# Constrained Smoothing Splines Revisited

Berwin A. Turlach
CMA and ACSys CRC
The Australian National University
Canberra ACT 0200
Australia

September 1, 1997

**Abstract**

In some regression settings one would like to combine the flexibility of nonparametric smoothing with some prior knowledge about the regression curve. Such prior knowledge may come from a physical or economic theory, leading to shape constraints such as the underlying regression curve being positive, monotone or convex-concave. We propose a new method for calculating smoothing splines that fulfill these kinds of constraints. Our approach leads to a quadratic programming problem and the infinite number of constraints are replaced by a finite number of constraints that are chosen adaptively. We show that the resulting problem can be solved using the algorithm of Goldfarb and Idnani (1982, 1983) and illustrate our method on several real data sets.

**KEY WORDS AND PHRASES**. Concavity, constrained smoothing, convexity, monotonicity, nonparametric smoothing, quadratic programming, splines.

**AMS (1991) SUBJECT CLASSIFICATION**. Primary 62G07, Secondary 65D10.

# 1 Introduction

Nonparametric smoothing techniques, developed over the last decades, give researchers flexible tools for analysing data. These methods are based, among others, on spline smoothers, kernel smoothers (including local polynomial smoothers) and orthogonal series smoothers. The properties of these smoothing techniques are well known and details can be found in several monographs (Eubank, 1988; Fan and Gijbels, 1996; Green and Silverman, 1994; Härdle, 1990; Müller, 1988; Scott, 1992; Simonoff, 1996; Thompson and Tapia, 1990; Wahba, 1990; Wand and Jones, 1995) and the references given therein.

However, in some situations the researcher would like to combine the flexibility of nonparametric smoothing techniques with prior knowledge about the underlying regression curve given by, say, a physical or economic theory. For example, economic theory predicts that when producers are maximising profit, the observed relationship between input and output will be concave and non-decreasing. Matzkin (1991) studies the estimation of monotone and concave utility functions while Hildreth (1954) considers the problem of estimating production functions and Engel curves. A recent review is given by Matzkin (1994). Applications in which one wishes to impose monotonicity constraints include, among others, nonparametric calibration (see, e.g. Knafl *et al.*, 1984), the estimation of dose-response curves (see, e.g. Kelly and Rice, 1990) and the estimation of cumulative distribution functions (see, e.g. Gaylord and Ramirez, 1991).

Mammen and Thomas-Agnan (1996) study the behaviour of smoothing splines under various shape restrictions. Specifically, given data $\{(t_i, y_i)\}$, $t_i \in [a, b]$ for $i = 1, \ldots, n$, they investigate the behaviour of the solution $\hat{g}$ of the following minimisation problem:

$$\text{minimise} \qquad \sum_{i=1}^{n} \left\{ y_i - g(t_i) \right\}^2 + \lambda \int_a^b \left\{ g^{(m)}(u) \right\}^2 du, \tag{1.1a}$$

$$\text{where} \qquad g^{(r)}(t) \geq 0 \quad t \in [a, b]. \tag{1.1b}$$

Mammen and Thomas-Agnan (1996) derive convergence rates for $\hat{g}$ for the general case and give a detailed analysis for the case $m = r = 2$. They also show that the infinite number of constraints (1.1b) can be replaced by a finite number of constraints with only a small loss of accuracy.

It is well known that the (unconstrained) minimiser $\hat{g}$ of (1.1a) (over a suitable function class) is a natural spline of order $2m$ with knots at the observation points $t_i$ (see, e.g. Eubank, 1988). For the case $m = 2$, we present in this paper an algorithm for calculating a natural cubic spline that fulfils the constraints (1.1b) for any $r \leq 2$ or any combination of such $r$'s, respectively. We comment on the more general problem (1.1) in Section 4.

Since the constraints (1.1b) are to be satisfied on an infinite set, problem (1.1) is a so-called semi-infinite optimisation problem. In Section 2 we shall show that the minimiser $\hat{g}$ of (1.1a) can be written as a solution of a quadratic programming problem. We shall replace the infinite number of constraints (1.1b) by a finite number of constraints that are chosen adaptively. This reduction to a finite number of constraints leads us back to a quadratic programming problem and we shall show that it is feasible to calculate the solution of the resulting quadratic program. The constraints that we impose are in the spirit of Hawkins (1994) and will guarantee that (1.1b) holds.

Clearly, our algorithm does not calculate the solution of (1.1) as its result is a cubic $C^2$-spline with knots at the $t_i$'s and fulfilling (1.1b). By way of contrast, Utreras (1985) shows, for $m = 2$ and $r = 1$ (monotonicity), that the solution to (1.1) is a piecewise cubic polynomial with knots at the points $t_i$ and at most $k = 2[n/2]+2$ additional knots whose location is unknown. He suggests that an exchange algorithm could be devised to find the solution to (1.1) but to the best of our knowledge such an algorithm is not yet proposed. Likewise, for the case $m = 2$ and $r = 2$ (convexity-concavity), Elfving and Andersson (1988) give a complete characterisation theorem for the solution of (1.1). This leads to a nonlinear minimisation problem that the authors propose to solve by Newton's method.

Given these difficulties for finding the exact solution of (1.1) most algorithms proposed calculate approximate solutions. Monotonicity constraints are considered in Ramsay (1988) and Gaylord and Ramirez (1991). Fritsch (1990) proposes an algorithm for calculating a monotone cubic $C^1$-spline. Tantiyaswasdikul and Woodroofe (1994) investigate problem (1.1) with $m = 1$ and $r = 1$ (monotonicity). Algorithms for estimating a convex-concave function are given in Dierckx (1980), Irvine *et al.* (1986) (see also Micchelli *et al.*, 1985), Schmidt (1987) and Schmidt and Scholz (1990). Schwetlick and Kunert (1993) propose a method for calculating an approximate solution of (1.1) for general $m$ and $r$ but concentrate their exposition on the convex-concave case. The estimation of monotone convex-concave functions is discussed in Dole (1996).

Typically, these approaches involve the choice of an appropriate $B$-spline basis which translates the minimisation problem (1.1a) into a least-squares problem for the coefficients of the basis functions. The constraints (1.1b) will impose some constraints on these coefficients which typically leads to a quadratic programming problem that has to be solved. Some of these algorithms are quite sophisticated and allow automatic location of knot points (Dierckx, 1980; Schwetlick and Kunert, 1993).

Our approach differs in several ways from those cited above. First, instead of choosing a suitable $B$-spline basis and identifying the necessary constraints on the coefficients of the basis functions, we fit an (unconstrained) smoothing spline in the first step. Next, we verify whether this estimate has the desired shape. If there are any violations, we add constraints that ensure that the smoothing spline will have the desired shape at those points where it currently does not fulfill the shape restrictions. Then the (constrained) smoothing spline is updated and the process of verifying-and-adding-new-constraints is iterated. Secondly, most of the above proposals treat problem (1.1) for one $r$ only. By way of contrast, our proposal is easily extended to the case in which (1.1b) should hold for several $r$ simultaneously.

Another approach to constrained smoothing splines is described in Ramsay (1997). The author proposes to impose the smoothness penalty in (1.1a) on a function, say $w$, which is related to the function $g$ via a differential equation. By choosing an appropriate differential equation one can ensure that $g$ has desired properties such as positivity, monotonicity or convexity. This approach seems to be especially attractive if one wants to ensure that (1.1b) holds with strict inequality almost everywhere. Again, it is not obvious how to use this approach if (1.1b) should hold for several $r$ simultaneously (i.e. what is the appropriate differential equation) and also the numerical implementation of this approach seems to be difficult.

