Beliakov, Gleb 2000, Shape preserving approximation using least squares splines, *Analysis in theory and applications*, vol. 16, no. 4, pp. 80-98.

Shape preserving approximation using least squares splines

Gleb Beliakov

School of Computing and Mathematics
Deakin University
Rusden campus, 662 Blackburn Rd.
Clayton 3168, Australia

Fax. +61 3 9244 7460
gleb@deakin.edu.au

Abstract
Least squares polynomial splines are an effective tool for data fitting, but they may fail to preserve essential properties of the underlying function, such as monotonicity or convexity. The shape restrictions are translated into linear inequality conditions on spline coefficients. The basis functions are selected in such a way that these conditions take a simple form, and the problem becomes non-negative least squares problem, for which effective and robust methods of solution exist. Multidimensional monotone approximation is achieved by using tensor-product splines with the appropriate restrictions. Additional interpolation conditions can also be introduced. The conversion formulas to traditional B-spline representation are provided.

Keywords: Least squares splines, monotone splines, monotone approximation, restricted least squares

Introduction

Interpolation and approximation with spline functions under monotonicity and convexity constraints have attracted substantial interest in the literature, specifically in computer aided design [1-3,6,7,15,16,19-21,24-27,31-45]. For certain sets of data, interpolating polynomial splines introduce extraneous inflection points [10,21], whereas in the problems of smoothing the data itself may not posses the desired characteristics due to observation errors. Monotonicity and convexity of the splines can be enforced by various methods. One of the early approaches to constrained splines (or splines in tension) is based on piecewise exponential interpolation [34,38,39]. The exponents arise as solutions to certain differential equations describing the physical model of constrained spline, the elastic band passing through the interpolation knots, pulled until all the extraneous inflection points have been straightened out.

The approaches involving polynomial splines [2,3,6,7,11,15,18,20,24-26,31,35,37] rely on introduction of additional interpolation knots or increasing spline deficiency. Typically, quadratic [15,26,31,37] or cubic C^2 or C^1 interpolants [2,6,7,11,21] are constructed. The corresponding algorithms and FORTRAN source code are widely available [16,17,26]. Projection approach was recently discussed in [40]. When the data is not monotone or convex, constrained smoothing splines can be used [41]. As with interpolation, additional approximation knots are introduced [3,16,18]. Extensions to bi-variate approximation of data on a rectangular grid have also been developed [7,11,15,25,27]. A review of these methods is given in [21].

The bi-variate and multivariate approximation of scattered data are substantially more complicated. One method is to use Powell-Sabin splines, with the additional shape restrictions if necessary [12,44]. Variational approach to spline approximation results in thin plate splines, and the constraints can be introduced there as well [42,43]. This approach is very general, but it requires at least as many basis functions and coefficients to represent the spline as the number of data points, which could be large, and the solution of the corresponding restricted quadratic programming problem of that size is numerically expensive [42].

A different approach to spline approximation, advocated by P. Dierckx [16], is to use the least squares splines. The approximation knots do not coincide with the data, and usually the number of spline segments is less than the number of data points. The coefficients of the spline are found as a solution to the linear least squares problem. When the knots of approximation coincide with the data, the least squares spline becomes the usual interpolating spline with the appropriate conditions imposed on the derivatives at the ends of the interpolation interval. One advantage of least squares splines is that they require substantially less coefficients and basis functions to represent the spline, and are easy to extend for multivariate case using tensor products. On the negative side is that fact that the quality of approximation critically depends on the position of the approximation knots [16,32]. The most simple uniformly distributed knots are often a bad choice. The methods of automatic optimisation of knots positions exist, but they involve minimisation of a nonlinear function with many local minima [10,16].

It is also possible to impose monotonicity or convexity conditions on the least squares splines. In the univariate case these conditions were discussed in [16,32]. Less attention has been given to shape preserving multivariate spline approximation.

The case of monotone multidimensional least squares spline approximation seems to be important for the following reasons. Firstly, in many problems the data is scattered rather than given on a rectangular grid. Powell-Sabin and thin plate splines can be used to solve this problem [42-44], but the construction is rather complicated. Tensor product least squares splines provide a much simpler solution, although not without its drawbacks [24,32]. Secondly, the scattered data usually contain observation errors, and the use of interpolation is not appropriate. Thirdly, the least squares splines significantly reduce the number of basis functions and coefficients required to approximate a given function, specifically in the case of well behaved functions and large number of data. However, even linear least squares splines may not preserve the monotonicity of the data (Fig. 1), let alone the cases of higher order splines and non-monotone data. The monotonicity or convexity properties of the underlying function may be semantically important (e.g. monotone transformation, dose-response, cost and growth curves [24,32,33], utility or membership functions, aggregation operators [8,9,28,46]), and enforcing these properties is critical.

