SIAM J. NUMER. ANAL.
Vol. 22, No. 2, April 1985

010

MONOTONE PIECEWISE BICUBIC INTERPOLATION*

R. E. CARLSON† AND F. N. FRITSCH†

Abstract. In a 1980 paper [SIAM J. Numer. Anal., 17 (1980), pp. 238–246] the authors developed a univariate piecewise cubic interpolation algorithm which produces a monotone interpolant to monotone data. This paper is an extension of those results to monotone $\mathscr{C}^1$ piecewise bicubic interpolation to data on a rectangular mesh. Such an interpolant is determined by the first partial derivatives and first mixed partial (twist) at the mesh points. Necessary and sufficient conditions on these derivatives are derived such that the resulting bicubic polynomial is monotone on a single rectangular element. These conditions are then simplified to a set of sufficient conditions for monotonicity. The latter are translated to a system of linear inequalities, which form the basis for a monotone piecewise bicubic interpolation algorithm.

1. Introduction. In [4] the authors developed a univariate piecewise cubic interpolation algorithm that produces a monotone interpolant to monotone data. This paper is a bivariate extension of those results. That is, given a set of monotone data values $\{f_{i,j}: i=1,2,\cdots,nx; j=1,2,\cdots,ny\}$ arranged over a rectangular grid, an algorithm is presented that produces a $\mathscr{C}^1$ piecewise bicubic function $p(x, y)$ which interpolates to the given data and which is monotone.

The development of such an algorithm was motivated by the need for accurately representing surfaces generated by equation of state (EOS) data. These surfaces must be smooth ($\mathscr{C}^1$) and monotone for the intended applications. Because the data typically vary over several orders of magnitude and change behavior radically in very small subdomains, existing higher order interpolation algorithms, such as bicubic spline or bicubic Bessel (three-point difference), tend to produce "ringing" or oscillations. Attempts to use the monotone cubic interpolation algorithm of [4] to compute first partial derivatives at the nodes significantly improved the results, but nonmonotonic subdomains occurred regardless of how the mixed partial derivatives were computed. The algorithm outlined here guarantees monotonicity and solves these problems.

In § 2 various univariate results are reviewed, and sufficient conditions for a cubic polynomial to be of constant sign on an interval are derived. In § 3 necessary and sufficient conditions are derived for a bicubic polynomial defined on a single rectangular element to be monotone. These are then modified to more tractable sufficient conditions. The latter are translated to pointwise form in § 4, where they provide the basis for the development of a numerical algorithm for the monotone piecewise bicubic interpolation problem. Such an algorithm is outlined here, but details of its implementation are deferred to a subsequent paper. An example using a subset of an EOS data table is included in the final section.

The methods used here are "tensor product" methods which involve only values and derivatives at the original data points. Thus, they are somewhat different from the piecewise quadratic algorithm with triangular elements recently proposed by Beatson and Ziegler [1] (this issue, pp. 401–411).

* Received by the editors June 30, 1981, and in revised form September 29, 1983. This work was supported by the U.S. Department of Energy under contract W-7405-Eng-48 and in part by its Office of Basic Energy Sciences, Applied Mathematical Sciences Program.

† Mathematics and Statistics Division, Lawrence Livermore National Laboratory, Livermore, California 94550.

2. Univariate results.

2.1. Necessary and sufficient conditions for monotonicity. Let u be a cubic polynomial defined on the interval $[a, b]$. Then u may be represented as follows:

$$u(x) = u(a)H_1(x) + u(b)H_2(x) + u'(a)H_3(x) + u'(b)H_4(x), \tag{1}$$

where the H_i are the usual cubic Hermite basis functions for $[a, b]$ (see Appendix A). Let

$$\Delta = \frac{u(b) - u(a)}{b - a}$$

be the slope of the line segment joining $(a, u(a))$ and $(b, u(b))$. When Δ is not zero, let

$$\alpha = \frac{u'(a)}{\Delta} \quad \text{and} \quad \beta = \frac{u'(b)}{\Delta}.$$

In [4] the authors derived the following set of necessary and sufficient conditions such that u is monotonic on $[a, b]$.

LEMMA 1. *Let u be a cubic polynomial on $[a, b]$ given by* (1). *Then u is monotone if and only if*

(a) $u'(a) = u'(b) = 0$ *if* $\Delta = 0$, *or*

(b) $(\alpha, \beta) \in \mathcal{M}$ *if* $\Delta \neq 0$,

where $\mathcal{M}$ is the region shown in Fig. 1.

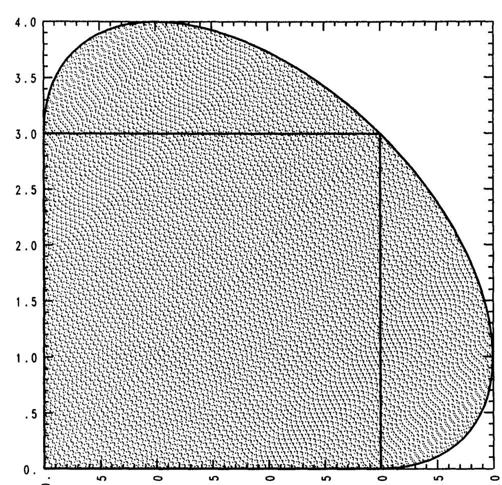


FIG. 1. *The monotonicity region $\mathcal{M}$ consists of the entire shaded area. The subregion $\mathcal{S} = [0, 3] \times [0, 3]$ is utilized in the corollary to Lemma* 1.

Remark. In essence this lemma says that nonmonotonicity occurs if the magnitude of the derivative at either node is too large in comparison with the slope of the line joining the data points. Reducing the magnitude of one or both of these derivatives will yield a monotone interpolant.

COROLLARY. *The following are sufficient conditions for the cubic polynomial u to be monotone on* $[a, b]$:

$$0 \leqq s \cdot u'(a) \leqq s \cdot 3\Delta \quad \textit{and}$$

$$0 \leqq s \cdot u'(b) \leqq s \cdot 3\Delta,$$

where

$$s=\begin{cases}+1 & \text{if } u(a)\leqq u(b),\\ -1 & \text{if } u(a)>u(b).\end{cases}$$

Proof. If $\Delta=0$, this requires $u'(a)=u'(b)=0$. Otherwise, these conditions are equivalent to restricting (α,β) to the square $\mathcal{S}=[0,3]\times[0,3]$ contained in $\mathcal{M}$.

