SIAM J. NUMER ANAL.
Vol. 26, No. 2, pp. 480-502, April 1989

POST-PROCESSING PIECEWISE CUBICS FOR MONOTONICITY*

R. K. BEATSON† AND H. WOLKOWICZ‡

Abstract. Monotone interpolation schemes of the *fit and modify* type are considered. Algorithms with optimal order error properties are developed. Numerical examples are given comparing these methods with others. The theoretical underpinnings of the algorithms include sufficient conditions for the existence of monotone splines with specified error and smoothness properties, and a result concerning approximate projections onto convex sets.

Key words. monotonicity constraint, cubic spline, post-processing

AMS(MOS) subject classifications. 65D07, 65D10, 41A25

1. Introduction. Often in interpolating data, $\{(t_i, y_i)\}_{i=0}^n$, it is important that the interpolant, s, inherit some geometric property of the data. This is a requirement when violation of the geometric property implies that the situation represented by the interpolated curve is physically impossible. It is also desirable when imposition of the constraint will lead to stability of some numerical algorithm, or mechanical/electrical device, built upon the interpolant. On the other hand, the point of interpolating data is surely to approximate behavior at points x where we have no data. Thus the error in approximation $\|f-s\|$ should be small. In this paper we present three new algorithms for monotone interpolation by piecewise cubics. These impose the geometric constraint without sacrificing the optimal $\mathcal{O}(\delta^j\omega(f^{(j)}, \delta))$ order of approximation to C^j functions, $1 \leqq j \leqq 3$. In particular this implies $\mathcal{O}(\delta^4)$ convergence to monotone C^4 functions.

The earliest work on such problems appears to be that of Chebyshev [6], [7]. This surprisingly little-known work was motivated by an application to the design of a governor for a steam engine. Monotonicity of the fitted curve was required in order that the governor be stable. More recently algorithm developers have been motivated by diverse applications including various chemical problems, VLSI, and CAD/CAM.

The paper is organised as follows. In § 2 we discuss various criteria for choosing among interpolation algorithms, and in particular the relevance of error estimates. In § 3 sufficient conditions for the existence of monotone spline interpolants with specified smoothness and error properties are developed. One application is an existence result for C^2 monotone cubic spline interpolants. In § 4 we present a general structure, such that all algorithms with this structure will have optimal order error properties. Section 5 concerns approximate projections onto convex sets. In § 6 we review previous work on monotone piecewise cubic interpolation algorithms. In § 7 we develop three new algorithms of this type, all with optimal order error properties. Some numerical comparisons of these algorithms with others are presented in § 8.

Throughout this paper $\|\cdot\|$ without any subscript denotes the uniform norm on the interval $[t_0, t_n]$. $C_1, C_2, C_3, \cdots$ denote absolute constants not depending on the function f being approximated, the nodes of interpolation, or the knots of the spline. By $S_{t,k}$ we mean the splines of order k with knots t, where these knots are supplemented by sufficient extra knots, to the left of t_0 and the right of t_n, so that there exist B-splines with support in $(-\infty, t_1]$ and $[t_{n-1}, \infty)$.

* Received by the editors February 9, 1987; accepted for publication (in revised form) February 19, 1988.

† Mathematics Department, University of Canterbury, Private Bag, Christchurch 1, New Zealand. This research was supported by National Science Foundation grant MCS-8305023.

‡ Mathematics Department, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. This author's research was supported by the National Science and Engineering Research Council of Canada grant A3388.

2. Choosing a curve-fitting algorithm. Consider the problem of choosing among several algorithms which given data $\{(t_i, y_i)\}_{i=0}^n$ fit a curve through these points satisfying certain specified constraints. The criterion of "visual pleasure," which is sometimes used, is inherently subjective and based on a few example data sets. Thus, while the action of an algorithm on example data sets is certainly relevant, other factors should also be considered. A possible solution is a "shopping list" of desirable properties over and above satisfaction of the constraints. For example, numerical stability, acceptable computational cost, smoothness (preferably measured quantitatively), and moderately sized operator norms (again preferably measured quantitatively) are all desirable. Choosing an algorithm can then be thought of as a mixed objective nonlinear programming problem. At least for general purpose algorithms, intended for use in a "black-box" manner by nonspecialists, the mixed nature of the objective is probably essential. For example, the natural cubic spline minimizes $\int (s'')^2$ over all unconstrained interpolants, and thus is optimal with respect to this "pure" smoothness criteria, but has unsatisfactory error properties. Indeed for such general purpose use, the algorithm with the longer/better "shopping list" is probably to be preferred even if it is a little more expensive computationally.

In our opinion *optimal order error estimates* deserve a place on the list of desirable properties. We will call an approximation method, defined for all data sets $\{(t_i, y_i)\}_{i=0}^n$ with the t_i's distinct, *j-optimal* if the error in approximating f, and f' satisfies

$$\|f^{(l)}-s^{(l)}\|_{[t_0,t_n]}\leqq C_1\delta^{j-l}\omega(f^{(j)},\delta), \qquad l=0,1$$

whenever the data values y_i come from a C^j function f, where δ is the meshsize of $\mathbf{t}$. Then for some set of j's, and the larger the set the better, we desire that curve-fitting algorithms be j-optimal. This is an obvious requirement when the data is known to come from a smooth function. Even when the data does not come from a smooth function such estimates are important as they imply that the size of the derivatives of the interpolant reflects the size of the differences of the data. More precisely,

(i) s depends on $\mathbf{t}$ and $[f(t_0),\cdots,f(t_n)]^T$ only, and

(ii) $\|f^{(j)}-(s(f))^{(j)}\|_{[t_0,t_n]}\leqq C_2\omega(f^{(j)},\delta)$ whenever $f\in C^j[t_0,t_n]$ together imply

$$\|(s(f))^{(j)}\|_{[t_0,t_n]}\leqq C_3 \max_l |f[t_l,\cdots,t_{l+j}]|$$

whether f is in C^j or not. This follows from the well-known theorem of Favard concerning the existence of smooth interpolants g satisfying $\|g^{(j)}\|\leqq C_4\max_l|f[t_l,\cdots,t_{l+j}]|$. This theorem is normally stated with $g\in L^j_\infty$ but simple smoothing arguments then yield the analogous result for C^j interpolants. The above can be restated as saying that the operator norms of the derived operators $s^{(j)}$, defined by $s^{(j)}(f^{(j)})=(s(f))^{(j)}$, are bounded.

3. Existence of monotone spline interpolants with good approximation properties. In this section we discuss sufficient conditions on the knots for the existence of monotone interpolants with good approximation properties. Consideration of step-function data reveals that the existence of "Heaviside" splines is a necessary condition. It is also a large part of the sufficient conditions below.

LEMMA 3.1. *Let $k\geqq 3$ and $\mathbf{r}$: $r_0\leqq r_1\leqq\cdots\leqq r_m$ be a superset of a strictly increasing knot set $\mathbf{t}$. Suppose $r_{i+k-2}>r_i$, for $0\leqq i\leqq m-k+2$, and that each interval $[t_i,t_{i+1}]$ contains at least k knots r_l. Then there exists a "Heaviside" spline $H_i\in S_{\mathbf{r},k}\subset C^1[t_0,t_n]$*

such that

$$H_i(t)=\begin{cases}0, & t\leqq t_i,\\ 1, & t\geqq t_{i+1},\end{cases} \tag{3.1}$$

$$\|H_i'\|\leqq (k-1)/(t_{i+1}-t_i) \tag{3.2}$$

and H_i is monotone.

Proof. Fix i, $0\leqq i<n$. If necessary by thinning the sequence r we may assume the interval $[t_i, t_{i+1}]$ contains exactly k of the knots r, namely $t_i=r_l,\cdots,r_{l+k-1}=t_{i+1}$. Let $N_{l,k-1}$ be the B-spline of order $k-1$ for the knots r supported on $[r_l, r_{l+k-1}]$. The normalization is such that $\sum_{p\in Z} N_{p,k-1}(x)=1$. Then let

$$H_i(x)=\int_{t_0}^{x}\frac{k-1}{t_{i+1}-t_i}N_{l,k-1}(u)\,du.$$

That H_i has the desired properties follows from the formula for the integral of a B-spline, and the positivity of a B-spline on the interior of its support. □

THEOREM 3.2. *Let $k\geqq 3$. Let a mesh* t: $t_0<t_1<\cdots<t_n$ *be given with mesh size δ. Let* r: $r_0\leqq r_1\leqq\cdots\leqq r_m$ *be a superset of* t *with $r_{l+k-2}>r_l$ for $0\leqq l\leqq m-k+2$. Further suppose that each interval $[t_i, t_{i+1}]$ contains at least k knots r_l. Let $1\leqq j\leqq k-1$. Then for each monotone $f\in C^j[t_0, t_n]$, there exists a spline $s\in S_{r,k}\subset C^1[t_0, t_n]$ such that*

(i) $s(t_i)=f(t_i)$, $0\leqq i\leqq n$;

(ii) *s is monotone on* $[t_0, t_n]$;

(iii) $\|f^{(l)}-s^{(l)}\|_{[t_0,t_n]}\leqq C_5\delta^{j-l}\omega(f^{(j)},\delta)$, $l=0, 1$.

Remark. If knots of multiplicity greater than one are avoided, then the guaranteed smoothness increases as $S_{r,k}\subset C^{k-2}$.

Proof. Suppose, without loss of generality, that f is increasing. Let $g=f'$. Then $g\in C^{j-1}[t_0, t_n]$ is a nonnegative function. Applying Theorem 1 of [2] there exists a spline $s_0'\in S_{t,k-1}$ such that

$$0\leqq s_0'(x)\leqq g(x),\qquad x\in[t_0, t_n], \tag{3.3}$$
$$\|g-s_0'\|\leqq C_6\delta^{j-1}\omega(f^{(j)},\delta).$$

Set $s_0(x)=f(t_0)+\int_{t_0}^{x}s_0'(u)\,du$.