Present paper addresses the issues of shape preserving univariate and multivariate least squares splines. Initially we consider the cases of univariate monotone and convex approximation. We will formulate the least squares problem using a basis different from traditional B-splines, but closely related to it. The key advantage of doing this is to express the restrictions on spline coefficients in the most simple way. Then, multivariate approximation will be considered. We will impose monotonicity conditions in respect to one variable only and in respect to all variables. Finally we will illustrate our method on a useful application of approximating aggregation operators in fuzzy sets theory.

Univariate monotone approximation

Suppose, there is a given set of data points $\{(x_i, y_i)\}_{i=1}^{I}$ on the interval $[a,b]$, and a prescribed set of approximation knots $\{t_j\}_{j=-k}^{N+k+1}$, such that $t_0 = a$, $t_{N+1} = b$ and $t_{-k} \leq \ldots \leq t_0 < t_1 < \ldots < t_{N+1} \leq \ldots \leq t_{N+k+1}$. The position of the knots outside $[a,b]$ is arbitrary [16]. Let $N_j^{k+1}(x)$ denote normalised B-spline of order k+1 (degree k) with knots $t_j, \ldots, t_{j+k+1}$. The recursive relation for B-splines is well known [10,16,36]

$$N_j^{l+1}(x) = \frac{x - t_j}{t_{j+1} - t_j} N_j^l(x) + \frac{t_{j+l+1} - x}{t_{j+l+1} - t_{j+1}} N_{j+1}^l(x),$$

$$N_j^1(x) = \begin{cases} 1, \text{ if } x \in [t_j, t_{j+1}), \\ 0 \text{ otherwise.} \end{cases}$$

The least squares spline $S(x)$ is a piecewise polynomial of order k+1

$$S(x) = \sum_{j=-k}^{N} a_j N_j^{k+1}(x) \qquad (1)$$

which minimises the least squares criterion

$$\sum_{i=1}^{I}\left(S(x_i)-y_i\right)^2 .$$

The linear space of polynomial splines of order k+1 defined on the knots $\{t_j\}_{j=-k}^{N+k+1}$ is denoted by $\mathcal{S}^{k+1}[t_{-k},...,t_{N+k+1}]$. Its dimension is N+k+1.

Functions $N_j^{k+1}(x)$ are well known in the literature, and they possess many useful properties, including local support, partition of unity, numerical stability, etc. When the knots $\{t_j\}$ are fixed, the problem of least square approximation is linear, and can be solved by standard methods, such as QR decomposition.

Even if the data $\{(x_i,y_i)\}_{i=1}^{I}$ is monotone, the r esulting least squares spline does not necessarily preserve this property. This is illustrated in Fig.1 for linear spline. To enforce monotonicity, additional restrictions should be imposed on the coefficients a_j. In the case of linear spline, it amounts to $a_j \le a_{j+1},\ j=-1,..,N-1$, that trivially follows from the partition of unity and local support properties. However in a more general case the restrictions on a_j are not simple.

In order to express the restrictions on spline coefficients in a more suitable form, we change the basis of $\mathcal{S}^{k+1}[t_{-k},...,t_{N+k+1}]$ to a new set of functions $T_j^{k+1}(x)$, defined by

$$T_j^{k+1}(x)=\sum_{m=j}^{N}N_m^{k+1}(x),\ j=-k,\dots,N . \qquad (2)$$

Clearly, the functions $T_j^{k+1}(x)$, which we call trapezoidal, or T-splines because of their form (Fig.2), are linearly independent, and therefore form a basis in $\mathcal{S}^{k+1}[t_{-k},...,t_{N+k+1}]$. Every spline $S(x)$ of order k+1 can be represented as

$$S(x)=\sum_{j=-k}^{N}b_jT_j^{k+1}(x) . \qquad (3)$$

The new splines possess a list of useful properties:

1. The support of $T_j^{k+1}(x)$ is $[t_j,t_{N+k+1}]$;
2. The derivative $(T_j^{k+1}(x))'=k\left(\dfrac{N_j^k(x)}{t_{j+k}-t_j}-\dfrac{N_{N+1}^k(x)}{t_{N+k+1}-t_{N+1}}\right)$, which means that $(T_j^{k+1}(x))'$ is proportional to $N_j^k(x)$ for $x\in[t_{-k},t_{N+1}]$;
3. $T_j^{k+1}(x)=1$ for $x\in[t_{j+k},t_{N+1}]$ and $j=-k,\dots,N-1$;
4. They are easily computed from B-splines.