Finally, we would like to note that spline smoothers are not the only nonparametric smoothers that can be used when estimating functions with shape restrictions (see, e.g. Marron *et al.*, 1996). Especially for monotone smoothing several alternative approaches were proposed (see, e.g. Delecroix *et al.*, 1995, 1996; Friedman and Tibshirani, 1984). Nonparametric estimation of convex-concave functions are discussed in Dent (1973), Holm and Frisén (1985) and Fraser and Massam (1989). A review of nonparametric smoothing under shape restrictions is given in Delecroix and Thomas-Agnan (1997).

In Section 2 we shall describe our approach for calculating constrained smoothing splines. This approach is illustrated on several data sets in Section 3. Conclusions and a discussion of further generalisations of our approach are given in Section 4.

# 2 Proposed Estimator

We assume, without loss of generality, that $a = t_0 < t_1 < t_2 < \cdots < t_n < t_{n+1} = b$. The (unconstrained) minimiser $\hat{g}$ of (1.1a) over a suitable function class is a natural spline of order $2m$ with knots at the observation points $t_i$. This means that $\hat{g}$ (i) is piecewise polynomial of order $2m$ on any subinterval $[t_i, t_{i+1})$, (ii) has $2m-2$ continuous derivatives, (iii) has an $(2m-1)$-st derivative that is a step function with jumps at $t_1, \ldots, t_n$ and (iv) is a polynomial of order $m$ outside of $[t_1, t_n]$.

In this section we shall restrict our discussion to the case $m = 2$ and $r \leq 2$ as these are the most important in practice. A general approach to find an approximate solution to (1.1) is discussed in Section 4. For $m = 2$, the piecewise polynomial representation of a natural cubic $C^2$-spline $g$ is:

$$g(t) = \sum_{i=0}^{n} I_{[t_i, t_{i+1})}(t) S_i(t), \tag{2.1a}$$

where $$S_i(t) = a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3, \quad 1 \leq i \leq n-1, \tag{2.1b}$$

$$S_0(t) = a_1 + b_1(t - t_1) \quad \text{and} \quad S_n(t) = S_{n-1}(t_n) + S'_{n-1}(t_n)(t - t_n).$$

The coefficients in (2.1b) have to fulfill the following equations for $g$ to be a natural cubic $C^2$-spline:

$$\begin{aligned} S_{i-1}(t_i) &= S_i(t_i) \quad \text{for } i = 1, \ldots, n \\ S'_{i-1}(t_i) &= S'_i(t_i) \quad \text{for } i = 1, \ldots, n \\ S''_{i-1}(t_i) &= S''_i(t_i) \quad \text{for } i = 1, \ldots, n \end{aligned} \tag{2.2}$$

The piecewise polynomial representation (2.1) is convenient to describe the constraints that we impose to ensure that the shape restriction (1.1b) hold. However, a direct implementation would lead to an unnecessarily large quadratic programming problem and we propose to use the *value-second derivative representation* (see Green and Silverman, 1994, chapter 2) for the actual implementation. In Appendix A we give transformation formulae for changing from one representation to the other.

To derive the value-second derivative representation we follow the notation of Green and Silverman (1994). For $i = 1, \ldots, n$, define $g_i = g(t_i)$ and $\gamma_i = g''(t_i)$. By definition, a natural cubic $C^2$-spline has $\gamma_1 = \gamma_n = 0$. Let $\boldsymbol{g}$ denote the vector $(g_1, \ldots, g_n)^T$ and $\boldsymbol{\gamma} = (\gamma_2, \ldots, \gamma_{n-1})^T$. Note that for notational simplicity later on the entries of $\boldsymbol{\gamma}$ are numbered in a non-standard way, starting at $i = 2$. The vectors $\boldsymbol{g}$ and $\boldsymbol{\gamma}$ specify the natural cubic spline $g$ completely.

However, not all possible vectors $\boldsymbol{g}$ and $\boldsymbol{\gamma}$ represent natural cubic splines. To derive sufficient (and necessary) conditions for $\boldsymbol{g}$ and $\boldsymbol{\gamma}$ to represent a cubic spline we define the following matrices $Q$ and $R$. Define $h_i = t_{i+1} - t_i$ for $i = 1, \dots, n-1$. Let $Q$ be the $n \times (n-2)$ matrix with entries $q_{i,j}$, for $i = 1, \dots, n$ and $j = 2, \dots, n-1$, given by

$$q_{j-1,j} = h_{j-1}^{-1}, \quad q_{j,j} = -h_{j-1}^{-1} - h_j^{-1}, \quad \text{and} \quad q_{j,j+1} = h_j^{-1},$$

for $j = 2, \dots, n-1$, and $q_{i,j} = 0$ for $|i-j| \geq 2$. Note, that the columns of $Q$ are numbered in the same non-standard way as the entries of $\boldsymbol{\gamma}$.

The $(n-2) \times (n-2)$ matrix $R$ is symmetric with elements $\{r_{i,j}\}_{i,j=2}^{n-1}$ given by

$$\begin{aligned} r_{i,i} &= \tfrac{1}{3}(h_{i-1} + h_i) \quad \text{for } i = 2, \cdots, n-1, \\ r_{i,i+1} = r_{i+1,i} &= \tfrac{1}{6} h_i \quad \text{for } i = 2, \cdots, n-2, \end{aligned}$$

and $r_{i,j} = 0$ for $|i-j| \geq 2$. Note, that $R$ is strictly diagonal dominant and, hence, it follows from standard arguments in numerical linear algebra, that $R$ is strictly positive-definite.

We are now able to state the following key result.

**Proposition 2.1** *The vectors $\boldsymbol{g}$ and $\boldsymbol{\gamma}$ specify a natural cubic spline $g$ if and only if the condition*

$$Q^T \boldsymbol{g} = R\boldsymbol{\gamma} \tag{2.3}$$

*is satisfied. If* (2.3) *is satisfied then we have*

$$\int_a^b \{g''(t)\}^2 \, dt = \boldsymbol{\gamma}^T R \boldsymbol{\gamma}. \tag{2.4}$$

For a proof see Green and Silverman (1994, section 2.5).

This result allows us to state problem (1.1a) as a quadratic programming problem. Let $\mathfrak{y}$ denote the $(2n-2)$-vector $(y_1, \dots, y_n, 0, \dots, 0)^T$, $\mathfrak{g}$ the $(2n-2)$-vector $(\boldsymbol{g}^T, \boldsymbol{\gamma}^T)^T$, $A$ the $(2n-2) \times (n-2)$-matrix $\begin{pmatrix} Q & -R^T \end{pmatrix}$, $I_n$ the $n \times n$ unit matrix and

$$\mathfrak{D} = \begin{pmatrix} I_n & 0 \\ 0 & \lambda R \end{pmatrix}. \tag{2.5}$$

Then the solution of (1.1a) is given by the solution of the following quadratic program:

$$\text{minimise} \qquad -\mathfrak{y}^T\mathfrak{g} + \tfrac{1}{2}\mathfrak{g}^T\mathfrak{D}\mathfrak{g}, \tag{2.6a}$$

$$\text{where} \qquad A^T\mathfrak{g} = 0. \tag{2.6b}$$

In (2.6b) the constraints have to be read elementwise for the vector $A^T\mathfrak{g}$. Green and Silverman (1994) describe a fast algorithm to solve (2.6) which goes back to work of Reinsch (1967, 1971). This algorithm calculates the solution of (2.6) in $O(n)$ operations.

