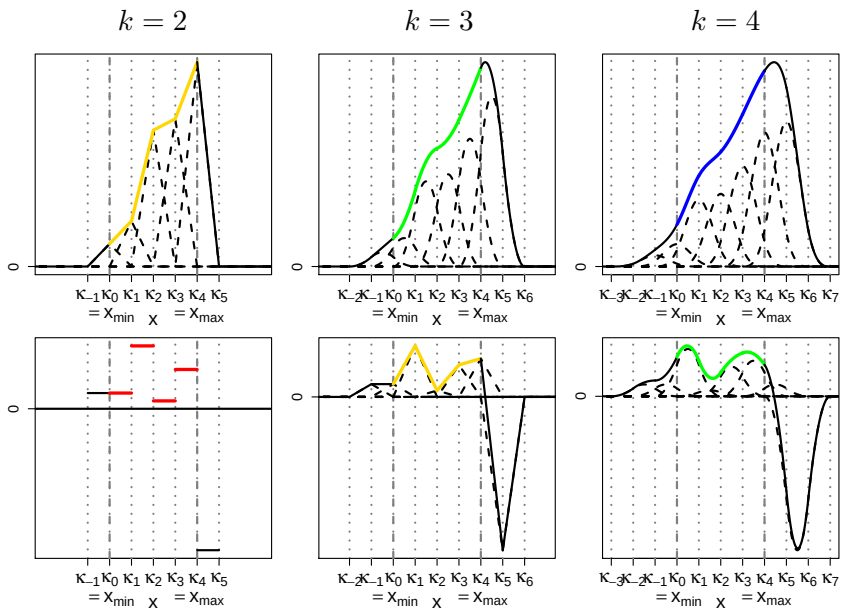

Figure 5: **Top**: monotonically weighted B-spline basis functions of orders 2, 3 and 4, and their sum. **Bottom**: first derivative of the spline from the top with underlying respectively weighted B-spline basis functions.

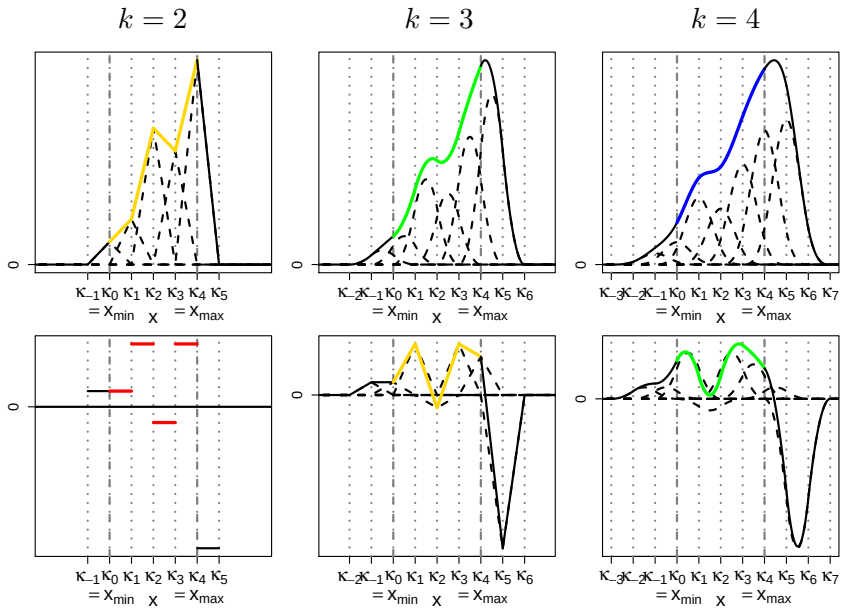

Figure 6: **Top**: apart from one time monotonically weighted B-spline basis functions of orders 2, 3 and 4, and their sum. **Bottom**: first derivative of the spline from the top with underlying respectively weighted B-spline basis functions.

regression splines denotes splines as in Equation (6) in the regression context. Penalized splines are regression splines with an additional penalty on the parameters. These penalties

are detailed further down. Finally, smoothing splines can be shown to be penalized splines with knots at all distinct sample values of the covariate x.

Order of the spline For regression splines and penalized splines, the order and the knot sequence have to be specified in advance of the estimation. Though Ruppert et al. (2003, p. 124f.) state that the order of the spline basis nearly does not matter as long as enough knots are used, there might exist some requirements to the resulting fitted spline function. For example, to construct a spline with continuous first derivative, at least the order $k = 3$ has to be chosen. Ruppert's (2002, p. 742) opinion is that $k = 3$ is enough and his experience shows that the results for $k = 3$ and $k = 4$ are usually similar. He & Shi (1998, p. 644) recommend to use $k = 3$ since then the monotonicity constraints (11) and (12) are "if and only if" constraints (cf. the explanation on page 12). Many studies use cubic splines which corresponds to $k = 4$. This is also the order that is recommended by Dierckx (1993, p. 45). He argues that splines of order $k = 4$ are computationally efficient and provide a good fit. Further, they allow the researcher to implement constraints on the parameters to guarantee monotonicity or convexity of the fitted regression curve (Dierckx, 1993, p. 119f.). According to Wegman & Wright (1983, p. 354), $k = 4$ is also the smallest order yielding visual smoothness. If more than two continuous derivatives are required, higher order splines with $k > 4$ are to be used (e.g. Dierckx, 1993, p. 45).

Knot sequence for regression splines In specifying the order of the spline basis, regression splines and penalized splines are treated alike. But this is different for specifying the knots. For regression splines the choice of the knots is very important. A trivial choice is to use equidistant knots or knots at equally spaced sample quantiles of x. Thus only the number of the knots or equivalently the number of the inner knots m has to be specified. For the choice of the latter, information criteria or cross-validation can be applied. For example Huang & Shen (2004), Huang et al. (2004) and Landajo et al. (2008) follow this approach. Alternatively, a rule of thumb can be applied. Asymptotic results for regression splines as for example given in the works of He & Shi (1998) or Huang et al. (2004) suggest

$$m \approx n^{1/5} - 1 \tag{15}$$

for cubic splines as a rule of thumb. Note that for knots at the equally spaced sample quantiles of x, the exterior knots usually are chosen to coincide with the boundary knots

since no obvious other positions exist (as opposed to equidistant knots). On $[\kappa_0, \kappa_{m+1}]$ the positions of the exterior knots do not matter for the resulting spline (though the basis functions, for which one of the boundary knots is included in the support, differ). For $\kappa_{-(k-1)} = \ldots = \kappa_0$ and $\kappa_{m+1} = \ldots = \kappa_{m+k}$, it holds that $B_j^{\boldsymbol{\kappa},k}(x) = B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}(x) = 0$ for $x \notin [\kappa_0, \kappa_{m+1}]$. This discussion analogously holds for other non-equidistant knot sequences.

For non-equidistant knots there exist many proposals to choose the knot sequence (i.e. the number and positions of the knots). On the one hand, knot selection algorithms based on information criteria can be applied. Then knots are stepwise deleted from or inserted to a given starting knot sequence which is usually equidistant or consists of sample quantiles of x. Examples for knot selection algorithms can be found in He & Shi (1998), Lee (2000) and Wand (2000). On the other hand, the knot positions can be estimated together with the other parameters. Dierckx (1993, sec. 4.2), Eubank (1984, sec. 4) and Huang et al. (2004) give an overview.

Penalized splines The elaborate and often computationally intensive search for the knot sequence can be circumvented if a penalty term is added in the optimization. Then a rather long knot sequence can be chosen since the penalty term avoids a too rough fit of the estimated regression curve. Usually the knot sequence is taken to be equidistant or the knots are placed at equally spaced sample quantiles of x where quite many knots are contained in the knot sequence $\boldsymbol{\kappa}$. However, it is not clear whether equidistant knots or sample quantiles are to be preferred (cf. e.g. the discussion between Eilers & Marx, 2010 and Crainiceanu et al., 2007). Lu et al. (2009, p. 1064) find nearly no differences in their study. As a rule of thumb for the number of knots, about $\min(n/4, 35)$ knots (Ruppert, 2002, p. 753) or 20-40 knots (e.g. Ruppert et al., 2003, p. 125 or Lang & Brezger, 2004, p. 186) can be employed.

Instead of a penalty term depending on the (second) derivative of the fitted function as in the work of O'Sullivan (1986, 1988), Eilers & Marx (1996) propose to penalize high second-order differences (or differences of another order) of the estimated parameters and thus introduce P-splines as a computationally advantageous special case of penalized splines. The respective minimization problem is given by

$$\min_{\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^{m+k}} \sum_{i=1}^{n} \left(y_i - \sum_{j=-(k-1)}^{m} \tilde{\alpha}_j B_j^{\boldsymbol{\kappa},k}(x_i) \right)^2 + \lambda \sum_{j=-(k-1)+2}^{m} (\Delta^2 \tilde{\alpha}_j)^2 \tag{16}$$

for an equidistant knot sequence. The penalty term in (16) can also be stated in matrix notation as $\lambda\,\tilde{\boldsymbol{\alpha}}^T\,\mathbf{D}^T\mathbf{D}\,\tilde{\boldsymbol{\alpha}}$ with

$$\mathbf{D} = \begin{pmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & -2 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{D}^T\mathbf{D} = \begin{pmatrix} 1 & -2 & 1 & & & & & \\ -2 & 5 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & \cdot & & & \\ & 1 & -4 & 6 & \cdot & \cdot & & \\ & & 1 & -4 & \cdot & \cdot & 1 & \\ & & & 1 & \cdot & \cdot & -4 & 1 \\ & & & & \cdot & \cdot & 6 & -4 & 1 \\ & & & & & \cdot & -4 & 5 & -2 \\ & & & & & & 1 & -2 & 1 \end{pmatrix}$$

where $\mathbf{D}$ is a $((m+k-2)\times(m+k))$-matrix and $\mathbf{D}^T\mathbf{D}$ is a $((m+k)\times(m+k))$-matrix. The matrix $\mathbf{D}$ can be derived using $\Delta^2\tilde{\alpha}_j = \tilde{\alpha}_j - 2\,\tilde{\alpha}_{j-1} + \tilde{\alpha}_{j-2} = \begin{pmatrix} 0 & \dots & 0 & 1 & -2 & 1 & 0 & \dots & 0 \end{pmatrix}\tilde{\boldsymbol{\alpha}}$ with entries at the positions $j-2, j-1, j$, $\begin{pmatrix} \Delta^2\tilde{\alpha}_{-(k-1)+2} & \cdots & \Delta^2\tilde{\alpha}_m \end{pmatrix}^T = \mathbf{D}\,\tilde{\boldsymbol{\alpha}}$ and $\sum_{j=-(k-1)+2}^{m}(\Delta^2\tilde{\alpha}_j)^2 = \left|\left|\begin{pmatrix} \Delta^2\tilde{\alpha}_{-(k-1)+2} & \cdots & \Delta^2\tilde{\alpha}_m \end{pmatrix}^T\right|\right|^2 = \tilde{\boldsymbol{\alpha}}^T\,\mathbf{D}^T\mathbf{D}\,\tilde{\boldsymbol{\alpha}}$.

