ROCKY MOUNTAIN
JOURNAL OF MATHEMATICS
Volume 14, Number 2, Spring 1984

NEW USER-TRANSPARENT EDGE CONDITIONS FOR BICUBIC SPLINE SURFACE FITTING

L. H. SEITELMAN

0. Introduction. Surface fitting is an important element in many problems in applied mathematics. In particular, in aircraft engine design and development, the accurate representation of geometric part data for use in engineering analysis programs or for determination of appropriate processing operations is a key requirement in computer-aided design and computer-aided manufacturing (CAD/CAM) applications. This paper is concerned with the development of practical techniques for surface representation.

1. The problem. We consider the problem of producing a bicubic spline surface fit which interpolates to given function data on a rectangular grid. This problem, the two-dimensional analog of univariate cubic spline curve fitting, arises in a very broad spectrum of engineering applications, such as the mathematical modeling of airfoils in aircraft engine design.

2. Background: the one-dimensional case. Because our solution to the problem is a natural generalization of the one-dimensional case, we begin with a brief review of our work on the univariate problem. A more rigorous and complete discussion can be found in Ahlberg, Nilson, and Walsh [1] or deBoor [2]; our focus here is primarily on practical procedures.

In [1], the cubic spline fit is presented as the mathematical analog of the draftsman's spline, and the need to prescribe additional boundary conditions (i.e., other than interpolation) in order to determine this fit is introduced as a natural consequence of the underlying physical model. In [4], using a natural or "cardinal" set of basis functions for the case of equally spaced data on an infinite grid, Nilson showed constructively that, for a set of n equally spaced data points, the restrictions of $n + 2$ of these basis functions to the data interval could be used to develop a two-parameter family of cubic spline interpolants to the given data; additional constraints, the so-called "end" conditions, are necessary to determine a unique interpolating spline. Specification of end slopes, or

Received by the editors on August 5, 1982.

second derivatives, or other quantities, are among the possible choices to resolve the nonuniqueness.

In a large number of curve fitting applications, the user has no *a priori* information about the curve fit behavior other than the function values at the data points. As a result, it is in practice convenient to use an adaptive procedure to prescribe suitable end conditions in order to determine the appropriate spline fit to the data. In [5], Seitelman developed a procedure for this purpose, and showed its superiority to several other methods for a reasonable spectrum of curve fitting problems. The same procedure was demonstrated in [6] to produce more accurate curve fits for the always difficult problem of extrapolation.

3. User-transparent end conditions for spline fitting. The automatic procedure for specifying spline end conditions is derived in [5] as follows: Given data points (x_i, y_i), $i = 1, \ldots, n$, and assuming that the interpolat-

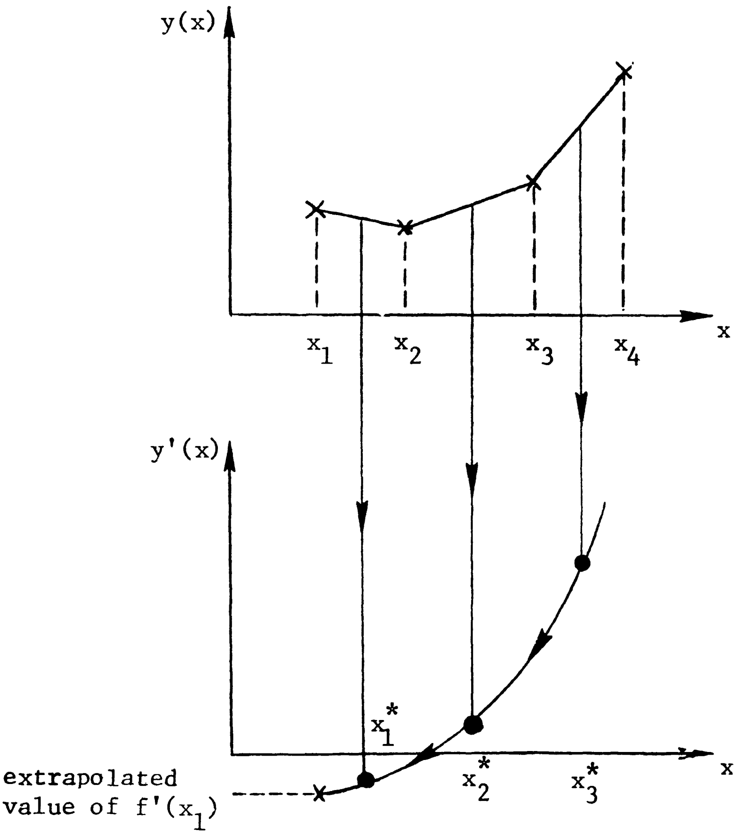


Figure 1: User-transparent end condition procedure.