2.2. Constant sign in $[a, b]$. A second univariate result involves conditions under which u does not change sign on $[a, b]$ (assuming $u(a)$ and $u(b)$ agree in sign). Although necessary and sufficient conditions are quite cumbersome, the following lemma contains a readily computable set of sufficient conditions.

LEMMA 2. *Let u be a cubic polynomial on $[a, b]$ given by* (1) *where $u(a)$ and $u(b)$ are of the same sign.*[1]

Then u does not change sign on $[a, b]$ if

a. *$u'(a)$ and $u'(b)$ are of opposite sign when $u(a)=u(b)=0$,*

b. $$\frac{-\hat{s}\cdot 3u(a)}{b-a}\leqq \hat{s}\cdot u'(a) \quad \text{and} \quad \hat{s}\cdot u'(b)\leqq\frac{\hat{s}\cdot 3u(b)}{b-a}$$

where

$$\hat{s}=\begin{cases}+1 & \text{if } u(a)\geqq 0 \text{ and } u(b)\geqq 0 \text{ (not both zero)},\\ -1 & \text{if } u(a)\leqq 0 \text{ and } u(b)\leqq 0 \text{ (not both zero)}.\end{cases}$$

Further, $u(x)$ is nowhere zero in (a, b) unless $u\equiv 0$ on $[a, b]$.

Proof. For $u(a)=u(b)=0$, $u(x)=u'(a)\cdot H_3(x)+u'(b)\cdot H_4(x)$. Because $H_3(x)>0$ and $H_4(x)<0$ for $x\in(a, b)$, $u(x)$ does not change sign when $u'(a)$ and $u'(b)$ are of opposite sign, and $u(x)$ is nowhere zero in (a, b) unless $u\equiv 0$.

For $u(a)$ and $u(b)$ not both zero, consider the case in which $u(a)$ and $u(b)$ are nonnegative ($\hat{s}=+1$). Construct two cubic polynomials c_1 and c_2 as follows:

$$c_1(x)=u(a)H_1(x)+\frac{-3u(a)}{b-a}H_3(x), \tag{2}$$

$$c_2(x)=u(b)H_2(x)+\frac{3u(b)}{b-a}H_4(x). \tag{3}$$

Note that $u(a)=0$ implies $c_1\equiv 0$ and $u(b)=0$ implies $c_2\equiv 0$. For $u(a)$ and $u(b)$ positive, define

$$\Delta_1=\frac{c_1(b)-c_1(a)}{b-a}=\frac{-u(a)}{b-a}$$

and

$$\Delta_2=\frac{c_2(b)-c_2(a)}{b-a}=\frac{u(b)}{b-a}.$$

From Lemma 1, c_1 is monotone decreasing ($\alpha=3, \beta=0$) and c_2 is monotone increasing ($\alpha=0, \beta=3$). Thus, c_1 and c_2 are either positive on (a, b) or identically zero.

Using (1), (2) and (3) we obtain

$$u(x)-[c_1(x)+c_2(x)]=\left[u'(a)+\frac{3u(a)}{b-a}\right]H_3(x)+\left[u'(b)-\frac{3u(b)}{b-a}\right]H_4(x). \tag{4}$$

[1] $u(a)=0$ implies no restriction on $u(b)$ and conversely.

Since $u'(a) \geqq -3u(a)/(b-a)$ and $u'(b) \leqq 3u(b)/(b-a)$, the coefficient of H_3 is nonnegative and the coefficient of H_4 is nonpositive. Therefore, if $u(a) > 0$ and/or $u(b) > 0$, $u(x) \geqq c_1(x) + c_2(x) > 0$ for $x \in (a, b)$. The proof for $u(a) \leqq 0$ and $u(b) \leqq 0$ ($\hat{s} = -1$) follows similarly.

2.3. Piecewise cubic interpolation. The univariate interpolation problem may be described as follows: Given a partition π: $a = x_1 < x_2 < \cdots < x_n = b$ of $[a, b]$ and a set of values $\{f_i\}_{i=1}^n$, construct a function p with domain $[a, b]$ such that

$$p(x_i) = f_i, \qquad i = 1, 2, \cdots, n. \tag{5}$$

For the special case in which p is a piecewise cubic function in $\mathscr{C}^1[a, b]$ with breakpoints $\{x_i\}_{i=1}^n$, p is uniquely determined by $\{f_i\}_{i=1}^n$ and $\{d_i\}_{i=1}^n$ where

$$d_i = p'(x_i) \qquad i = 1, 2, \cdots, n. \tag{6}$$

Since we assume the $\{f_i\}$ are given, a cubic Hermite interpolation algorithm refers to a procedure for computing the $\{d_i\}$. In [4], given a set of monotone data ($f_i \leqq f_{i+1}$, $i = 1, 2, \cdots, n-1$, or $f_i \geqq f_{i+1}$, $i = 1, 2, \cdots, n-1$), an algorithm is presented which guarantees that p is monotone. This algorithm is a two-step process for computing the $\{d_i\}$. In subsequent work, Fritsch and Butland [3] have developed a one-step algorithm for computing the $\{d_i\}$ which automatically satisfy the conditions of the corollary to Lemma 1.

3. Monotone bicubic functions. For univariate functions monotonicity is defined as follows:

DEFINITION 1. A function f with domain $[a, b]$ is monotone if, for any $x_1 \leqq x_2$, $s \cdot f(x_1) \leqq s \cdot f(x_2)$ where

$$s = \begin{cases} +1 & \text{if } f \text{ is monotone increasing,} \\ -1 & \text{if } f \text{ is monotone decreasing.} \end{cases}$$

Monotonicity for bivariate functions is somewhat less clear. Based on the physical application which prompted the need for this interpolation algorithm, we define a bivariate function f to be monotone on $D = [a, b] \times [c, d]$ if all univariate "slices" of f taken parallel to the coordinate axes are monotone. Specifically,

DEFINITION 2. A function f with domain D is monotone on D if

a. For each fixed $x = x^*$ and $y_1 \leqq y_2$,

$$s_1 \cdot f(x^*, y_1) \leqq s_1 \cdot f(x^*, y_2)$$

where

$$s_1 = \begin{cases} +1 & \text{for } f \text{ monotone increasing in } y, \\ -1 & \text{for } f \text{ monotone decreasing in } y. \end{cases}$$

b. For each fixed $y = y^*$ and $x_1 \leqq x_2$,

$$s_2 \cdot f(x_1, y^*) \leqq s_2 \cdot f(x_2, y^*)$$

where

$$s_2 = \begin{cases} +1 & \text{for } f \text{ monotone increasing in } x, \\ -1 & \text{for } f \text{ monotone decreasing in } x. \end{cases}$$

It is understood that s_1 and s_2 are constants. That is, if, at one value of x^*, $s_1 = +1$, then $s_1 = +1$ at all values of $x^* \in [a, b]$. However, it is permissible for $s_1 \neq s_2$.