We propose to use the algorithm of Goldfarb and Idnani (1982, 1983) to solve (2.6). This algorithm is well-suited for our purposes as it handles equality as well as inequality constraints and calculates the optimal solution of the quadratic programming problem iteratively by successively making violated constraints active. Specifically, the algorithm starts with setting the set of active constraints to the null set and calculating the solution of the unconstrained problem (which is $\mathfrak{y}$ in our setup). After calculating this initial solution the algorithm iterates as follows. All constraints are checked and if none is violated we have found the optimal solution of our quadratic program. If any constraints are violated, choose one of them, add it to the set of active constraints and update the solution such that it is the minimiser of the (quadratic) target function under the constraints given in the active set. If a constraint in the active set is automatically fulfilled by the addition of a new constraint, it is dropped from the set of active constraints during this updating step. Goldfarb and Idnani (1983) show that their algorithm calculates the (unique) solution of a quadratic programming problem in finitely many steps as long as the matrix appearing in the quadratic criterion (which is $\mathfrak{D}$ in our setup) is positive definite and the constraints are consistent.

The algorithm of Goldfarb and Idnani is especially attractive for our purposes as its implementation allows to add adaptively constraints during the iterations. This changes of course the underlying quadratic program that is solved. However, as long as only a finite number of constraints are added, the "final" quadratic program is identical to the one we would have had if we would start with all constraints (initial ones and adaptively added ones) in the first place. Hence, the algorithm of Goldfarb and Idnani will still calculate in finitely many steps the solution of the "final" quadratic program. We shall use this feature to add constraints to (2.6b) which ensure that (1.1b) will hold for the final solution. These constraints are described in the following sections and it will be clear that in each case we add only a finite number of constraints. Moreover, the complete set of constraints will be consistent at each step.

## 2.1 Convex-concave smoothing

Since a natural cubic splines has by definition a piecewise linear second derivative it is sufficient to constrain the second derivatives of the solution at all knot points to obtain a convex-concave smoothing splines. Hence, for an approximate solution of (1.1) with $r = 2$, i.e. to impose convexity or concavity, we add the constraints

$$\gamma_i \geq 0 \quad \text{for} \quad i = 2, \ldots, n-1, \tag{2.7}$$

respectively

$$\gamma_i \leq 0 \quad \text{for} \quad i = 2, \ldots, n-1, \tag{2.8}$$

to (2.6b). Recall that $\gamma_1 = \gamma_n = 0$ by definition and we do not need to constrain them. The solution $\hat{\mathfrak{g}} = (\hat{\boldsymbol{g}}^T, \hat{\boldsymbol{\gamma}}^T)^T$ of this quadratic program defines our smoothing spline estimate $\hat{g}$.

## 2.2 Monotone smoothing

For the sake of simplicity, we will concentrate here on monotone increasing smoothers, i.e. (1.1b) with $r = 1$. The case of monotone decreasing smoothers can be treated analogously. To impose monotonicity the simple approach of Section 2.1 does not work. Imposing that $g'$ is non-negative at all knot points will not necessarily guarantee that this holds for $g'$ between the knot points. However, for $g'$ to be non-negative, it is necessary that $g'(t_i) \geq 0$ for $i = 1, \ldots, n$. Hence, we add the following constraints to (2.6b):

$$\begin{aligned} b_i &= \frac{g_{i+1} - g_i}{h_i} - \frac{h_i}{6}(2\gamma_i + \gamma_{i+1}) \geq 0 \quad \text{for} \quad i = 1, \ldots, n-1, \quad \text{and} \\ b_n &= \frac{g_n - g_{n-1}}{h_{n-1}} + \frac{h_i}{6}\gamma_{n-1} \geq 0 \end{aligned} \tag{2.9}$$

Note, that these constraints ensure that $g'$ is non-negative on $[a, t_1]$ and $[t_n, b]$. To ensure that $g'$ is also non-negative on $[t_1, t_n]$ we have to add further constraints. The constraints that we add are in the spirit of Hawkins (1994). Specifically, we solve (2.6a) initially under the constraints given by (2.6b) and (2.9). For the solution of this problem we check whether $g'$ is positive on each subinterval. Since $g'$ is quadratic on each subinterval $[t_i, t_{i+1})$, $i = 1, \ldots, n-1$, this is easily done by the following steps.

1. If $d_i = (\gamma_{i+1} - \gamma_i)/(6h_i) \leq 0$ then $g'$ is concave on the subinterval and hence

positive by (2.9).

2. Otherwise, we calculate the critical point $t_{0i} = t_i - (c_i/3d_i)$ of $g'$. If $t_{i0}$ lies outside the subinterval then $g'$ is positive on the subinterval by (2.9).

3. Otherwise, we have to determine the sign of the extreme value of $g'$, this is given by the sign of $3\,d_i\,b_i - c_i^2$. If the extreme value is non-negative then this is also true for $g'$ on the subinterval by (2.9). Otherwise $g'$ becomes negative on the subinterval.

For the subintervals on which $g'$ becomes negative we introduce further constraints, namely, that the extremum of $g'$ will still occur at the point $t_{0i}$ and that the extreme value will be non-negative. Note, that the constraints $g'(t_i) \geq 0$, $g'(t_{0i}) \geq 0$, $g'(t_{i+1}) \geq 0$ and $g''(t_{0i}) = 0$ ensure that $g'$ is non-negative on the interval $[t_i, t_{i+1})$. Hence, for each subinterval we impose at most two additional constraints. We also remark that these constraints can be motivated by the observation that we force $g'$ to respect (1.1b) at the point where it "most" violates (1.1b) while maintaining the "shape" of $g'$ as defined by the location of its critical point.

Clearly, if we add a pair of constraints for the interval $[t_i, t_{i+1})$ at least the constraint $g'(t_{0i}) \geq 0$ will become active. Hence, the current solution $\hat{\mathfrak{g}}$ to the quadratic program will change. To avoid that our final result depends on the sequence in which we impose the additional constraints, we add at each iteration all necessary additional constraints simultaneously. Thus our approach to calculate a monotone smoothing spline is as follows.

1. Solve the quadratic program given by (2.6) and (2.9) using the algorithm of Goldfarb and Idnani (1982, 1983).

2. The solution $\hat{\mathfrak{g}} = (\hat{\boldsymbol{g}}^T, \hat{\boldsymbol{\gamma}}^T)^T$ of the current quadratic program defines our smoothing spline estimate $\hat{g}$. Check for each subinterval $[t_i, t_{i+1})$, $i = 1, \ldots, n-1$, whether $\hat{g}'$ is non-negative. Let $\mathcal{J}$ denote the set of all indices $j$ such that $\hat{g}'$ becomes negative on $[t_j, t_{j+1})$.

3. If $\mathcal{J} = \emptyset$ then stop. Otherwise add the following constraints to the quadratic program:

$$\begin{aligned} b_j + 2\,c_j\,\Delta_j + 3\,d_j\,\Delta_j^2 \geq 0 \quad &\text{for all} \quad j \in \mathcal{J} \\ 2\,c_j + 6\,d_j\,\Delta_j = 0 \quad &\text{for all} \quad j \in \mathcal{J} \end{aligned} \tag{2.10}$$

where $\Delta_j = t_{0j} - t_j$. (Use the formulae in Appendix A to transform these constraints into constraints on $\mathfrak{g}$, i.e. on $\boldsymbol{g}$ and $\boldsymbol{\gamma}$.)