ing function $y(x)$ to the data is smooth, the mean value theorem tells us that there exist points x_i^*, $i = 1, \ldots, n-1$, with $x_i < x_i^* < x_{i+1}$, such that

$$y'(x_i^*) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i},$$

$i = 1, \ldots, n-1$. If we assume that $x_i^* = (1/2)(x_i + x_{i+1})$, $i = 1, 2, 3$, i.e., if we assume that the mean value theorem points for the first three intervals lie at the center of each of those intervals, then we can use a parabolic extrapolation of the (assumed) values of the slope $y'(x)$ at these points to estimate the end slope $y'(x_1)$. The situation is depicted in Figure 1.

To be precise, we have

$$\begin{aligned} y'(x_1) = {} & \frac{y_2 - y_1}{x_2 - x_1} \frac{(2x_1 - x_2 - x_3)}{(x_1 - x_3)} \frac{(2x_1 - x_3 - x_4)}{(x_1 + x_2 - x_3 - x_4)} \\ & + \frac{y_3 - y_2}{x_3 - x_2} \frac{(x_1 - x_2)(2x_1 - x_3 - x_4)}{(x_3 - x_1)(x_2 - x_4)} \\ & + \frac{y_4 - y_3}{x_4 - x_3} \frac{(x_1 - x_2)(2x_1 - x_2 - x_3)}{(x_3 + x_4 - x_1 - x_2)(x_4 - x_2)}. \end{aligned}$$

A similar procedure is used to estimate $y'(x_n)$.

4. Visualization and computation. Although cardinal splines are extremely useful in visualizing curve fitting problems, the set of cardinal spline functions derived in [4] is of little computational interest, since all of the basis functions are nonzero except at the data points. However, the great value of this formulation is the explicit representation of the effect of the data points of the interpolating fit, and the clear representation of the need for supplementary end conditions to adjust the potentially "wiggly" behavior of the fit at the ends of the data set. This explicitness meshes naturally with the physical model underlying the mathematical representation of the spline fit.

5. B-splines and end conditions. A computationally efficient approach to the interpolation problem is provided by the use of splines with compact support. This alternate to the cardinal spline basis set for representing spline functions is provided by the B-spline family described extensively in [2]. For the cubic case, this family consists of cubic spline functions each of which is nonzero over only four data intervals. The great utility of this family is the fact that the compact support feature guarantees that the evaluation of a linear combination of B-splines involves the evaluation of at most four non-zero terms.

Let $x_1, x_2, \ldots, x_n$ be the data points, with $x_i < x_j$ if $i < j$. We define

$B_i(x)$, the B-spline centered at a data point x_i, $i = 1, \ldots, n$, to be that B-spline with support in the interval (x_{i-2}, x_{i+2}).

(We set

$$x_0 = x_1 - h_1, \quad x_{n+1} = x_n + h_n$$
$$x_{-1} = x_0 - h_1, \quad x_{n+2} = x_{n+1} + h_n$$

where

$$h_1 = x_1 - x_0, \quad h_n = x_n - x_{n-1}$$

to define the B-splines at the edges of the data sets.)

It is easily shown that we can find a unique linear combination of the n functions $B_i(x)$ which interpolates to the given function data $f(x_i)$ at the x_i, i.e., that there exists a unique linear combination $\sum_{k=1}^{n} \alpha_k B_k(x)$ of $B_1(x)$, $B_2(x)$, $\ldots$, $B_n(x)$ which satisifies

$$f(x_j) = \sum_{k=1}^{n} \alpha_k B_k(x_j), \; j = 1, \ldots, n. \tag{1}$$

This is an interesting development, in view of the siutation discussed in §2, i.e., the fact that there is a two-parameter family of cubic splines that interpolates to given data. Where did those two degrees of freedom "go"?