T-splines are closely related to integrated I-splines from [32], which have been also proposed for monotone approximation. The difference merely lies in the way they were defined, as a linear combination of B-splines of the same order, or integrals of M-splines (differently normalised B-splines) of lower order. They coincide on [*a*,*b*] but differ on (b,∞).

T-splines basis is well conditioned for numerical calculations (although slightly less than B-splines). Indeed, the matrix $\mathbf{T}_{ij} = T_j(t_i)$ can be obtained from the (well conditioned [10,36]) basis $\mathbf{N}_{ij} = N_j(t_i)$ by multiplication $\mathbf{T} = \mathbf{NL}$, where **L** is a lower triangular matrix with unit entries below the diagonal. Its inverse is the two-diagonal matrix with 1 on the diagonal and -1 on the lower co-diagonal, and its condition number is 2*M*, *M* is the size of **L**.

Let us utilise the property 2 to obtain the necessary and sufficient conditions for spline monotonicity (we consider monotone non-decreasing splines, for monotone non-increasing splines the results are analogous). When *k*=1 (linear spline), its derivative is defined by a linear combination of piecewise constant functions with non-intersecting supports, and consequently by the sign of the coefficient of the only non-zero B-spline $N_j^1(x)$ for $x \in [t_j, t_{j+1})$. Therefore, the necessary and sufficient condition for monotonicity in this case is that all b_j be non-negative, except b_{-1}, which by itself determines the value of the spline at $a = t_0$.

The non-negativity of the coefficients is a sufficient condition of monotonicity of higher order splines as well [32] (indeed, a non-negative linear combination of non-negative functions, such as B-splines, is non-negative). Let us now establish the necessary condition for quadratic splines.

Proposition.
For quadratic spline

$$S(x) = \sum_{j=-2}^{N} b_j T_j^3(x)$$

the necessary and sufficient condition for monotonicity is
$b_j \geq 0$, *j*=-1,…,*N*.

Proof.
The derivative of *S*(*x*) on [*a*,*b*] is a piecewise linear spline whose positivity on [*a*,*b*] follows from its positivity at the knots $\{t_j\}_{j=0}^{N+1}$. The only spline which is not zero at t_j is $N_{j-1}^2(x)$, and therefore the sign of the derivative $S'(x)$ at t_j is determined by the sign of the coefficient b_{j-1}, *j*=0,…,*N*+1. ♦

Thus, in order to construct non-decreasing linear or quadratic spline, one has to solve the linear least squares problem with linear restrictions on the coefficients:

Minimise $\sum_{i=1}^{I} \left[\sum_{j=-k}^{N} b_j T_j^{k+1}(x_i) - y_i \right]^2$, subject to $b_j \geq 0$, j= -*k*+1,…,*N*.

This problem of non-negative least squares has been thoroughly studied, and efficient algorithms have been developed [14,22,23,29,30]. NNLS algorithm or its successor, BVLS, are available from NETLIB [17,22] or in printed form in [22,23]. In [13,14] an improvement to branch-and-bound technique [4], which further reduces the search space, is described. The algorithm LSEI [22,23] is more general and allows one to handle inequality and equality constraints simultaneously, which is useful to force the spline to pass through certain points.

If strict monotonicity is required, say $S'(x) \geq d > 0$ (different *d* may be required at different knots t_j), it can immediately be translated into $b_j \geq d$, j=-*k*+1,…,*N* (or $b_j \geq d_j$, if $S'(t_j) \geq d_j$). Moreover, if upper and lower bounds on the derivative are specified, $d_j \leq S'(t_j) \leq e_j$, then the restrictions will be $d_j \leq b_j \leq e_j$. The problem then can be solved using standard BVLS algorithm from [29].

Monotonicity of higher order spline can also be imposed by requiring non-negativity of the coefficients, however, because this is not a necessary condition, the resulting monotone spline is not guaranteed to be the best approximation in least squares sense.

Fig. 3 illustrates approximation of monotone and non-monotone data using T-splines. Calculation of the coefficients has been performed using LSEI algorithm. The additional restrictions on the function, $f(a) = 0$ and $f(b) = 1$, can be derived from noticing that the only basis function not zero at *a* is $T_1(x)$, whereas at *b* all basis functions are equal to 1. It implies that $b_{-k} = 0$ and $\sum_{j=-k}^{N} b_j = 1$.