While for $\lambda = 0$ the fit from (16) corresponds to that from the unpenalized model which is potentially overfitting, for $\lambda \to \infty$ the fit turns into a straight line (Eilers & Marx, 1996, p. 93) which may be oversmoothed. Hence, the selection of the smoothing/penalty parameter λ is an important and demanding task since the estimation results are sensitive with respect to λ. Eilers & Marx (1996) suggest to use the Akaike information criterion or (generalized) cross-validation, but many other criteria are possible, too (e.g. Imoto & Konishi, 2003).

The term $\sum_{j=-(k-1)+2}^{m}(\Delta^2\tilde{\alpha}_j)^2$ of the penalty in Equation (16) is motivated by the second derivative of $\sum_{j=-(k-1)}^{m}\tilde{\alpha}_j B_j^{\boldsymbol{\kappa},k}(x)$ (with equidistant knots) which can be derived using Equation (10) and is given by

$$\frac{\partial^2 B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}(x)}{\partial x^2} = \frac{1}{h^2}\sum_{j=-(k-1)+2}^{m}\left(\Delta^2\alpha_j\right)B_j^{\boldsymbol{\kappa},k-2}(x).$$

Hence, the term $\sum_{j=-(k-1)+2}^{m}(\Delta^2\tilde{\alpha}_j)^2$ penalizes high second derivatives (which can also be interpreted as changes in the first derivative) of the fitted regression function. Analogously, the second derivative for non-equidistant knot sequences,

$$\frac{\partial^2 B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}(x)}{\partial x^2} = (k-1)(k-2)\sum_{j=-(k-1)+2}^{m}\frac{\frac{\alpha_j-\alpha_{j-1}}{\kappa_{j+k-1}-\kappa_j} - \frac{\alpha_{j-1}-\alpha_{j-2}}{\kappa_{j+k-2}-\kappa_{j-1}}}{\kappa_{j+k-2}-\kappa_j}\,B_j^{\boldsymbol{\kappa},k-2}(x)$$

can be used to formulate the respective term in the penalty as

$$\sum_{j=-(k-1)+2}^{m}\left(\frac{\frac{\Delta\alpha_j}{\kappa_{j+k-1}-\kappa_j} - \frac{\Delta\alpha_{j-1}}{\kappa_{j+k-2}-\kappa_{j-1}}}{\kappa_{j+k-2}-\kappa_j}\right)^2. \tag{17}$$

Applying (17) to equidistant knots results in

$$\sum_{j=-(k-1)+2}^{m} \left(\frac{\frac{\Delta\alpha_j}{(k-1)\,h} - \frac{\Delta\alpha_{j-1}}{(k-1)\,h}}{(k-2)\,h} \right)^2 = \frac{1}{(k-1)^2\,(k-2)^2\,h^4} \sum_{j=-(k-1)+2}^{m} (\Delta^2\alpha_j)^2.$$

To make sure that $\sum_{j=-(k-1)+2}^{m}(\Delta^2\alpha_j)^2$ results from (17) for equidistant knots, the constant factor $(k-1)^2\,(k-2)^2\,h_p^4$ with $h_p = \frac{\kappa_{m+1}-\kappa_0}{m+1}$ may optionally be multiplied to (17) since it does not influence the optimization process. The matrix $\mathbf{D}$ for non-equidistant knots is analogously constructed to the equidistant case using $\frac{\frac{\Delta\alpha_j}{\kappa_{j+k-1}-\kappa_j} - \frac{\Delta\alpha_{j-1}}{\kappa_{j+k-2}-\kappa_{j-1}}}{\kappa_{j+k-2}-\kappa_j} = \frac{1}{\kappa_{j+k-2}-\kappa_j}\begin{pmatrix} 0 & \cdots & 0 & \frac{1}{\kappa_{j+k-2}-\kappa_{j-1}} & -(\frac{1}{\kappa_{j+k-2}-\kappa_{j-1}} + \frac{1}{\kappa_{j+k-1}-\kappa_j}) & \frac{1}{\kappa_{j+k-1}-\kappa_j} & 0 & \cdots & 0 \end{pmatrix} \tilde{\boldsymbol{\alpha}}$ again with entries at the positions $j-2, j-1, j$. To include the correction factor, $\mathbf{D}$ has to be multiplied by $(k-1)\,(k-2)\,h_p^2$. Note that for equidistant knots it holds that $h_p = h$.

The minimization of the objective function in (16) (or analogously inserting (17) for non-equidistant knots) can be performed like in the case of an ordinary least squares problem what is presented in the following. Defining $\mathbf{B}$ to be the $(n \times (m+k))$-matrix with elements $B_{i,j'} = B^{\boldsymbol{\kappa},k}_{j'-k}(x_i)$, $j' = 1, \ldots, m+k$, $i = 1, \ldots, n$ and $\mathbf{y} = \begin{pmatrix} y_1 & \cdots & y_n \end{pmatrix}^T$, then

$$\sum_{i=1}^{n} u_i^2 = \sum_{i=1}^{n} \left(y_i - \sum_{j=-(k-1)}^{m} \alpha_j B_j^{\boldsymbol{\kappa},k}(x_i) \right)^2$$

can alternatively be formulated in matrix notation as

$$(\mathbf{y} - \mathbf{B}\,\boldsymbol{\alpha})^T\,(\mathbf{y} - \mathbf{B}\,\boldsymbol{\alpha}) = \mathbf{y}^T\mathbf{y} - 2\,\mathbf{y}^T\,\mathbf{B}\,\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\,\mathbf{B}^T\mathbf{B}\,\boldsymbol{\alpha}.$$

The penalty term is $\lambda\,\boldsymbol{\alpha}^T\,\mathbf{D}^T\mathbf{D}\,\boldsymbol{\alpha}$ as defined above for equidistant or non-equidistant knots. Hence, the complete minimization problem is given by

$$\min_{\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^{m+k}} \left(\mathbf{y}^T\mathbf{y} - 2\,\mathbf{y}^T\,\mathbf{B}\,\tilde{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\alpha}}^T\left(\mathbf{B}^T\mathbf{B} + \lambda\,\mathbf{D}^T\mathbf{D}\right)\tilde{\boldsymbol{\alpha}} \right). \tag{18}$$

From this minimization, the OLS-estimator $\hat{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha}$ results as

$$\hat{\boldsymbol{\alpha}} = \left(\mathbf{B}^T\mathbf{B} + \lambda\,\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{B}^T\mathbf{y}$$

and the corresponding hat matrix is $\mathbf{B}\left(\mathbf{B}^T\mathbf{B} + \lambda\,\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{B}^T$ (see also Eilers & Marx, 1996). An OLS-estimation with $((n+m+k)\times 1)$-response $\mathbf{y}^* = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix}$ and $((n+m+$

$k) \times (m + k)$)-regressor matrix $\mathbf{B}^* = \begin{pmatrix} \mathbf{B} \\ \sqrt{\lambda}\,\mathbf{D} \end{pmatrix}$ results in the exact same minimization problem (18), the same OLS-estimator $\hat{\boldsymbol{\alpha}}$ and the same hat matrix. Hence, penalized spline regression with difference penalty can easily be carried out as ordinary least squares regression (see also Eilers & Marx, 1996, p. 101).

Smoothing splines The selection of the order k and the knot sequence for smoothing splines is different compared to that for regression splines and penalized splines. Kimeldorf & Wahba (1970a,b) showed that $\hat{f}$ from the minimization of

$$\sum_{i=1}^{n} \left(y_i - \tilde{f}(x_i)\right)^2 + \lambda \int_{\min x_i}^{\max x_i} \left(\frac{\partial^\gamma \tilde{f}(x)}{\partial x^\gamma}\right)^2 dx \tag{19}$$

with respect to $\tilde{f}$ for some integer $\gamma > 0$ is a spline of order 2γ with possible knots at the distinct observations x_i (Wegman & Wright, 1983, p. 353). That is, the order (e.g. linear or cubic) of the smoothing spline results from the choice of γ and only splines with even order can be obtained from the minimization of (19), such as linear ($k = 2$) or cubic ($k = 4$) splines. Further, the knots do not have to be chosen since every (distinct) observation constitutes a knot. The smoothing parameter λ can be chosen as for penalized splines using several information criteria.

Hence, penalized splines are in-between regression splines and smoothing splines (e.g. Claeskens et al., 2009, p. 529). On the one hand, for $\lambda = 0$ penalized splines correspond to regression splines. If on the other hand the knots are chosen to be at the distinct values of x and the penalty is based on a derivative of the estimated spline, then penalized splines correspond to smoothing splines.

Note that in a regression analysis with only one covariate x, smoothing splines with $\lambda = 0$ can only be applied with order $k = 2$ (and $k = 1$ but by the above definition, smoothing splines always are of even order). This is due to the fact that $m+k$ parameters have to be estimated and only $m + 2$ distinct values of x are available for the estimation since distinct values of x constitute the two boundary and m inner knots. Hence for $k > 2$, the regressor matrix does not have full rank since the row rank is $m + 2$ which is smaller than the number of columns, $m + k$, in this case. Note that multiple observations at a knot have identical rows in the regression matrix and thus cannot increase the rank. In multiple regressions, the same problem occurs in the corresponding columns of the spline

component.

For $\lambda > 0$, the rank deficit usually can be avoided as for example described by Eilers & Marx (1996, p. 101) and Eilers & Marx (2010, p. 639) since the penalty term may be interpreted as additional observations leading to more different rows in the regressor matrix (see also the discussion in the above paragraph on penalized splines and the motorcycle example in Section 5).

Monotonicity For many applications a monotone relationship between x and y is assumed a priori. Applying the monotonicity constraint (11) or (12) for the parameter estimation, ensures a monotone fit. As already explained on page 13, for $k \geq 4$ condition (11) or (12) is not necessary for $B_{\boldsymbol{\alpha}}^{\boldsymbol{\kappa},k}$ to be monotone on $[\kappa_0, \kappa_{m+1}]$. Hence, Wood (1994) provides necessary conditions for monotonicity but these are not as easy to implement as constraining the parameters by (11) or (12). Another approach for applications with assumed monotone relationship is to use a second penalty term which penalizes derivations from a monotone fit (Bollaerts et al., 2006, p. 193). In addition, imposing a monotonicity constraint has a smoothing effect on the estimated regression function (Dierckx, 1993, p. 119). This can also be observed in the applications of Haupt et al. (2012).