The answer, of course, is: "Nowhere!" For if we define $x_{-2} = x_{-1} - h_1$, $x_{n+3} = x_{n+2} + h_n$, then it can easily be shown that there is a two-parameter family of solutions to the system of equations,

$$f(x_j) = \sum_{k=0}^{n+1} \beta_k B_k(x_j), \; j = 1, \ldots, n. \tag{2}$$

In this case, if the end conditions are varied, then in general *all* the β_k will be affected. This is in sharp contrast to the cardinal spline formulation, in which the point data interpolation requirement fixes all but two of the basis function coefficients.

The abbreviated form (1) of the interpolating B-spline is equivalent to specifying two particular end conditions on the cubic spline function. These conditions will now be determined.

If we differentiate (1), we obtain $f'(x) = \sum_{k=1}^{n} \alpha_k B_k'(x)$. By construction, the B-splines satisfy $B_k'(x_j) = 0$, if $j \neq k - 1, k, k + 1$, and therefore $f'(x_1) = \alpha_1 B_1'(x_1) + \alpha_2 B_2'(x_1)$. If $x_3 - x_2 = x_2 - x_1$, then $B_1'(x_1) = 0$, and so, for the case of equally spaced data, we have $f'(x_1) = \alpha_2 B_2'(x_1)$. A similar development at the right-hand end of the data set yields, for the equally spaced data case, $f'(x_n) = \alpha_{n-1} B_{n-1}'(x_n)$. Therefore, it is clear, that for the case of equally spaced data, the use of the interpolating B-spline (1) is equivalent to specifying end slopes which depend only on the B-spline coefficient at the second and next-to-last data points. The contrast between these constraints and the user-transparent adaptive conditions developed in [5] is striking.

The analysis required to determine the B-spline coefficients for the case (2), where the user-transparent end conditions are prescribed as in [5], is presented in the Appendix (§13). We are primarily interested here in the extension of this work to data fitting in two dimensions.

6. Extensions to surface fitting. With the above as background, we proceed now to the problems of two-dimensional surface fitting. Specifically, we consider the case of data defined on a rectangular grid, a not uncommon practical case. A typical example of such an application is the parametric fitting of data defined on a network of curves in space, where the same number of points is identified on each curve, thereby defining a "rectangular" grid in the parametric variables u and v, with equal spacings in the point numbers in each of the two curve 'directions". An example of this parametrization is given in the figure below. Fitting each variable of the point coordinate data as a function of these parameters, i.e., representing points on the surface as the set of points of the form $(x(u, v), y(u, v), z(u, v))$, produces a parametric surface.

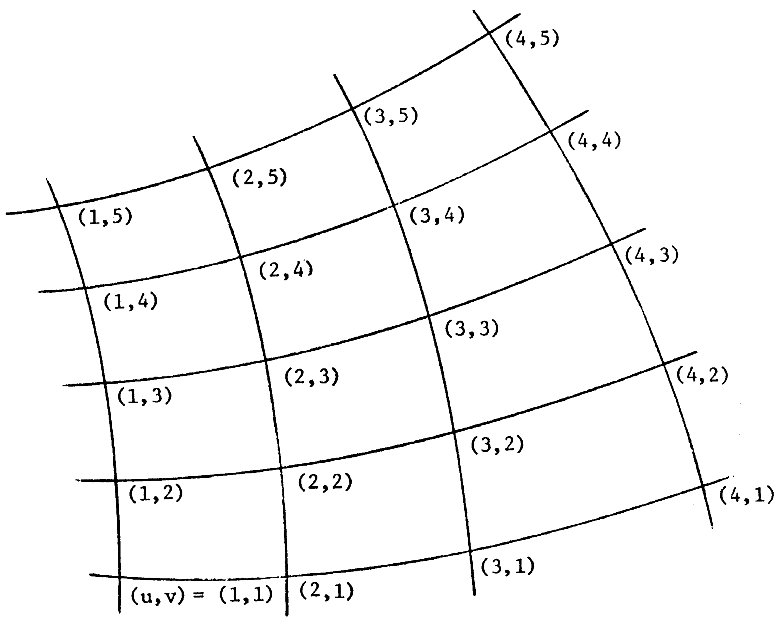


Figure 2: Identifying the same number of points on each member of a family of curves parametrizes that network of points.

We develop the ideas for surface fitting on a rectangular grid. For simplicity of presentation, we consider the case of equal spacing in each coordinate direction, although our remarks and results clearly generalize to the non-equally spaced case.