Once the coefficients of the spline $\{b_j\}$ are found, the value of *S*(*x*) can be calculated using the Eq.(3). However, this may not be as efficient as using B-splines, because the sum in (1) involves at most *k*+1 terms (due to local support of $N_j(x)$). It could be better to return to the traditional B-spline representation using the following conversion formulas:

$$a_j = \sum_{m=-k}^{j} b_m, \quad j = -k, \ldots, N$$

These conversion formulas follow from

$$S(x) = \sum_{j=-k}^{N} b_j T_j(x) = \sum_{j=-k}^{N} b_j \sum_{m=j}^{N} N_m(x) = \sum_{j=-k}^{N} N_j(x) \sum_{m=1}^{j} b_m = \sum_{j=-k}^{N} a_j N_j(x).$$

Convex least squares splines

The convexity condition for least squares splines $S''(x) \geq 0$ is not readily translated into simple restrictions on the coefficients. Let us again change the basis for $\mathcal{S}^{k+1}[t_{-k}, \ldots, t_{N+k+1}]$. The new basis functions $\{\overline{T}_j^{k+1}\}_{j=-k}^{N}$ are defined as

$$\overline{T}_j^{k+1}(x) = \sum_{m=-k}^{N} T_m^{k+1}(x)(t_{i+k} - t_i)\,, \tag{4}$$

where $T_j^{k+1}(x)$ are given in (2). Fig. 4 shows the quadratic basis functions. It is clear that $\{\overline{T}_j^{k+1}\}_{j=-k}^{N}$ are linearly independent and form a basis in $S^{k+1}[t_{-k},...,t_{N+k+1}]$. The second derivatives of the new functions are calculated using the formula

$$(\overline{T}_j^{k+1}(x))'' = k(k-1)\left(\frac{N_j^{k-1}(x)}{t_{j+k-1} - t_j} - \frac{N_{N+1}^{k-1}(x)}{t_{N+k} - t_{N+1}}\right).$$

It is clear that on [*a*,*b*] only the part involving $N_j^{k-1}(x)$ is not zero. If we use quadratic spline (*k*=2), the second derivative is expressed as linear combination of constant splines (of order 1), and in the case of cubic splines, the second derivative is a piecewise linear function. Similarly to the monotone splines, the necessary and sufficient condition for convexity of the spline, represented as

$$S(x) = \sum_{j=-3}^{N} c_j \overline{T}_j^4(x)\,,$$

is non-negativity of the coefficients $c_j \geq 0$, $j = -1,\dots,N$.

Again, the problem becomes non-negative least squares problem, and it can be solved using existing techniques, such as NNLS, BVLS or LSEI algorithms [14,22,23,29]. Higher order splines can also be used, however, as with monotone splines, the non-negativity of the coefficients may be too strong a restriction, and the spline will not be the best approximation to the data in the least squares sense.

Interpolation and other conditions

Least squares splines can also interpolate the data, provided that at least the same number of basis functions as the number of data is selected (Whitney-Schoenberg conditions are presumed throughout this paper) . If monotonicity or convexity restrictions are imposed, they effectively reduce the number of degrees of freedom, and the resulting spline may fail to satisfy some of the interpolation conditions. Similar to interpolating or smoothing splines, where additional knots or additional discontinuities (cf. cubic C^1 splines [6,7,11], quartic C^2 splines [25]) are introduced, additional basis functions are required. If we select a bigger number of interpolation knots *N*, then we can construct an interpolating monotone (or convex) spline by solving the least squares problem with both equality and inequality constraints (provided that the data itself is monotone or convex). This is most easily done using LSEI algorithm [22,23].

The LSEI algorithm solves the following problem

$$\mathbf{Eb} = \mathbf{e}\,,\ \mathbf{Ab} \approx \mathbf{y}\,,\ \mathbf{Gb} \geq \mathbf{g}\,,$$

where the matrix **E** and vector **e** define the strict equality conditions, matrix **A** and vector **y** define equations to be approximately satisfied (least squares sense), **G** and **g** define inequality constraints (in our case **G=I** and **g=0**), and **b** is the vector of unknown coefficients.

The LSEI algorithm is very robust with respect to matrices **E,A** and **G** not being full column rank, and returns minimal length solution if $\mathbf{Eb} = \mathbf{e}$ is inconsistent [22,23]. Fig. 5 shows the results of monotone interpolation with the least squares splines whose knots coincide with the data (additional knots are in between data points).

On Fig. 6 convex cubic spline is presented. Interpolation can also be achieved using additional knots and basis functions. On Fig. 7 11 basis functions were used to interpolate 6 points with a convex spline, one extra knot per interval, like in McAllister and Roulier algorithm [26,31].