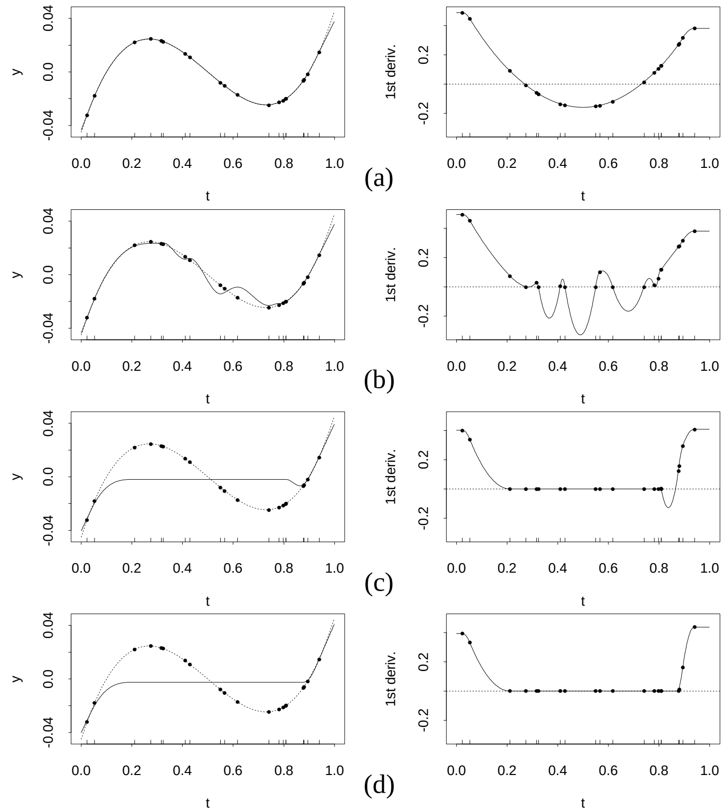


***Figure 2.1:*** *Example for calculating a monotone smoothing spline. The pictures in the left column show the 20 observations (points), the true function (dashed line) and the smoothing spline estimates (solid line). The pictures in the right column show the corresponding first derivative of the spline estimates (solid line) and the horizontal line $y = 0$ (dashed line). The observation points, which are also the knots of the spline estimates, were drawn from an uniform distribution and are indicated as a "rug" in each picture. Panel (a) is the solution of* (2.6)*. In Panel (b) the constraints* (2.9) *are added. Panel (c) depicts the result after adaptively adding constraints* (2.10) *the first time. Panel (d) shows the final results after two iterations.*

4. Continue with the algorithm of Goldfarb and Idnani (1982, 1983) to find the solution of the augmented quadratic program and goto Step 2.

In our experience a monotone smoothing spline is usually calculated after one iteration of Steps 2–4. In rare occasions a second iteration is needed. Figure 2.1 demonstrates the proposed algorithm. Here, we have chosen an underlying function that is not monotone, observations with no noise and a very small value of $\lambda$ such that the (unconstrained) smoothing spline is practically interpolating the data. Although this setup is a kind of worst-case scenario we had to generate several samples before we found one in which two iterations of Steps 2–4 were necessary. This sample is presented in Figure 2.1.

## 2.3 Positive smoothing

An application in which one would like to impose positivity constraints is density estimation. By using binning techniques the density estimation problem can be transformed into a regression problem and positive smoothing splines can be used to estimate the probability density function (see, e.g. Eilers and Marx, 1996). Other approaches to density estimation using splines are discussed in Kooperberg and Stone (1991) and Terrell (1993).

To fit a positive smoothing spline we proceed analogously to Section 2.2. First we impose the following necessary conditions additionally to (2.6b):

$$a_i = g_i \geq 0 \quad \text{for} \quad i = 1, \ldots, n. \tag{2.11}$$

Here, we also have to decide which behaviour we would like to have outside the interval $[t_1, t_n]$. If we wish that the smoothing spline is positive on $(-\infty, t_1]$ or $[t_n, \infty)$, then the additional constraints $b_1 \geq 0$ or $b_n \geq 0$, respectively, are necessary and sufficient. If positivity is only requested for the intervals $[a, t_1]$ or $[t_n, b]$, then the additional constraints $a_1 + b_1(a - t_1) \geq 0$ or $a_n + b_n(b - t_n) \geq 0$, respectively, are necessary and sufficient. Depending on the application, we may also choose to impose no constraints outside the interval $[t_1, t_n]$.

Again, we calculate the solution of the quadratic program given by (2.6) and (2.11) first. Then we check whether the solution $\hat{g}$ is positive on each subinterval $[t_i, t_{i+1})$, $i = 1, \ldots, n-1$, as follows:

1. Calculate the critical points of $\hat{g}$ in the subinterval: Define $\alpha_i = b_i - 2\,c_i\,t_i + 3\,t_i^2$, $\beta_i = 2\,(c_i - 3\,d_i\,t_i)$ and $\delta_i = 3\,d_i$. Set $\tau_i = \beta_i^2 - 4\,\alpha_i \delta_i$. If $\tau_i < 0$, then $\hat{g}$ has no critical points in the subinterval and is positive by (2.11). If $\tau_i = 0$, then $\hat{g}$ has only one critical point which is necessarily an inflection point. Hence $\hat{g}$ is monotone on the subinterval and positive by (2.11).

2. Otherwise the two critical points are

$$t_{1i} = -\frac{\beta_i + \sqrt{\tau_i}}{2\delta_i} \quad \text{and} \quad t_{2i} = -\frac{\beta_i - \sqrt{\tau_i}}{2\delta_i}.$$

If none of the critical points lies in the subinterval then $\hat{g}$ is monotone on the subinterval and hence positive by (2.11).

3. Otherwise, we have to determine the value(s) of $\hat{g}$ at the critical point(s) that lie within the subinterval. If these extreme-values are non-negative then this is also true for $\hat{g}$ on the subinterval by (2.11). Otherwise $\hat{g}$ becomes negative on the subinterval.

Note that there is at most one critical point in each interval at which the value of $\hat{g}$ may be negative. Without loss of generality let this critical point be $t_{1i}$. To calculate a positive spline fit we again add adaptively constraints to the quadratic program which will ensure that (1.1b) holds at $t_{1i}$ and conserve the current "shape" of $\hat{g}$, i.e. $t_{1i}$ will remain the critical point at which $g$ takes its minimum on $[t_i, t_{i+1})$. Specifically, we propose the following procedure to calculate a positive smoothing spline:

1. Solve the quadratic program given by (2.6) and (2.11) using the algorithm of Goldfarb and Idnani (1982, 1983).

2. Check for each subinterval $[t_i, t_{i+1})$, $i = 1, \ldots, n-1$, whether $\hat{g}$ is non-negative. Let $\mathcal{J}$ denote the set of all indices $j$ such that $\hat{g}$ becomes negative on $[t_j, t_{j+1})$ and let $t_{1j}$ denote the critical point at which $\hat{g}$ is negative.

3. If $\mathcal{J} = \emptyset$ then stop. Otherwise add the following constraints to the quadratic program

$$\begin{aligned} a_j + b_j\,\Delta_j + c_j\,\Delta_j^2 + d_j\,\Delta_j^3 &\geq 0 \quad \text{for all} \quad j \in \mathcal{J} \\ b_j + 2\,c_j\,\Delta_j + 3\,d_j\,\Delta_j^2 &= 0 \quad \text{for all} \quad j \in \mathcal{J} \\ 2\,c_j + 6\,d_j\,\Delta_j &\geq 0 \quad \text{for all} \quad j \in \mathcal{J} \end{aligned} \tag{2.12}$$

   where $\Delta_j = t_{1j} - t_j$. (Use the formulae in Appendix A to transform these constraints into constraints on $\mathfrak{g}$, i.e. on $\boldsymbol{g}$ and $\boldsymbol{\gamma}$.)

4. Continue with the algorithm of Goldfarb and Idnani (1982, 1983) to find the solution of the augmented quadratic program and goto Step 2.

Note, that we have to impose at most once constraints of type (2.12) in each subinterval. An alternative approach would be to impose only the first two constraints in (2.12). Since we impose that $t_{1j}$ remains a critical value this ought to be sufficient as, except for extreme cases, one would not expect that $t_{1j}$ changes from the minimiser of $\hat{g}$ on $[t_i, t_{i+1})$ to the maximiser of $\hat{g}$. In any case, also with this alternative approach we would at most add twice additional constraints in each subinterval.