We will correct $s_0(x)$ to interpolate to f using the functions $H_i(x)$ of Lemma 3.1. Given $s_i(x)$ interpolating to f at $t_0,\cdots,t_i$ define

$$e_i=f(t_{i+1})-s_i(t_{i+1})\quad\text{and}\quad s_{i+1}(x)=s_i(x)+e_iH_i(x).$$

Clearly s_{i+1} interpolates to f at $t_0,\cdots,t_{i+1}$. Also since $s_i'(x)=s_0'(x)$, for $x\geqq t_i$,

$$\begin{aligned}0\leqq e_i&\leqq f(t_{i+1})-s_i(t_{i+1})\\ &=\int_{t_i}^{t_{i+1}}f'(x)-s_0'(x)\,dx\\ &\leqq C_7(t_{i+1}-t_i)\delta^{j-1}\omega(f^{(j)},\delta).\end{aligned} \tag{3.4}$$

Take $s(x)=s_n(x)$. Then because H_i' is supported on $[t_i, t_{i+1}]$

$$\begin{aligned}\|f'-s'\|_{[t_i,t_{i+1}]}&\leqq\Big\|f'-s_0'-\sum_{i=1}^{n-1}e_iH_i'\Big\|_{[t_i,t_{i+1}]}\\ &\leqq\|f'-s_0'\|_{[t_i,t_{i+1}]}+e_i\|H_i'\|_{[t_i,t_{i+1}]}\\ &\leqq C_8\delta^{j-1}\omega(f^{(j)},\delta),\end{aligned}$$

by (3.2), (3.3), and (3.4). Since s interpolates f at all the t_i's we find

$$\|f-s\|_{[t_i,t_{i+1}]} \leq C_9 h_i \delta^{j-1} \omega(f^{(j)}, \delta). \qquad \square$$

Note that it is not possible, in general, to find a monotone cubic spline interpolant with *simple knots* only at the nodes of interpolation **t**. Existence of "Heaviside" splines fails! Indeed something stronger is true. Namely that each interval $[t_i, t_{i+1}]$ must contain at least four knots counting multiplicities in order that monotone cubic spline interpolants exist. If not, then consider that step-function data s' is a nontrivial quadratic spline supported within $[t_i, t_{i+1}]$. This is a contradiction as quadratic B-splines are quadratic splines of minimal support and their support spans four knots counting multiplicities.

However, if we add some extra knots, then Theorem 3.2 applies and monotone spline interpolants exist. In particular adding $2n$ extra knots, two in each interval (t_i, t_{i+1}), we obtain Corollary 3.3.

COROLLARY 3.3. *Let* **t**: $t_0 < t_1 < \cdots < t_n$ *and a monotone function* $f \in C^j[t_0, t_n]$, *for some* $1 \leq j \leq 3$, *be given. Let* **r**: $r_0 < r_1 < \cdots < r_{3n}$ *be such that*

$$t_i = r_{3i} < r_{3i+1} < r_{3i+2} < r_{3i+3} = t_{i+1}, \qquad i = 0, \cdots, n-1.$$

Then there exists a C^2 *piecewise cubic with knots* **r** *such that*

(i) $s(t_i) = f(t_i)$, $0 \leq i \leq n$;

(ii) s *is monotone on* $[t_0, t_n]$;

(iii) $\|f^{(l)} - s^{(l)}\|_{[t_0,t_n]} \leq C_{10} \delta^{j-l} w(f^{(j)}, \delta)$, $l = 0, 1$,

where $\delta = \max_i (t_{i+1} - t_i)$.

Similarly adding $n+1$ extra knots, one at each node of interpolation, thereby decreasing the guaranteed continuity to C^1, we obtain Corollary 3.4.

COROLLARY 3.4. *Let* **t**: $t_0 < t_1 < \cdots < t_n$ *and a monotone function* $f \in C^j[t_0, t_n]$, *for some* $1 \leq j \leq 3$, *be given. Define* **r** *by*

$$t_0 = r_0 = r_1 < t_1 = r_2 = r_3 < \cdots < t_n = r_{2n} = r_{2n+1}.$$

Then there exists a C^1 *piecewise cubic with knots* **r** *such that*

(i) $s(t_i) = f(t_i)$, $0 \leq i \leq n$;

(ii) s *is increasing on* $[t_0, t_n]$;

(iii) $\|f^{(l)} - s^{(l)}\|_{[t_0,t_n]} \leq C_{11} \delta^{j-l} \omega(f^{(j)}, \delta)$, $l = 0, 1$,

where $\delta = \max_i (t_{i+1} - t_i)$.

4. Meta-algorithms. Several constrained spline-fitting algorithms proposed in recent years (see, e.g., McAllister and Roulier [13], Fritsch and Carlson [11], Yan [14]) have the following *fit and modify* structure.

META-ALGORITHM 1. Given data $\{(t_i, f(t_i))\}_{i=0}^n$ and $k \geq 3$, choose a knot sequence **r**, $\mathbf{r} \supset \mathbf{t}$. Then

Step 1. Fit an initial interpolant $s_1 \in S_{\mathbf{r},k}$ possibly not satisfying the constraints.

Step 2. Modify s_1 to obtain a spline $s_2 \in S_{\mathbf{u},k}$ satisfying the constraints, where $\mathbf{u} \supset \mathbf{r}$.

We will classify all such algorithms as being of the *fit and modify type*. All monotonicity preserving interpolation algorithms with the related structure below are *j-optimal* for $1 \leq j \leq k-1$. That is, they satisfy the optimal order error estimates

$$\|f^{(l)} - s^{(l)}\|_{[t_0,t_n]} \leq C_{12} \delta^{j-l} \omega(f^{(j)}, \delta), \qquad l = 0, 1, \tag{4.1}$$

whenever $f \in C^j$ for some $1 \leq j \leq k-1$. In this meta-algorithm certain natural approximation properties are required in Step 1 and Step 2.

META-ALGORITHM 2. Given monotone data $\{(t_i, f(t_i))\}_{i=0}^n$ and $k \geqq 3$ let δ denote the meshsize of t. Choose a knot sequence $\mathbf{r}$, $\mathbf{r} \supset \mathbf{t}$, satisfying the hypotheses of Lemma 3.1.

Step 1. Fit an initial interpolant $s_1 \in S_{\mathbf{r},k}$ with, for some subset of the j's in $1 \leqq j \leqq k-1$, the optimal order approximation property $\|f' - s_1'\| = O(\delta^{(j-1)}\omega(f^{(j)}, \delta))$.

Step 2. Modify s_1 to obtain a monotone interpolant $s_2 \in \mathscr{C} \cap S_{\mathbf{u},k}$ with

$$\|s_1' - s_2'\| \leqq A \inf_{s \in \mathscr{C} \cap S_{\mathbf{r},k}} \|s_1' - s'\|,$$

where $\mathbf{u}$, $\mathbf{u} \supset \mathbf{r}$, satisfies the hypotheses of Lemma 3.1, A is a constant not depending on $\mathbf{t}$, f or s_1, and $\mathscr{C}$ is the space of monotone functions interpolating the given data values $f(t_i)$ at the nodes t_i.

To see the error estimates let s_3 be the monotone spline interpolant whose existence is guaranteed in Theorem 3.2. Then

$$\begin{aligned}\|s_1' - s_3'\| &\leqq \|s_1' - f'\| + \|f' - s_3'\| \\ &\leqq C_{13}\delta^{j-1}\omega(f^{(j)}, \delta).\end{aligned}$$

Hence

$$\begin{aligned}\|f' - s_2'\| &\leqq \|f' - s_1'\| + \|s_1' - s_2'\| \\ &\leqq \|f' - s_1'\| + A\|s_1' - s_3'\| \\ &\leqq C_{14}\delta^{j-1}\omega(f^{(j)}, \delta).\end{aligned}$$

The estimate of $\|f - s_2\|$ follows from this since s_2 interpolates f at all the t_i's.

5. Approximate projections onto convex sets. Suppose we have a convex set $\mathscr{C}$ in $\mathbf{R}^n$ and a point α outside it. Then provided the boundary, or at least the relevant part of it, is smooth, the shortest path to $\mathscr{C}$ is a line segment that intersects its boundary at right angles. It turns out that the distance from α to $\mathscr{C}$ along suboptimal line segments can be bounded in terms of the angle of intersection and the optimal distance.

Recall that the angle θ, $0 \leqq \theta \leqq \pi$, between two vectors x and y is given by $\cos\theta = x^T y/(\|x\|_2\|y\|_2)$. If γ is on the surface of a body and α is outside it, define the angle θ between $\overline{\gamma\alpha}$ and the surface to be the angle between $\overline{\gamma\alpha}$ and its projection on the tangent plane to the surface at γ. Clearly, $0 \leqq \theta \leqq \pi/2$.

LEMMA 5.1. *Let $\mathscr{C}$ be any closed bounded convex set in $\mathbf{R}^n$ with C^1 boundary $\partial\mathscr{C}$. Let α be a point outside $\mathscr{C}$ and β be the nearest point in $\mathscr{C}$ to α. Let γ be any point on $\partial\mathscr{C}$ such that $\overline{\gamma\alpha}$ meets $\mathscr{C}$ only at γ and at an angle θ, $0 < \theta \leqq \pi/2$. Then*

$$\|\alpha - \gamma\|_2 \leqq \frac{1 + \cos\theta}{\sin\theta}\|\alpha - \beta\|_2.$$

In order to prove this lemma we need the following technical result.

LEMMA 5.2. *Let $0 < \theta \leqq \pi/2$ and $\mathbf{D}$ be the 2×2 matrix such that*

$$\mathbf{D}^{-1} = \begin{bmatrix} 1 & \cos\theta \\ 0 & \sin\theta \end{bmatrix}.$$

Then

$$\|\mathbf{D}\|_2 = \frac{\sqrt{1 + \cos\theta}}{\sin\theta} \quad \text{and} \quad \|\mathbf{D}^{-1}\|_2 = \sqrt{1 + \cos\theta}.$$

Proof. Note $\|\mathbf{B}\|_2 = \sqrt{\rho(\mathbf{B}^T\mathbf{B})}$ and calculate.

Proof of Lemma 5.1. Consider the change of coordinates $\kappa' = \mathbf{A}\kappa$, where $\mathbf{A}$ is a nonsingular matrix. This transformation takes points to points, lines to lines, and convex sets to convex sets. In particular $\partial\mathscr{C}' = \mathbf{A}\partial\mathscr{C}$. Suppose $\mathbf{A}$ is chosen so that the angle between $\overline{\gamma'\alpha'}$ and $\mathscr{C}'$ is $\pi/2$, as in Fig. 1. Then γ' is the closest point in $\mathscr{C}'$ to α'. Hence

FIG. 1

$$\|\alpha-\gamma\|_2/\|\mathbf{A}^{-1}\|_2 \leqq \|\alpha'-\gamma'\|_2 \leqq \|\alpha'-\beta'\|_2 \leqq \|\mathbf{A}\|_2\|\alpha-\beta\|_2,$$

implying

$$\|\alpha-\gamma\|_2 \leqq \|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2\|\alpha-\beta\|_2.$$

It remains to show that there exists a matrix $\mathbf{A}$ with the required angle spreading property and $\|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2 \leqq (1+\cos\theta)/\sin\theta$.