The following lemma is an immediate consequence of Definition 2.

LEMMA 3. *If f is monotone on D, then*

$$\operatorname{sign}\left[\frac{f(a,d)-f(a,c)}{d-c}\right]=\operatorname{sign}\left[\frac{f(b,d)-f(b,c)}{d-c}\right]=s_1 \tag{7}$$

and

$$\operatorname{sign}\left[\frac{f(b,c)-f(a,c)}{b-a}\right]=\operatorname{sign}\left[\frac{f(b,d)-f(a,d)}{b-a}\right]=s_2. \tag{8}$$

Viewed from an interpolation standpoint, this lemma states that in order to construct a monotone interpolant, the data must be consistent in the sense that (7) and (8) are satisfied.

In the remainder of this section we restrict our attention to a bicubic polynomial u with domain d; u may be written as follows:

$$\begin{aligned} u(x,y)=\ & u(a,c)H_1(x)G_1(y)+u(b,c)H_2(x)G_1(y)\\ &+u(a,d)H_1(x)G_2(y)+u(b,d)H_2(x)G_2(y)\\ &+u_x(a,c)H_3(x)G_1(y)+u_x(b,c)H_4(x)G_1(y)\\ &+u_x(a,d)H_3(x)G_2(y)+u_x(b,d)H_4(x)G_2(y)\\ &+u_y(a,c)H_1(x)G_3(y)+u_y(b,c)H_2(x)G_3(y)\\ &+u_y(a,d)H_1(x)G_4(y)+u_y(b,d)H_2(x)G_4(y)\\ &+u_{xy}(a,c)H_3(x)G_3(y)+u_{xy}(b,c)H_4(x)G_3(y)\\ &+u_{xy}(a,d)H_3(x)G_4(y)+u_{xy}(b,d)H_4(x)G_4(y) \end{aligned} \tag{9}$$

where the H_i are as in § 2 and the G_j are the cubic Hermite basis functions on $[c,d]$.

As a consequence of (9) we observe that $u(x,y^*)$ and $u(x^*,y)$ are univariate cubic polynomials in x and y respectively. Therefore, from Definition 2, u is monotone if and only if, $u(x,y^*)$ and $u(x^*,y)$ are monotone.

Analogous to the univariate case, define for $u(x,y)$

$$\begin{aligned} \Delta_1(x)&=\frac{u(x,d)-u(x,c)}{d-c}\\ &=\Delta_1(a)\cdot H_1(x)+\Delta_1(b)\cdot H_2(x)+\Delta_1'(a)\cdot H_3(x)+\Delta_1'(b)\cdot H_4(x), \end{aligned} \tag{10}$$

where

$$\Delta_1'(x)=\frac{u_x(x,d)-u_x(x,c)}{d-c}. \tag{11}$$

If $\Delta_1(x)\neq 0$, let

$$\alpha_1(x)=\frac{u_y(x,c)}{\Delta_1(x)} \quad\text{and}\quad \beta_1(x)=\frac{u_y(x,d)}{\Delta_1(x)}. \tag{12}$$

Similarly, define

$$\begin{aligned} \Delta_2(y)&=\frac{u(b,y)-u(a,y)}{b-a}\\ &=\Delta_2(c)\cdot G_1(y)+\Delta_2(d)\cdot G_2(y)+\Delta_2'(c)\cdot G_3(y)+\Delta_2'(d)\cdot G_4(y), \end{aligned} \tag{13}$$

where

$$\Delta_2'(y) = \frac{u_y(b, y) - u_y(a, y)}{b - a}. \tag{14}$$

If $\Delta_2(y) \neq 0$, let

$$\alpha_2(y) = \frac{u_x(a, y)}{\Delta_2(y)} \quad \text{and} \quad \beta_2(y) = \frac{u_x(b, y)}{\Delta_2(y)}. \tag{15}$$

The following theorem is a direct consequence of (9)-(15) and Lemma 1.

THEOREM 1. *Let u be a bicubic polynomial as given by* (9) *such that* $\{u(a, c), u(b, c), u(a, d), u(b, d)\}$ *satisfy* (7) *and* (8). *Then, on* $D = [a, b] \times [c, d]$:

(i) *u is monotone in y if and only if*

(a) *$\Delta_1(x)$ does not change sign for $x \in [a, b]$*;

(b) *$u_y(x, c) = u_y(x, d) = 0$ whenever $\Delta_1(x) = 0$*;

(c) *$(\alpha_1(x), \beta_1(x)) \in \mathcal{M}$ whenever $\Delta_1(x) \neq 0$.*

(ii) *u is monotone in x if and only if*

(d) *$\Delta_2(y)$ does not change sign for $y \in [c, d]$*;

(e) *$u_x(a, y) = u_x(b, y) = 0$ whenever $\Delta_2(y) = 0$*;

(f) *$(\alpha_2(y), \beta_2(y)) \in \mathcal{M}$ whenever $\Delta_2(y) \neq 0$.*

As one might suspect, the necessary and sufficient conditions of Theorem 1 are not easily translated into a viable algorithm. However, by replacing conditions (a)-(f) by appropriate sufficient conditions, a set of linear inequalities among the partial derivatives at the nodes can be derived. These inequalities, which form the basis for a computational algorithm, are derived by applying the Corollary to Lemma 1 to Theorem 1 to obtain the following.