B-spline basis and truncated power basis A basis which is equivalent to the B-spline basis of order k with knot sequence $\boldsymbol{\kappa}$ where $\kappa_{-(k-1)} < \ldots < \kappa_{m+k}$ and with basis functions $B_j^{\boldsymbol{\kappa},k}$, $j = -(k-1), \ldots, m$, is given by the truncated power basis of degree $k-1$ with basis functions $1, x, \ldots, x^{k-1}, (x-\kappa_1)_+^{k-1}, \ldots, (x-\kappa_m)_+^{k-1}$ (de Boor, 2001, ch. IX). For knot sequences with multiple knots, the respective truncated power basis functions are as described on page 5. Both bases have the same number of basis functions and hence the same number of parameters to estimate. Note that regression splines, penalized splines as well as smoothing splines can be generated from either of the bases where both bases have clear advantages and drawbacks. Eilers & Marx (2010) compare the B-spline basis and the truncated power basis. They clearly favor the B-spline basis, especially due to its computational advantages (see also Eubank, 1984, p. 440 and the discussion on page 6). Apart from computational aspects, the truncated power basis can be explained more intuitively since the piecewise character of the resulting spline is immediately obvious. Further, testing whether a knot is active (i.e. whether it is necessary for the fitted spline

and with it the respective estimated parameter) or testing for a polynomial regression function of order k, is much easier when the truncated power basis is applied. In this case, it is sufficient to test whether the respective parameter(s) are significant (e.g. Eubank, 1984, p. 443, Landajo et al., 2008, p. 236f.). But monotonicity constraints are not as easy to obtain compared to estimations using the B-spline basis. When applying P-splines based on the truncated power basis, the second order differences in the penalty term of Equation (16) have to be replaced by the squares of the parameters of the truncated power functions (e.g. Eilers & Marx, 2010, p. 638, Kauermann, 2005, p. 57).

Multiple regression The concept of using a B-spline basis to formulate a flexible regression model can be extended to the multiple regression framework. Frequently, the multiple regression model is assumed to be additive and hence one-dimensional splines as presented here can be applied. Multidimensional splines can be constructed by using tensor-product splines. The computational costs of those tensor-product splines rapidly increase with the number of estimated spline dimensions and are usually not applied for more than two dimensions. Some references concerning that issue which is not further discussed here are Dierckx (1993), Ruppert et al. (2003, ch. 13) and He & Shi (1996). For the additive models one or more covariates can be modeled using splines. Suppose that the covariates $x_1, \ldots, x_s$ have a nonlinear conditional effect on the response y which is approximated by splines and the remaining covariates $x_{s+1}, \ldots, x_q$ have linear conditional influence on y. That is, a semiparametric model is specified where the covariates $x_1, \ldots, x_s$ constitute the nonparametric part of the model and the remaining covariates $x_{s+1}, \ldots, x_q$ form the parametric part. Since the B-spline basis represents a partition of unity (see Equation (5)), incorporating a separate basis for each of the covariates $x_1, \ldots, x_s$ in a model leads to multicollinearity. To avoid this, the bases for $x_1, \ldots, x_s$ have to be slightly modified. From Equation (5), each basis function $B_j^{\boldsymbol{\kappa},k}$ can be written as

$$B_j^{\boldsymbol{\kappa},k}(x) = 1 - \sum_{\substack{j'=-(k-1) \\ j' \neq j}}^{m} B_{j'}^{\boldsymbol{\kappa},k}(x).$$

Hence, a basis that is equivalent to $B_j^{\boldsymbol{\kappa},k}$, $j = -(k-1), \ldots, m$, is given by 1, $B_j^{\boldsymbol{\kappa},k}$, $j = -(k-1)+1, \ldots, m$, and (14) can be reformulated as

$$f(x) = \alpha_{-(k-1)} + \sum_{j=-(k-1)+1}^{m} (\alpha_j - \alpha_{-(k-1)}) B_j^{\boldsymbol{\kappa},k}(x), \tag{20}$$

for the regression case with only one covariate x. To ease notation, Equation (20) is written as

$$f(x) = \beta_0' + \sum_{j=-(k-1)+1}^{m} \alpha_j B_j^{\boldsymbol{\kappa},k}(x) \tag{21}$$

though the parameters α_j, $j = -(k-1)+1, \ldots, m$, are not the same as in Equation (14). Analogously, the conditional expectation $\mathrm{E}(y|x_1, \ldots, x_q) = f(x_1, \ldots, x_q)$ in the multivariate case of Equation (1) is assumed to be given by

$$\begin{aligned} f(x_1, \ldots, x_q) = \beta_0 + & \sum_{j=-(k_1-1)+1}^{m_1} \alpha_{1j} B_{1j}^{\boldsymbol{\kappa}_1,k_1}(x_1) \\ & + \ldots + \sum_{j=-(k_s-1)+1}^{m_s} \alpha_{sj} B_{sj}^{\boldsymbol{\kappa}_s,k_s}(x_s) + \sum_{j=s+1}^{q} \beta_j \, x_j \end{aligned} \tag{22}$$

where $\beta_0 = \beta_{10}' + \ldots + \beta_{s0}'$. Further, monotonicity constraints and penalties have to be adjusted. The monotonicity constraints (11) and (12) are rendered to

$$\alpha_j \geq \alpha_{j-1} \geq 0, \quad j = -(k-1)+2, \ldots, m, \qquad \text{with} \qquad \mathbf{C} = \begin{pmatrix} 0 & 1 & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \\ & & & \ddots & \ddots \end{pmatrix},$$

and

$$\alpha_j \leq \alpha_{j-1} \leq 0, \quad j = -(k-1)+2, \ldots, m, \qquad \text{with} \qquad \mathbf{C} = \begin{pmatrix} 0 & -1 & & & \\ & 1 & -1 & & \\ & & 1 & -1 & \\ & & & \ddots & \ddots \end{pmatrix}, \tag{23}$$

for the case with only one covariate (21) and analogous reasoning can be applied for the multivariate case (22). For the penalty term of (16) and (17) it can be shown that $\alpha_{-(k-1)}$ has to be replaced by 0 for the case with only one covariate. But note that the parameters α_j from (16) and (17) are not the same as those from (20) or (22). The matrices $\mathbf{D}$ and $\mathbf{D}^T\mathbf{D}$ modify to

$$\mathbf{D} = \begin{pmatrix} 0 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & 1 & -2 & 1 & \\ & & & \ddots & \ddots & \ddots \\ & & & & 1 & -2 & 1 \end{pmatrix} \qquad \text{and} \qquad \mathbf{D}^T\mathbf{D} = \begin{pmatrix} 0 & 0 & 0 & & & & & \\ 0 & 5 & -4 & 1 & & & & \\ 0 & -4 & 6 & -4 & \cdot & & & \\ & 1 & -4 & 6 & \cdot & \cdot & & \\ & & 1 & -4 & \cdot & \cdot & 1 & \\ & & & 1 & \cdot & \cdot & -4 & 1 \\ & & & & \cdot & \cdot & 6 & -4 & 1 \\ & & & & & \cdot & -4 & 5 & -2 \\ & & & & & & 1 & -2 & 1 \end{pmatrix}. \tag{24}$$

This can be applied analogously to the multivariate case (22) with a separate penalty matrix $\mathbf{D}_1$, …, $\mathbf{D}_s$ and a separate smoothing parameter λ_1, …, λ_s for each of the covariates x_1, …, x_s. Note that in the multivariate regression, the dimension of $\mathbf{C}$ and $\mathbf{D}$ (or $\mathbf{D}_1$, …, $\mathbf{D}_s$) has to be adapted according to the number of estimated parameters by enlarging the matrices appropriately with zero-elements.

4. Computational aspects using R

R Spline estimations can be carried out using the open-source software R (R Core Team, 2012, www.r-project.org) and supplementary packages. For the examples in this work, version 2.15.1 is applied.

Appendix A contains the R-code which is necessary to reproduce the results from the examples in Section 5.

splineDesign B-splines and their derivatives can be evaluated at a value x using the function splineDesign from the base package splines, where the knot sequence and the order have to be specified. The B-splines evaluated at the observed x are treated as regressors in the function lm for least squares regressions. More precisely, the complete spline term resulting from splineDesign may be treated as one regressor in the formula within lm and a parameter is estimated for each basis function. As an alternative for splineDesign there is the option of using bs within lm. But bs has the disadvantage of being more complicated to use unless B-splines with quantile knots and all exterior knots equal to the boundary knots are employed.

If only a penalty term but no monotonicity constraint (or another inequality constraint) is imposed, the function lm can be applied to estimate the parameters as described on page 19 by including additional observations with are zero at the response side and $\sqrt{\lambda}\,\mathbf{D}$ on the covariate side.

pcls To implement penalties and/or inequality constraints, the function pcls from the package mgcv of Wood (2012, version 1.7-19) can be applied for least squares estimations. For this function several arguments have to be constructed. The response vector is put in the argument y. For the estimations explained in this paper, the weights in w are all 1. The columns of the design matrix X contain the spline component(s) and the covariates not modeled with splines. Here, no equality constraints are imposed, hence the matrix C is a (0×0)-matrix and the vector of feasible initial parameters p is not such important but it has to satisfy the inequality constraints. Since the monotonicity constraints as for example presented for the bivariate case in Equation (13) are inequality constraints, the matrix Ain equals this matrix $\mathbf{C}$ and the corresponding right-hand side vector bin is $\mathbf{0}$. If only one spline term is penalized, the penalty part is not difficult to implement. Then,

the penalty matrix $\mathbf{D}$ is just the only element of the list `S`, `off` is 0 and `sp` is equal to λ. Note that for `pcls` the penalty matrix may only contain the rows and columns of the relevant, i.e. penalized, parameters. For more than one penalized spline term, the list `S` has to contain just as many penalty matrices and `sp` just as many smoothing parameters. The vector `off` then contains the position numbers within the parameter vector which correspond to the first penalized parameter for each of the penalty terms, each number minus 1.

Since `pcls` provides no standard errors, these may be bootstrapped, for example. This is not done here.

5. Examples

In this section, two exemplary data sets are analyzed. These chosen data sets have been used in many studies and are available in the `R`-package `MASS` (see `?mcycle` and `?Boston`) from Venables & Ripley (2002, version 7.3-17). Hence, all results from this section can be reproduced easily using the `R`-script in the appendix. The first example is a bivariate analysis with only one covariate modeled by splines with no monotonicity constraint. The second example is multivariate where two of the three covariates are modeled using additive spline components one of which is monotonicity constrained.

Motorcycle data The motorcycle data set with $n = 133$ observations contains only two variables describing a simulated motorcycle accident for a test of crash helmets. The original data set is used by Silverman (1985). The independent variable used as the only covariate is the time after the impact (`times`, in milliseconds). The dependent variable is the head acceleration (`accel`, in g).