In certain problems the interpolation and approximation conditions are mixed. For instance, probability and possibility functions are monotone and have range [0,1]. Thus, to reconstruct such functions from data one needs to solve non-negative least squares problem with additional interpolation conditions $f(a)=0$ and $f(b)=1$. In multidimensional case, which we consider in the next section, similar but more complicated conditions are imposed. The solution to this problem is straightforward using LSEI algorithm. The matrices **E** and **A** contain the values of T-splines at corresponding data points, whereas the matrix **G** is the identity matrix.

Tensor product splines

In this section we extend the univariate monotone least squares splines to the bivariate and multivariate cases. We will use tensor product of linear and quadratic *T*-splines for this purpose [5,10,16,27].

The tensor product *T*-spline is the construction

$$T_{j_1 j_2 \cdots j_V}(x_1,\dots,x_V) = \prod_{v=1}^{V} T_{j_v}(x_v)\,.$$

The function of V arguments is approximated with

$$S(x_1,\dots,x_V) = \sum_{j_1=-k}^{N_1} \cdots \sum_{j_K=-k}^{N_V} b_{j_1 \cdots j_V} T_{j_1 \cdots j_V}(x_1,\dots,x_V)\,.$$

We assume that all the univariate splines have the same order, although this restriction can be dropped. In the two-dimensional case these formulas take the form

$$S(x_1,x_2) = \sum_{m=-k}^{M} \sum_{n=-k}^{N} b_{mn} T_{mn}(x_1,x_2)\,.$$

The monotonicity condition implies that all partial derivatives of the spline have to be non-negative at every point. Because functions $T_{mn}(x_1,x_2)$ are tensor products, their partial derivatives are multiples of the derivatives of the univariate *T*-splines, which are non-negative in the region of approximation:

$$\frac{\partial S(x_1,x_2)}{\partial x_1} = \sum_{n=-k}^{N} \frac{kb_{mn}}{t_{m+k}-t_m} N_m^k(x_1)T_n(x_2)\,,\ x_1 = t_{m+1}\,.$$

In follows that the coefficients $\{b_{mn}\}$ have to satisfy the following inequalities

$$\sum_{l=-k}^{n} b_{ml} \geq 0\,,\ n=-k,...,N\,,\ m=-k+1,...,M\ . \quad (5a)$$

Differentiation with respect to the other argument results in

$$\sum_{l=-k}^{m} b_{ln} \geq 0\,,\ m=-k,...,M\,,\ n=-k+1,...,N\,. \quad (5b)$$

These inequalities form part of the constrained quadratic programming problem:

$$\text{minimise}\ \sum_{i=1}^{I}\left[\sum_{m=-k}^{M}\sum_{n=-k}^{N} b_{mn}T_{mn}(x_{1i},x_{2i}) - y_i\right]^2$$

subject to (5a) and (5b). x_{ji} denotes *j*-th coordinate of *i*-th observation.

This problem can be solved by traditional methods [29,30], in particular using BVLS [14,29] or LSEI [22,23] algorithms, both available via NETLIB [17].

For multidimensional case the linear inequality restrictions are similar:

$$\sum_{j_1=-k}^{J_1}\cdots\sum_{j_V=-k}^{J_V} b_{j_1\ldots j_v\ldots j_V} \geq 0\,,\ j_v = -k+1,...N_v\,,\ J_l = -k,...,N_l\,,\ l \neq v\,;\ v = 1,2,...,V\,.$$

The sums are taken over all combinations of upper index limits N_l, $l \neq v$. Altogether there are at most $V \times N_1 \times ... \times N_V$ inequalities (some are redundant). They can be easily represented in matrix form, and the matrix consists of 0s and 1s arranged in a fashion consistent with the indexing system.

Let now suppose that monotonicity is required with respect to one argument only, say x_1. Consider bi-linear case. In this case the problem can be reduced to non-negative least squares problem by the following artifice. The tensor product spline will be represented as

$$S(x_1,x_2) = \sum_{m=-1}^{M}\sum_{n=-1}^{N} a_{mn}T_m(x_1)N_n(x_2)\,.$$

We mix two bases here, T-splines are used to express monotonicity condition, whereas B-splines are used because of their localised support. Differentiating with respect to the first argument results in

$$\frac{\partial S(x_1, x_2)}{\partial x_1} = \frac{ka_{mn}}{t_{m+1} - t_m} N_m^1(x_1) N_n^2(x_2)\,, \quad x_1 \in [t_m, t_{m+1}]\,,\ x_2 = \bar{t}_{n+1}\,.$$

Therefore, monotonicity with respect to x_1 implies $a_{mn} \geq 0$.