COROLLARY. *Let u be as in Theorem* 1. *Sufficient conditions for u to be monotone in y are*:

(a′) sign $(\Delta_1(x)) = s_1$, *all* $x \in [a, b]$;

(b′) $0 \leqq s_1 \cdot u_y(x, \eta) \leqq s_1 \cdot 3\Delta_1(x)$ *for $\eta = c$ and d, for all $x \in [a, b]$.*

Sufficient conditions for u to be monotone in x are:

(c′) sign $(\Delta_2(y)) = s_2$, *all* $y \in [c, d]$;

(d′) $0 \leqq s_2 \cdot u_x(\xi, y) \leqq s_2 \cdot 3\Delta_2(y)$ *for $\xi = a$ and b, for all $y \in [c, d]$.*

Note that we have replaced (b) and (c) by the simpler (b′), (e) and (f) by (d′).

As a first step, we require that the cubic be monotone along the boundaries of D. This is accomplished by requiring the first partial derivatives at the corners to satisfy

$$\left.\begin{aligned} &0 \leqq s_1 \cdot u_y(\xi, \eta) \leqq s_1 \cdot 3\Delta_1(\xi) \quad (16)\\ &0 \leqq s_2 \cdot u_x(\xi, \eta) \leqq s_2 \cdot 3\Delta_2(\eta) \quad (17) \end{aligned}\right\} \quad \xi = a \text{ and } b,\ \eta = c \text{ and } d.$$

Conditions (7) and (8) guarantee sign $(\Delta_1(a)) =$ sign $(\Delta_1(b)) = s_1$ and sign $(\Delta_2(c)) =$ sign $(\Delta_2(d)) = s_2$. Applying Lemma 2 to (10) we obtain the following sufficient conditions for (a′):

$$-s_1 \cdot \frac{3\Delta_1(a)}{b - a} \leqq s_1 \cdot \Delta_1'(a) \quad \text{and} \quad s_1 \cdot \Delta_1'(b) \leqq s_1 \cdot \frac{3\Delta_1(b)}{b - a}. \tag{18}$$

Similarly, sufficient conditions for (c′) are:

$$-s_2 \cdot \frac{3\Delta_2(c)}{d - c} \leqq s_2 \cdot \Delta_2'(c) \quad \text{and} \quad s_2 \cdot \Delta_2'(d) \leqq s_2 \cdot \frac{3\Delta_2(d)}{d - c}. \tag{19}$$

Because $u_y(x, \eta)$ is a cubic polynomial in x and (16) implies $\operatorname{sign}(u_y(a, \eta)) = \operatorname{sign}(u_y(b, \eta)) = s_1$, we can apply Lemma 2 to obtain the following sufficient conditions for the first inequality of (b′):

$$(20)\qquad -s_1 \cdot \frac{3u_y(a, \eta)}{b-a} \leqq s_1 \cdot u_{xy}(a, \eta) \quad \text{and} \quad s_1 \cdot u_{xy}(b, \eta) \leqq s_1 \cdot \frac{3u_y(b, \eta)}{b-a}, \quad \eta = c \text{ and } d.$$

The second inequality of (b′) can be rewritten as

$$(21)\qquad 0 \leqq s_1[3\Delta_1(x) - u_y(x, \eta)].$$

Note that $\phi(x) = 3\Delta_1(x) - u_y(x, \eta)$ is also a cubic polynomial, and the second inequality of (16) implies $\operatorname{sign}(\phi(a)) = \operatorname{sign}(\phi(b)) = s_1$ so we can apply Lemma 2 to obtain sufficient conditions for (21):

$$(22)\qquad \begin{aligned} &-s_1 \cdot 3\left[\frac{3\Delta_1(a) - u_y(a, \eta)}{b-a}\right] \leqq s_1[3\Delta_1'(a) - u_{xy}(a, \eta)] \quad \text{and} \\ &s_1[3\Delta_1'(b) - u_{xy}(b, \eta)] \leqq s_1 \cdot 3\left[\frac{3\Delta_1(b) - u_y(b, \eta)}{b-a}\right], \quad \eta = c \text{ and } d. \end{aligned}$$

Rearranging the first inequality of (22) and combining it with the first inequality of (20) we obtain the following bounds on $u_{xy}(a, \eta)$:

$$(23)\qquad \begin{aligned} -s_1 \cdot \frac{3u_y(a, \eta)}{b-a} &\leqq s_1 \cdot u_{xy}(a, \eta) \\ &\leqq s_1 \cdot 3\left[\Delta_1'(a) + \frac{3\Delta_1(a) - u_y(a, \eta)}{b-a}\right], \quad \eta = c \text{ and } d. \end{aligned}$$

Similarly,

$$(24)\qquad \begin{aligned} s_1 \cdot 3\left[\Delta_1'(b) - \frac{3\Delta_1(b) - u_y(b, \eta)}{b-a}\right] &\leqq s_1 \cdot u_{xy}(b, \eta) \\ &\leqq s_1 \cdot \frac{3u_y(b, \eta)}{b-a}, \quad \eta = c \text{ and } d. \end{aligned}$$

A similar treatment of (d′) leads to the conditions

$$(25)\qquad -s_2 \cdot \frac{3u_x(\xi, c)}{d-c} \leqq s_2 \cdot u_{xy}(\xi, c) \leqq s_2 \cdot 3\left[\Delta_2'(c) + \frac{3\Delta_2(c) - u_x(\xi, c)}{d-c}\right],$$

$$(26)\qquad s_2 \cdot 3\left[\Delta_2'(d) - \frac{3\Delta_2(d) - u_x(\xi, d)}{d-c}\right] \leqq s_2 \cdot u_{xy}(\xi, d) \leqq s_2 \cdot \frac{3u_x(\xi, d)}{d-c},$$

which are to be satisfied for $\xi = a$ and b.

We have established:

THEOREM 2. *If u is a bicubic polynomial on $D = [a, b] \times [c, d]$ which satisfies* (7), (8), (16)–(19), (23)–(26), *then u is monotone on D.*

Remark. If only monotonicity in y is required, then only conditions (7), (16), (18), (23) and (24) need be imposed. Similarly, if only monotonicity in x is required, the relevant conditions are (8), (17), (19), (25) and (26).

4. A monotone bicubic interpolation algorithm. Let π_1: $a = x_1 < x_2 < \cdots < x_{nx} = b$ be a partition of $[a, b]$ and let π_2: $c = y_1 < y_2 < \cdots < y_{ny} = d$ be a partition of $[c, d]$. Let $\{f_{ij}\}$ be a set of $nx \cdot ny$ data values. The bivariate interpolation problem is to construct a function p on $D=[a, b]\times[c, d]$ such that

$$p(x_i, y_j)=f_{ij}, \qquad i=1,2,\cdots,nx, \quad j=1,2,\cdots,ny \tag{27}$$

and $p\in \mathscr{C}^1[D]$.

