Least squares data fitting using shape preserving piecewise approximations

G.H. Elliott

School of Mathematical Studies, University of Portsmouth, Hampshire Terrace, Portsmouth, PO1 2EG, UK

In recent years there has been a great deal of interest in the preservation of data properties in an interpolating function, and many good algorithms are available for this problem.

In this paper a basis is constructed for a tensioned spline that gives a numerically stable algorithm for the L_2 fitting of data that can preserve monotonicity and/or convexity. The motivation for this work is the fitting of data from a sewerage farm.

1. Introduction

Polynomials are widely used in data approximation, but among their few undesirable properties is that even in the simple case of quadratic interpolation to three given function values $\{x_k, y_k\}$ $(k = 0, 1, 2)$, with $x_0 < x_1 < x_2$, the quadratic may fail to be monotone on $[x_0, x_2]$ even if the y values are monotone. Similarly the cubic which interpolates a convex function may not itself be convex.

Unfortunately the same situation occurs in cubic spline interpolation. This has been recognised for many years, and several attempts have been made to modify the spline in order to preserve data properties. Among the best known are:

- using splines in which the cubic segments have their gradients restricted so as to preserve monotonicity; see for example [5];
- tensioning the spline using exponential functions as in [7] or [8];
- tensioning the spline using a denominator in each segment which contains a single parameter that can be increased until the resulting approximation has the required property; see for example [3,4,6].

All of these methods can deal correctly with some classic "difficult" data sets such as the Akima data [1]. They are difficult in the sense that there is a sudden change from a small to a large gradient over a short interval.

The work described in this paper was motivated by the need to fit a curve to some data collected from a new type of sewerage bed (Gravel Bed Hydroponics). A sample set of data collected on one day is given in fig. 1.

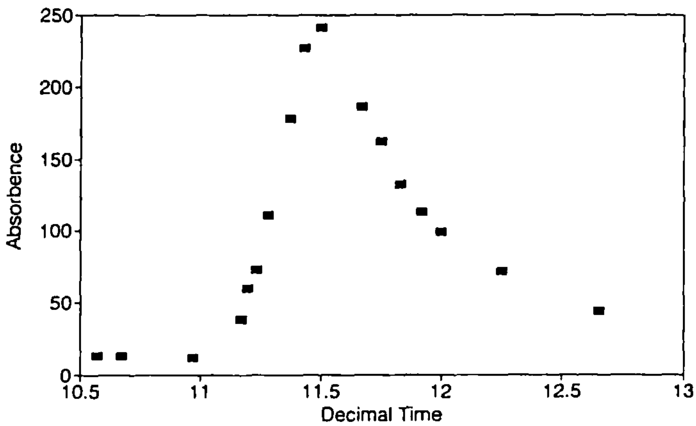

Fig. 1. Bed BD2, original data.

Although the measurements are made as accurately as possible, the data is not sufficiently accurate to warrant interpolation and a least squares fit is appropriate. A simple least squares cubic spline approximation suffers the same fate as would an interpolant, in that it fails to be monotone where the data is monotone (fig. 2). The knots are denoted by the symbol $\times$.

The loss of monotonicity in the spline approximation is remedied using a method based on the tensioning idea of Delbourgo and Gregory [4] which is outlined in section 2.

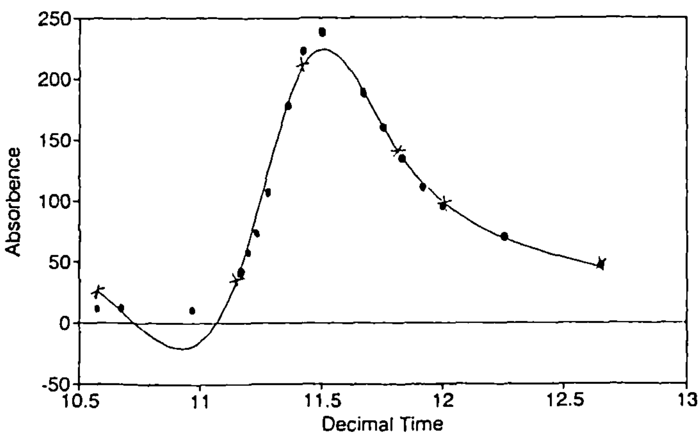

Fig. 2. Bed BD2, cubic spline. SS=2181.