Figure 7 shows the estimated regression curves of the model

$$\texttt{accel} = \sum_{j=-(k-1)}^{m} \alpha_j \, B_j^{\boldsymbol{\kappa},k}(\texttt{times}) + u$$

for varying k with a fixed number of $m = 8$ inner knots in an equidistant knot sequence $\boldsymbol{\kappa}$. It can be seen that for $k = 1$ the regression curve consists of horizontal straight lines with jumps at the knots and for $k = 2$ the regression curve is constructed from straight lines with non-zero slopes which are continuously connected. For $k = 3$ and $k = 4$ the quadratic

and cubic pieces can be observed but for $k > 5$ the orders cannot well be distinguished visually anymore. Further, for $k \geq 3$ the position of the knots cannot be visually detected since the pieces are connected at least once continuously differentiable.

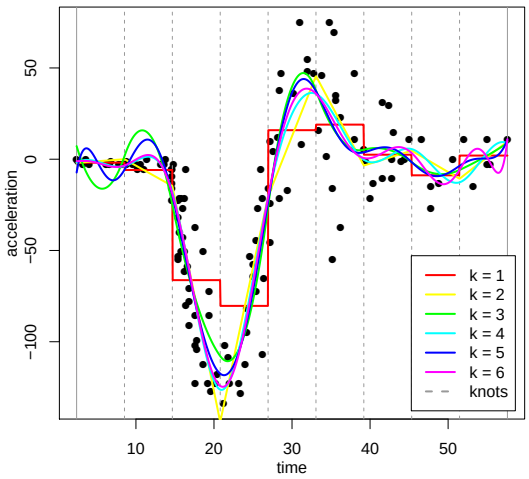

Figure 7: B-spline regression functions with $m = 8$ equidistant inner knots and different orders $k = 1, \ldots, 6$.

In contrast, in Figure 8 the order of the spline is fixed ($k = 4$) and the number of the equidistant inner knots m varies from 0 to 9. For $m = 0$ a cubic polynomial can be observed. The curve for $m = 1$ consists of two cubic pieces which are connected twice continuously differentiable at the only inner knot. With increasing m the number of cubic polynomial pieces also increases. The knots cannot be displayed in Figure 8 since they differ for each value of m. Note that $k = 4$ in Figure 7 and $m = 8$ in Figure 8 show the same curve.

For $m = 0$ and $k = 4$ the same fit can be generated by regressing the response on a polynomial of degree $p = 3$. Both variants result in the same amount of degrees of freedom. Hence, one could think about using polynomials of higher degrees instead of B-splines with higher m, that is estimating the model

$$\texttt{accel} = \sum_{j=0}^{p} \beta_j \, \texttt{times}^j + u$$

for some degree $p \geq 0$. To compare the corresponding results, Figure 9 shows the regression curves for polynomials of degree $p = 3, \ldots, 12$. While for lower values of m and p, the curves look quite similar, for higher values of m and p, the polynomial regression curves exhibit an undesirable behavior by oscillating more and more especially at the boundaries.

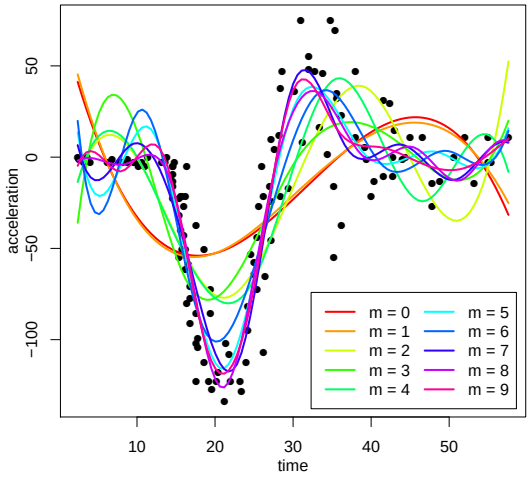

Figure 8: B-spline regression functions of order $k = 4$ with $m = 0, \ldots, 9$ equidistant inner knots.

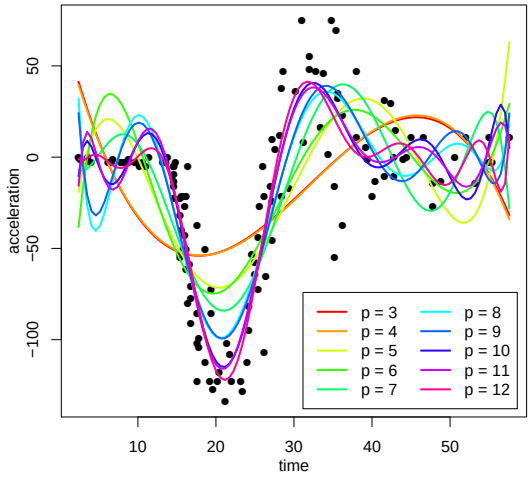

Figure 9: Regression functions for polynomial regression with $p = 3, \ldots, 12$.

In a next step, the regression curve is estimated using P-splines. In the presented example, the equidistant knot sequence contains $m = 20$ inner knots and the smoothing parameter λ takes different values. For $\lambda = 0$, the fit is unpenalized and for $\lambda \to \infty$ the regression curve is a straight line as described on page 17. This can also be observed in Figure 10. To decide which value of the smoothing parameter λ is to be used, selection criteria like the Schwarz or Akaike information criterion or (generalized) cross-validation can be applied, where

$$SIC = \log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right) + \frac{\log n \cdot \mathrm{tr}(\mathbf{H})}{n} \qquad CV = \sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - \mathbf{H}_{ii}}\right)^2$$

$$AIC = \log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right) + \frac{2 \cdot \mathrm{tr}(\mathbf{H})}{n} \qquad GCV = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{(1 - \mathrm{tr}(\mathbf{H})/n)^2}$$

with $\mathbf{H}$ being the hat matrix of the regression (the trace of which gives the dimension of the estimated model) and $\mathbf{H}_{ii}$ its ith diagonal element (e.g. Ruppert et al., 2003, sec. 5.3.1 f. for $(G)CV$). Figure 11 plots values of λ in a suitable interval against the corresponding results for SIC, AIC, CV and GCV. To be able to plot the results for all selection criteria into a single graphic, the values of the criteria are normed to the interval $[0, 1]$. This transformation does not matter for the decision based on these criteria since their magnitude is not important but only where the minimum lies. Further, the values of the selection criteria for $\lambda = 0$ are not plotted since they are huge and hence the values for

the other λs were not distinguishable in the graphic. The four selection criteria do not propose the exact same value, but most of them have their minimum at around $\lambda = 0.75$ and the resulting regression function is one of the curves plotted in Figure 10.

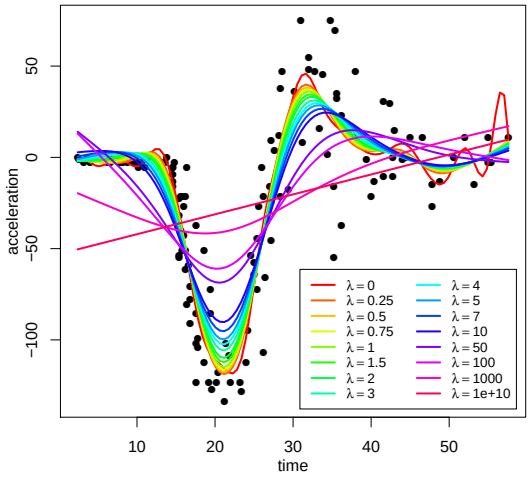

Figure 10: P-spline regression functions of order $k = 4$ with $m = 20$ equidistant inner knots and varying values for λ.

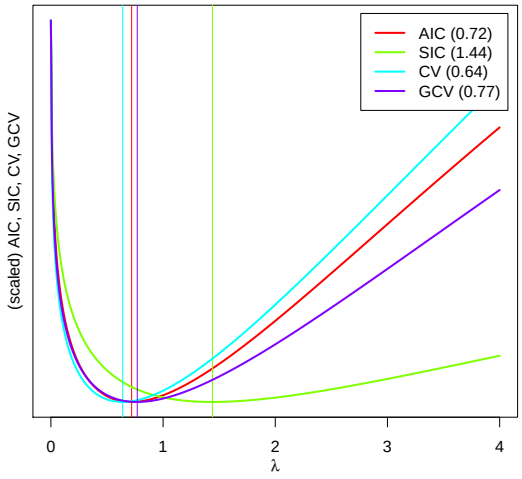

Figure 11: Smoothing parameter λ vs. resulting selection criteria for P-spline regressions of order $k = 4$ with $m = 20$ equidistant inner knots.

Finally, smoothing splines are applied. Analogously to the presentation of P-splines, Figure 12 shows the regression functions for several values of λ and Figure 13 plots values of λ against the corresponding selection criteria. The unconstrained (i.e. $\lambda = 0$) smoothing spline estimation cannot be carried out for this example with $k = 4$ due to the fact that the corresponding regressor matrix does not have full rank for $k \geq 3$ as already addressed on page 19. Hence instead of the completely unconstrained estimation a nearly unconstrained estimation using $\lambda = 10^{-16}$ is carried out to obtain Figure 12.

As an overview, one estimated regression function is plotted for each of B-spline, P-spline or smoothing spline regression. For all three regressions, the order is $k = 4$. For B-splines, the equidistant knot sequence contains $m = 8$ inner knots and for the P-splines again $m = 20$ is chosen. The smoothing parameter is $\lambda = 0.75$ for the P-spline estimation and for the smoothing spline estimation it is $\lambda = 75$. The three regression functions appear quite similar, especially the P-spline and smoothing spline curves with appropriately chosen values for λ. Overall, the specifications for the three curves seem all well chosen.

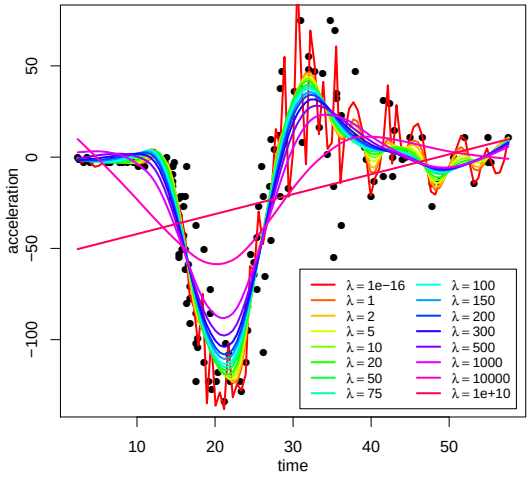

Figure 12: Smoothing spline regression functions of order $k = 4$ with varying values for λ.