We may assume without loss of generality that $\gamma = 0$. Also the result is trivial when $\theta = \pi/2$. Hence we may assume $0<\theta<\pi/2$. Then $\alpha-\gamma=\alpha$ can be resolved into two nonzero components, ψ, lying in the tangent plane T_γ and η perpendicular to the tangent plane. Let δ_1 and δ_2 be two unit vectors in the directions of ψ and η, respectively and choose $\delta_3,\cdots,\delta_n$ so as to complete an orthonormal basis for $\mathbf{R}^n$. Change to coordinates with respect to this basis given by $\hat{\kappa} = \mathbf{B}\kappa$ where $\mathbf{B} = [\delta_1:\delta_2:\cdots:\delta_n]^T$ is an orthogonal matrix. In the new coordinates $\hat{\psi}$ and $\hat{\eta}$, the components of $\hat{\alpha}-\hat{\gamma}=\hat{\alpha}$ in and perpendicular to the tangent plane, $\hat{T}_{\hat{\gamma}}$, may be written

$$\hat{\psi} = [a\cos\theta, 0, \cdots, 0]^T \quad\text{and}\quad \hat{\eta} = [0, a\sin\theta, 0, \cdots, 0]^T,$$

where a is positive.

Now choose $\mathbf{D}$ as the 2×2 matrix, which as a premultiplier opens the angle between $[1,0]^T$ and $[\cos\theta, \sin\theta]^T$ to $\pi/2$. More precisely, choose

$$\mathbf{D}^{-1} = \begin{bmatrix} 1 & \cos\theta \\ 0 & \sin\theta \end{bmatrix}.$$

Then, by Lemma 5.2, $\|\mathbf{D}\|_2\|\mathbf{D}^{-1}\|_2 = (1+\cos\theta)/\sin\theta$. Let

$$\mathbf{C} = \begin{bmatrix} \mathbf{D} & O \\ O & \mathbf{I}_{n-2} \end{bmatrix}$$

and $\mathbf{A} = \mathbf{C}\mathbf{B}$. Then in the $\kappa' = \mathbf{A}\kappa = \mathbf{C}\hat{\kappa}$ coordinate system the angle between $\overline{\gamma'\alpha'}$ and $T'_{\gamma'}$ is $\pi/2$ as required. Also $\|\mathbf{A}\|_2 = \|\mathbf{D}\|_2$ and $\|\mathbf{A}^{-1}\|_2 = \|\mathbf{D}^{-1}\|_2$ so $\|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2 = (1+\cos\theta)/\sin\theta$ and the result follows. □

6. Previous work on monotone piecewise cubic interpolation algorithms. Fritsch and Carlson [11] obtained the following set of necessary and sufficient conditions for a cubic to be monotone on an interval. Define the "monotonicity region" $\mathscr{M}$ to be the closure of that part of the first quadrant bounded by the coordinate axes and the "upper half" of the ellipse

$$\alpha^2+\beta^2+\alpha\beta-6\alpha-6\beta+9=0, \tag{6.1}$$

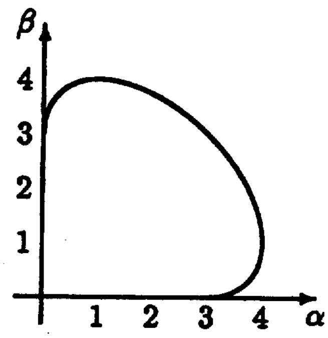

FIG. 2. *The monotonicity region* $\mathcal{M}$.

as shown in Fig. 2. We note that the boundary $\partial\mathcal{M}$ is horizontal at $(1,4)$ and $(3,0)$ and vertical at $(0,3)$ and $(4,1)$. Let s be a C^1 piecewise cubic, with knots only at the t_i's, interpolating the given data $\{(t_i, y_i)\}_{i=0}^n$. Define $d_i = s'(t_i)$ and $S_i = (y_{i+1} - y_i)/(t_{i+1} - t_i)$. Then [11] s is monotone on $[t_i, t_{i+1}]$ if and only if

$$(d_i, d_{i+1}) \in \mathcal{M}_i = S_i\mathcal{M}.$$

Motivated by these necessary and sufficient conditions Fritsch and Carlson [11], and later Hyman [12], developed *fit and modify* cubic spline interpolation algorithms with Step 2 implemented as follows. They choose a subset $\mathcal{S} \subset \mathcal{M}$ and if $(\alpha, \beta) = (d_i/S_i, d_{i+1}/S_i)$ does not belong to $\mathcal{S}$ then modify (d_i, d_{i+1}) until it does. They recommend choosing $\mathcal{S}$ as that part of $\mathcal{M}$ within the disc $\alpha^2 + \beta^2 \leq 9$ and the square $[0,3] \times [0,3]$, respectively. Such a procedure has the advantage of being very simple to implement. However, it limits the order of approximation. Indeed, Eisenstat, Jackson, and Lewis [9], [10] show that if $(4,1) \notin \mathcal{S}$ then the method is at best third-order accurate. It follows that the Fritsch–Carlson, and Hyman methods are at best third-order accurate, and thus not three-optimal. Using the methods of [9], [10] we obtain Lemma 6.1.

LEMMA 6.1. *Consider a monotone interpolation algorithm of the type described above. Then if the elliptical part of the boundary of* $\mathcal{M}$ *is not contained in* $\mathcal{S}$ *the algorithm is at best third-order accurate. That is, there exists at least one monotone function* $f \in C^\infty[-1,1]$ *and a sequence of meshes* $\{\mathbf{t}_\delta\}$, $\mathbf{t}_\delta$ *having mesh length* δ, *for which the error* $\|f - s_\delta\| \neq o(\delta^3)$ *as* $\delta \to 0$. *Consequently, the method is not three-optimal in the sense defined in* § 2.

Proof. Consider $f(t) = t^3$ on $[-1,1]$. Fix a point γ on the elliptical boundary of $\mathcal{M}$ but not in $\mathcal{S}$. Then for each δ sufficiently small and each $0 \leq \theta \leq 1$, there exists a mesh $\mathbf{t}$, with mesh length δ, such that for some i, $t_i = -\theta\delta$ and $t_{i+1} = (1-\theta)\delta$. But as θ varies between zero and 1

$$(\alpha, \beta) = \left(\frac{f'(t_i)}{S_i}, \frac{f'(t_{i+1})}{S_i}\right) = \left(\frac{3\theta^2}{(1-\theta)^3 + \theta^3}, \frac{3(1-\theta)^2}{(1-\theta)^3 + \theta^3}\right)$$

sweeps out the upper half of the ellipse (6.1). Hence, for a suitable choice of θ this point coincides with γ. Thus the first derivative approximations produced by the algorithm satisfy $\|(d_i, d_{i+1}) - (f'(t_i), f'(t_{i+1}))\|_2 > C\delta^2$ for some $C > 0$ not depending on δ. Markov's inequality then implies $\|s - f\|_{[t_i, t_{i+1}]} \geq C_{15}\delta^3$. □

Eisenstat, Jackson, and Lewis [9], [10] obtain several other results relating the choice of $\mathcal{S}$ to the order of approximation. They also present a *fit and modify type* algorithm that is shown to give fourth-order approximation to C^4 monotone functions under certain additional assumptions. The extra hypothesis in [9] takes the form of a stronger smoothness assumption in the neighbourhood of multiple zeros of f. In [10] this is replaced by the more pleasant condition that the local mesh ratio be bounded away from zero and infinity. Their algorithm uses cubic splines with double knots at the t_i's. Yan [14] develops a C^1 monotone cubic spline interpolation algorithm that gives fourth-order approximation to C^4 monotone functions f. In this *fit and modify* type algorithm two extra knots, which may have multiplicity greater than 1, are judiciously inserted in every subinterval within which the initial interpolant is not monotone. This allows monotone interpolation on $[t_i, t_{i+1}]$ without changing the initial values of $s'(t_i)$ and $s'(t_{i+1})$, thereby decomposing the problem of finding a monotone interpolant into small and easily solved subproblems.

7. New monotonicity preserving cubic spline interpolation algorithms. In this section we will present three algorithms that, given monotone data $\{(t_i, y_i)\}_{i=0}^{n}$, fit a C^1 monotone piecewise cubic interpolant. The algorithms are of the *fit and modify* type described in § 4. They are constructed to be *j-optimal* for $1 \leq j \leq 3$.[1] Recall from § 2 that the error is then $\mathcal{O}(\delta^j \omega(f^{(j)}, \delta))$ whenever the data comes from a C^j function f, $1 \leq j \leq 3$. In particular this implies $\mathcal{O}(\delta^4)$ approximation to C^4 functions. Of the three we recommend the third, which is labeled Algorithm 2, as a reasonable compromise between approximation and smoothness considerations. The first algorithm was announced in [3].

Before constructing the algorithms we need some preliminary material. The reader will recall that there is a two-parameter family of C^2 cubic splines s with simple knots only at the data points satisfying the interpolation conditions $s(t_i) = y_i$, $0 \leq i \leq n$. s can be uniquely determined by specifying two more *end conditions*. In the absence of derivative data some suitable choices are the "not-a-knot" condition

$$s_-^{(3)}(t_1) = s_+^{(3)}(t_1) \quad \text{and} \quad s_-^{(3)}(t_{n-1}) = s_+^{(3)}(t_{n-1}) \tag{7.1}$$

of de Boor, and the conditions

$$s'(t_0) = c_1'(t_0) \quad \text{and} \quad s'(t_n) = c_2'(t_n), \tag{7.2}$$

or

$$s''(t_0) = c_1''(t_0) \quad \text{and} \quad s''(t_n) = c_2''(t_n), \tag{7.3}$$

where c_1 and c_2 are cubic polynomials such that

$$c_1(t_i) = y_i \quad \text{and} \quad c_2(t_{n-i}) = y_{n-i}, \quad 0 \leq i \leq 3.$$

These end conditions are analyzed in [4] and the resulting splines are j-optimal for $1 \leq j \leq 3$.[1] Subsequent work [5] indicates that the condition (7.2) is significantly better than the other two for general purpose use.

We now present the first algorithm. Because this is not presented in a sufficiently concrete form so as to be immediately implementable, we will call it a meta-algorithm. It involves a nonlinear programming approach that is not as yet competitive in terms of computational cost with the algorithms to come later. However, it has a flexibility the others do not, and could easily be modified to accommodate mixed approximation and smoothing criteria in the objective.

[1] For $j = 1$ this requires the additional assumption that $(t_1 - t_0)/(t_2 - t_1)$ and $(t_n - t_{n-1})/(t_{n-1} - t_{n-2})$ be bounded above as the mesh changes. However, some mesh dependence is required of all (unconstrained) interpolation schemes with error $\mathcal{O}(\delta^3 \omega(f^{(3)}, \delta))$ that use only function value information. See [4] for details.

META-ALGORITHM 3. Given monotone data $\{(t_i, y_i)\}_{i=0}^n$:

Step 1a. Fit a C^2 cubic spline s corresponding to one of the sets of end conditions (7.1), (7.2), or (7.3).

Step 1b. For $i := 0$ to n if $s'(t_i)$ has the wrong sign set

$$s'(t_i) := \min\{-s'(t_i), S_{i-1}, S_i\}. \tag{7.4}$$

Step 2. Further modify the new vector of $s'(t_i)$'s the least amount possible in the l^∞ norm to obtain monotonicity on $[t_0, t_n]$.