2. Shape preserving piecewise rational interpolation

Let (x_k, y_k) $(k = 0, \ldots, n)$ be a given set of data points with $x_0 < x_1 < \ldots < x_n$. Define $h_k = x_{k+1} - x_k$ and $\Delta_k = (y_{k+1} - y_k)/h_k$. A piecewise rational cubic function is then defined on $[x_k, x_{k+1}]$ as $s(x) = P_k(t)/Q_k(t)$, $t = (x - x_k)/h_k$, where

$$P_k(t) = y_{k+1}t^3 + (r_k y_{k+1} - h_k d_{k+1})t^2(1 - t) + (r_k y_k + h_k d_k)t(1 - t)^2 + y_k(1 - t)^3$$

and

$$Q_k(t) = 1 + (r_k - 3)t(1 - t)\,. \tag{2.1}$$

It has the interpolation properties $s(x_k) = y_k$, $s(x_{k+1}) = y_{k+1}$, $s'(x_k) = d_k$, $s'(x_{k+1}) = d_{k+1}$.

If the d_k are given, we have immediately $s \in C^1[x_0, x_n]$. Alternatively, the d_k can be chosen to make $s \in C^2[x_0, x_n]$ by solving the tridiagonal diagonally dominant system

$$h_k d_{k-1} + d_k\{h_k(r_{k-1} - 1) + h_{k-1}(r_k - 1)\} + h_{k-1}d_{k+1} = h_{k-1}r_k\Delta_k + h_k r_{k-1}\Delta_{k-1},$$
$$k = 1, \ldots, n - 1\,.$$

If $r_k = 3$, the function s reduces to a cubic spline. If

$$r_k > (d_k + d_{k+1})/\Delta_k\,, \tag{2.2}$$

then s is monotone on $[x_k, x_{k+1}]$ provided Δ_k, d_k and d_{k+1} are positive. Another inequality on r_k can be given to ensure that the data is convex if the data is consistent with convexity. The function s can therefore in general be regarded as a tensioned spline. Derivations and further details can be found in the papers of Delbourgo and Gregory.

3. A tensioned B-spline

A least squares approximation requires a set of basis functions, ideally with compact support, on the knot set $\{x_0, \ldots, x_n\}$. A typical choice is the set $\{B_k(x)\}$, where $B_k(x)$ is a natural cubic spline which is identically zero for $x \leqslant x_{k-2}$ and $x \geqslant x_{k+2}$ (the knot set needs to be extended slightly at the ends), and normalised so that $B_k(x_k) = 1$.

We propose a piecewise approximation using a set of tensioned splines $\{\bar{B}_k(x)\}$ as basis. The function $\bar{B}_k(x)$ is similar to $B_k(x)$ but is divided in each segment $[x_k, x_{k+1}]$ by a denominator term

$$1 + (r_k - 3)t(1 - t)$$

in which the parameter r_k depends only on the *segment* and not on the particular basis function under consideration. For a given set of tension parameters, the resulting functions $\bar{B}_k(x)$ therefore still span a linear space and can be used as a basis for a least squares approximation in which tensioning can be applied. Figure 3 shows a typical tensioned basis spline on an equally spaced knot set in which the tension parameters for the non-zero segments are $r = 3, 10, 3$ and 6.

There seem to be no simple recurrences relating tensioned splines $\bar{B}_k(x)$ of different orders corresponding to the very efficient relations for polynomial splines given by Cox [2]. Nevertheless, we shall see that the $\bar{B}_k(x)$ can still be computed by a quick and stable algorithm. Adopting the normalisation $\bar{B}_k(x_k) = 1$, the tensioned B-spline is completely described if we can calculate its three gradients $d1_k, d2_k, d3_k$, at x_{k-1}, x_k and x_{k+1} respectively. Defining $a_k = \bar{B}_k(x_{k-1}), c_k = \bar{B}_k(x_{k+1})$ and $R_k = r_k/h_k$, the conditions $\bar{B}_k''(x_{k-2}) = \bar{B}_k''(x_{k+2}) = 0$ imply

$$a_k = d1_k/R_{k-2} \text{ and } c_k = -d3_k/R_{k+1}\,,$$

so that the second derivative continuity equations from section 2 simplify to

$$\{h_{k-1}(r_{k-2} - 2) + h_{k-2}(r_{k-1} - 1) + h_{k-2}R_{k-1}/R_{k-2}\}d1_k + h_{k-2}d2_k = h_{k-2}R_{k-1}\,,$$

$$h_k(1 + R_{k-1}/R_{k-2})d1_k + \{h_k(r_{k-1} - 1) + h_{k-1}(r_k - 1)\}d2_k$$
$$+ h_{k-1}(1 + R_k/R_{k+1})d3_k = h_kR_{k-1} + h_{k-1}R_k\,,$$

$$h_{k+1}d2_k + \{h_{k+1}(r_k - 1) + h_k(r_{k+1} - 2) + h_{k+1}R_k/R_{k+1}\}d3_k = -h_{k+1}R_k\,.$$