For our purposes p will be a piecewise bicubic function. Thus, an interpolation algorithm is a procedure for computing the set of values $\{p_x(x_i, y_j), p_y(x_i, y_j), p_{xy}(x_i, y_j): i=1,2,\cdots,nx; j=1,2,\cdots,ny\}$. An algorithm is desired which chooses these values such that p is monotone if the data are monotone.

In order to develop such an algorithm, it will be convenient to recast some of the inequalities developed in the previous section in a pointwise, rather than elementwise, form. We will need some slightly modified notation:

$$\Delta_{1,j}(x)=\frac{p(x, y_{j+1})-p(x, y_j)}{y_{j+1}-y_j}, \qquad \Delta_{2,i}(y)=\frac{p(x_{i+1}, y)-p(x_i, y)}{x_{i+1}-x_i}. \tag{28}$$

Here, and in what follows, quantities with indices outside the allowable ranges $1\leqq i\leqq nx$, $1\leqq j\leqq ny$ are to be omitted. For example, $\Delta_{1,j}$ is defined only for $1\leqq j\leqq ny-1$.

In terms of the functions defined by (28), conditions (16) and (17) become, respectively

$$0\leqq s_1\cdot p_y(x_i, y_j)\leqq 3\cdot \min\{s_1\cdot\Delta_{1,j-1}(x_i), s_1\cdot\Delta_{1,j}(x_i)\}, \tag{29}$$

$$0\leqq s_2\cdot p_x(x_i, y_j)\leqq 3\cdot \min\{s_2\cdot\Delta_{2,i-1}(y_j), s_2\cdot\Delta_{2,i}(y_j)\}. \tag{30}$$

[As noted above, (29) involves only a single upper bound if $j=1$ or $j=ny$, and similarly for (30).]

These are bounds on the magnitudes of the first partial derivatives in terms of changes in the data in the corresponding direction.

From (18) we obtain

$$-s_1\cdot\frac{3\Delta_{1,j}(x_i)}{x_{i+1}-x_i}\leqq s_1\cdot\Delta'_{1,j}(x_i)\leqq s_1\cdot\frac{3\Delta_{1,j}(x_i)}{x_i-x_{i-1}}. \tag{31}$$

These are bounds on the differences between the p_x-values in the y direction based on differences in the data in that direction. Similarly, (19) yields

$$-s_2\cdot\frac{3\Delta_{2,i}(y_j)}{y_{j+1}-y_j}\leqq s_2\cdot\Delta'_{2,i}(y_j)\leqq s_2\cdot\frac{3\Delta_{2,i}(y_j)}{y_j-y_{j-1}}. \tag{32}$$

It will be convenient to introduce the notation

$$A_{ij}=s_1\left[\Delta'_{1,j}(x_i)-\frac{3\Delta_{1,j}(x_i)}{x_i-x_{i-1}}\right], \qquad B_{ij}=s_1\left[\Delta'_{1,j}(x_i)+\frac{3\Delta_{1,j}(x_i)}{x_{i+1}-x_i}\right], \tag{33a}$$

$$C_{ij}=s_2\left[\Delta'_{2,i}(y_j)-\frac{3\Delta_{2,i}(y_j)}{y_j-y_{j-1}}\right], \qquad D_{ij}=s_2\left[\Delta'_{2,i}(y_j)+\frac{3\Delta_{2,i}(y_j)}{y_{j+1}-y_j}\right]. \tag{33b}$$

In terms of this notation, (31) and (32) are equivalent, respectively, to

$$A_{ij}\leqq 0\leqq B_{ij}, \tag{31'}$$

$$C_{ij}\leqq 0\leqq D_{ij}. \tag{32'}$$

Conditions (23) and (24) lead to bounds on the p_{xy}-values. Let $R_{ij}=[x_i, x_{i+1}]\times[y_j, y_{j+1}]$ denote the (i,j) mesh element. Using the abbreviations

$$p_y^+(i,j)=s_1\cdot\frac{p_y(x_i, y_j)}{x_{i+1}-x_i}, \qquad p_y^-(i,j)=s_1\cdot\frac{p_y(x_i, y_j)}{x_i-x_{i-1}}, \tag{34}$$

the application of (23) to R_{ij} with $a=x_i$, $b=x_{i+1}$, $\eta=c=y_j$ yields

$$-3p_y^+(i,j)\leqq s_1\cdot p_{xy}(x_i,y_j)\leqq 3B_{ij}-3p_y^+(i,j). \tag{35a}$$

Similarly, (23) applied to $R_{i,j-1}$ with $a=x_i$, $b=x_{i+1}$, $\eta=d=y_j$ yields

$$-3p_y^+(i,j)\leqq s_1\cdot p_{xy}(x_i,y_j)\leqq 3B_{i,j-1}-3p_y^+(i,j). \tag{35b}$$

Applying (24) to $R_{i-1,j}$ and $R_{i-1,j-1}$, on the other hand, produces

$$3A_{ij}+3p_y^-(i,j)\leqq s_1\cdot p_{xy}(x_i,y_j)\leqq 3p_y^-(i,j), \tag{36a}$$

$$3A_{i,j-1}+3p_y^-(i,j)\leqq s_1\cdot p_{xy}(x_i,y_j)\leqq 3p_y^-(i,j). \tag{36b}$$

Conditions (35) and (36) can be summarized as

$$\begin{aligned}&3\cdot\max\{-p_y^+(i,j),\max\{A_{i,j-1},A_{ij}\}+p_y^-(i,j)\}\\&\qquad\leqq s_1\cdot p_{xy}(x_iy_j)\leqq 3\cdot\min\{p_y^-(i,j),\min\{B_{i,j-1},B_{ij}\}-p_y^+(i,j)\}.\end{aligned} \tag{37}$$

It follows from (31′) that each of the individual inequalities in (35) and (36) is consistent, but there is no a priori reason to believe that the entire set (37) should be consistent. In fact, the authors have constructed examples in which inconsistency occurs. Thus, we must impose the following *consistency conditions*:

$$\max\{A_{i,j-1},A_{ij}\}+p_y^-(i,j)\leqq\min\{B_{i,j-1},B_{ij}\}-p_y^+(i,j). \tag{38}$$

Referring to (34), this can be expressed as a bound on the magnitude of p_y:

$$s_1\cdot p_y(x_i,y_j)\leqq\frac{h_{i-1}h_i}{h_{i-1}+h_i}[\min\{B_{i,j-1},B_{ij}\}-\max\{A_{i,j-1},A_{ij}\}], \tag{39}$$

where $h_i=x_{i+1}-x_i$. Referring to (31′) and (33a), this bound is nonnegative and depends on the p_x-values (but not p_y).