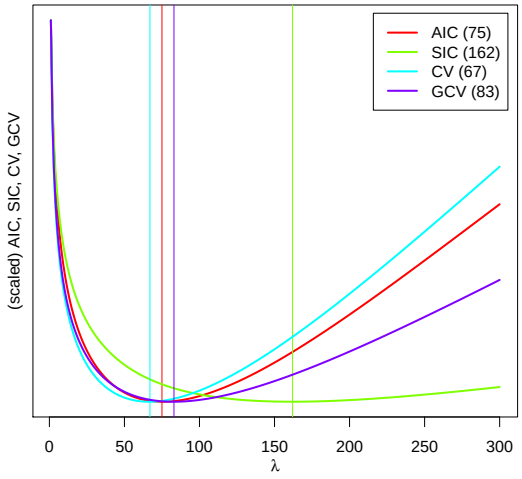

Figure 13: Smoothing parameter λ vs. resulting selection criteria for smoothing spline regressions of order $k = 4$.

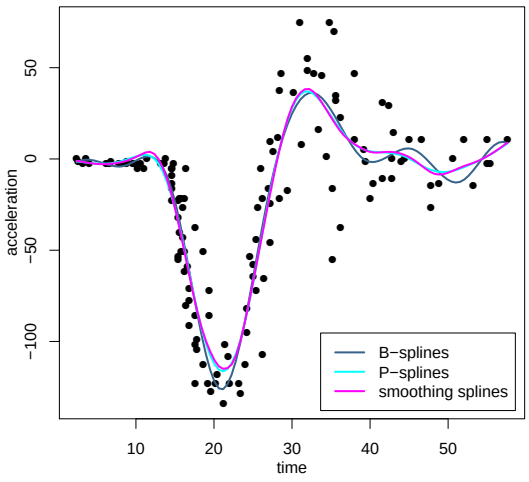

Figure 14: Regression functions from B-spline ($k = 4$, $m = 8$, equidistant knots), P-spline ($k = 4$, $m = 20$, equidistant knots, $\lambda = 0.75$) and smoothing spline ($k = 4$, $\lambda = 75$) estimation.

Boston housing data The data set on housing values in the suburbs of Boston contains many variables but only four of them are used for the following example to keep the exemplary R-code short. It has $n = 506$ observations, each of which constitutes a suburb of Boston. The median value of houses in a suburb (`medv`, in 1000$) is explained by the percentage of lower status population in this suburb (`lstat`), the weighted mean of distances from the suburb to five employment centers in Boston (`dis`) and the average room number of houses in the suburb (`rm`).

The following model is estimated using B-splines, P-splines and smoothing splines:

$$\log(\texttt{medv}) = \beta_0 + \sum_{j=-(k_l-1)+1}^{m_l} \alpha_{lj}\, B_j^{\boldsymbol{\kappa}_l,k_l}(\log(\texttt{lstat})) + \sum_{j=-(k_d-1)+1}^{m_d} \alpha_{dj}\, B_j^{\boldsymbol{\kappa}_d,k_d}(\log(\texttt{dis})) + \alpha^r\, \texttt{rm} + u, \tag{25}$$

where l, d and r correspond to `lstat`, `dis` and `rm`. Like in the motorcycle example, cubic splines are used (i.e. $k_l = k_d = k = 4$), the equidistant knot sequence for the B-spline basis contains $m = 8$ inner knots and for the P-spline estimation, it contains $m = 20$ inner knots. For the smoothing spline estimation a problem occurs. There are 455 distinct observations for the covariate `lstat` and 412 for `dis`. For $k = 4$ this would result in $1 + 456 + 413 + 1 = 871$ parameters to be estimated for $n = 506$ observations which is too much for the estimation, even with the penalty term since for `pcls` the number of rows of the regressor matrix has to exceed the number of columns. As a solution, the knot positions for this example (except the boundary knots) are chosen to have only one digit (instead of two and four), what results in $1 + 217 + 282 + 1 = 501$ parameters. Note that strictly speaking this is no smoothing spline estimation anymore.

In Equation (25) it is not possible to include the full spline bases for both covariates log(`lstat`) and log(`dis`) since they would both be a partition of unity what would cause perfect collinearity (cf. also the discussion leading to Equation (22)). Hence, one basis function from one of the bases has to be removed or one basis function of each and instead an intercept is included. Here, the latter approach is followed. For the covariate `lstat`, a monotone decreasing relationship is assumed, hence the corresponding parameters are restricted according to Equation (23). The resulting matrix $\mathbf{C}$ for the inequality constraint $\mathbf{C}\boldsymbol{\alpha} \geq \mathbf{0}$ is

$$\mathbf{C} = \begin{pmatrix} 0 & -1 & & & & 0 & \cdots & 0 \\ & 1 & -1 & & & 0 & \cdots & 0 \\ & & 1 & -1 & & 0 & \cdots & 0 \\ & & & \ddots & \ddots & \vdots & & \vdots \\ & & & & 1 & -1 & 0 & \cdots & 0 \end{pmatrix}$$

and has dimension $(r \times c)$ with $r = m_l + k - 1$ and $c = 1 + (m_l + k - 1) + (m_d + k - 1) + 1$ where c is the number of parameters to be estimated. Note that m_l and m_d differ for the B-spline, P-spline and smoothing spline estimations.

The penalty matrix $\mathbf{D}$ for the P-spline estimation is the one from Equation (24) enlarged with zero-elements to be of dimension $((c-2) \times c)$, where the "-2" is due to the second-order difference penalty. For the smoothing spline estimation, the matrix

$\mathbf{D}$ is constructed analogously according to Equation (17). Since this example is mainly intended to provide exemplary `R`-code and present an example with two spline components, no search for optimal smoothing parameters is performed for the Boston housing data. Instead, the smoothing parameters are chosen visually to be 3 and 4 for the P-spline estimation and 1 and 30 for the smoothing spline estimation.

Figure 15 and 16 show the estimated regression curves in the `lstat`-dimension and the `dis`-dimension. Since this is a multivariate example, no observational points are shown in the two-dimensional graphics and the remaining two covariates have to be fixed. Here, the median values are chosen for the two non-plotted covariates. Like in the bivariate example for the motorcycle data, all three estimation variants lead to quite similar graphical results for appropriately chosen spline parameters (k, $\boldsymbol{\kappa}$, λ).

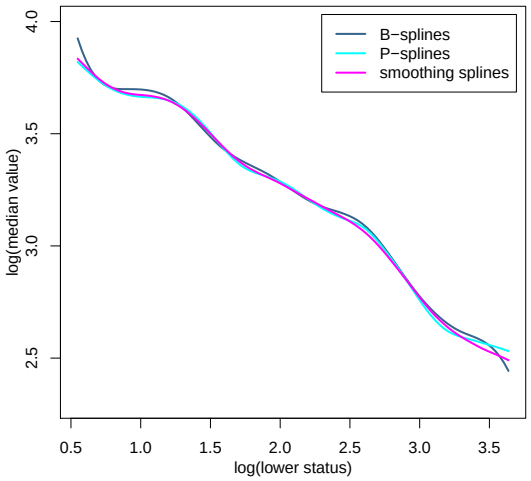

Figure 15: Regression functions for covariate log(`lstat`) and fixed `dis` and `rm` (each on its median value) from B-spline ($k_l = 4$, $m_l = 8$, equidistant knots), P-spline ($k_l = 4$, $m_l = 20$, equidistant knots, $\lambda_l = 3$) and smoothing spline ($k_l = 4$, $\lambda_l = 4$) estimation.

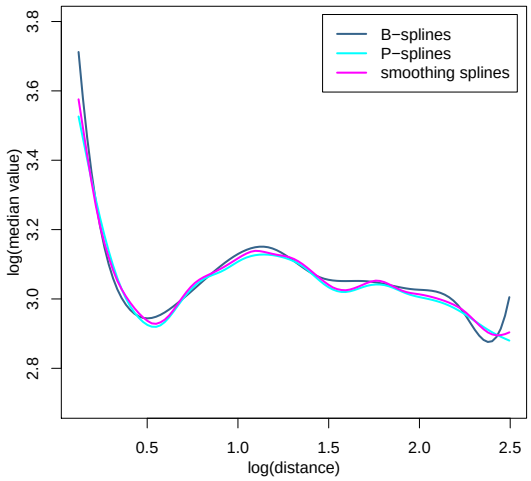

Figure 16: Regression functions for covariate log(`dis`) and fixed `lstat` and `rm` (each on its median value) from B-spline ($k_d = 4$, $m_d = 8$, equidistant knots), P-spline ($k_d = 4$, $m_d = 20$, equidistant knots, $\lambda_d = 1$) and smoothing spline ($k_d = 4$, $\lambda_d = 30$) estimation.

6. Conclusion

Three variants of spline regression are discussed and an overview of them is given: regression splines (here based on B-splines), penalized splines (here using second-order differ-

ences) and smoothing splines (here also using second-order differences). All these spline variants have in common that they are very flexible since they facilitate the choice of the functional form which is mostly determined by the data when using splines. Further, they are practicable as they are implemented in many statistical software packages as for example in `R`. For the three variants of spline regression, different spline parameters have to be chosen. The order of the splines often is taken to be $k = 4$ in practice and in the presented examples. Additionally the knot sequence (i.e. number and position of the knots) has to be specified. Concerning this point, the variants differ as discussed in the paper. For penalized splines and smoothing splines, the penalty term and its weight (λ) in the minimization problem need to be chosen.

B-splines have the advantage that no smoothing parameter has to be chosen but only the number and position of the knots. However, the choice of the knot sequence becomes very important for the fit instead. Using P-splines or smoothing splines, the knot sequence does not matter that much but the smoothing parameter has to be well chosen, where different selection criteria can guide this choice. Smoothing splines in practice have the additional disadvantage of leading to a large number of parameters to be estimated which may exceed the number of observations very fast, especially for multivariate regressions, and the costs (computation and time) of the corresponding optimization increase.

In multivariate settings, several covariates may be estimated using splines. If the components are combined additively in the model, the number of parameters to be estimated does not increase much for B-splines and P-splines. But the search for suitable smoothing parameters is elaborate and time-consuming since it has to be performed multidimensional. Further, in multivariate regressions, for example, dummy variables may be interacted with the spline component(s).

Since the spline basis functions can be interpreted as usual covariates in a regression analysis, the estimations can be carried out using ordinary least squares methods even with additional penalty terms (as for example carried out in the motorcycle analysis in Section 5). If inequality constraints (as for example monotonicity) are imposed, other estimation methods have to be applied (using for example the function `pcls` from the `mgcv`-package for `R` which is based on quadratic programming).

References

Bollaerts, K., Eilers, P. H. C., & Aerts, M. (2006). Quantile regression with monotonicity restrictions using P-splines and the L_1-norm. *Statistical Modelling*, 6(3), 189–207.