## 2.4 Mixed constraints

In some applications one wishes to impose constraints of type (1.1b) for several $r$, e.g. Dole (1996) investigates the estimation of monotone convex-concave functions. These cases are easily treated with our approach. Depending on the combination of constraints that one wishes to impose there may be some simplifications that we shall discuss now.

If we impose concavity-convexity constraints then the first derivative of $\hat{g}$ will be automatically monotone. Hence, to fit for example a monotone increasing convex smoothing spline we only have to solve the quadratic program given by (2.6), (2.7) and $b_1 = \frac{1}{h_1}(g_2 - g_1) - \frac{h_1}{6}\gamma_2 \geq 0$. In this case no constraints have to be added adaptively. Other combinations of monotone increasing/decreasing and convexity-concavity can be treated analogously.

Likewise, if we wish to estimate a positive concave function with our approach the smoothing spline is given by the solution of the quadratic program defined by (2.6), (2.8) and (2.9). Here, additional constraints for the behaviour of the spline outside of $[t_1, t_n]$ may be imposed as discussed after (2.9).

A positive convex function assumes its minimum either at a single point or on a (connected) interval and has no other critical points. Unfortunately, the point or interval, respectively, at which the minimum is attained are not know a priori. Hence, to estimate a positive convex function we would start with the quadratic program given by (2.6), (2.7) and (2.9). If the solution of this problem is positive we stop. Otherwise we can either just add additional constraints as described in Section 2.3 or determine the global minimiser of the solution and impose additional constraints similar to (2.12) in the subinterval where the global minimum is attained only. In either case, we would have to perform at most one iteration of adding constraints.

Finally, to estimate a positive monotone function on $[t_1, t_n]$, we would just follow the procedure described in Section 2.2 but add the constraint $g_1 \geq 0$ or $g_n \geq 0$ to (2.9) depending on whether we want to estimate a monotone increasing or monotone decreasing function. (Again, we might choose to add further constraints to control the behaviour of the smoothing spline outside the interval $[t_1, t_n]$.)

# 3 Applications

In this section we shall give several examples of the performance of our method by applying it to several data sets.

The first example is the volumetric moisture content data given in table 2 of Dierckx

(1980). To compare our approach with the method of Dierckx (1980) we need a slight modification since the observations in this data set have different weights $w_i$, i.e. we would like to solve the minimisation problem

$$\text{minimise} \quad \sum_{i=1}^{n} w_i \left\{ y_i - g(t_i) \right\}^2 + \lambda \int_a^b \left\{ g''(u) \right\}^2 du$$

with the constraint that $g$ is concave. To take the weights into account, we just have to define the vector $\mathfrak{y}$ as the $(2n-2)$ vector $(w_1 y_1, \dots, w_n y_n, 0 \dots, 0)^T$ and put the weights $w_i$ into the diagonal of the upper $n \times n$-submatrix of $\mathfrak{D}$ in (2.5). Dierckx (1980) presents three fits to this data using $B$-splines based on 2, 3 and 5 knots. The sum of squared residuals of his constrained fits are 0.10991, 0.033478 and 0.00019375, respectively. Using the same knots but without the concavity constraints, his fits have a sum of squared residuals of 0.075099, 0.023544 and 0.000111, respectively. By careful choice of $\lambda$ we obtained concave smoothing splines using our method which have similar sum of squared errors as the fits of Dierckx. These fits are shown in Figure 3.1. The sum of squared errors for the constrained fits are 0.1098721 $(\lambda = 17)$, 0.03371643 $(\lambda = 0.4)$ and 0.0001925 $(\lambda = 0.00055)$. If we drop the concavity constraints, the sum of squared residuals are 0.1098721, 0.03358064 and 0.0001023, respectively. Comparing Figure 3.1 with figure 7.a in Dierckx (1980) we see that the two approaches give similar fits. However, it seems that our fits stay “closer” to the data for large values of $\lambda$ than the corresponding fits of Dierckx. This is due to the fact that our approach allows the smoothing spline more flexibility as it has knots at each observation point. The larger flexibility is also manifested by the observation that for large $\lambda$ the difference between the sum of squared residual for the constrained and unconstrained fit differ less than in Dierckx’s approach.

Our second example are the titanium data of DeBoor (1978) which are also considered in Elfving and Andersson (1988) and Schwetlick and Kunert (1993). Following the lead of Schwetlick and Kunert we use an extremely small smoothing parameter ($\lambda = 10^{-7}$) to calculate a “quasi interpolating” smoothing spline that we restrict to be convex on the subintervals $[595, 835)$ and $[955, 1075]$. The result is depicted in Figure 3.2(a). Note that the unconstrained smoothing spline fits the central part of the data nicely but displays oscillating behaviour in both tails due to its interpolating nature. The constrained smoothing spline forces these oscillations to disappear which is clearly visible in the magnification of the right part of Figure 3.2(a) shown in Figure 3.2(b). This result demonstrates that concavity-convexity constraints (and/or monotonicity constraints) force the curve estimate to be smooth. This was also noticed

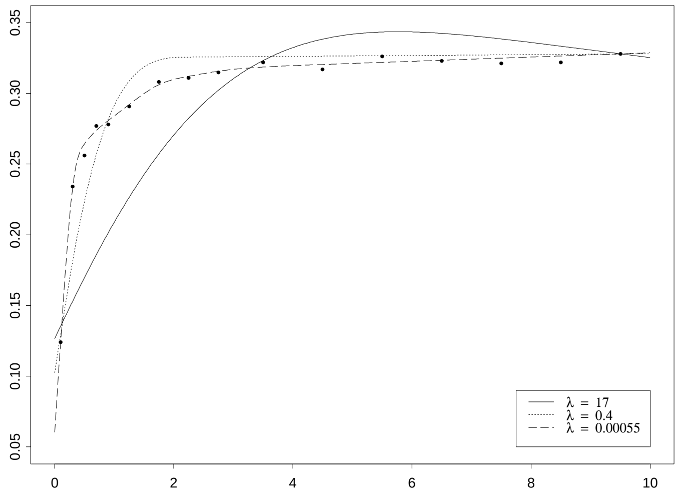


***Figure 3.1:*** *Three fits to the volumetric moisture content data (shown as points) given in table 2 of Dierckx (1980). The smoothing parameters $\lambda$ were chosen such that the fits have a similar sum of squared residuals as the fits presented in figure 7.a of Dierckx (1980).*

by Dole (1996), who remarks that "bounds on the first and second derivatives appear to be sufficient penalty for roughness" (page 2) and "lower (or upper) bounds on the first and second derivatives can be thought of as smoothing parameters, analogous to $\lambda$" (page 8). Comparing Figure 3.2 with figure 1 and figure 2 of Schwetlick and Kunert (1993) we see that our result are practically identical to the results in their example 1. This is not surprising since in this example Schwetlick and Kunert use a $B$-spline basis with knots at every observation point.

Our final example is a data set from DASL (1992). We selected the data set with the gold medal performances in the men's high jump, discus throw and long jump for the modern Olympic games from 1900 to 1984. All measurements are given in inches. We shall fit monotone smoothing splines to these data as it is often asserted that sportive performances are continuously improving. However, the documentation of this data set states that it "has been suggested that the Mexico City Olympics in 1968 saw unusually good track and field performances, possibly because of the high altitude" and that "scatterplots [ . . . ] show fall-off of performance with each of the two World Wars of the 20th century".