As in the one-dimensional case, the concepts are most easily presented by using cardinal splines (in this case, cardinal bicubic spline functions) as basic building blocks. Although computationally inefficient, these functions provide a very convenient tool for visualizing the solution to the surface interpolation problem.

Let $x_i = x_1 + (i - 1)\Delta_x$, $i = 1, \ldots, m$, $y_j = y_1 + (j - 1)\Delta_y$, $j = 1, \ldots, n$, with Δ_x, $\Delta_y > 0$ define the (equally spaced) grid of given data points. A bicubic spline on the rectangle $\mathscr{R} = [x_1, x_n] \times [y_1, y_n]$ is a function which is doubly cubic (in both x and y) on each subrectangle $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$, and which is twice continuously differentiable over $\mathscr{R}$, i.e., which is in the class $C_2^4(\mathscr{R})$, where $C_r^n(\mathscr{R})$ is the family of functions $f(x, t)$ whose n-th order partial derivatives, involving no more than r-th order differentiation with respect to a single variable, exist and are continuous.

DeBoor [3] showed that there exists a unique bicubic spline which interpolates to the given function data $f_{ij} = f(x_i, y_j)$, $i = 1, \ldots, m, j = 1, \ldots, n$, and which also satisfies the slope constraints

$$\left.\frac{\partial f}{\partial x}\right|_{x=x_1,\, x=y_j} = a_j, j = 1, 2, \ldots, n,$$

$$\left.\frac{\partial f}{\partial x}\right|_{x=x_m,\, y=y_j} = b_j, j = 1, 2, \ldots, n,$$

$$\left.\frac{\partial f}{\partial y}\right|_{x=x_i,\, y=y_1} = c_i, \mathrm{i} = 1, 2, \ldots, m,$$

and

$$\left.\frac{\partial f}{\partial y}\right|_{x=x_i,\, y=y_n} = d_i, i = 1, 2, \ldots, m,$$

where the a_j, b_j, c_i, and d_i are given, and the twist constraints at the corners,

$$\left.\frac{\partial^2 f}{\partial x \partial y}\right|_{x=x_1,\, y=y_1} = e_1,$$

$$\left.\frac{\partial^2 f}{\partial x \partial y}\right|_{x=x_1,\, y=y_n} = e_2,$$

$$\left.\frac{\partial^2 f}{\partial x \partial y}\right|_{x=x_m,\, y=y_1} = e_3,$$

and

$$\left.\frac{\partial^2 f}{\partial x \partial y}\right|_{x=x_m,\, y=y_n} = e_4.$$

These edge (and corner) conditions determine a so-called Type *I* bicubic spline.

As in the one-dimensional case, in a large number of curve fitting applications, the user has no *a priori* information about the surface fit behavior other than the function values at the data points. As a result, we shall develop an adaptive procedure to derive suitable edge and corner conditions in order to determine the appropriate bicubic spline fit to the data.

7. Specifying user-transparent conditions for surface representation. To determine the bicubic spline fit, we shall derive values for $\partial f/\partial x$ at $x = x_1$ and $x = x_m$, for each y_j, for $\partial f/\partial y$ at $y = y_1$ and $y = y_m$ for each x_i, and for $\partial^2 f/\partial x \partial y$ at the corners of the rectangular grid.

Let $A(x)$ be the univariate cardinal cubic spline [3] centered at $x = 0$; if $\lambda = -2 + \sqrt{3}$, then

$$A(x) = \begin{cases} 1 - 3(\lambda + 1)x^2 + (3\lambda + 2)x^3, & 0 \leqq x \leqq 1 \\ 3\lambda^n(x - n - (\lambda + 2)(x - n)^2 + (\lambda + 1)(x - n)^3), & n \leqq x \leqq n+1 \text{ and } n > 0 \text{ an integer}, \\ A(-x), & x < 0 \quad , \end{cases}$$