For a given knot set and estimates for suitable tensions, these equations are solved for $k = 0, \ldots, n$. Each tensioned B-spline $\bar{B}_k(x)$ is thus computed from three tridiagonal equations, the solution of which gives its gradients $d1_k, d2_k$ and $d3_k$ at x_{k-1}, x_k and x_{k+1}, and stored in terms of the five quantities $d1_k, d2_k, d3_k, a_k$ and c_k. The basis function $\bar{B}_k(x)$ is therefore by construction zero for $x \geqslant x_{k+2}$ and $x \leqslant x_{k-2}$, and has a continuous second derivative.

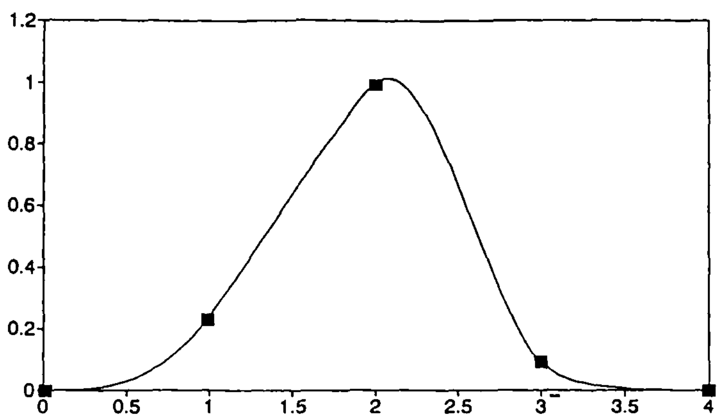

Fig. 3. Tensioned B-spline. Tensions 3, 10, 3, 6.

4. The computation of the piecewise approximation

A least squares approximation to the data set $\{(u_i,f_i), i = 1,\ldots,m\}$ is now sought in the form

$$s(x) = \sum_{k=0}^{n} \alpha_k \bar{B}_k(x)$$

by solving the overdetermined system

$$s(u_i) = f_i, \quad i = 1,\ldots,m. \tag{4.1}$$

Since there will typically be several data points in each segment, we need the formulae for those $\bar{B}_k(x)$ which are not zero in $[x_k, x_{k+1}]$, viz.

$$D_k\bar{B}_{k-1}(x) = c_{k-1}(1-t)^3,$$

$$D_k\bar{B}_k(x) = t^2(\{tc_k + (r_kc_k - h_kd3_k)(1-t)\} + (1-t)^2\{(r_k + h_kd2_k)t + (1-t)\},$$

$$D_k\bar{B}_{k+1}(x) = t^2\{t + (r_k - h_kd2_{k+1})(1-t)\} + (1-t)^2\{(r_ka_{k+1} + h_kd1_{k+1})t + a_{k+1}(1-t)\},$$

$$D_k\bar{B}_{k+2}(x) = a_{k+2}t^3,$$

where as before

$$t = (x - x_k)/h_k \text{ and } D_k = 1 + (r_k - 3)t(1-t).$$

For a given segment there are just an initial 6 multiplications needed, plus 18 multiplications and 4 divisions to compute all four non-zero basis functions for each data point u_i which compares favourably with the computation of the corresponding cubic B-spline basis by recurrence.

Equations (4.1) are then solved by Householder reduction to obtain the coefficients. Again lacking recurrence for the $\bar{B}_k(x)$, it is easier to compute the values of the resulting approximation $s(x)$ by first evaluating $s(x)$ and its derivatives at the knots, where only three basis functions are non-zero. We have

$$s(x_k) = \alpha_{k-1}c_{k-1} + \alpha_k + \alpha_{k+1}a_{k+1}$$

and

$$s'(x_k) = \alpha_{k-1}d3_{k-1} + \alpha_kd2_k + \alpha_{k+1}d1_{k+1}.$$

The values of $s(x)$ can then be found directly from (2.1).