Cao, Y., Lin, H., Wu, T. Z., & Yu, Y. (2010). Penalized spline estimation for functional coefficient regression models. *Computational Statistics & Data Analysis*, 54(4), 891–905.

Claeskens, G., Krivobokova, T., & Opsomer, J. D. (2009). Asymptotic properties of penalized spline estimators. *Biometrika*, 96(3), 529–544.

Crainiceanu, C. M., Ruppert, D., Carroll, R. J., Joshi, A., & Goodner, B. (2007). Spatially adaptive Bayesian penalized splines with heteroscedastic errors. *Journal of Computational & Graphical Statistics*, 16(2), 265–288.

Davis, C. H. (1857). *Theory of the motion of the heavenly bodies moving about the sun in conic sections: a translation of Gauss's "Theoria Motus"*. Little, Brown and Company.

de Boor, C. (2001). *A Practical Guide to Splines*, volume 27 of *Applied Mathematical Sciences*. Berlin: Springer, revised edition.

Dierckx, P. (1993). *Curve and Surface Fitting with Splines*. Numerical Mathematics and Scientific Computation. Oxford: Oxford University Press.

Eilers, P. H. C. & Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2), 89–121.

Eilers, P. H. C. & Marx, B. D. (2010). Splines, knots, and penalties. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2, 637–653.

Eubank, R. L. (1984). Approximate regression models and splines. *Communications in Statistics - Theory and Methods*, 13(4), 433–484.

Gauss, C. F. (1809). Theoria motus corporum coelestium. In *Carl Friedrich Gauss – Werke*. Königliche Gesellschaft der Wissenschaften zu Göttingen (1906).

Haupt, H., Kagerer, K., & Schnurbus, J. (2011). Cross-validating fit and predictive accuracy of nonlinear quantile regressions. *Journal of Applied Statistics*, 38(12), 2939–2954.

Haupt, H., Kagerer, K., & Steiner, W. J. (2012). Smooth quantile based modeling of brand sales, price and promotional effects from retail scanner panels. Under revision at the *Journal of Applied Econometrics*.

He, X. & Shi, P. (1998). Monotone B-spline smoothing. *Journal of the American Statistical Association*, 93(442), 643–650.

He, X. M. & Shi, P. (1996). Bivariate tensor-product B-splines in a partly linear model. *Journal of Multivariate Analysis*, 58(2), 162–181.

Huang, J. Z. & Shen, H. (2004). Functional coefficient regression models for non-linear time series: A polynomial spline approach. *Scandinavian Journal of Statistics*, 31(4), 515–534.

Huang, J. Z., Wu, C. O., & Zhou, L. (2004). Polynomial spline estimation and inference for varying coefficient models with longitudinal data. *Statistica Sinica*, 14(3), 763–788.

Imoto, S. & Konishi, S. (2003). Selection of smoothing parameters in B-spline nonparametric regression models using information criteria. *Annals of the Institute of Statistical Mathematics*, 55(4), 671–687.

Kauermann, G. (2005). A note on smoothing parameter selection for penalized spline smoothing. *Journal of Statistical Planning and Inference*, 127(1-2), 53–69.

Kimeldorf, G. S. & Wahba, G. (1970a). A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics*, 41(2), 495–502.

Kimeldorf, G. S. & Wahba, G. (1970b). Spline functions and stochastic processes. *Sankhya: The Indian Journal of Statistics (A)*, 32, 173–180.

Landajo, M., De Andrés, J., & Lorca, P. (2008). Measuring firm performance by using linear and non-parametric quantile regressions. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 57(2), 227–250.

Lang, S. & Brezger, A. (2004). Bayesian P-splines. *Journal of Computational & Graphical Statistics*, 13(1), 183–212.

Lee, T. C. M. (2000). Regression spline smoothing using the minimum description length principle. *Statistics & Probability Letters*, 48(1), 71–82.

Legendre, A.-M. (1805). *Nouvelles méthodes pour la détermination des orbites des comètes.*

Lu, M., Zhang, Y., & Huang, J. (2009). Semiparametric estimation methods for panel count data using monotone B-splines. *Journal of the American Statistical Association*,

104(487), 1060–1070.

Mackenzie, M. L., Donovan, C., & McArdle, B. (2005). Regression spline mixed models: A forestry example. *Journal of Agricultural, Biological and Environmental Statistics*, 10(4), 394–410.

O'Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems. *Statistical Science*, 1(4), 502–527.

O'Sullivan, F. (1988). Fast computation of fully automated log-density and log-hazard estimators. *SIAM Journal on Scientific and Statistical Computing*, 9(2), 363–379.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Ruppert, D. (2002). Selecting the number of knots for penalized splines. *Journal of Computational and Graphical Statistics*, 11(4), 735–757.

Ruppert, D., Wand, M. P., & Carroll, R. J. (2003). *Semiparametric regression*, volume 12 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge: Cambridge University Press.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society Series B - Methodological*, 47(1), 1–52.

Venables, W. N. & Ripley, B. D. (2002). *Modern Applied Statistics with S*. New York: Springer, fourth edition.

Wand, M. P. (2000). A comparison of regression spline smoothing procedures. *Computational Statistics*, 15(4), 443–462.

Wegman, E. J. & Wright, I. W. (1983). Splines in statistics. *Journal of the American Statistical Association*, 78(382), 351–365.

Wood, S. N. (1994). Monotonic smoothing splines fitted by cross-validation. *SIAM Journal on Scientific Computing*, 15(5), 1126–1133.

Wood, S. N. (2012). *mgcv: Mixed GAM Computation Vehicle with GCV/AIC/REML smoothness estimation*. R package version 1.7-19.