The results of fitting a monotone smoothing spline to the data using our approach are depicted in Figures 3.3–3.5. The smoothing parameter $\lambda$ was chosen visually in

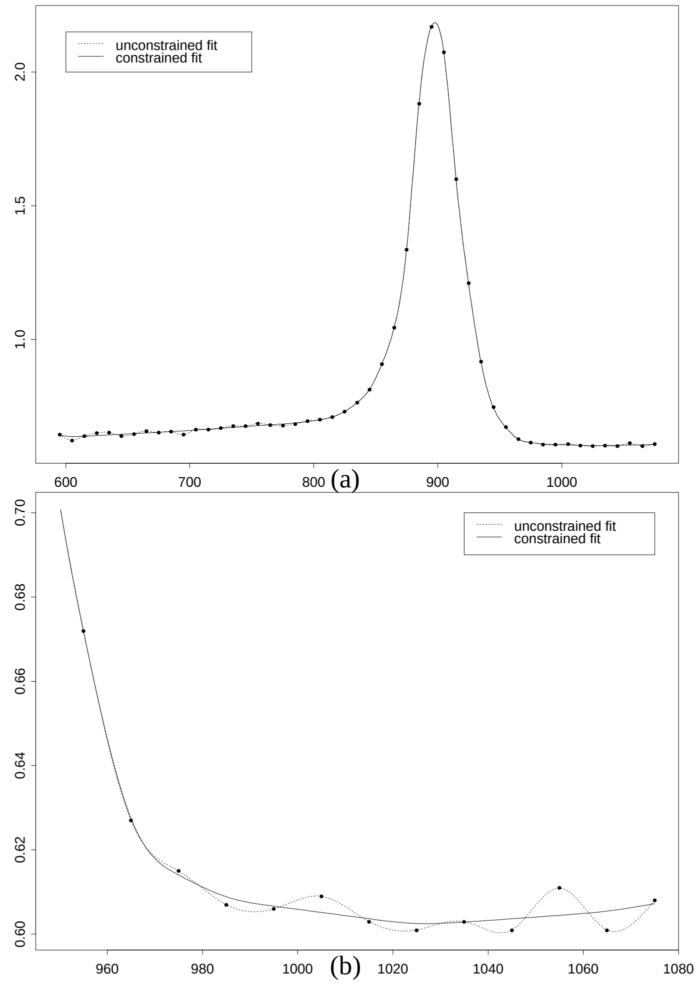


***Figure 3.2:*** *Fits to the Titanium data of DeBoor (1978). Panel (a) shows the data (points), an unconstrained smoothing spline estimate (dotted line) and the smoothing spline estimate constrained to be convex on* $[595, 835)$ *and* $[955, 1075]$ *(solid line). Panel (b) displays the enlarged right part of Panel (a). Similar plots are given in Schwetlick and Kunert (1993).*

each case. The upper left picture in each figure shows the observations as points, the solution to (2.6) as dotted line and the solution of the quadratic program given by (2.6) and (2.9) as solid line. The upper right pictures show the corresponding first derivatives. In each case we had to add only once additional constraints of type (2.10). The final solution is presented in the lower row of each figure. On the left we show the

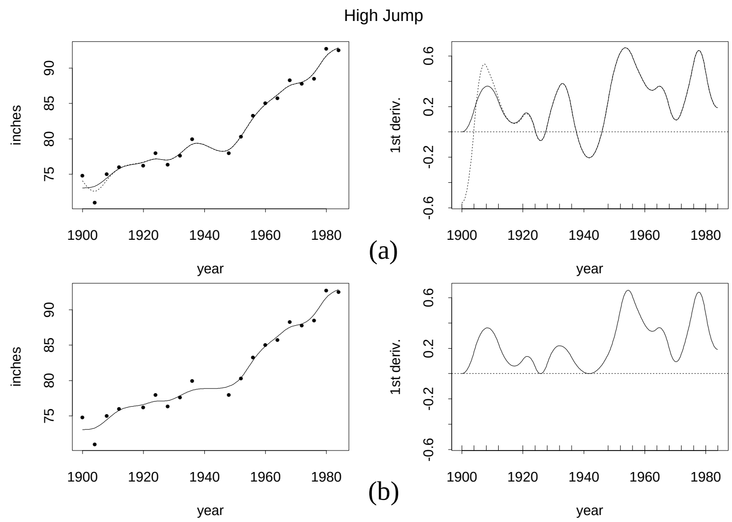


***Figure 3.3:*** *A monotone smoothing spline fitted to the high jump data. The pictures in the left column show the observations (points) and smoothing spline fit(s). The pictures in the right column show the corresponding first derivative of the spline fit(s) and the horizontal line* $y = 0$*. The observation points, which are also the knots of the spline fit, are indicated as a "rug" in each picture in the right column. Panel (a) shows the solution of* (2.6) *as dotted line and the solid line depicts the solution when constraints* (2.9) *are added. Panel (b) depicts the solution after adaptively adding constraints* (2.10)*. The smoothing parameter was* $\lambda = 10$*.*

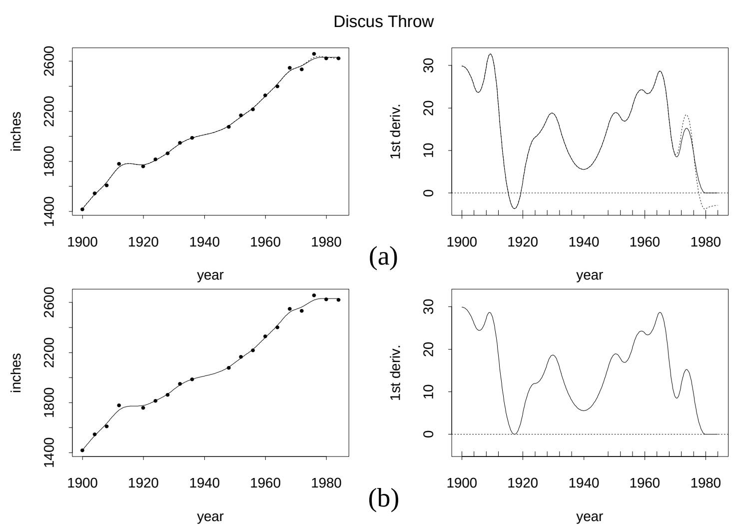


***Figure 3.4:*** *Same caption as for Figure 3.3 but depicting results for the discus throw data. The smoothing parameter was* $\lambda = 5$*.*

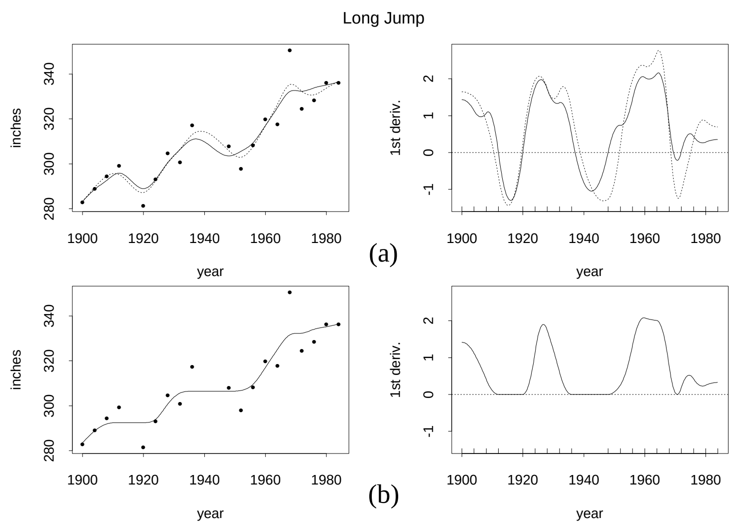


***Figure 3.5:*** *Same caption as for Figure 3.3 but depicting results for the long jump data. The smoothing parameter was* $\lambda = 20$.

observations together with the final fit (as solid line) and on the right the corresponding first derivative.