Step 3. Construct the piecewise cubic Hermite interpolant to the final data $\{(t_i, y_i, s'(t_i))\}_i$.

Meta-algorithm 3 fits a monotone C^1 cubic spline interpolant. It is easily seen to be an instance of Meta-algorithm 2 and to be j-optimal for $1 \leq j \leq 3$.[2] Step 1b is implemented as shown, rather than simply setting $s'(t_i)$ to zero, to try and avoid flat spots in the fitted curve when this is possible. Given this, and noting that if $s'(t_i)$ has the wrong sign it is in error at least by $|s'(t_i)|$, (7.4) seems a reasonable way of correcting the sign.

In Step 2 there are a lot of choices to be made. The nonlinear programming problem underlying this is:

Find $[d_0, \cdots, d_n]^T$ minimizing

$$\max_i |d_i - s_1'(t_i)|$$

subject to

$$(d_i, d_{i+1}) \in S_i\mathcal{M}, \qquad i = 0, 1, \cdots, n-1,$$

where s_1 is the spline interpolant at the end of Step 1.

This is, of course, a convex programming problem with a nondifferentiable objective and, for most data sets, a nonunique solution. However, it has a lot of structure that could be exploited by a special purpose code. Thus solving is not inherently excessively expensive. Also if $s'(t_i)$'s are not changed unnecessarily the C^2 continuity will be maintained where possible. This is since C^2 continuity is maintained at the knot t_i if $s'(t_{i-1})$, $s'(t_i)$, and $s'(t_{i+1})$ are unchanged from their initial values. General purpose nonlinear programming algorithms can be used if the problem above is restated in the differentiable form:

Minimize λ subject to

$$-\lambda \leq s_1'(t_i) - d_i \leq \lambda, \qquad 0 \leq i \leq n,$$

and

$$(d_i, d_{i+1}) \in S_i\mathcal{M}, \qquad 0 \leq i \leq n-1.$$

Actually, in implementing such an approach we recommend first projecting the $s_1'(t_i)$'s onto some suitable superset of $\mathcal{M}_i$, say $[0, 4S_i] \times [0, 4S_i]$, and then finding the nearest feasible vector $\mathbf{d}$ to this new vector of derivatives. This maintains the desirable error properties of the algorithm, while giving a better-conditioned optimization problem. With this refinement we have had success both with a simple, but very slow pattern search and with the Harwell routine VF03.

[2] This is subject to the earlier restriction on the local mesh ratios at both ends of the interval when $j = 1$.

We now turn to the second algorithm. This is based on the idea of projecting $(s'(t_i), s'(t_{i+1}))$ onto a superset of the monotonicity region $\mathcal{M}$ for each interval (t_i, t_{i+1}) and then adding an extra knot where necessary to allow monotonicity without further modifying the derivatives at the data points.[3] The superset chosen is $\mathscr{E}_i = S_i\mathscr{E}$, where $\mathscr{E}$ is the union of monotonicity region $\mathcal{M}$ with the squares $[0, 1]\times[3, 4]$ and $[3, 4]\times[0, 1]$, as shown in Fig. 3. The geometry of $\mathscr{E}$ is such that the approximate projection onto $\mathscr{E}$ can be chosen to never increase α or β, and then projections corresponding to neighbouring intervals will not move $(s'(t_i), s'(t_{i+1}))$ outside $\mathscr{E}_i$. The algorithm is derived from the following technical lemma giving one method for adding an extra breakpoint.

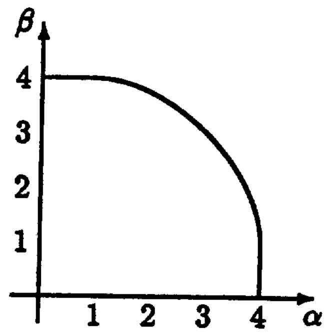

FIG. 3. *The extended monotonicity region* $\mathscr{E}$.

LEMMA 7.1. *Let* $1\leqq\gamma\leqq 2$ *and data* $g(0)$, $g(1)$, $g'(0)$, $g'(1)$ *be given with* $g(1)-g(0)=1$ *and* $g'(0), g'(1)\geqq 0$. *Let* $(\alpha, \beta)=(g'(0), g'(1))$, *and suppose that* $(\alpha, \beta)\in\mathscr{E}-\mathcal{M}$. *Then the cubic polynomial c interpolating the given Hermite data at* 0 *and* 1 *is not monotone increasing and* c' *has a negative minimum,* $-\varepsilon$, *at a unique point* x^* *in either* $(0, \frac{1}{3})$ *or* $(\frac{2}{3}, 1)$ *accordingly as* (α, β) *belongs to* $[0, 1]\times[3, 4]$ *or* $[3, 4]\times[0, 1]$. *In the first case define*

$$r(x)=\begin{cases}\dfrac{\varepsilon x(x-2x^*)}{(x^*)^2}, & 0\leqq x\leqq 2x^*,\\[2ex] \varepsilon\left(\dfrac{2x^*}{1-2x^*}\right)\left(\dfrac{2}{1-2x^*}\right)^2(x-2x^*)(1-x), & 2x^*<x\leqq 1,\end{cases}$$

and

$$\hat{c}(x)=c(x)-\gamma\int_0^x r(u)\,du.$$

Then $\hat{c}$ *is monotone increasing on* $[0, 1]$, *interpolates the same function and first derivative data at zero and* 1, *and satisfies* $\|\hat{c}'-c'\|_{[0,1]}\leqq C_{16}\varepsilon$. *When* $(\alpha, \beta)\in[3, 4]\times[0, 1]$, *r and* $\hat{c}$ *may be constructed analogously.*

Proof. Let $c'(x)=q(x)$. The functionals $f(0)$, $f(1)$, and $\int_0^1 f(x)\,dx$ form a basis for π_2' dual to the basis $\{3(x-\frac{1}{3})(x-1), 3x(x-\frac{2}{3}), 6x(1-x)\}$ for π_2. Hence

$$q(x)=3(\alpha+\beta-2)x^2+(6-4\alpha-2\beta)x+\alpha. \tag{7.5}$$

Since q is nonnegative at zero and 1, and negative somewhere in between, it has a

[3] In the context of monotone cubic spline interpolation, the device of adding extra knots is due to Yan [14]. It has also been used for constrained quadratic spline interpolation by McAllister and Roulier [13], and for convex cubic spline interpolation by de Boor [8].

minimum value $-\varepsilon$ on $[0, 1]$. Elementary calculus shows that this minimum value occurs at

$$x^* = \frac{2\alpha+\beta-3}{3(\alpha+\beta-2)} \tag{7.6}$$

and that

$$\varepsilon = \frac{(2\alpha+\beta-3)^2}{3(\alpha+\beta-2)} - \alpha. \tag{7.7}$$

Using the expression for x^* above, we see that if $(\alpha, \beta) \in [0, 1] \times [3, 4]$ then $x^* \in [0, \frac{1}{3}]$. Substituting in the expression for q, we see $q(\frac{1}{3}) = \frac{1}{3}(4-\beta) \geqq 0$. Rewriting q in the alternative form

$$q(x) = a(x-x^*)^2 - \varepsilon$$

where $a, \varepsilon > 0$ if $(\alpha, \beta)) \in ([0, 1] \times [3, 4]) \setminus \mathcal{M}$, $q(\frac{1}{3}) \geqq 0$ implies $\varepsilon \leqq a(\frac{1}{3} - x^*)^2$.

Now consider correcting q to be nonnegative on $[0, 1]$ with a piecewise quadratic correction $-\gamma r$ as in Fig. 4 and the statement of the lemma. The integral $\int_0^1 r$, is zero

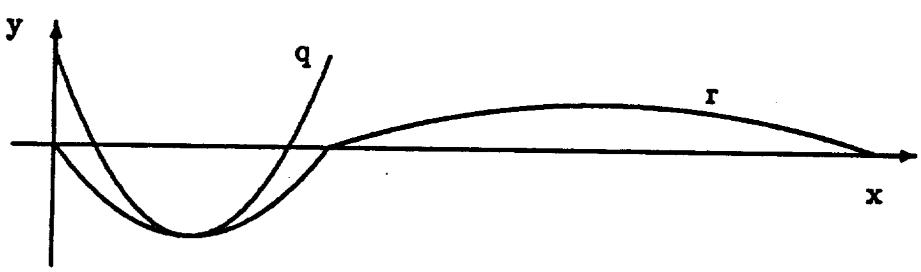

FIG. 4. $q = c'$ *and the corrector* r.

since if p is a quadratic with zeros at a and b, and value κ at $(a+b)/2$ then $\int_a^b p = 2(b-a)\kappa/3$. Clearly r is convex on $(0, 2x^*)$, concave on $(2x^*, 1)$, and $r(x) \leqq q(x)$ for $x \in (0, 2x^*)$. Hence $q - \gamma r$ is nonnegative on $(0, 2x^*)$. It remains to show that $q - \gamma r$ is nonnegative on $(2x^*, 1)$.

Since $q - \gamma r$ is convex on $(2x^*, 1)$, and $(q - \gamma r)(2x^*) \geqq 0$, it suffices to show $(q - \gamma r)'(2x^*+) \geqq 0$. Now

$$\begin{aligned}(q-\gamma r)'(2x^*+) &\geqq (q-2r)'(2x^*+)\\ &= 2x^*\left(a - \frac{8\varepsilon}{(1-2x^*)^2}\right)\\ &\geqq 2ax^*\left[1 - \frac{8}{9}\left(\frac{1-3x^*}{1-2x^*}\right)^2\right]\\ &\geqq 0\end{aligned}$$

since $0 \leqq (1-3x^*)/(1-2x^*) \leqq 1$ for $x^* \in [0, \frac{1}{3}]$. This completes the proof of the lemma. □

Lemma 7.1 can be applied to the general situation. We obtain from it the following algorithm fragment for adding an extra breakpoint and "artificial" data $s(u)$, $s'(u)$ so as to correct a C^1 piecewise cubic Hermite interpolant for monotonicity on what was, before the extra breakpoint was added, a single subinterval.