Straightforward calculation gives

$$A'(x) = \begin{cases} -6(\lambda + 1)x + 3(3\lambda + 2)x^2, & 0 \leqq x \leqq 1 \quad , \\ 3\lambda^n(1 - 2(\lambda + 2)(x - n) + 3(\lambda + 1)(x - n)^2), & n \leqq x \leqq n+1 \text{ and } n > 0 \text{ an integer}, \\ -A'(-x), & x < 0 \quad , \end{cases}$$

and

$$A''(x) = \begin{cases} -6(\lambda + 1) + 6(3\lambda + 2)x, & 0 \leqq x \leqq 1 \quad , \\ 6\lambda^n(-(\lambda + 2) + 6(\lambda + 1)(x - n)), & n \leqq x \leqq n+1 \text{ and } n > 0 \text{ an integer}, \\ A''(-x), & x < 0 \quad , \end{cases}$$

and so, in particular,

$$A'(x) = \begin{cases} 0 \;, & \text{if } x = 0, \\ 3\lambda^n \;, & \text{if } x = n > 0,\ n \text{ an integer}, \\ -3\lambda^n, & \text{if } x = -n < 0,\ n \text{ an integer}, \end{cases}$$

and

$$A''(x) = \begin{cases} -6(\lambda + 1) & , \text{ if } x = 0, \\ -6\lambda^n(\lambda + 2), & \text{if } x = n > 0, n \text{ an integer}, \\ -6\lambda^n(\lambda + 2), & \text{if } x = -n < 0, n \text{ an integer}. \end{cases}$$

We define the interpolating bivariate cubic spline as a linear combination of products of one-dimensional cardinal splines,

$$(*) \qquad t(x, y) = \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} t_{ij} \, A\left(\frac{x - x_i}{\Delta_x}\right) A\left(\frac{y - y_j}{\Delta_y}\right).$$

It follows from the relations for the cardinal spline that

$$\begin{aligned} t(x_r, y_s) &= \sum_i \sum_j t_{ij} \, A\left(\frac{x_r - x_i}{\Delta_x}\right) A\left(\frac{y_s - y_j}{\Delta_y}\right) \\ &= \sum_i \sum_j t_{ij} \, A(r - i) \, A(s - j) \\ &= \sum_i \sum_j t_{ij} \delta_{ri} \delta_{sj} \\ &= t_{rs}, \end{aligned}$$

and so the interpolation conditions require that $t_{rs} = f_{rs}$, $r = 1, \ldots, m$; $s = 1, \ldots, n$. To determine the remaining coefficients, we can use the previously developed extrapolation techniques to determine slope-like end conditions at the boundary of the rectangular region $[x_1, x_m] \times [y_1, y_n]$. Indeed, if $y = y_s$ is fixed, then we have

$$\begin{aligned} t(x, y_s) &= \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} t_{ij} \, A\left(\frac{x - x_i}{\Delta_x}\right) A\left(\frac{y_s - y_j}{\Delta_y}\right) \\ &= \sum_i \sum_j t_{ij} \, A\left(\frac{x - x_i}{\Delta_x}\right) A(s - j) \\ &= \sum_i \sum_j t_{ij} \, A\left(\frac{x - x_i}{\Delta_x}\right) \delta_{sj} \\ &= \sum_i t_{is} \, A\left(\frac{x - x_i}{\Delta_x}\right). \end{aligned}$$

Therefore,

$$\frac{\partial t(x, y_s)}{\partial x} = \frac{1}{\Delta_x} \sum_i t_{is} \, A'\left(\frac{x - x_i}{\Delta_x}\right).$$

Similarly,

$$\frac{\partial t(x_r, y)}{\partial y} = \frac{1}{\Delta_y} \sum_j t_{rj} \, A'\left(\frac{y - y_j}{\Delta_y}\right).$$

It follows that

$$\frac{\partial t(x_r, y_s)}{\partial x} = \frac{1}{\Delta_x} \sum_i t_{is} A'\left(\frac{x_r - x_i}{\Delta_x}\right)$$
$$= \frac{1}{\Delta_x} \sum_i t_{is} A'(r - i)$$
$$= \frac{1}{\Delta_x} \sum_{i=0}^{r-1} t_{is} \cdot 3\lambda^{r-i} - \frac{1}{\Delta_x} \sum_{i=r+1}^{m+1} t_{is} \cdot 3\lambda^{i-r},$$

and the quantity on the left-hand side of the above can be estimated at $x = x_1$ from the data at (x_1, y_s), (x_2, y_s), (x_3, y_s), and (x_r, y_s) by the techniques described in [5]. Similarly, at $x = x_m$, the data points at (x_{m-3}, y_s), (x_{m-2}, y_s), (x_{m-1}, y_s), and (x_m, y_s) can be used to estimate the left-hand side.