It is easy to verify directly from the defining equations that if $r_k \geqslant 3$, then $d1_k > 0$ and $d3_k > 0$, and further that $\bar{B}_k(x) > 0$ for $x_{k-2} < x < x_{k+2}$. This suggests that the tensioned basis functions share the same numerical stability as the cubic B-spline basis.

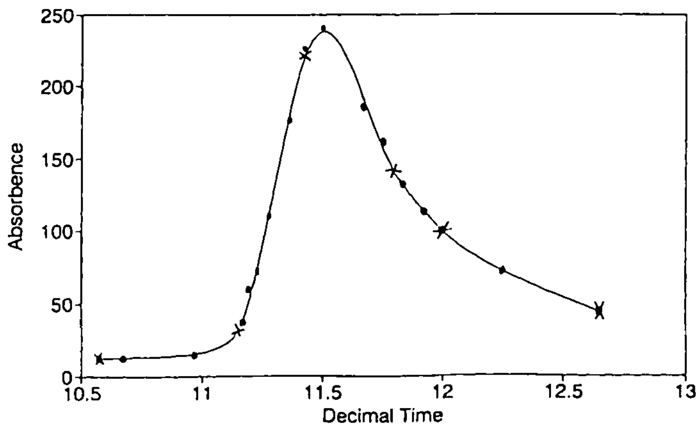

Fig. 4. Bed BD2, tensioned spline. SS=198.

5. Practical considerations and examples.

To obtain a least squares fit to a given data set using the tensioned basis there remain two choices, namely the choice of knots and tensions. The choice of knots has been considered by many authors. Numerical experience shows that the goodness of fit can vary dramatically with the choice of knots. All that has been done in these examples is a small movement of each knot about an initial trial set, and the insertion of an extra knot in any segment where the current approximation is unsatisfactory.

The choice of tensions is made partly interactively, with the proviso that condition (2.2) guarantees a monotone approximation where needed. The tensioning can also simply be used to improve the fit by monitoring the sum of squared errors. Figure 4 illustrates how the tensioning can correct the lack of monotonicity of the cubic spline approximation in fig. 2, the tensioned approximation values being given in table 1.

Table 1
Tensioned approximation values of fig. 4.

x_k	$s(x_k)$	$s'(x_k)$	r_k
10.57	12.22	10.73	10.0
11.16	33.56	353.90	4.0
11.42	217.25	515.36	3.0
11.80	139.91	−287.88	5.0
12.00	98.97	−161.63	5.0
12.65	43.95	−66.47	

6. Concluding remarks

The algorithm has been shown to have shape preserving possibilities but there remain two areas of weakness. The choice of tension parameters r_k could be made automatic but they depend on the gradients $s(x_k)$ which in turn depend on the parameters r_k. Secondly, experience in allowing knots to move shows that in some cases knots appear to want to coincide. This suggests that we should allow for a reduction in smoothness of $s(x)$ by allowing for coincident knots as is normal in polynomial spline approximation. It may be possible to modify the computation of the basis functions to allow for coincident knots.

References

[1] H. Akima, A new method of interpolation and smooth curve fitting based on local procedures, J. ACM 17 (1970) 589–602.

[2] M.G. Cox, Practical spline approximation, *Lecture Notes in Mathematics 965: Topics in Numerical Analysis*, ed. P.R. Turner (Springer, Berlin, 1982) pp.79–112.

[3] R. Delbourgo and J.A. Gregory, C^2 rational quadratic spline interpolation to monotonic data, IMA J. Numer. Anal. 3 (1983) 141–152.

[4] R. Delbourgo and J.A. Gregory, Shape preserving piecewise rational interpolation, University of Brunel report TR/10/83 (1983).

[5] F.N. Fritsch and R.E. Carlson, Monotone piecewise cubic interpolation, SIAM J. Numer. Anal. 17 (1980) 238–246.

[6] J.A. Gregory and R. Delbourgo, Piecewise rational quadratic interpolation, IMA J. Numer. Anal. 2 (1982) 123–130.

[7] S. Pruess, Properties of splines in tension, J. Approx. Theory 17 (1976) 86–96.

[8] P. Rentrop, An algorithm for the computation of the exponential spline, Numer. Math. 35 (1980) 81–93.