Thus we again arrived to the non-negative least squares problem which can be solved using a somewhat faster NNLS algorithm from [29]. The extension of this method for V dimensions is straightforward.

Additional constraints $f(\mathbf{0}) = 0$ and $f(\mathbf{1}) = 1$ can also be imposed. The first condition implies $a_{-k\ldots-k} = 0$ and the second implies $\sum_{j_1=-k}^{N_1} \ldots \sum_{j_V=-k}^{N_V} b_{j_1 \cdots j_V} = 1$. Another restriction important for triangular conorms [8,9,28,46] is $f(0,0,\ldots,x_i,\ldots 0) = x_i$. It can be achieved by forcing the spline to interpolate the values $f(0,0,\ldots,t_j,\ldots 0) = t_j$, where t_j are the knots of approximation in respect to coordinate x_i. The restriction $f(1,1,\ldots,x_i,\ldots 1) = 1$ is imposed by setting the appropriate interpolation conditions $f(1,1,\ldots,t_j,\ldots 1) = 1$, easily specified as the input of LSEI algorithm [22,23].

Algorithm and numerical results

Given that calculation of spline coefficients is reduced to standard quadratic programming problem with linear inequality constraints, the implementation of the algorithm is straightforward. In matrix form it can be written as

Solve $\mathbf{Tb} \approx \mathbf{y}$, given that $\mathbf{Gb} \geq \mathbf{0}$,

with $\approx$ standing for “approximately equal”. **T** is $I \times N$ matrix with the entries given by the values of basis functions $T_j(x_i)$ at data points x_i, and **G** is the identity matrix **I**. Both matrices serve as the input of the algorithm LSEI (or NNLS, which requires only **T**).

Computation of the entries of **T** can be performed in a very effective way using B-splines basis $N_j(x)$. Let the entries of matrix **N** be $N_{ij} = N_j(x_i)$. Then

$$\mathbf{T} = \mathbf{NL}\,,$$

whete **L** is the lower triangular matrix of size $M \times M$ consisting of 0s and 1s, and $M = N + k + 1$ is the number of basis functions. Once the coefficients **b** have been found, they can be transformed to the vector of coefficients **a** of B-spline representation (1) by

$\mathbf{a} = \mathbf{Lb}$.

For convex splines the calculations are similar. The matrix $\overline{\mathbf{T}}$ consisting of the values of $\overline{T}_j(x_i)$ at data points can be calculated using

$$\overline{\mathbf{T}} = \mathbf{T\Delta} = \mathbf{NL\Delta},$$

where $\mathbf{\Delta}$ is the lower triangular matrix with the elements at each row given by $\Delta_{ij} = t_{i+k} - t_i$.

The vector **a** of B-spline representation is calculated using

$$\mathbf{a} = \mathbf{L\Delta b}.$$

It should be noted that the numbering starts at -*k*, so that the upper left element of the matrices is (-*k*,-*k*).

In multivariate case the matrices have more complicated structures. **T** consists of the values of products

$$T_{j_1 j_2 \dots j_V}(x_1, \dots, x_V) = \prod_{v=1}^{V} T_{j_v}(x_v)$$

at data points, whereas **G** consists of 0s and 1s, and its structure depends on the way of indexing of the basis functions. Both matrices serve as input to LSEI.

Figures 3 - 9 illustrate the results of univariate and bivariate monotone spline approximation.

Conclusion

Least squares splines provide a rather simple method of data approximation and interpolation, easily expandable for multivariate case using tensor products. The quality of approximation demands very much on the positioning of the knots, and even in the simplest cases such splines may fail to preserve some of the properties of the data or underlying function. Optimising knot positions is one way of improving the quality of approximation, but it often runs into the problem of many local maxima.

Another way to achieve this is to enforce known properties of the function, such as monotonicity or convexity. These properties translate into certain restrictions on spline coefficients, and the problem becomes linear restricted least squares problem. The form of restrictions depends on the choice of the basis in the space of polynomial splines. The T-spline basis presented in this paper expresses monotonicity and convexity conditions in the most simple way, as non-negativity of coefficients, and the problem becomes the non-negative least squares problem, for which the methods of solution as well as program code are readily available. Since T-splines are simple linear combinations of B-splines, effective methods of computing matrix entries and B-spline coefficients can be implemented by using matrix multiplication. On the other hand, generic restricted least squares algorithms allow one to specify additional interpolation conditions, useful when the function is supposed to pass through certain points.