In like fashion, we have the result

$$\frac{\partial t(x_r, y_s)}{\partial y} = \frac{1}{\Delta_y} \sum_{j=0}^{n+1} t_{rj} A'\left(\frac{y_s - y_j}{\Delta_y}\right)$$
$$= \frac{1}{\Delta_y} \sum_{j=0}^{s-1} t_{rj} \cdot 3\lambda^{s-j} - \frac{1}{\Delta_y} \sum_{j=s+1}^{n+1} t_{rj} \cdot 3\lambda^{j-s},$$

and the values of $\partial t/\partial y$ can be extrapolated from the function values near the $y = y_1$ and $y = y_s$ edges of the data rectangle.

Specifically, we have

$$\left.\begin{aligned} \frac{\partial t(x_1, y_s)}{\partial x} &= \frac{1}{\Delta_x}[3\lambda t_{0s} - 3\sum_{i=2}^{m+1} t_{is}\lambda^{i-1}] \\ \frac{\partial t(x_m, y_s)}{\partial x} &= \frac{1}{\Delta_x}[\sum_{i=0}^{n-1} 3t_{is}\lambda^{m-i} - 3\lambda t_{m+1,s}] \end{aligned}\right\} s = 1, \ldots, n,$$

$$\left.\begin{aligned} \frac{\partial t(x_r, y_1)}{\partial y} &= \frac{1}{\Delta_y}[3\lambda t_{r0} - 3\sum_{j=2}^{m+1} t_{rj}\lambda^{j-1}] \\ \frac{\partial t(x_r, y_n)}{\partial y} &= \frac{1}{\Delta_y}[3\sum_{j=0}^{n-1} t_{rj}\lambda^{n-j} - 3\lambda t_{r,n+1}] \end{aligned}\right\} r = 1, \ldots, m,$$

while the extrapolated estimates are given by

$$\frac{\partial t(x_1, y_s)}{\partial x} \cong \frac{1}{8\Delta_x}(-15f_{1,s} + 25f_{2,s} - 13f_{3,s} + 3f_{4,s})$$

$$\frac{\partial t(x_m, y_s)}{\partial x} \cong \frac{1}{8\Delta_x}(15f_{m,s} - 25f_{m-1,s} + 13f_{m-2,s} - 3f_{m-3,s})$$

$$\frac{\partial t(x_r, y_1)}{\partial y} \cong \frac{1}{8\Delta_y}(-15f_{r,1} + 25f_{r,2} - 13f_{r,3} + 3f_{r,4})$$

$$\frac{\partial t(x_r, y_n)}{\partial y} \cong \frac{1}{8\Delta_y}(15f_{r,n} - 25f_{r,n-1} + 13f_{r,n-2} - 3f_{r,n-3}).$$

These relationships permit the simultaneous solution for pairs of unknown coefficients; to be precise, the relations for $\partial t(x_1, y_s)/\partial x$ and $\partial t(x_m, y_s)/\partial x$ determine $t_{0,s}$ and $t_{m+1,s}$ as the solution of the equations

$$3\lambda t_{0,s} - 3\lambda^m t_{m+1,s} = 3\sum_{i=2}^{m} t_{is}\lambda^{i-1} + (1/8)(-15f_{1,s} + 25f_{2,s} - 13f_{3,s} + 3f_{4,s})$$

and

$$3\lambda^m t_{0,s} - 3\lambda t_{m+1,s}$$
$$= -3\sum_{i=1}^{m-1} t_{i,s}\lambda^{m-i} + (1/8)(15f_{m,s} - 25f_{m-1,s} + 13f_{m-2,s} - 3f_{m-3,s})$$

In like fashion, the relations for $\partial t(x_r, y_1)/\partial y$ and $\partial t(x_r, y_n)/\partial y$ determine $t_{r,0}$ and $t_{r,n+1}$ as the solution of the equations