A. R-codes

```
#######
# load packages
#######

library(MASS)           # data sets
library(splines)        # B-splines
library(mgcv)           # constrained estimation (pcls)


#######
# define some functions
#######

# equidistant knot sequence

knots_eq    <- function(x, k, m)
                {
                  c(min(x) - ((k-1):0) * (max(x)-min(x))/(m+1),
                     seq(from=min(x), to=max(x), length.out=m+2)[-c(1,m+2)],
                     max(x) + (0:(k-1)) * (max(x)-min(x))/(m+1))
                }

# functions for polynomial regression

polynom     <- function(x, p)
                {
                  if (p==0) { return("1") }
                  if (p >0) {
                    return(paste(polynom(x=x, p=p-1), " + I(", x, "^",
                                                    p, ")", sep="")) }
                }
polynom_m <- function(x, p)
                {
                  if (p==0) { return(as.matrix(rep(1,length(x)))) }
                  if (p >0) { return(cbind(polynom_m(x=x,p=p-1),x^p)) }
                }
```

```
# functions for information criteria

AIC   <- function(res, H)
      {
        log(sum(res^2)/length(res)) + 2*sum(diag(H))/length(res)
      }

SIC   <- function(res, H)
      {
        log(sum(res^2)/length(res)) +sum(diag(H))*log(length(res))/length(res)
      }

CV    <- function(res, H)
      {
        sum((res/(1-diag(H)))^2)
      }

GCV   <- function(res, H)
      {
        sum(res^2)/(1-sum(diag(H))/length(res))^2
      }

normed      <- function(x) { (x-min(x))/(max(x) - min(x)) }
```

```
#######
# bivariate example: motorcycle accidents
#######

# data

times       <- mcycle$times
accel       <- mcycle$accel


# B-spline estimations with different k (m=8) (FIGURE 7)

m           <- 8
ks          <- 1:6
cols_k      <- rainbow(length(ks))

par(mai=c(0.65,0.6,0.1,0.1), mgp=c(2,1,0))
plot(times, accel, xlab="time", ylab="acceleration", pch=16)
abline(v=knots_eq(times, 1, m), lty=c(1,rep(2,m),1), col="grey60")

for (kk in 1:length(ks))
{
  k         <- ks[kk]
  est       <- lm(accel ~ -1 +
                  splineDesign(x=times, knots=knots_eq(times, k, m), ord=k))
  plot(function(x) splineDesign(x=x, knots=knots_eq(times, k, m), ord=k) %*%
                  est$coef,
       from=min(times), to=max(times), n=1001, lwd=2, col=cols_k[kk], add=TRUE)
}
legend("bottomright", inset=0.02, legend=c(paste("k = ",ks,sep=""), "knots"),
       lwd=2, lty=c(rep(1,length(ks)),2), col=c(cols_k, "grey60"), bg="white")
```

```
# B-spline estimations with different m (k=4) (FIGURE 8)

k           <- 4
ms          <- 0:9
cols_m      <- rainbow(length(ms))

par(mai=c(0.65,0.6,0.1,0.1), mgp=c(2,1,0))
plot(times, accel, xlab="time", ylab="acceleration", pch=16)

for (mm in 1:length(ms))
{
  m         <- ms[mm]
  est       <- lm(accel ~ -1 +
                  splineDesign(x=times, knots=knots_eq(times, k, m), ord=k))
  plot(function(x) splineDesign(x=x, knots=knots_eq(times, k, m), ord=k) %*%
                  est$coef,
            from=min(times), to=max(times), add=TRUE, lwd=2, col=cols_m[mm])
}
legend("bottomright", inset=0.02, legend=paste("m = ", ms, sep=""),
            col=cols_m, lwd=2, ncol=2)


# polynomial estimations with different p (FIGURE 9)

ps          <- 3:12
cols_p      <- rainbow(length(ps))

par(mai=c(0.65,0.6,0.1,0.1), mgp=c(2,1,0))
plot(times, accel, xlab="time", ylab="acceleration", pch=16)

for (pp in 1:length(ps))
{
  p         <- ps[pp]
  est       <- lm(formula=as.formula(paste("accel ~ -1 + ",
                                        polynom("times",p=p),sep="")))
  plot(function(x) polynom_m(x, p=p) %*% est$coef,
            from=min(times), to=max(times), add=TRUE, lwd=2, col=cols_p[pp])
}
legend("bottomright", inset=0.02, legend=paste("p = ", ps, sep=""),
            col=cols_p, lwd=2, ncol=2)
```

```
# P-spline estimation with different lambda (FIGURE 10)

k           <- 4
m           <- 20

D           <- matrix(0, nrow=m+k-2, ncol=m+k)
for (j in 1:(m+k-2))
{
  d_j       <- c(rep(0,j-1),1,-2,1,rep(0,(m+k)-3-(j-1)))
  e_j       <- c(rep(0,j-1),  1   ,rep(0,(m+k)-3-(j-1)))
  D         <- D + e_j%*%t(d_j)
}

y_star      <- c(accel, rep(0,m+k-2))
X_spl       <- splineDesign(x=times, knots=knots_eq(times,k,m), ord=k)

ls          <- c(0,0.25,0.5,0.75,1,1.5,2,3,4,5,7,10,50,100,1000,10^10)
cols_l      <- rainbow(length(ls))

par(mai=c(0.65,0.6,0.1,0.1), mgp=c(2,1,0))
plot(times, accel, xlab="time", ylab="acceleration", pch=16)

for (ll in 1:length(ls))
{
  l         <- ls[ll]
  X_star    <- rbind(X_spl, sqrt(l)*D)
  est       <- lm(y_star ~ -1 + X_star)
  plot(function(x) splineDesign(x=x, knots=knots_eq(times, k, m), ord=k) %*%
                 est$coef,
            from=min(times), to=max(times), add=TRUE, lwd=2, col=cols_l[ll])
}
legend("bottomright", inset=0.02, col=cols_l, lwd=2, ncol=2, cex=0.8,
            legend=parse(text=paste("lambda==", ls, sep="")))
```

```
# lambda vs. SIC,... for P-splines from above (FIGURE 11)

lambdas     <- c(10^{-5}, 1:400/100)

AICs        <- rep(NA, length(lambdas))
SICs        <- AICs
GCVs        <- AICs
CVs         <- AICs

for (ll in 1:length(lambdas))
{
  l         <- lambdas[ll]
  X_star    <- rbind(X_spl, sqrt(l)*D)
  est       <- lm(y_star ~ -1 + X_star)
  Hat       <- X_spl %*% solve(t(X_star)%*%X_star) %*% t(X_spl)
  Res       <- accel - X_spl%*% est$coef
  AICs[ll]  <- AIC(res=Res, H=Hat)
  SICs[ll]  <- SIC(res=Res, H=Hat)
  GCVs[ll]  <- GCV(res=Res, H=Hat)
  CVs[ll]   <- CV(res=Res, H=Hat)
}

par(mai=c(0.65,0.4,0.1,0.1), mgp=c(2,1,0))
plot(range(lambdas), 0:1, type="n", yaxt="n", xlab=expression(lambda),ylab="")
axis(2, at=0.5, label="(scaled) AIC, SIC, CV, GCV", tick=FALSE, line=-0.5)

SCs   <- c("AICs", "SICs", "CVs", "GCVs")
for (SC in 1:length(SCs))
{
  points(lambdas, normed(get(SCs[SC])),
                              type="l", lwd=2, col=rainbow(4)[SC])
  abline(v=lambdas[which(get(SCs[SC])==min(get(SCs[SC])))],col=rainbow(4)[SC])
}
legend("topright", inset=0.02,
      legend=c(paste("AIC (", lambdas[which(AICs==min(AICs))], ")", sep=""),
               paste("SIC (", lambdas[which(SICs==min(SICs))], ")", sep=""),
               paste("CV (",  lambdas[which(CVs ==min(CVs ))], ")", sep=""),
               paste("GCV (", lambdas[which(GCVs==min(GCVs))], ")", sep="")),
      col=rainbow(4), lwd=2, bg="white")
```

```
# smoothing spline estimation with different lambda (FIGURE 12)

k           <- 4
kappas      <- c(rep(min(times),k-1),sort(unique(times)),rep(max(times),k-1))
m           <- length(kappas) - 2*k

D           <- matrix(0, nrow=m+k-2, ncol=m+k)
for (j in 1:(m+k-2))
{
  denom     <- (diff(kappas, lag=k-2))[2+j]
  nom_1     <- (diff(kappas, lag=k-1))[1+c(j,j+1)]
  nom_2     <- matrix(c(1,-1,0,0,-1,1),ncol=2)%*%(1/nom_1)
  d_j       <- c(rep(0,j-1),nom_2,rep(0,(m+k)-3-(j-1)))
  e_j       <- c(rep(0,j-1),  1  ,rep(0,(m+k)-3-(j-1)))
  D         <- D + e_j%*%t(d_j)/denom
}

cor_fact    <- (k-1)*(k-2)*((max(times)-min(times))/(m+1))^2
D           <- cor_fact*D

y_star      <- c(accel, rep(0,m+k-2))
X_spl       <- splineDesign(x=times, knots=kappas, ord=k)

ls       <- c(10^{-16},1,2,5,10,20,50,75,100,150,200,300,500,1000,10000,10^10)
cols_l   <- rainbow(length(ls))

par(mai=c(0.65,0.6,0.1,0.1), mgp=c(2,1,0))
plot(times, accel, xlab="time", ylab="acceleration", pch=16)

for (ll in 1:length(ls))
{
  l         <- ls[ll]
  X_star    <- rbind(X_spl, sqrt(l)*D)
  est       <- lm(y_star ~ -1 + X_star)
  plot(function(x) splineDesign(x=x, knots=kappas, ord=k) %*% est$coef,
            from=min(times), to=max(times), add=TRUE, lwd=2, col=cols_l[ll])
}
legend("bottomright", inset=0.02, col=cols_l, lwd=2, ncol=2, cex=0.8,
            legend=parse(text=paste("lambda==", ls, sep="")))
```

```
# lambda vs. SIC,... for smoothing splines from above (FIGURE 13)

lambdas     <- c(1:300)

AICs        <- rep(NA, length(lambdas))
SICs        <- AICs
GCVs        <- AICs
CVs         <- AICs

for (ll in 1:length(lambdas))
{
  l         <- lambdas[ll]
  X_star    <- rbind(X_spl, sqrt(l)*D)
  est       <- lm(y_star ~ -1 + X_star)
  Hat       <- X_spl %*% solve(t(X_star)%*%X_star) %*% t(X_spl)
  Res       <- accel - X_spl%*% est$coef
  AICs[ll]  <- AIC(res=Res, H=Hat)
  SICs[ll]  <- SIC(res=Res, H=Hat)
  GCVs[ll]  <- GCV(res=Res, H=Hat)
  CVs[ll]   <- CV(res=Res, H=Hat)
}

par(mai=c(0.65,0.4,0.1,0.1), mgp=c(2,1,0))
plot(range(lambdas), 0:1, type="n", yaxt="n", xlab=expression(lambda),ylab="")
axis(2, at=0.5, label="(scaled) AIC, SIC, CV, GCV", tick=FALSE, line=-0.5)

SCs   <- c("AICs", "SICs", "CVs", "GCVs")
for (SC in 1:length(SCs))
{
  points(lambdas, normed(get(SCs[SC])),
                              type="l", lwd=2, col=rainbow(4)[SC])
  abline(v=lambdas[which(get(SCs[SC])==min(get(SCs[SC])))],col=rainbow(4)[SC])
}
legend("topright", inset=0.02,
      legend=c(paste("AIC (", lambdas[which(AICs==min(AICs))], ")", sep=""),
               paste("SIC (", lambdas[which(SICs==min(SICs))], ")", sep=""),
               paste("CV (",  lambdas[which(CVs ==min(CVs ))], ")", sep=""),
               paste("GCV (", lambdas[which(GCVs==min(GCVs))], ")", sep="")),
      col=rainbow(4), lwd=2)
```

```
# B-spline, P-spline and smoothing spline estimation (FIGURE 14)

k           <- 4
mb          <- 8
mp          <- 20
ms          <- length(unique(times))-2
lp          <- 0.75
ls          <- 75

kappas      <- c(rep(min(times),k-1),sort(unique(times)),rep(max(times),k-1))

D_p         <- matrix(0, nrow=mp+k-2, ncol=mp+k)
for (j in 1:(mp+k-2))
{
  d_j       <- c(rep(0,j-1),1,-2,1,rep(0,(mp+k)-3-(j-1)))