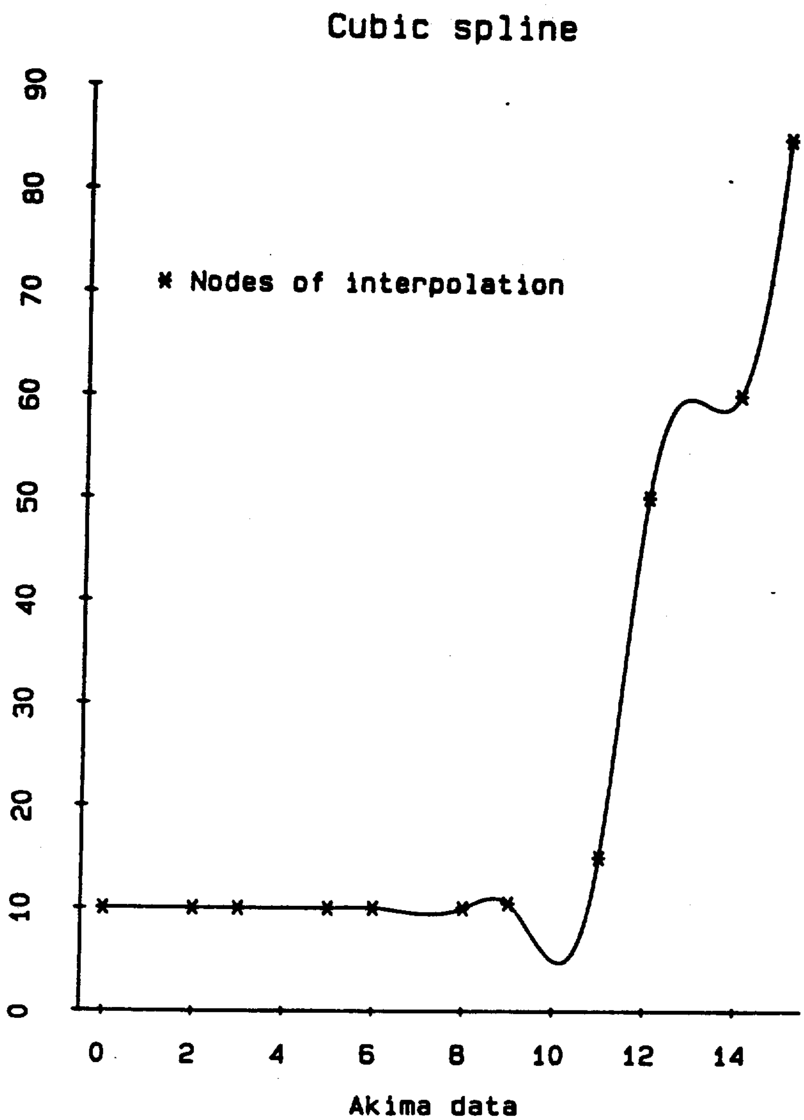

FIG. 5

ADDING A BREAKPOINT. Let s be a C^1 piecewise cubic with breakpoints t. Suppose $s(t_{i+1}) > s(t_i)$ and $s'(t_i), s'(t_{i+1}) \geqq 0$. Set $h = t_{i+1} - t_i$, $S_i = (s(t_{i+1}) - s(t_i))/h$, $\alpha = s'(t_i)/S_i$, and $\beta = s'(t_{i+1})/S_i$. Further suppose $1 \leqq \gamma \leqq 2$ and $(\alpha, \beta) \in \mathscr{C}/\mathscr{M}$. Then

Step 1. Let

$$\delta = \frac{h(2\alpha + \beta - 3)}{3(\alpha + \beta - 2)} \quad \text{and} \quad \varepsilon = S_i * \left(\frac{(2\alpha + \beta - 3)^2}{3(\alpha + \beta - 2)} - \alpha \right).$$

Step 2. If $\alpha < 1$ then insert an additional breakpoint at $u := t_i + 2\delta$ with $s(u) := s(u) + 4\gamma\varepsilon\delta/3$ and $s'(u)$ unchanged.

Else (i.e., when $\beta < 1$) let $\delta := h - \delta$ and insert an additional breakpoint at $u := t_{i+1} - 2 * \delta$ with $s(u) := s(u) - 4\gamma\varepsilon\delta/3$ and $s'(u)$ unchanged.

Step 3. Modify the original piecewise cubic replacing the single cubic span on (t_i, t_{i+1}) by the two cubic spans forming the Hermite cubic interpolant to the modified s and unmodified s' data at t_i, u, and t_{i+1}.

The derivation of this algorithm fragment from Lemma 7.1 is by means of a change of variable of the form $\hat{x} = a(x - t_i)$, $\hat{y} = by$. Step 1 comes directly from (7.6) and (7.7), and Step 2 from writing things in terms of the integral of the derivative corrector r.

Finally before presenting the algorithm we need to define a method for approximate projection onto $\mathscr{C}_i$. This is stated in terms of the scaled variables α and β occurring in the definition of the monotonicity region $\mathscr{M}$ (see § 6).

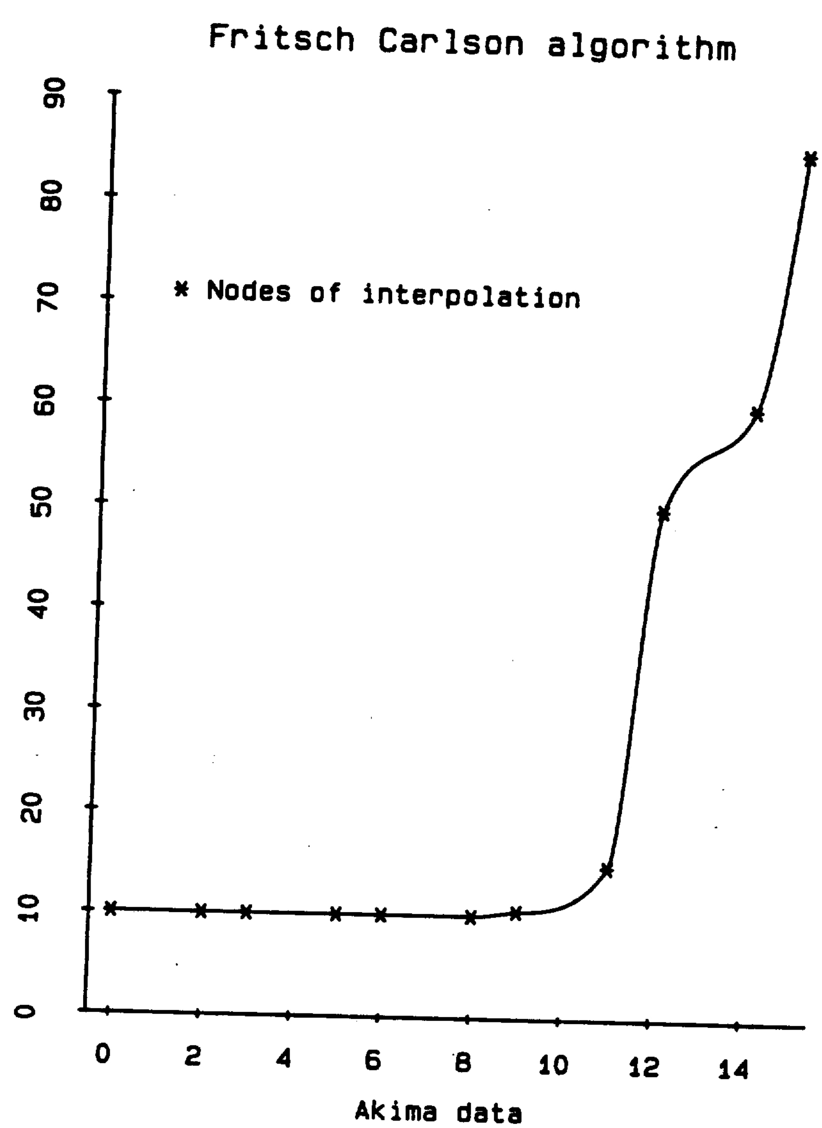

FIG. 6

APPROXIMATE PROJECTION ONTO $\mathscr{C}$. Given (α, β) outside $\mathscr{C}$:
If $\alpha \leqq 1$ let $\beta := 4$
else if $\beta \leqq 1$ let $\alpha := 4$
else project (α, β) to $\partial\mathscr{C}$ along the ray joining (α, β) to $(1, 1)$.

Since rays joining $(1, 1)$ to $\partial\mathscr{C}$ are never tangential to $\partial\mathscr{C}$, we can see from Lemma 5.1 that this moves points outside $\mathscr{C}_i$ an amount less than some constant times the minimum amount to bring them to $\partial\mathscr{C}_i$. We are now ready to present Algorithm 1.

ALGORITHM 1. Given monotone data $\{(t_i, y_i)\}_{i=0}^{n}$ and γ, $1 \leqq \gamma \leqq 2$:
Step 1a. Fit a C^2 cubic spline s corresponding to one of the sets of end conditions (7.1), (7.2), or (7.3).
Step 1b. For $i := 0$ to n, if $s'(t_i)$ has the wrong sign, set $s'(t_i) = -s'(t_i)$.

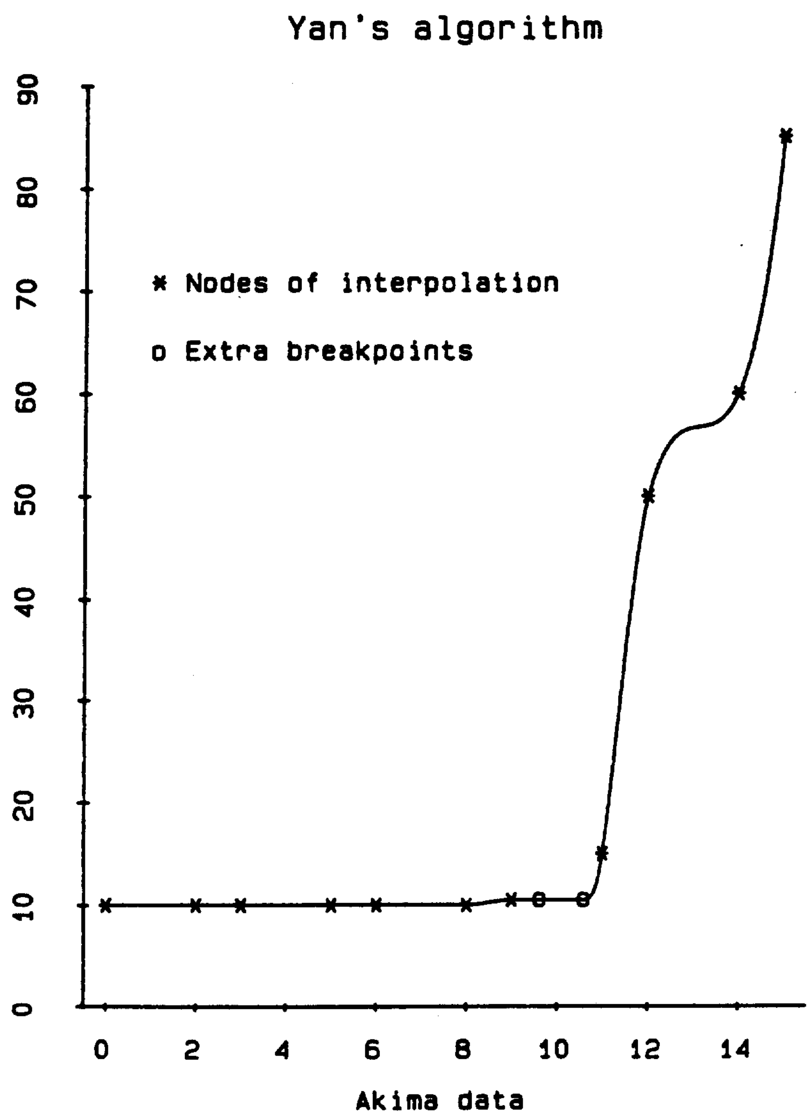

FIG. 7

Step 2a. For $i := 0$ to $n-1$ by twos, if $(s'(t_i), s'(t_{i+1})) \notin \mathscr{C}_i$, approximately project $(s'(t_i), s'(t_{i+1}))$ onto $\mathscr{C}_i$.