Observe that for the high jump and discus throw data the initial fit has already a positive derivative at all knots points except for the boundary. In each case the first derivative takes negative values between some knot points. Imposing constraints (2.10) we get monotone fits that seem to be reasonable for these two data sets given the initial unconstrained smooth. Note that, except during WWII in case of the high jump and WWI in case of the discus throw, the first derivatives of the initial smooths drop during the time of the World Wars and 1968 but remain positive. Hence, we would conclude that during these periods the performance in these two fields stagnated but did not drop. The performances in 1968 in these two categories should not be viewed as unusually good. Varying the smoothing parameter over a sensible range leads to the same conclusions but it would be of course desirable to develop some kind of tests for this kind of statements. For the case of monotone kernel smoothing some work is done by Bowman *et al*. (1996).

The fits to the long jump data show a completely different picture. First, this data set seems to have a much higher variability than the other two data sets. The derivative of the initial spline smoother becomes negative during each of the World Wars and around 1968. Of course, the performance in the long jump was quite exceptional in

1968 but there is also clear indication for a fall-off in performance during the World Wars. Again, it would be nice if we had a formal test for monotonicity, however, for these data it is fairly obvious that the underlying function is not monotone.

# 4 Further Generalisations and Conclusions

Our approach for calculating constrained smoothing splines is easily extended to other situations. Fisher *et al.* (1997) discuss the estimation of the boundary of a convex set from data on its support function $g$. Such a function $g$ is periodic on the interval $[-\pi, \pi)$. Furthermore, it is necessary (and sufficient) that $g + g'' \geq 0$. With our approach we would have to change $Q$ and $R$ such that they reflect a periodic spline function. Constraints that ensure that $g + g'' \geq 0$ could be imposed analogously to those discussed in Section 2.2.

When estimating a cumulative distribution function (see, e.g. Gaylord and Ramirez, 1991) one would also like to restrict the curve estimate such that $0 \leq g \leq 1$. This is easily done with our approach by fitting a monotone curve as described in Section 2.2 with the additional constraints $g_1 \geq 0$ and $g_n \leq 1$. Villalobos and Wahba (1987) consider a similar problem for the bivariate case. They impose the restrictions on a finite grid. This approach does not guarantee that the smoothing spline will respect the restrictions everywhere. However, if the grid on which one imposes the constraints is fine enough, one can be reasonable sure in practice that the final smooth will fulfill the (infinite number of) constraints. However, a fine grid leads to a large number of constraints and most of them will typically be unnecessary in the sense that they are not active in the solution of the quadratic program. Our approach tries to impose a minimum number of additional constraints and we illustrated it for the univariate case. The extension of our approach to the multivariate case is currently under study.

In Section 2, we discussed how to find an approximate solution of (1.1) for $m = 2$. This choice of $m$ allowed us to use a special representation of the smoothing spline which reduced the number of parameters in the quadratic program significantly. For general $m$ such a representation does not seem to be readily available. On the other hand, the value-second derivative representation that we used makes the implementation of the additional constraints (2.10) and (2.12) a little cumbersome. For general $m$ we propose to use the piecewise polynomial representation similar to (2.1). Using this representation it is easy to evaluate the spline and its derivatives on each subinterval using standard algorithms. Hence, for general $m$ the procedure to find an approximate solution to (1.1) would be to solve first the quadratic program given by constraints similar to (2.2). After this initial step one could use for each $r$, for which one would like to impose (1.1b), a generic root-finding algorithm to determine on each subinterval the critical points of $\hat{g}^{(r)}$. If (1.1b) is violated at any of these critical points then we add constraints that ensure that in the next iteration (a) (1.1b) is fulfilled at the points in question and (b) that these points remain critical points of $\hat{g}^{(r)}$. Note that this approach leads to a solution with continuous $(2m-2)$-nd derivative. This is in contrast to the remark of Utreras (1985) that with his technique "the continuity of the $(2m-2)$-st derivative doesn't seem possible for $m > 2$".

In summary, we presented a new way of calculating constrained smoothing splines. Our method leads initially to a quadratic programming problem with further constraints being added adaptively to ensure that the final solution fulfils (1.1b). The examples in Section 3 show that our method works well in practice and it is easily adapted to different kinds of constraints as discussed in Section 2 and above. In our experience the resulting quadratic program (2.6) is numerically stable and can be easily solved with the algorithm described in Goldfarb and Idnani (1982, 1983), although it may be worthwhile to develop an algorithm that takes advantage of the sparse structure of $\mathfrak{D}$ in (2.6).

# Acknowledgements

I am grateful to Ming-Yen Cheng for helpful discussions. This work was carried out within the Cooperative Research Centre for Advanced Computational Systems established under the Australian Government's Cooperative Research Centres Program.

# References

Bowman, A., Jones, M.C. and Gijbels, I. (1996). Testing monotonicity of regression, *Discussion Paper 9610*, Institut de Statistique, Université Catholique de Louvain, Voie du Roman Pays 20, 1348 Louvain-la-Neuve, Belgium.

DASL (1992). The Data and Story Library, accessible electronically at StatLib (CMU). **URL:** *http://www.stat.cmu.edu/DASL/*

DeBoor, C. (1978). *A Practical Guide to Splines*, Vol. 27 of *Applied Mathematical Sciences*, Springer-Verlag, Berlin/Heidelberg.

Delecroix, M. and Thomas-Agnan, C. (1997). Kernel and spline smoothing under shape restrictions. Unpublished manuscript.

Delecroix, M., Simioni, M. and Thomas-Agnan, C. (1995). A shape constrained smoother: Simulation study, *Computational Statistics* **10**(2): 155–175.

Delecroix, M., Simioni, M. and Thomas-Agnan, C. (1996). Functional estimation under shape constraints, *Journal of Nonparametric Statistics* **6**(1): 69–89.

Dent, W. (1973). A note on least squares fitting of functions constrained to be either nonnegative, nondecreasing or convex, *Management Science* **20**(1): 130–132.

Dierckx, P. (1980). An algorithm for cubic spline fitting with convexity constraints, *Computing* **24**: 349–371.

Dole, D. (1996). Scatterplot smoothing subject to monotonicity and convexity. Unpublished manuscript.
**URL:** *http://www.general.uwa.edu.au/u/ddole/cosmo/links.html*

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

Elfving, T. and Andersson, L.E. (1988). An algorithm for computing constrained smoothing spline functions, *Numerische Mathematik* **52**: 583– 595.

Eubank, R.L. (1988). *Smoothing Splines and Nonparametric Regression*, Marcel Dekker, New York and Basel.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Application*, Vol. 66 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, New York.

Fisher, N.I., Hall, P., Turlach, B.A. and Watson, G.S. (1997). On the estimation of a convex set from noisy data on its support function, *Journal of the American Statistical Association* **92**(437): 84–91.

Fraser, D.A.S. and Massam, H. (1989). A mixed primal-dual bases algorithm for regression under inequality constraints. Application to convex regression, *Scandinavian Journal of Statistics* **16**: 65–74.

Friedman, J. and Tibshirani, R. (1984). The monotone smoothing of scatterplots, *Technometrics* **26**(3): 243–250.

Fritsch, F.N. (1990). Monotone piecewise cubic data fitting, *in* J.C. Mason and M.G. Cox (eds), *Algorithms for Approximation II*, Chapman and Hall, London, pp. 99–106.

Gaylord, C.K. and Ramirez, D.E. (1991). Monotone regression splines for smoothed bootstrapping, *Computational Statistics Quarterly* **6**(2): 85–97.