  e_j       <- c(rep(0,j-1),  1   ,rep(0,(mp+k)-3-(j-1)))
  D_p       <- D_p + e_j%*%t(d_j)
}

D_s         <- matrix(0, nrow=ms+k-2, ncol=ms+k)
for (j in 1:(ms+k-2))
{
  denom     <- (diff(kappas, lag=k-2))[2+j]
  nom_1     <- (diff(kappas, lag=k-1))[1+c(j,j+1)]
  nom_2     <- matrix(c(1,-1,0,0,-1,1),ncol=2)%*%(1/nom_1)
  d_j       <- c(rep(0,j-1),nom_2,rep(0,(ms+k)-3-(j-1)))
  e_j       <- c(rep(0,j-1),  1  ,rep(0,(ms+k)-3-(j-1)))
  D_s       <- D_s + e_j%*%t(d_j)/denom
}
cor_fact    <- (k-1)*(k-2)*((max(times)-min(times))/(ms+1))^2
D_s         <- cor_fact*D_s

yp_star     <- c(accel, rep(0,mp+k-2))
Xp_spl      <- splineDesign(x=times, knots=knots_eq(times,k,mp), ord=k)
Xp_star     <- rbind(Xp_spl, sqrt(lp)*D_p)

ys_star     <- c(accel, rep(0,ms+k-2))
Xs_spl      <- splineDesign(x=times, knots=kappas, ord=k)
Xs_star     <- rbind(Xs_spl, sqrt(ls)*D_s)
```

```
cols_3      <- c("steelblue4", "cyan", "magenta")

par(mai=c(0.65,0.6,0.1,0.1), mgp=c(2,1,0))
plot(times, accel, xlab="time", ylab="acceleration", pch=16)

est_b       <- lm(accel ~ -1 +
                  splineDesign(x=times, knots=knots_eq(times, k, mb), ord=k))
plot(function(x) splineDesign(x=x, knots=knots_eq(times, k, mb), ord=k) %*%
                  est_b$coef,
            from=min(times), to=max(times), add=TRUE, lwd=2, col=cols_3[1])

est_p       <- lm(yp_star ~ -1 + Xp_star)
plot(function(x) splineDesign(x=x, knots=knots_eq(times, k, mp), ord=k) %*%
                  est_p$coef,
            from=min(times), to=max(times), add=TRUE, lwd=2, col=cols_3[2])

est_s       <- lm(ys_star ~ -1 + Xs_star)
plot(function(x) splineDesign(x=x, knots=kappas, ord=k) %*%
                  est_s$coef,
            from=min(times), to=max(times), add=TRUE, lwd=2, col=cols_3[3])

legend("bottomright", inset=0.02, col=cols_3, lwd=2,
            legend=c("B-splines", "P-splines", "smoothing splines"))
```

```
#######
# multivariate example: Boston housing
#######

# data

medv        <- Boston$medv
lstat       <- Boston$lstat
dis         <- Boston$dis
rm          <- Boston$rm


# order for B-splines, P-splines and smoothing splines

k     <- 4


# B-splines

mb    <- 8

X_B   <- cbind(1, splineDesign(x=log(lstat),
                      knots=knots_eq(log(lstat),k=k,m=mb), ord=k)[,-1],
                  splineDesign(x=log(dis),
                      knots=knots_eq(log(dis),k=k,m=mb), ord=k)[,-1],
                  rm)

C_B   <- matrix(0, nrow=mb+k-1, ncol=1+(mb+k-1)+(mb+k-1)+1)
for (j in 1:(mb+k-1))
{
  C_B[j,j]        <- (j>1)
  C_B[j,j+1]      <- -1
}

M_B   <- list(y=log(medv),    w=rep(1, length(medv)),
            X=X_B,            C=matrix(0,0,0),
            S=list(),         off=array(0,0),
            sp=array(0,0),    p=c(0,-(2:(mb+k)), rep(0,(mb+k-1)+1)),
            Ain=C_B,          bin=rep(0,mb+k-1))

est_B <- pcls(M_B)
```

```
# P-splines

mp          <- 20
lp_l        <- 3
lp_d        <- 4

DTD_p       <- matrix(0, nrow=mp+k-2, ncol=mp+k)
for (j in 1:(mp+k-2))
{
  d_j       <- c(rep(0,j-1),1*(j>1),-2,1,rep(0,(mp+k)-3-(j-1)))
  e_j       <- c(rep(0,j-1),      1     ,rep(0,(mp+k)-3-(j-1)))
  DTD_p     <- DTD_p + e_j%*%t(d_j)
}
DTD_P <- t(DTD_p)%*%DTD_p

X_P   <- cbind(1, splineDesign(x=log(lstat),
                       knots=knots_eq(log(lstat),k=k,m=mp), ord=k)[,-1],
                  splineDesign(x=log(dis),
                       knots=knots_eq(log(dis),k=k,m=mp), ord=k)[,-1],
                  rm)

C_P   <- matrix(0, nrow=mp+k-1, ncol=1+(mp+k-1)+(mp+k-1)+1)
for (j in 1:(mp+k-1))
{
  C_P[j,j]        <- (j>1)
  C_P[j,j+1]      <- -1
}

M_P   <- list(y=log(medv),          w=rep(1, length(medv)),
            X=X_P,                  C=matrix(0,0,0),
            S=list(DTD_P[-1,-1], DTD_P[-1,-1]),        off=c(1,1+mp+k-1),
            sp=c(lp_l,lp_d),        p=c(0,-(2:(mp+k)), rep(0,(mp+k-1)+1)),
            Ain=C_P,                bin=rep(0,mp+k-1))

est_P <- pcls(M_P)
```

```
# smoothing splines

lstat_tmp1  <- sort(unique(lstat))
lstat_tmp2  <- unique(round(lstat_tmp1[-c(1,length(lstat_tmp1))],1))
lstat_tmp3  <- lstat_tmp2[lstat_tmp2>min(lstat) & lstat_tmp2<max(lstat)]

dis_tmp1    <- sort(unique(dis))
dis_tmp2    <- unique(round(dis_tmp1[-c(1,length(dis_tmp1))],1))
dis_tmp3    <- dis_tmp2[dis_tmp2>min(dis) & dis_tmp2<max(dis)]

kappas_l    <- log( c( rep(min(lstat),k), lstat_tmp3, rep(max(lstat),k) ) )
kappas_d    <- log( c( rep(min(dis),k),
                       sort(unique(round(dis,2)[
                                  round(dis,2)>min(dis) &
                                  round(dis,2)<max(dis)])),
                       rep(max(dis),k) ) )

ms_l        <- length(kappas_l)-2*k
ms_d        <- length(kappas_d)-2*k
ls_l        <- 1
ls_d        <- 30

DTD_s_l     <- matrix(0, nrow=ms_l+k-2, ncol=ms_l+k)
for (j in 1:(ms_l+k-2))
{
  denom     <- (diff(kappas_l, lag=k-2))[2+j]
  nom_1     <- (diff(kappas_l, lag=k-1))[1+c(j,j+1)]
  nom_2     <- matrix(c(1,-1,0,0,-1,1),ncol=2)%*%(1/nom_1) * c(j>1,1,1)
  d_j       <- c(rep(0,j-1),nom_2,rep(0,(ms_l+k)-3-(j-1)))
  e_j       <- c(rep(0,j-1),  1  ,rep(0,(ms_l+k)-3-(j-1)))
  DTD_s_l   <- DTD_s_l + e_j%*%t(d_j)/denom
}
cor_fact_l  <- (k-1)*(k-2)*((max(lstat)-min(lstat))/(ms_l+1))^2
DTD_s_l     <- cor_fact_l*DTD_s_l
DTD_l       <- t(DTD_s_l) %*% DTD_s_l
```

```
DTD_s_d     <- matrix(0, nrow=ms_d+k-2, ncol=ms_d+k)
for (j in 1:(ms_d+k-2))
{
  denom     <- (diff(kappas_d, lag=k-2))[2+j]
  nom_1     <- (diff(kappas_d, lag=k-1))[1+c(j,j+1)]
  nom_2     <- matrix(c(1,-1,0,0,-1,1),ncol=2)%*%(1/nom_1) * c(j>1,1,1)
  d_j       <- c(rep(0,j-1),nom_2,rep(0,(ms_d+k)-3-(j-1)))
  e_j       <- c(rep(0,j-1),  1  ,rep(0,(ms_d+k)-3-(j-1)))
  DTD_s_d   <- DTD_s_d + e_j%*%t(d_j)/denom
}
cor_fact_d  <- (k-1)*(k-2)*((max(dis)-min(dis))/(ms_d+1))^2
DTD_s_d     <- cor_fact_d*DTD_s_d
DTD_d       <- t(DTD_s_d) %*% DTD_s_d

X_S   <- cbind(1, splineDesign(x=log(lstat), knots=kappas_l, ord=k)[,-1],
                  splineDesign(x=log(dis), knots=kappas_d, ord=k)[,-1], rm)

C_S   <- matrix(0, nrow=ms_l+k-1, ncol=1+(ms_l+k-1)+(ms_d+k-1)+1)
for (j in 1:(ms_l+k-1))
{
  C_S[j,j]        <- (j>1)
  C_S[j,j+1]      <- -1
}

M_S   <- list(y=log(medv),          w=rep(1, length(medv)),
            X=X_S,                  C=matrix(0,0,0),
            S=list(DTD_l[-1,-1], DTD_d[-1,-1]),        off=c(1,1+ms_l+k-1),
            sp=c(ls_l,ls_d),        p=c(0,-(2:(ms_l+k)), rep(0,(ms_d+k-1)+1)),
            Ain=C_S,                bin=rep(0,ms_l+k-1))

est_S <- pcls(M_S)
```

```
# graphic for lstat (FIGURE 15)

cols_3 <- c("steelblue4", "cyan", "magenta")

par(mai=c(0.65,0.6,0.1,0.1), mgp=c(2,1,0))
plot(log(lstat), log(medv), xlab="log(lower status)",ylab="log(median value)",
            ylim=c(2.3,4), type="n")

plot(function(x) est_B[1] +
      c(splineDesign(x=x, knots=knots_eq(log(lstat),k=k,m=mb), ord=k)[,-1] %*%
            est_B[2:(mb+k)]) +
      c(splineDesign(x=median(log(dis)),
                  knots=knots_eq(log(dis),k=k,m=mb), ord=k)[,-1] %*%
            est_B[mb+k + 1:(mb+k-1)]) +
      median(rm) * est_B[1+2*(mb+k-1)+1],
      from=min(log(lstat)), to=max(log(lstat)), add=TRUE, lwd=2,col=cols_3[1])

plot(function(x) est_P[1] +
      c(splineDesign(x=x, knots=knots_eq(log(lstat),k=k,m=mp), ord=k)[,-1] %*%
            est_P[2:(mp+k)]) +
      c(splineDesign(x=median(log(dis)),
                  knots=knots_eq(log(dis),k=k,m=mp), ord=k)[,-1] %*%
            est_P[mp+k + 1:(mp+k-1)]) +
      median(rm) * est_P[1+2*(mp+k-1)+1],
      from=min(log(lstat)), to=max(log(lstat)), add=TRUE, lwd=2,col=cols_3[2])

plot(function(x) est_S[1] +
      c(splineDesign(x=x, knots=kappas_l, ord=k)[,-1] %*%
            est_S[1 + 1:(ms_l+k-1)]) +
      c(splineDesign(x=median(log(dis)), knots=kappas_d, ord=k)[,-1] %*%
            est_S[1+(ms_l+k-1) + 1:(ms_d+k-1)]) +
      median(rm) * est_S[1+(ms_l+k-1)+(ms_d+k-1)+1],
      from=min(log(lstat)), to=max(log(lstat)), add=TRUE, lwd=2,col=cols_3[3])

legend("topright", inset=0.02, col=cols_3, lwd=2,
            legend=c("B-splines", "P-splines", "smoothing splines"))
```

```
# graphic for dis (FIGURE 16)

par(mai=c(0.65,0.6,0.1,0.1), mgp=c(2,1,0))
plot(log(dis), log(medv), xlab="log(distance)", ylab="log(median value)",
            ylim=c(2.7,3.8), type="n")

plot(function(x) est_B[1] +
      c(splineDesign(x=median(log(lstat)),
                  knots=knots_eq(log(lstat),k=k,m=mb), ord=k)[,-1] %*%
            est_B[2:(mb+k)]) +
      c(splineDesign(x=x, knots=knots_eq(log(dis),k=k,m=mb), ord=k)[,-1] %*%
            est_B[mb+k + 1:(mb+k-1)]) +
      median(rm) * est_B[1+2*(mb+k-1)+1],
      from=min(log(dis)), to=max(log(dis)), add=TRUE, lwd=2, col=cols_3[1])

plot(function(x) est_P[1] +
      c(splineDesign(x=median(log(lstat)),
                  knots=knots_eq(log(lstat),k=k,m=mp), ord=k)[,-1] %*%
            est_P[2:(mp+k)]) +
      c(splineDesign(x=x, knots=knots_eq(log(dis),k=k,m=mp), ord=k)[,-1] %*%
            est_P[mp+k + 1:(mp+k-1)]) +
      median(rm) * est_P[1+2*(mp+k-1)+1],
      from=min(log(dis)), to=max(log(dis)), add=TRUE, lwd=2, col=cols_3[2])

plot(function(x) est_S[1] +
      c(splineDesign(x=median(log(lstat)),knots=kappas_l,ord=k)[,-1] %*%
            est_S[1 + 1:(ms_l+k-1)]) +
      c(splineDesign(x=x, knots=kappas_d, ord=k)[,-1] %*%
            est_S[1+(ms_l+k-1) + 1:(ms_d+k-1)]) +
      median(rm) * est_S[1+(ms_l+k-1)+(ms_d+k-1)+1],
      from=min(log(dis)), to=max(log(dis)), add=TRUE, lwd=2, col=cols_3[3])

legend("topright", inset=0.02, col=cols_3, lwd=2,
      legend=c("B-splines", "P-splines", "smoothing splines"))
```