Tensor product T-splines allow one to enforce shape restrictions in $\mathbf{R}^n$, by using the same technique of non-negative least squares. This technique is very effective for scattered data approximation, where other techniques (Powell-Sabin splines, thin-plate splines) are rather complicated and require at least the same number of coefficients as data. Besides, various interpolation conditions can be imposed simultaneously as well. This is useful for reconstruction of functions with certain properties, such as triangular norms and co-norms. The extensive research in restricted least squares problem and robustness of corresponding algorithms make spline computation straightforward.

References

[1] L.-E. Andersson and P.-A. Ivert, Constrained interpolants with minimal $W^{k,p}$ norm, *J. Approx. Theory* **49** (1987), 283-288.

[2] L.-E. Andersson and T. Elfving, An algorithm for constrained interpolation, *SIAM J.Scient. Stat. Comput.* **8** (1987), 1012-1025.

[3] L.-E. Andersson and T. Elfving, Interpolation and approximation by monotone cubic splines, *J. Approx. Theory* **66** (1991), 302-333.

[4] R.D. Armstrong and E.L. Frome, A branch-and-bound solution of a restricted least squares problem, *Technometrics* **18** (1976), 447-450.

[5] G.A. Ateshian, A B-spline least-squares surface-fitting method for articular surfaces of diarthrodial joints, *J. Biomed. Eng.* **115** (1993) 366-373

[6] R.K. Beatson and H.Wolkowicz, Post-processing piecewise cubics for monotonicity, *SIAM J.Numer.Anal.* **26** (1989), 480-502.

[7] R.K. Beatson and Z. Ziegler, Monotonicity preserving surface interpolation, *SIAM J. Numer. Anal.* **22** (1985), 401-411.

[8] G. Beliakov, Definition of general aggregation operators through similarity relations, *Fuzzy Sets and Systems* (2000), in press.

[9] G. Beliakov, Aggregation Operators as Similarity Relations, in: *Information, Uncertainty, Fusion*, eds. R. Yager, B. Bouchon-Meunier and L. Zadeh, Kluwer, Boston 1999.

[10] C. de Boor, *A practical guide to splines*, Springer, Berlin-NewYork, 1978.

[11] R.E. Carlson and F.N. Fritsch, Monotone piecewise bicubic interpolation, *SIAM J. Numer. Anal.* **22** (1985), 386-400.

[12] J.M. Carnicer and W. Dahmen, Convexity preserving interpolation and Powell-Sabin elements, *Comp. Aided Geom. Design* **9** (1992), 279-289.

[13] D. Clark, An algorithm for solving the restricted least squares problem, *J. Austral. Math. Soc.* **21** (1980) (Series B), 345-356.

[14] D. Clark and M. Osborne, On linear and interval least-squares problems, *IMA J. Numer. Anal.* **8** (1988), 23-36.

[15] P. Constantini and F. Fontanella, Shape-preserving bivariate interpolation, *SIAM J. Numer. Anal.* **22** (1985), 488-506.

[16] P. Dierckx, *Curve and surface fitting with splines*, Clarendon press, Oxford, 1995.

[17] J.J. Dongarra and E. Grosse, Distribution of mathematical software via electronic mail, *Comm. ACM* **30** (1987), 403-440.

[18] T. Elfving and L.-E. Andersson, An algorithm for computing constrained smoothing spline functions, *Numer.Math.* **52** (1989), 583-595.

[19] R.L. Eubank, *Spline smoothing and nonparametric regression,* Marcel Dekker Inc, New York, 1988.

[20] S. Fredenhagen, H.J. Oberle and G. Opfer, On the construction of optimal monotone cubic spline interpolations, *J. Approx. Theory* **96** (1999), 182-201.

[21] H. Greiner, A survey on univariate data interpolation and approximation by splines of given shape, *Math. Comp. Modelling* **15** (1991), 97-106.

[22] R.J. Hanson and K.H. Haskell, Algorithm 587. Two algorithms for the linearly constrained least squares problem, *ACM Trans. Math. Software* **8** (1982) 323-333.

[23] K.H. Haskell and R.J. Hanson, An algorithm for linear least squares problems with equality and nonnegativity constraints, *Math. Progr.* **21** (1981), 98-118.

[24] X. He and P. Shi, Monotone B-spline smoothing, *J. Am. Stat. Ass.* **93** (1998), 643-650.