Step 2b. For $i := 1$ to $n-1$ by twos, if $(s'(t_i), s'(t_{i+1})) \notin \mathscr{C}_i$, approximately project $(s'(t_i), s'(t_{i+1}))$ onto $\mathscr{C}_i$.

Step 2c. For $i := 0$ to $n-1$, if $(s'(t_i), s'(t_{i+1})) \notin \mathscr{M}_i$, then add a breakpoint and extra data according to the scheme of the algorithm fragment so that the new spline is monotone on (t_i, t_{i+1}).

It is clear from the discussion above that Steps 2a and 2b ensure that the Hermite data for each subinterval (t_i, t_{i+1}) corresponds to a point in the extended monotonicity region. Extra breakpoints are then added as required to ensure monotonicity. Hence the output of Algorithm 1 is monotone. It remains to show the error estimates.

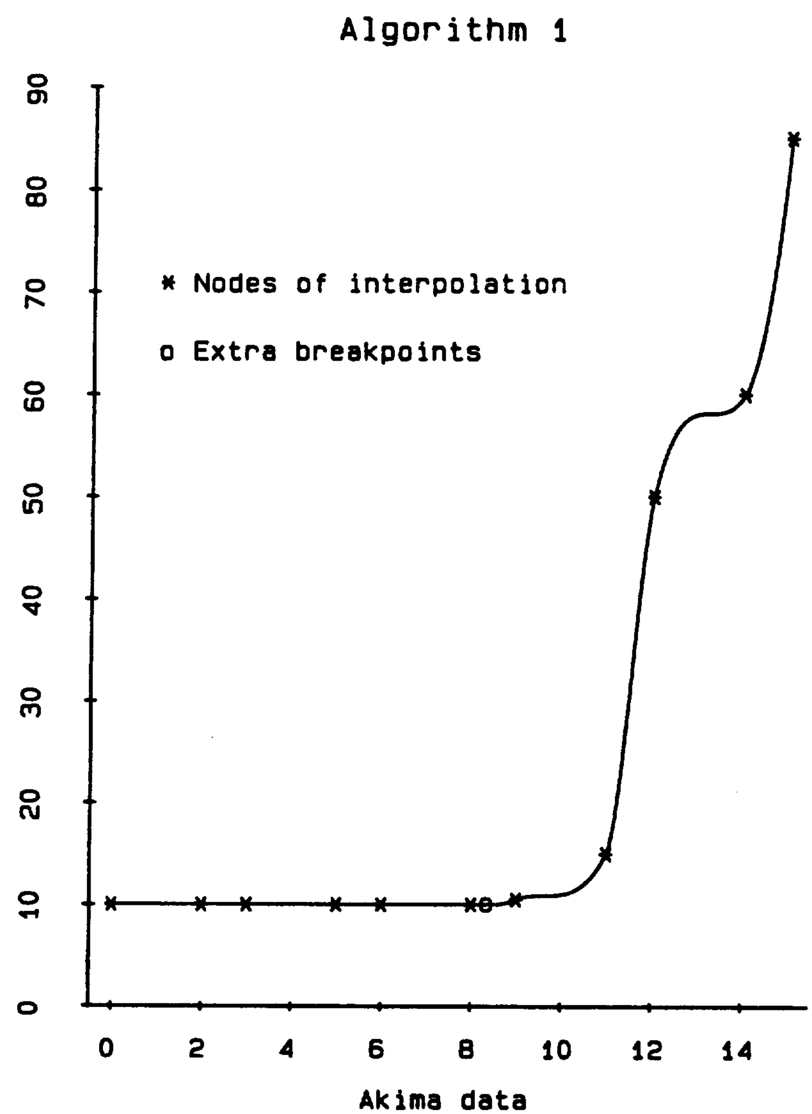

FIG. 8

Denote by s_{1a}, s_{1b}, etc. the piecewise cubic at the end of each of the respective steps. The j-optimality of s_{1b} for $1 \leq j \leq 3$ follows from [4]. The separation of the correction steps within 2a and 2b prevents possible error propagation—each correction being entirely local and corresponding to a single interval. Hence from the equivalence of norms in two-space and the properties of approximate projection discussed above, we see that

$$\inf_{s\in\mathscr{C}\cap S_{r,4}} \|s'_{2b}-s'\| \leq \|s'_{2b}-s'_{1b}\| \leq C_{17} \inf_{s\in\mathscr{C}\cap S_{r,4}} \|s'_{1b}-s'\|,$$

where $\mathscr{C}$ is the set of monotone functions interpolating the given data and $r_i = t_{[i/2]}$, $0 \leq i \leq 2n+1$. Also an application of Lemma 7.1 gives

$$\|s'_{2b}-s'_{2c}\| \leq C_{18} \inf_{s\in\mathscr{C}\cap S_{r,4}} \|s'_{2b}-s'\|.$$

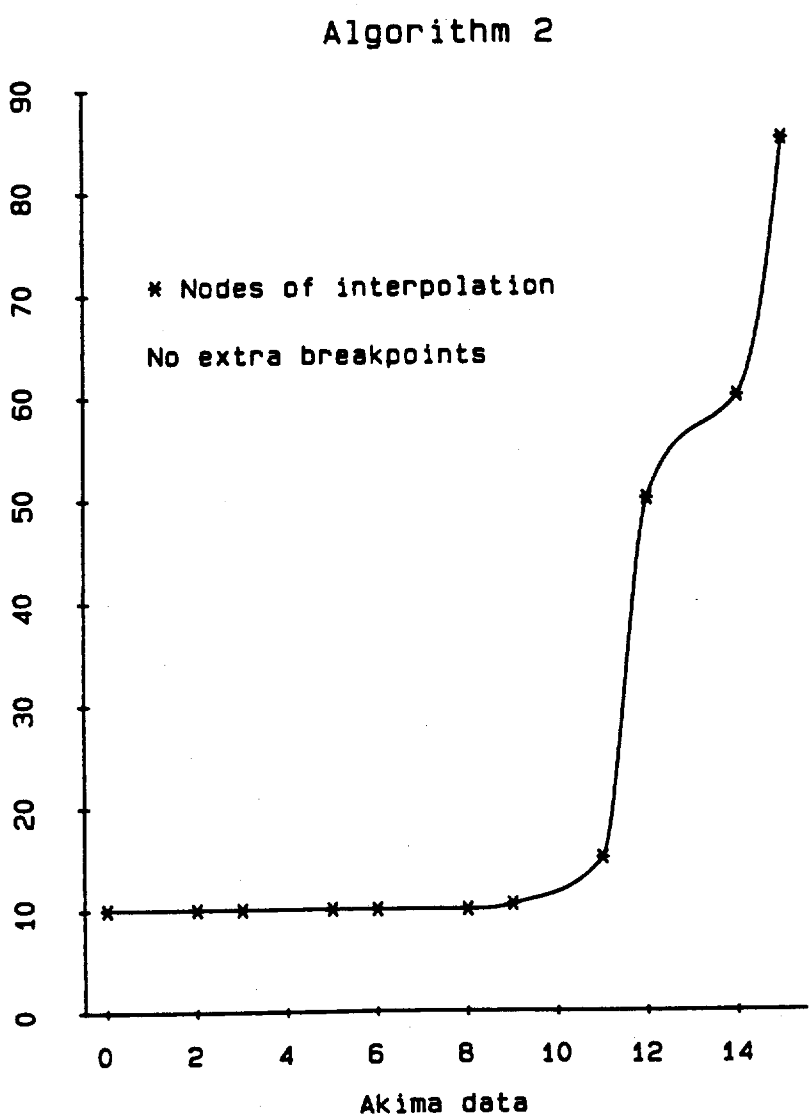

FIG. 9

Combining these results, we see

$$\|s'_{1b} - s'_{2c}\| \le C_{19} \inf_{s \in \mathscr{C} \cap S_{r,4}} \|s'_{1b} - s'\|.$$

Hence the algorithm is an instance of Meta-algorithm 2 and is j-optimal for $1 \le j \le 3$ by the argument given in § 4.

We now turn to Algorithm 2. This is based on the idea of relaxation—points outside of $\mathscr{M}_i$ will be moved further than necessary into the interior of $\mathscr{C}_i$. The primary motivation for trying this approach is that points on the boundary of $\mathscr{M}_i$ correspond to s' having a zero in $[t_i, t_{i+1}]$. As could be anticipated this algorithm adds fewer extra knots than Algorithm 1 in many cases.

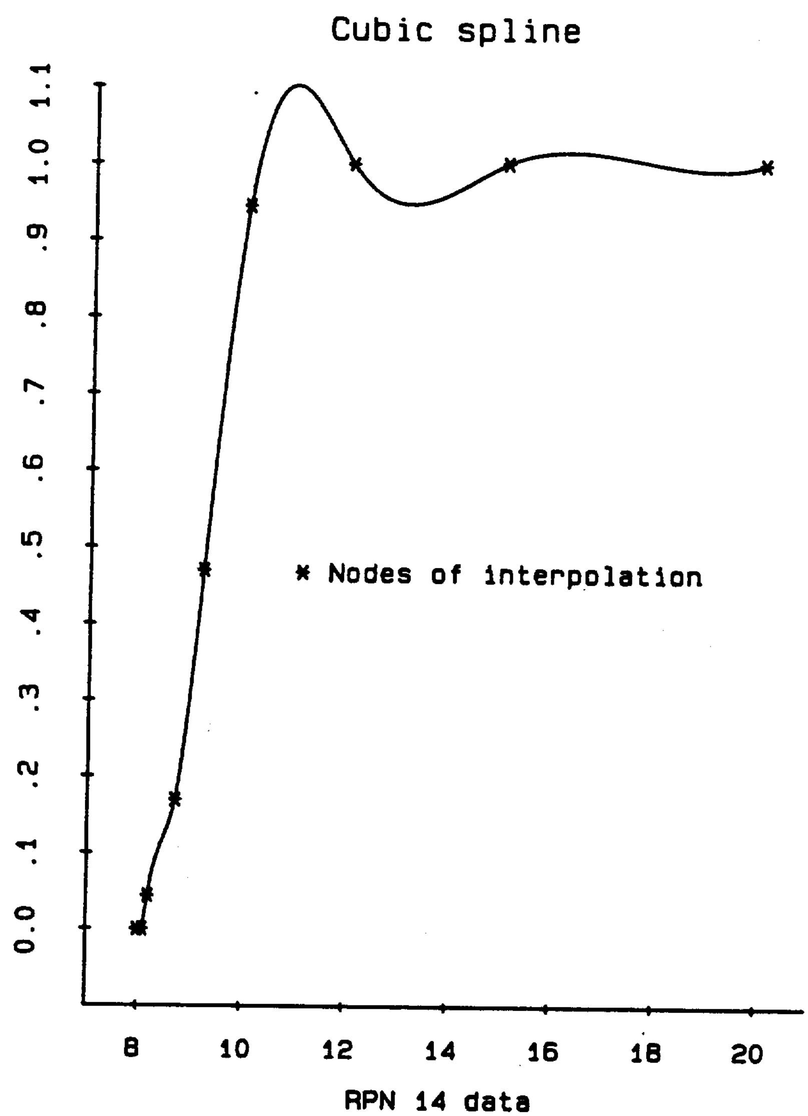

FIG. 10

We will call a nonnegative function $g \in C[0,1]$ a relaxation function if
(a) $g(x) \leq x$, for all $x \in [0,1]$,
(b) $(1-g(x))/(1-x)$ is bounded on $[0,1)$.
For example,

$$g(x)=\begin{cases} x/2, & x<\frac{2}{3}, \\ 2x-1, & x \geq \frac{2}{3}, \end{cases}$$

$$g(x)=\begin{cases} 2x/3, & x<\frac{2}{3}, \\ (5x-2)/3, & x \geq \frac{2}{3}, \end{cases}$$

and $g(x)=(x+x^2)/2$ are relaxation functions. With this definition in hand we can