A similar derivation from (25) and (26), with definitions

$$p_x^+(i,j)=s_2\cdot\frac{p_x(x_i,y_j)}{y_{j+1}+y_j},\qquad p_x^-(i,j)=s_2\cdot\frac{p_x(x_i,y_j)}{y_j-y_{j-1}}, \tag{40}$$

leads to the inequalities

$$\begin{aligned}&3\cdot\max\{-p_x^+(i,j),\max\{C_{i-1,j},C_{ij}\}+p_x^-(i,j)\}\\&\qquad\leqq s_2\cdot p_{xy}(x_i,y_j)\leqq 3\cdot\min\{p_x^-(i,j),\min\{D_{i-1,j},D_{ij}\}-p_x^+(i,j)\},\end{aligned} \tag{41}$$

with consistency conditions

$$\max\{C_{i-1,j},C_{i,j}\}+p_x^-(i,j)\leqq\min\{D_{i-1,j},D_{ij}\}-p_x^+(i,j) \tag{42}$$

or

$$s_2\cdot p_x(x_i,y_j)\leqq\frac{k_{j-1}k_j}{k_{j-1}+k_j}[\min\{D_{i-1,j},D_{ij}\}-\max\{C_{i-1,j}C_{ij}\}], \tag{43}$$

where $k_j=y_{j+1}-y_j$.

To summarize, sufficient conditions for monotonicity in y are (29), (31), (37), and (39); for monotonicity in x, (30), (32), (41), and (43).

There are many ways to set values of p_x, p_y, p_{xy} at the mesh points which satisfy the conditions derived above. Since all are linear inequalities (rather sparse ones!) in these $3\cdot nx\cdot ny$ parameters, and we know a feasible point (all zero), one might consider using linear (or quadratic) programming methods. The problem is that it is not at all clear what to use for an objective function. We would like to maximize some measure of "goodness" (physical reasonableness), but there is no agreement on such a measure. Further there is a rather large number of inequalities, so that such an algorithm is unlikely to be practical for problems of reasonable size (30×40 is not unusual).

Because of the problems mentioned above, we have chosen to use a more direct approach. A brief outline of an algorithm follows. Details will be given in a subsequent paper.

Step 1: Verify that the data are monotone and compute s_1 and s_2.

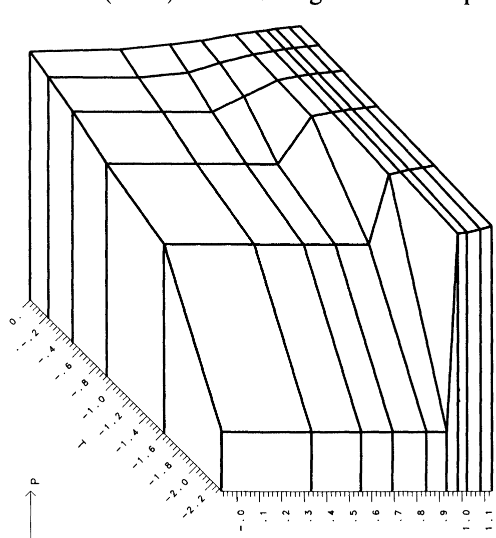
Step 2: Choose an initial set of values for $p_x(x_i, y_j)$ and $p_y(x_i, y_j)$ which satisfy (29) and (30). (For example, use subroutine PCHIM, from [2], along each of the mesh lines.) Subsequent modifications to p_x or p_y will only reduce their magnitudes without altering their signs, so these inequalities will remain valid.

Step 3: Make a pass through the mesh, reducing the magnitudes of p_x and/or p_y as necessary to satisfy (31) and (39). Repeat for (32) and (43). It may be necessary to iterate this procedure until no first partial derivatives need to be changed.

Step 4: Check that the inequalities (37) and (41) are consistent with one another. (That is, that the resulting bounds define a nonempty interval.) If the value $p_{xy}(x_i, y_j)=0$ fails to satisfy one of these inequalities, they may be inconsistent. In this event, an iterative procedure very similar to that of Step 3 can be employed to force zero into the interval. (Details to be given in subsequent paper.)

Step 5: Compute $p_{xy}(x_i, y_j)$ by applying the standard three point difference formula to $p_x(x_i, y)$ and $p_y(x, y_j)$. If the average of these two values is not in the acceptable range, the closer endpoint value is used. The resulting p satisfies all of the sufficient conditions, hence is monotone.

5. Example. In this section we demonstrate the problems encountered in representing an equation of state (EOS) surface. Fig. 2 shows a portion of EOS surface



Truncated section of aluminum EOS table

FIG. 2. EOS *surface for aluminum* (*pressure, P, as a function of density, ρ, and temperature, T, on a* log-log-log *scale*).

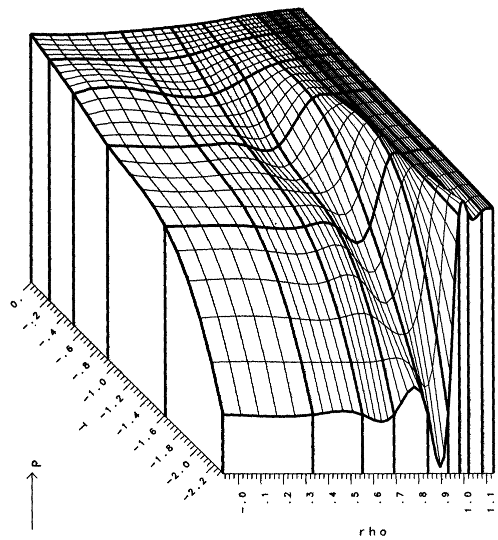


FIG. 3. *Bicubic spline interpolant, exhibiting highly nonmonotonic behavior.*

for aluminum with pressure as a function of density and temperature, on a log-log-log scale. The actual data values are given in Appendix B. This picture was formed by joining the data points with straight line segments. In each of the subsequent figures (Figs. 3-7), the interpolant has been evaluated on a uniform 5×5 refinement of the original rectangular partition and the resulting points joined with straight line segments for display purposes. The heavy lines are sections along the mesh lines.