[25] W. Hess and J.W. Schmidt, Positive quartic, monotone quintic C^2-spline interpolation in one and two dimensions, *J.Comp.Appl.Math.* **55** (1994), 51-67.

[26] R. Iqbal, A one-pass algorithm for shape-preserving quadratic spline interpolation, *J. Scientific. Comp*. **7** (1992), 359-376.
[27] B. Juttler, Surface fitting using convex tensor-product splines, *J. Comp. Appl. Math.* **84** (1997), 23-44.
[28] G. Klir and T. Folger, *Fuzzy sets, uncertainty, and information*, Prentice Hall, Singapore, 1992.
[29] C. Lawson and R.Hanson, *Solving least squares problems*, SIAM, Philadelphia, 1995.
[30] D.G. Luenberger, *Introduction to linear and nonlinear programming*, Addison-Wesley, Reading, Mass, 1973.
[31] D.F. McAllister and J.A. Roulier, An algorithm for computing a shape-preserving oscillatory quadratic spline, *ACM Trans. Math. Software* **7** (1981), 331-347.
[32] J.O. Ramsay, Monotone regression splines in action (with comments), *Stat. Science* **3** (1988) 425-461.
[33] P.A. Ruud, Restricted least squares subject to monotonicity and concavity constraints, *Advances in Economics and Econometrics: Theory and Applications, Vol.* 3, Eds. D. M. Kreps and K. F. Wallis, Cambridge, 1997, 166-187.
[34] N.S. Sapidis and P.D.Kaklis, An algorithm for constructing convexity and monotonicity preserving splines in tension, *Comp.Aided Geom.Des.* **5** (1988), 127-137.
[35] J.W. Schmidt and W. Hess, An always successful method in univariate convex C^2 interpolation, *Numer. Math*. **71** (1995) 237-252.
[36] L.L. Schumaker, *Spline functions: Basic theory,* Wiley, New York, 1981.
[37] L.L. Schumaker, On shape-preserving quadratic spline interpolation, *SIAM J.Numer.Anal.* **20** (1983), 854-864.
[38] D.G. Schweikert, An interpolation curve using a spline in tension, *J.Math. Phys.* **45** (1966), 312-317.
[39] H. Spath, Exponential spline interpolation, *Computing* **4** (1969), 225-233.
[40] B.A. Turlach, Constrained smoothing splines revisited, Statistics research report SRR 008-97, Australian National University, 1997.
[41] F.I. Utreras, Smoothing noisy data under monotonicity constraints. Existence, characterization and convergence rates, *Numer.Math.* **47** (1985), 611-626.
[42] F.I. Utreras and M.L. Varas, Monotone interpolation of scattered data in R^s, *Constr. Appr.* **7** (1991) 49-68.
[43] M. Villalobos and G. Wahba, Inequality-constrained multivariate smoothing splines with application to the estimation of posterior probabilities, *J. Am. Stat. Assoc.* **82** (1987) 239-248.
[44] K. Willemans and P. Dierckx, Smoothing scattered data with a monotone Powell-Sabin spline surface, *Numer. Algorithms* **12** (1996), 215-232.
[45] I.W. Wright and E.J. Wegman, Isotonic, convex and related splines, *Ann. Stat.* **8** (1980) 1023-1035.
[46] H.-J. Zimmermann, *Fuzzy set theory - and its applications*, Kluwer, Boston. 1996.

Figure captions

Figure 1. Linear least squares spline fails to preserve monotonicity of the data.

Figure 2. Trapezoidal basis functions.

Figure 3. Approximation of monotone (a) and non-monotone (b) data using linear and quadratic monotone splines. The underlying function is $f(x) = \min[1, (1.5x)^4]$. In case (b) the uniformly distributed in [-0.1,0.1] noise is added.

Figure 4. Quadratic basis functions for convex approximation.

Figure 5. Monotone interpolation of monotone data from [26,37].

Figure 6. Convex (a) and strictly convex ($f''(x) \geq 0.00008$) (b) cubic approximation of the first 31 points of titanium heat data [10,18].

Figure 7. Convex cubic interpolation using 5 additional knots (data from [26]).

Figure 8. Monotone bi-linear interpolation of scattered data. Six basis function for each variable and 50 noisy data points were used. The underlying function is

$$f(x,y) = \begin{cases} |x|y, \text{ if } xy > 0, \\ 0 \text{ elsewhere.} \end{cases}$$

Figure 9. Monotone bi-linear approximation of the triangular co-norm $f(x,y) = \min(1, \left(x^2 + y^2\right)^{1/2})$. The data is scattered and contains small random noise uniformly distributed in [-0.1,0.1].