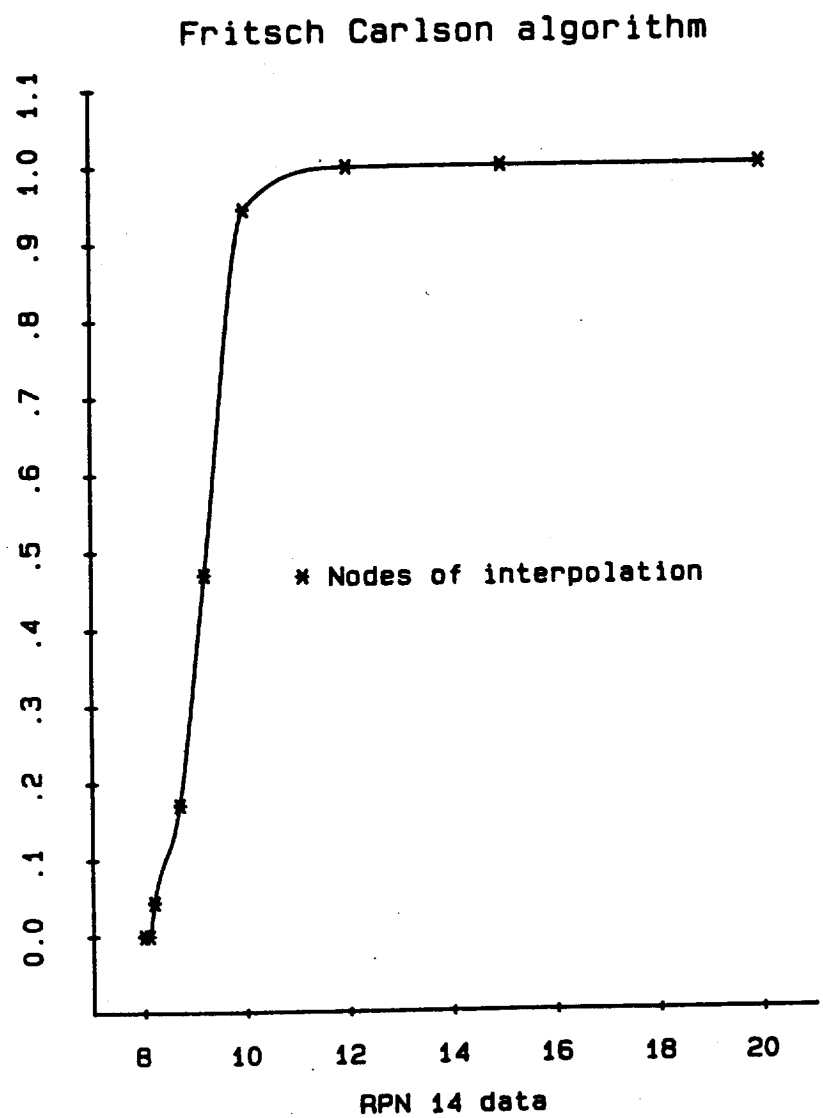

FIG. 11

define another method of projecting onto $\mathscr{E}_i$ in which the point outside of $\mathscr{M}_i$ is moved into the interior of $\mathscr{E}_i$.

RELAXED PROJECTION ONTO $\mathscr{E}$. Given a relaxation function g and a point (α, β) in the first quadrant and outside $\mathscr{M}$:

Step 1. Calculate $\lambda > 0$ such that $(1, 1) + \lambda(\alpha - 1, \beta - 1)$ lies on $\partial\mathscr{M}$.

Step 2. If $\alpha \leqq 1$ let $\beta := 1 + g(\lambda)(\beta - 1)$
else if $\beta \leqq 1$ let $\alpha := 1 + g(\lambda)(\alpha - 1)$
else let $(\alpha, \beta) = (1, 1) + g(\lambda)(\alpha - 1, \beta - 1)$.

Since rays joining $(1, 1)$ to $\partial\mathscr{M}$ are never tangential, Lemma 5.1 guarantees radial projection onto $\partial\mathscr{M}$, which moves points by a distance bounded by a multiple of the

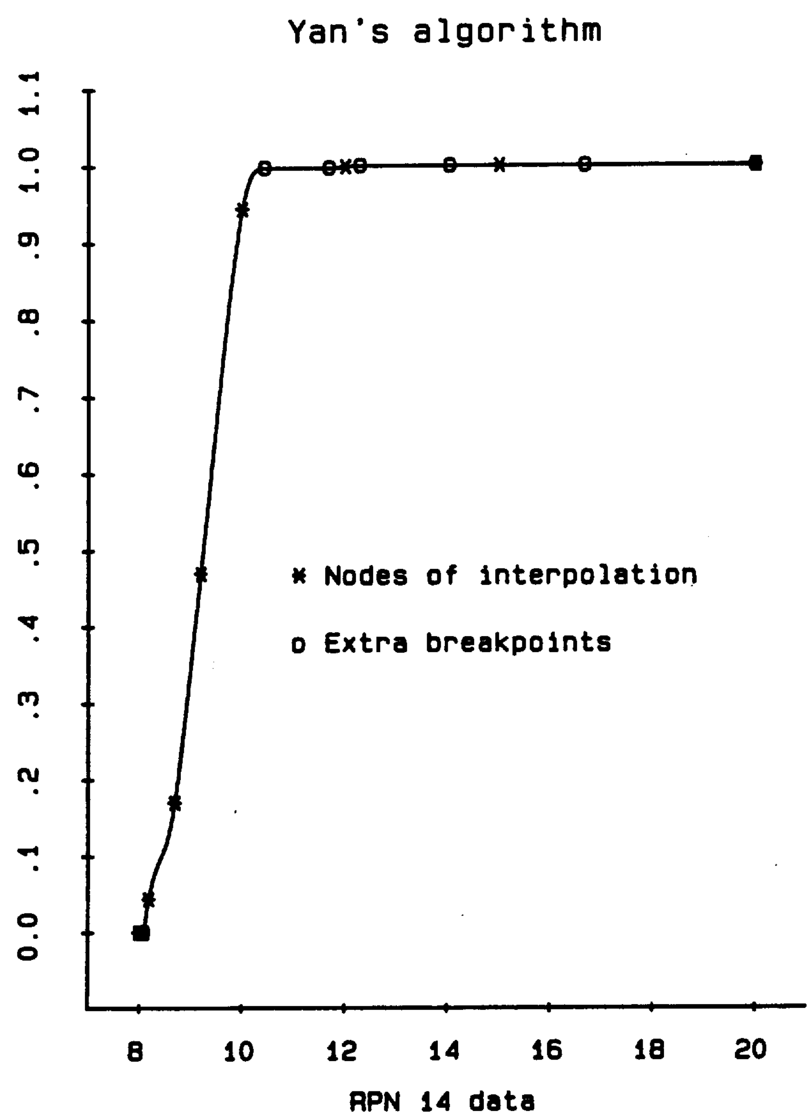

FIG. 12

minimal distance. Relaxed projection in turn moves points by a distance bounded by a multiple of the distance of radial projection. In terms of the parameter λ above points (α, β) with $\alpha, \beta > 1$ are moved $(1-g(\lambda))/(1-\lambda)$ times as far as they would be by radial projection. Thus the distance moved by relaxed projection is bounded by a multiple of the minimal distance. Using this result at the appropriate point, we find that the error estimates for Algorithm 2 below now follow by an argument analogous to that used for Algorithm 1.

ALGORITHM 2. Given monotone data $\{(t_i, y_i)\}_{i=0}^{n}$, γ, $1 \le \gamma \le 2$, and a relaxation function g:

Step 1a. Fit a C^2 cubic spline s corresponding to one of the sets of end conditions (7.1), (7.2), or (7.3).

Step 1b. For $i := 0$ to n, if $s'(t_i)$ has the wrong sign, set $s'(t_i) := -s'(t_i)$.