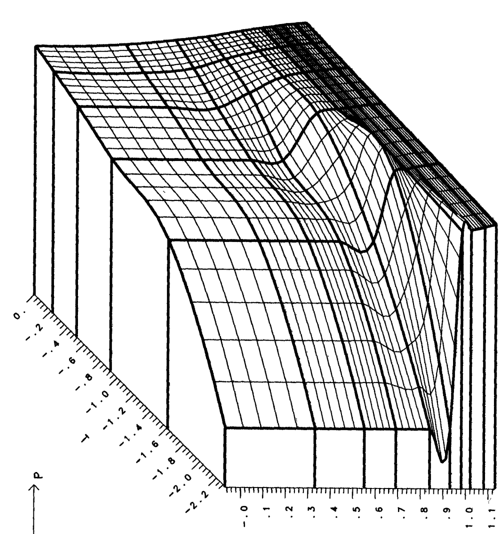


FIG. 4. *Bicubic Bessel interpolant, which also exhibits nonmonotonic behavior.*

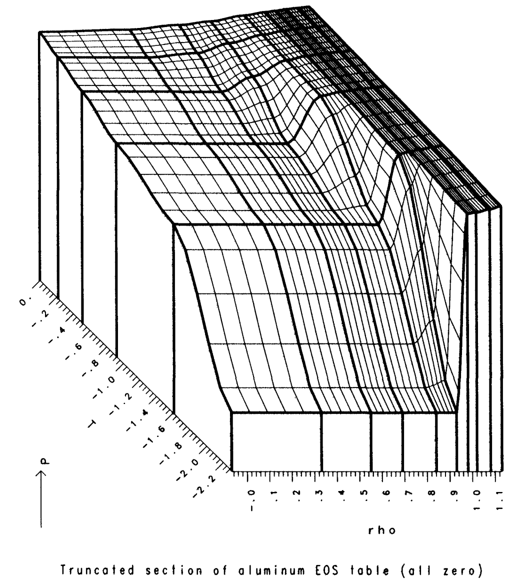


FIG. 5. *Bicubic Hermite surface with all nodal derivatives set to zero. This surface is monotone but not accurate near mesh lines.*

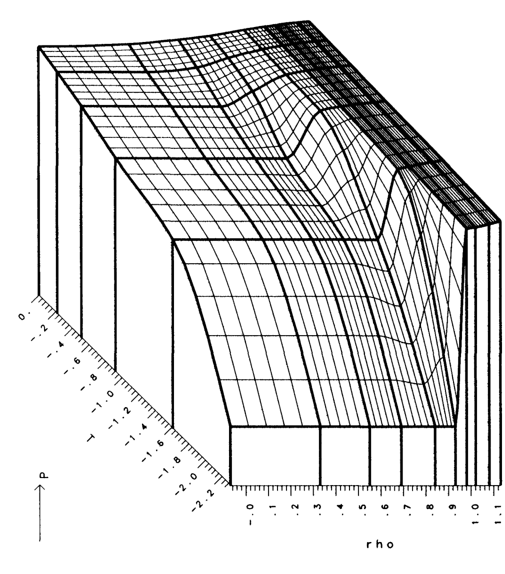


FIG. 6. *Bicubic Hermite surface using* PCHIM [2] *along the mesh lines and setting* $p_{xy}(x_i, y_j) = 0$. *Note the nonmonotonic behavior in the vicinity of* $\rho = 0.8$ *between the first two T-lines.*

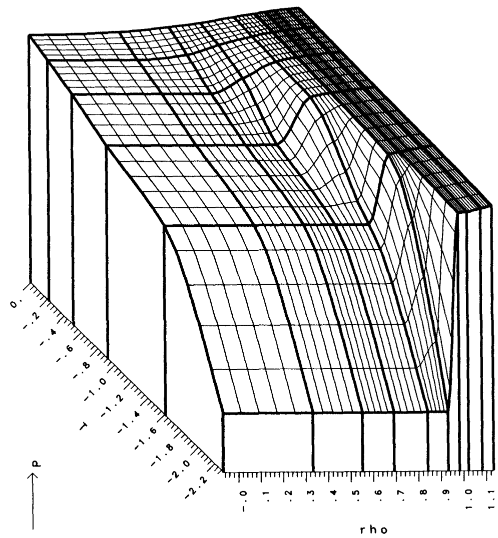


FIG. 7. *Bicubic Hermite surface obtained from the algorithm outlined in* § 4.

Appendix.

A. Cubic Hermite basis functions defined on the interval [0, *h*].

$$H_1(x)=\frac{1}{h^3}(2x^3-3hx^2+h^3),$$

$$H_2(x)=\frac{1}{h^3}(-2x^3+3hx^2),$$

$$H_3(x)=\frac{1}{h^2}(x^3-2hx^2+h^2x),$$

$$H_4(x)=\frac{1}{h^2}(x^3-hx^2).$$

B. Data set for example. The data listed in Table 1 were used in the example of § 5. The body of the table represents $P(T, \rho)$, pressure vs. temperature and normalized

TABLE 1
EOS *data of Fig.* 2.

	$P(T,\rho)$	Temperature (T) −2.30	−1.61	−0.92	−0.51	−0.22	0.00
	−0.07	−34.54	−13.82	−10.10	−7.26	−5.66	−4.53
	0.33	−34.54	−13.82	−10.10	−7.26	−5.66	−4.13
	0.55	−34.54	−13.82	−10.10	−7.26	−4.88	−3.35
	0.69	−34.54	−13.82	−10.10	−4.82	−3.34	−2.73
Density	0.84	−34.54	−13.82	−2.52	−2.22	−1.98	−1.78
(ρ)	0.93	−34.54	−2.68	−1.88	−1.56	−1.41	−1.28
	0.98	−3.06	−2.28	−1.63	−1.32	−1.15	−1.05
	1.02	−2.86	−1.92	−1.39	−1.10	−0.92	−0.81
	1.08	−2.37	−1.60	−1.17	−0.90	−0.72	−0.60
	1.13	−1.89	−1.30	−0.95	−0.71	−0.54	−0.41

TABLE 2
Parameters for surface of Fig. 7.