Goldfarb, D. and Idnani, A. (1982). Dual and Primal-Dual Methods for Solving Strictly Convex Quadratic Programs, *in* J.P. Hennart (ed.), *Numerical Analysis, Proceedings, Cocoyoc, Mexico 1981*, Vol. 909 of *Lecture Notes in Mathematics*, Springer-Verlag, Berlin, pp. 226–239.

Goldfarb, D. and Idnani, A. (1983). A numerically stable dual method for solving strictly convex quadratic programs, *Mathematical Programming* **27**: 1–33.

Green, P.J. and Silverman, B.W. (1994). *Nonparametric Regression and Generalized Linear Models*, Vol. 58 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, London.

Härdle, W. (1990). *Applied Non-parametric Regression*, Vol. 19 of *Econometric Society Monographs*, Cambridge University Press, Cambridge.

Hawkins, D.M. (1994). Fitting monotonic polynomials to data, *Computational Statistics* **9**(3): 233–247.

Hildreth, C. (1954). Point estimates of ordinates of concave functions, *Journal of the American Statistical Association* **49**: 598–619.

Holm, S. and Frisén, M. (1985). Nonparametric regression with simple curve characteristics, *Research Report 4*, Statistiska institutionen, Göteborgs Universitet, Viktoriagatan 13, S 41125 Göteborg, Sweden.

Irvine, L.D., Marin, S.P. and Smith, P.W. (1986). Constrained interpolation and smoothing, *Constructive Approximation* **2**(2): 129–151.

Kelly, C. and Rice, J. (1990). Monotone smoothing with application to dose-response curves and the assessment fo synergism, *Biometrics* **46**: 1071–1085.

Knafl, G., Sacks, J., Spiegelman, C. and Ylvisaker, D. (1984). Nonparametric calibration, *Technometrics* **26**(3): 233–241.

Kooperberg, C. and Stone, C.J. (1991). A study of logspline density estimation, *Computational Statistics & Data Analysis* **12**(3): 327–347.

Mammen, E. and Thomas-Agnan, C. (1996). Smoothing splines and shape restrictions, *Discussion Paper 87*, Sonderforschungsbereich 373, Humboldt-Universität zu Berlin.
**URL:** *ftp://amadeus.wiwi.hu-berlin.de/pub/papers/sfb373/sfb1996/dpsfb960087.ps.Z*

Marron, J.S., Turlach, B.A. and Wand, M.P. (1996). Local polynomial smoothing under qualitative constraints, Vol. 28 of *Computing Science and Statistics*, Interface Foundation of North America, Inc., pp. 647–652.
**URL:** *http://wwwmaths.anu.edu.au/˜berwin/psfiles/interface96.ps.gz*

Matzkin, R.L. (1991). Semiparametric estimation of monotone and concave utility functions for polychotomous choice models, *Econometrica* **59**(5): 1315–1327.

Matzkin, R.L. (1994). Restrictions of economic theory in nonparametric methods, *in* R.F. Engle and D.L. McFadden (eds), *Handbook of Econometrics*, Vol. 4, North Holland, Amsterdam, chapter 42, pp. 2523–2558.

Micchelli, C.A., Smith, P.W., Swetitis, J. and Ward, J.D. (1985). Constrained $L_p$ approximation, *Constructive Approximation* **1**(1): 93–102.

Müller, H.G. (1988). *Nonparametric Regression Analysis of Longitudinal Data*, Vol. 46 of *Lecture Notes in Statistics*, Springer-Verlag, New York.

Ramsay, J.O. (1988). Monotone regression splines in action (with discussion), *Statistical Science* **3**(4): 425–461.

Ramsay, J.O. (1997). Estimating smooth monotone functions. Unpublished manuscript.
**URL:** *ftp://ego.psych.mcgill.ca/pub/ramsay/monotone.ps.gz*

Reinsch, C.R. (1967). Smoothing by spline functions, *Numerische Mathematik* **10**: 177–183.

Reinsch, C.R. (1971). Smoothing by spline functions. II, *Numerische Mathematik* **16**: 451–454.

Schmidt, J.W. (1987). An unconstrained dual program for computing convex $C^1$-spline approximants, *Computing* **39**: 133–140.

Schmidt, J.W. and Scholz, I. (1990). A dual algorithm for convex-concave data smoothing by cubic $C^2$-splines, *Numerische Mathematik* **57**: 333–350.

Schwetlick, H. and Kunert, V. (1993). Spline smoothing under constraints on derivatives, *Bit* **33**: 512–528.

Scott, D.W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*, John Wiley & Sons, New York, Chichester.

Simonoff, J.S. (1996). *Smoothing Methods in Statistics*, Springer Series in Statistics, Springer-Verlag, New York.

Tantiyaswasdikul, C. and Woodroofe, M.B. (1994). Isotonic smoothing splines under sequential designs, *Journal of Statistical Planning and Inference* **38**: 75–88.

Terrell, G.R. (1993). Spline density estimators, *Proceedings of the Statistical Computing Section*, American Statistical Association, 1429 Duke Street, Alexandria, VA 22314.

Thompson, J.R. and Tapia, R.A. (1990). *Nonparametric Function Estimation, Modeling, and Simulation*, SIAM, Philadelphia.

Utreras, F.I. (1985). Smoothing noisy data under monotonicity constraints: Existence, characterization and convergence rates, *Numerische Mathematik* **47**: 611–625.

Villalobos, M. and Wahba, G. (1987). Inequality-constrained multivariate smoothing splines with application to the estimation of posterior probabilities, *Journal of the American Statistical Association* **82**(397): 239–248.

Wahba, G. (1990). *Spline Functions for Observational Data*, CBMS-NSF Regional Conference series, SIAM, Philadelphia.

Wand, M.P. and Jones, M.C. (1995). *Kernel smoothing*, Vol. 60 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, London.

# A Transformation formulae

Use the following formulae to change from the value-second derivative representation to the piecewise polynomial representation (2.1)

$$
\begin{aligned}
a_i &= g_i && \text{for} \quad i = 1, \ldots, n-1, \\
b_i &= \frac{g_{i+1} - g_i}{h_i} - \frac{h_i}{6}(2\gamma_i + \gamma_{i+1}) && \text{for} \quad i = 1, \ldots, n-1, \\
c_i &= \frac{\gamma_i}{2} && \text{for} \quad i = 1, \ldots, n-1, \quad \text{and} \\
d_i &= \frac{\gamma_{i+1} - \gamma_i}{6h_i} && \text{for} \quad i = 1, \ldots, n-1.
\end{aligned}
$$

Additionally,

$$
\begin{aligned}
&a_0 = a_1 = g_1, \quad a_n = g_n, \quad b_0 = b_1, \quad c_0 = d_0 = 0 = c_n = d_n = 0 \quad \text{and} \\
&b_n = S'_{n-1}(t_n) = b_{n-1} + 2c_{n-1}h_{n-1} + 3d_{n-1}h_{n-1}^2 \\
&\quad = \frac{g_n - g_{n-1}}{h_{n-1}} + \frac{h_i}{6}(\gamma_{n-1} + 2\gamma_n)
\end{aligned}
$$

Remember that $h_i = t_{i+1} - t_i$ for $i = 1, \ldots, n-1$ and $\gamma_1 = \gamma_n = 0$.

Changing from the piecewise polynomial representation (2.1) to the value-second derivative representation is trivially done by

$$
\begin{aligned}
g_i &= S_i(t_i) = a_i && \text{for} \quad i = 1, \ldots, n, \\
\gamma_i &= S_i''(t_i) = 2c_i && \text{for} \quad i = 2, \ldots, n-1, \quad \text{and} \\
\gamma_1 &= \gamma_n = 0.
\end{aligned}
$$