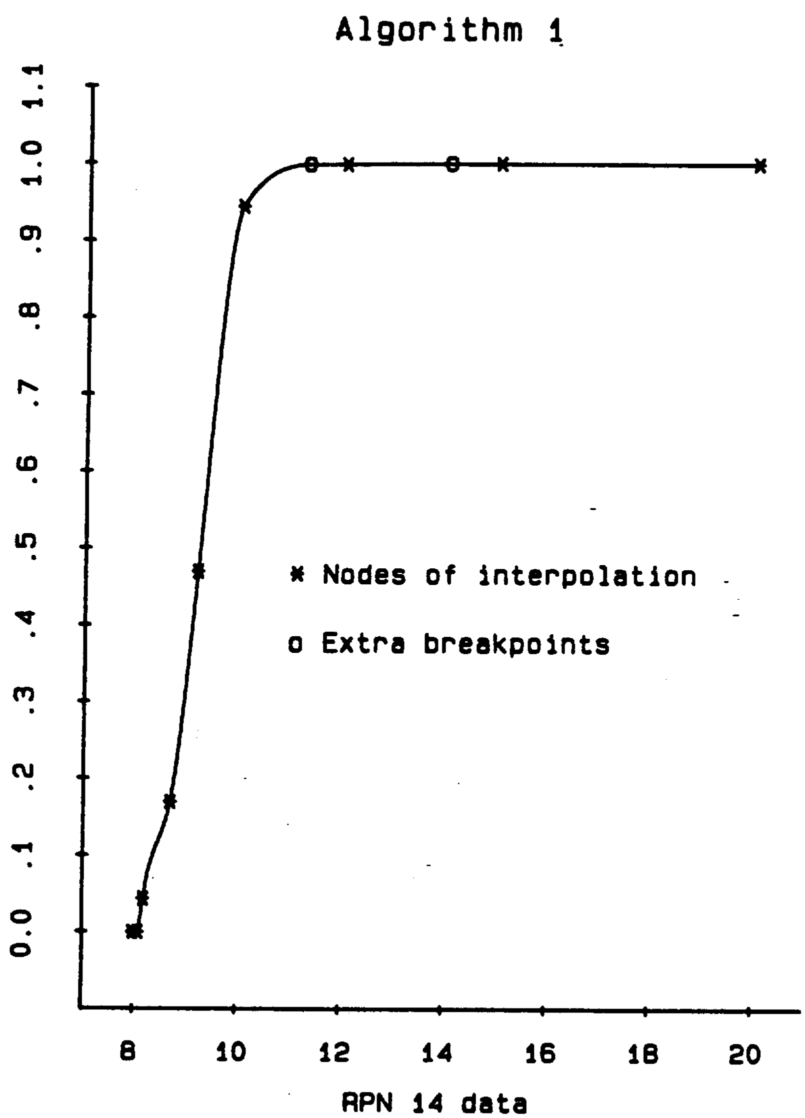

FIG. 13

Step 2a. For $i := 0$ to $n-1$ by twos, if $(s'(t_i), s'(t_{i+1})) \notin \mathcal{M}_i$, perform a relaxed projection of $(s'(t_i), s'(t_{i+1}))$ onto $\mathcal{C}_i$.

Step 2b. For $i := 1$ to $n-1$ by twos, if $(s'(t_i), s'(t_{i+1})) \notin \mathcal{M}_i$, perform a relaxed projection of $(s'(t_i), s'(t_{i+1}))$ onto $\mathcal{C}_i$.

Step 2c. For $i := 0$ to $n-1$, if $(s'(t_i), s'(t_{i+1})) \notin \mathcal{M}_i$, then add a knot and extra data according to the scheme of the algorithm fragment so that the new spline is monotone on (t_i, t_{i+1}).

8. Numerical results. In this section we compare various piecewise cubic interpolation algorithms on some example data sets. The algorithms used are the following:

Cubic spline. Cubic spline interpolation with end conditions (7.2).

Fritsch-Carlson. The Fritsch-Carlson algorithm with derivatives initialized using

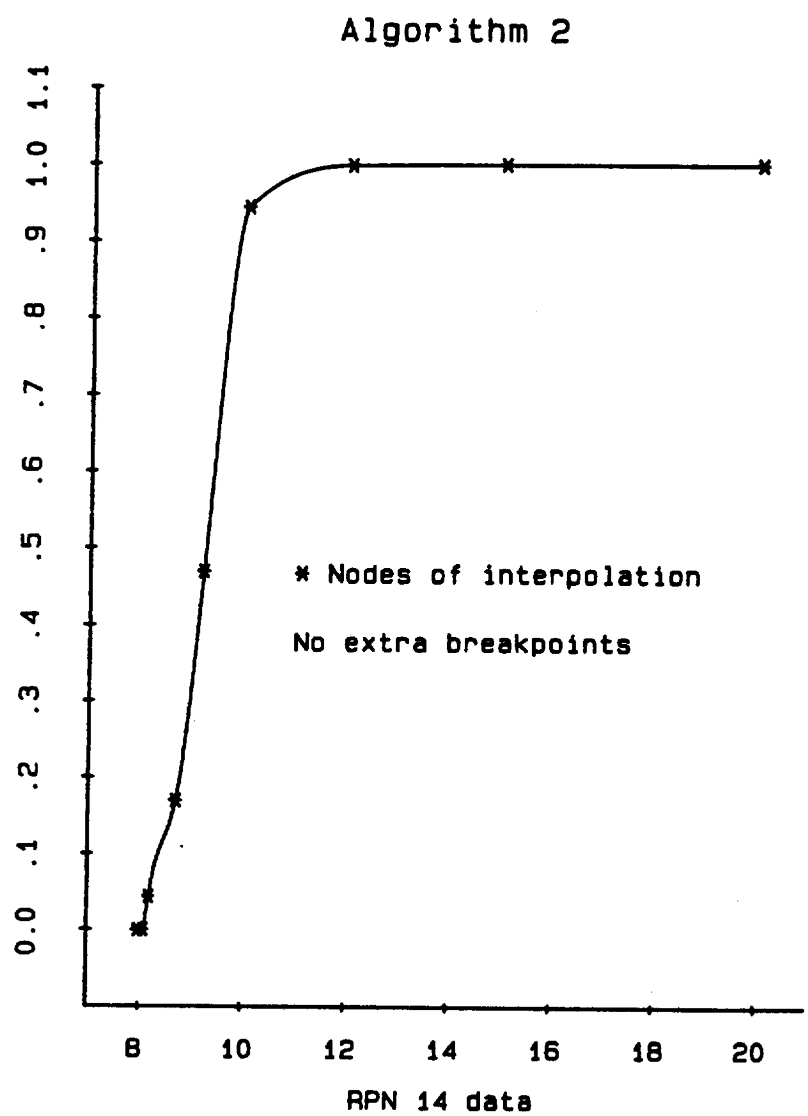

FIG. 14

three-point formulas and $\mathcal{S}$ a quarter of the disk center, the origin, and radius three, as recommended in [11].

Yan. Yan's algorithm as given in [14].

Algorithm 1. Algorithm 1 of this paper with $\gamma = 1$ and derivatives initialized using a cubic spline with end conditions (7.2).

Algorithm 2. Algorithm 2 of this paper with $\gamma = 1$, derivatives initialized using a cubic spline with end conditions (7.2), and

$$g(x) = \begin{cases} x/2, & x < \frac{2}{3}, \\ 2x - 1, & x \geqq \frac{2}{3}. \end{cases}$$

The first set of results show the error behaviour of the algorithms on data coming from the sigmoidal function

$$f(x) = \begin{cases} 0, & x \leqq 0.25, \\ \exp(-1/(4x-1)^2), & x > 0.25. \end{cases}$$

The interpolation mesh consists of $n+1$ uniformly spaced points on $[0, 1]$ including zero and 1. The error was approximated by evaluating $f - s$ on a 64-times finer mesh and calculating the maximum absolute error over this mesh. The numerical results are consistent with the Yan and Algorithm 2 methods being $\mathcal{O}(\delta^4)$ to smooth functions, while the Fritsch–Carlson method is only $\mathcal{O}(\delta^3)$. (See Table 1.)

TABLE 1

n	Fritsch–Carlson	Yan	Algorithm 2
4	$1.21940E-1$	$1.14144E-1$	$1.14295E-1$
8	$1.14952E-2$	$1.70767E-2$	$1.76598E-2$
16	$3.73562E-3$	$5.19677E-3$	$2.40882E-3$
32	$7.86227E-4$	$5.36681E-4$	$2.08481E-4$
64	$9.88770E-5$	$3.79518E-5$	$1.59501E-5$
128	$1.07709E-5$	$2.28413E-6$	$6.50118E-7$
256	$1.30483E-6$	$1.33182E-7$	$3.75526E-8$

Next we consider the action of the algorithms on some example data sets from the literature. The first is the data set

x	0	2	3	5	6	8	9	11	12	14	15
y	10	10	10	10	10	10	10.5	15	50	60	85

of Akima [1]. The second is the RPN 14 data of Fritsch and Carlson [11]:

x	7.99	8.09	8.19	8.7	9.2
y	0	$2.76429E-5$	$4.37498E-2$	0.169183	0.469428

x	10	12	15	20
y	0.943740	0.998636	0.999919	0.999994

The corresponding interpolants are shown in Figs. 5–9 and Figs. 10–14 respectively.

The unconstrained cubic spline is clearly nonmonotone on these data sets and is therefore unacceptable in applications where monotonicity is a requirement. The "rounder" look of the Fritsch–Carlson and Algorithm 2 fits, compared with the two other constrained methods, is almost certainly due to the avoidance of points on the boundary of $\mathcal{M}$.

9. Discussion. Of the algorithms developed in this paper Algorithm 2 seems to offer the best combination of computational cost, optimal order error properties, and smoothness of fitted monotone curve. If desired, it can be made into a local method by replacing the initial spline derivatives with four-point derivative approximations.

Pascal code for Algorithms 1 and 2 is available from the first author.

Several problems remain for the future. These include handling data of changing monotonicity, combining monotonicity and convexity constraints, and fitting subject to an explicit quantitative mix of smoothness and approximation properties. Work on these questions is in progress.

REFERENCES

[1] H. AKIMA, *A new method of interpolation and smooth curve fitting based on local procedures*, J. Assoc. Comput. Mach., 17 (1970), pp. 589–602.

[2] R. K. BEATSON, *Restricted range approximation by splines and variational inequalities*, SIAM J. Numer. Anal., 19 (1982), pp. 372–380.

[3] ———, *Constrained approximation and interpolation*, SIAM Conference on Geometric Modeling and Robotics, Albany, NY, July 1985.

[4] ———, *On the convergence of some cubic spline interpolation schemes*, SIAM J. Numer. Anal., 23 (1986), pp. 903–912.

[5] R. K. BEATSON AND E. CHACKO, *Which cubic spline should one use?* Tech. report, Mathematics Department, University of Canterbury, Christchurch, New Zealand, 1988.

[6] P. L. CHEBYSHEV, *Du régulateur centrifuge*, pp. 109–126 in Vol. 2 of his collected works. Reprinted by Chelsea in 1961.

[7] ———, *Sur les fonctions qui diffèrent le moins possible de zèro*, J. de Math. Pures et Appl., 19 (1874), pp. 319–346.

[8] C. DE BOOR, *A Practical Guide to Splines*, Springer-Verlag, Berlin, New York, 1978.

[9] S. C. EISENSTAT, K. R. JACKSON, AND J. W. LEWIS, *The order of monotone piecewise cubic interpolation*, Tech. Report 207, Computer Science Department, Yale University, 1981.

[10] ———, *The order of monotone piecewise cubic interpolation*, SIAM J. Numer. Anal., 22 (1985), pp. 1220–1237.

[11] F. N. FRITSCH AND R. E. CARLSON, *Monotone piecewise cubic interpolation*, SIAM J. Numer. Anal., 17 (1980), pp. 238–246.

[12] J. M. HYMAN, *Accurate monotonicity preserving interpolation*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 645–654.

[13] D. F. MCALLISTER AND J. A. ROULIER, *Interpolation by convex quadratic splines*, Math. Comp., 32 (1978), pp. 1154–1162.

[14] Z. YAN, *Piecewise cubic curve fitting algorithm*, Mathematics Department, University of South Carolina, Columbia, SC, 1986.