x	y	f	p_x	p_y	p_{xy}
-2.30	-0.07	-34.54	3.68551E+01	0.	0.
-2.30	0.33	-34.54	3.68551E+01	0.	0.
-2.30	0.55	-34.54	3.68551E+01	0.	0.
-2.30	0.69	-34.54	3.68551E+01	0.	0.
-2.30	0.84	-34.54	3.68551E+01	0.	0.
-2.30	0.93	-34.54	6.86812E+01	0.	0.
-2.30	0.98	-3.06	1.22464E+00	9.57227E+00	-4.16186E+01
-2.30	1.02	-2.86	1.65942E+00	6.10465E+00	3.83408E+00
-2.30	1.08	-2.37	1.36232E+00	8.84714E+00	-5.81455E+00
-2.30	1.13	-1.89	1.02899E+00	1.02515E+01	-7.44620E+00
-1.61	-0.07	-13.82	9.14139E+00	0.	0.
-1.61	0.33	-13.82	9.14139E+00	0.	0.
-1.61	0.55	-13.82	9.14139E+00	0.	0.
-1.61	0.69	-13.82	9.14139E+00	0.	0.
-1.61	0.84	-13.82	9.14139E+00	0.	0.
-1.61	0.93	-2.68	2.26204E+00	1.38682E+01	-3.00113E+01
-1.61	0.98	-2.28	1.02767E+00	8.48908E+00	-7.28605E+00
-1.61	1.02	-1.92	9.82352E-01	6.81388E+00	-1.48420E+00
-1.61	1.08	-1.60	7.99758E-01	5.65714E+00	-3.33296E+00
-1.61	1.13	-1.30	6.36756E-01	6.30303E+00	-3.67864E+00
-0.92	-0.07	-10.10	6.12818E+00	0.	0.
-0.92	0.33	-10.10	6.12818E+00	0.	0.
-0.92	0.55	-10.10	6.12818E+00	0.	0.
-0.92	0.69	-10.10	6.12818E+00	0.	0.
-0.92	0.84	-2.52	1.29996E+00	1.17313E+01	-5.54165E+00
-0.92	0.93	-1.88	9.17734E-01	5.77568E+00	-3.56691E+00
-0.92	0.98	-1.63	8.31163E-01	5.47297E+00	-2.13020E+00
-0.92	1.02	-1.39	7.33898E-01	4.62617E+00	-1.93785E+00
-0.92	1.08	-1.17	6.41877E-01	4.01105E+00	-1.65781E+00
-0.92	1.13	-0.95	5.46831E-01	4.73333E+00	-2.00959E+00
-0.51	-0.07	-7.26	6.10270E+00	0.	0.
-0.51	0.33	-7.26	6.10270E+00	0.	0.
-0.51	0.55	-7.26	6.10270E+00	0.	0.
-0.51	0.69	-4.82	7.13376E+00	1.73814E+01	-7.04250E+00
-0.51	0.84	-2.22	7.79438E-01	9.96950E+00	-1.20504E+01
-0.51	0.93	-1.56	6.15035E-01	5.68907E+00	-1.40145E-01
-0.51	0.98	-1.32	6.55659E-01	5.13915E+00	3.68657E-01
-0.51	1.02	-1.10	6.58723E-01	4.21995E+00	-1.74223E-01
-0.51	1.08	-0.90	6.37975E-01	3.55846E+00	-6.85581E-01
-0.51	1.13	-0.71	5.85810E-01	4.01212E+00	-1.33832E+00
-0.22	-0.07	-5.66	5.31131E+00	0.	0.
-0.22	0.33	-5.66	5.31131E+00	0.	0.
-0.22	0.55	-4.88	7.50065E+00	5.57411E+00	-4.62427E+00
-0.22	0.69	-3.34	3.54524E+00	9.95122E+00	-2.32785E+01
-0.22	0.84	-1.98	8.68290E-01	7.34874E+00	-8.27822E+00
-0.22	0.93	-1.41	5.53309E-01	5.65803E+00	-1.79799E+00
-0.22	0.98	-1.15	5.09102E-01	5.47137E+00	3.46142E-02
-0.22	1.02	-0.92	5.51131E-01	4.29639E+00	7.27890E-01
-0.22	1.08	-0.72	5.78936E-01	3.46558E+00	3.25504E-01
-0.22	1.13	-0.54	5.88656E-01	3.72121E+00	9.77412E-02
0.	-0.07	-4.53	4.97206E+00	0.	0.
0.	0.33	-4.13	7.57456E+00	1.64939E+00	4.81875E+00
0.	0.55	-3.35	6.41432E+00	3.97068E+00	-2.05296E+01
0.	0.69	-2.73	1.76732E+00	5.20181E+00	-1.98331E+01
0.	0.84	-1.78	9.44250E-01	5.88691E+00	-4.95170E+00
0.	0.93	-1.28	6.22687E-01	4.98812E+00	-4.24025E+00
0.	0.98	-1.05	3.97750E-01	5.23315E+00	-1.67324E+00
0.	1.02	-0.81	4.47938E-01	4.50000E+00	1.19880E+00
0.	1.08	-0.60	5.13000E-01	3.64838E+00	1.34622E+00
0.	1.13	-0.41	5.92937E-01	3.93636E+00	1.83255E+00

density. The tabulated values are the (natural) logarithms of the indicated quantities, truncated to two decimal places for this example.

C. Final monotone interpolant. The parameters that define the final monotone interpolant, as plotted in Fig. 7, are listed in Table 2.

Acknowledgment. The authors would like to thank R. P. Dickinson for valuable discussions in the early stages of this work and for developing the software used to produce Figs. 2-7.

REFERENCES

[1] R. BEATSON AND Z. ZIEGLER, *Monotonicity preserving surface interpolation*, this Journal, this issue, pp. 401-411.

[2] F. N. FRITSCH, *Piecewise cubic Hermite interpolation package*, Lawrence Livermore National Laboratory Preprint UCRL-87285 (July 1982). (Poster presented at SIAM 30th Anniversary Meeting, Stanford Univ., 19-20 July 1982.)

[3] F. N. FRITSCH AND J. BUTLAND, *A method for constructing local monotone piecewise cubic interpolants*, SIAM J. Sci. Stat. Comput., 5 (1984), pp. 300-304.

[4] F. N. FRITSCH AND R. E. CARLSON, *Monotone piecewise cubic interpolation*, this Journal, 17 (1980), pp. 238-246.