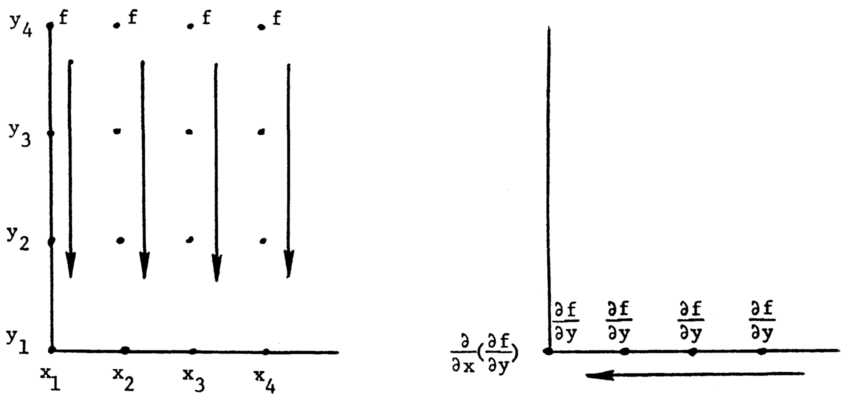
$$3\lambda t_{r,0} - 3\lambda^n t_{r,n+1} = 3\sum_{j=2}^{n} t_{rj}\lambda^{j-1} + (1/8)(-15f_{r,1} + 25f_{r,2} - 13f_{r,3} + 3f_{r,4})$$

and

$$3\lambda^n t_{r,0} - 3\lambda t_{r,n+1}$$
$$= -3\sum_{j=1}^{n-1} t_{rj}\lambda^{n-j} + (1/8)(15f_{r,n} - 25f_{r,n-1} + 13f_{r,n-2} - 3f_{rn-3}).$$

Finally, we determine the four remaining spline coefficients $t_{0,0}$, $t_{0,n+1}$, $t_{m+1,0}$, and $t_{m+1,n+1}$ by developing equations to estimate the twist at the corners of the grid.

To do this, we need only recognize that we have already developed



(a) Extrapolate in y to determine $\dfrac{\partial f}{\partial y}$ at edge.

(b) Extrapolate in x to determine $\dfrac{\partial}{\partial x}\left(\dfrac{\partial f}{\partial y}\right)$ at corner.

Figure 3: Extrapolation Procedures to Determine Twist.

estimates for $g = \partial f/\partial y$ at each of the data points (x_1, y_1), (x_2, y_1), (x_3, y_1) and (x_4, y_1). If we now consider the function g, we can develop an estimate of $\partial g/\partial x = (\partial/\partial x)(\partial f/\partial y)$. These extrapolations are indicated schematically in Figure 3.

For our grid, the equation for the twist is given as ($g|_{r,s}$ represents the quantity $g(x_r, y_s)$)

$$
\begin{aligned}
\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right) \text{at} (x_1 y_1) &\cong \frac{1}{8\Delta_x}\left(-15\frac{\partial f}{\partial y}\bigg|_{1,1} + 24\frac{\partial f}{\partial y}\bigg|_{2,1} - 13\frac{\partial f}{\partial y}\bigg|_{3,1} + 3\frac{\partial f}{\partial y}\bigg|_{4,1}\right)\\
&= \frac{1}{8\Delta_x}\left(-15\left(\frac{1}{8\Delta_y}[-15f_{1,1} + 25f_{1,2} - 13f_{1,3} + 3f_{1,4}]\right)\right.\\
&\quad + 25\left(\frac{1}{8\Delta_y}[-15f_{2,1} + 25f_{2,2} - 13f_{2,3} + 3f_{2,4}]\right)\\
&\quad - 13\left(\frac{1}{8\Delta_y}[-15f_{3,1} + 25f_{3,2} - 13f_{3,3} + 3f_{3,4}]\right)\\
&\quad \left.+ 3\left(\frac{1}{8\Delta_y}[-15f_{4,1} + 25f_{4,2} - 13f_{4,3} + 3f_{4,4}]\right)\right)\\
&= \frac{1}{8\Delta_y}\left(-15\left(\frac{1}{8\Delta_x}[-15f_{1,1} + 25f_{2,1} - 13f_{3,1} + 3f_{4,1}]\right)\right.\\
&\quad + 25\left(\frac{1}{8\Delta_x}[-15f_{1,2} + 25f_{2,2} - 13f_{3,2} + 3f_{4,2}]\right)\\
&\quad - 13\left(\frac{1}{8\Delta_x}[-15f_{1,3} + 25f_{2,3} - 13f_{3,3} + 3f_{4,3}]\right)\\
&\quad \left.+ 3\left(\frac{1}{8\Delta_x}[-15f_{1,4} + 25f_{2,4} - 13f_{3,4} + 3f_{4,4}]\right)\right)\\
&= \frac{1}{8\Delta_y}\left(-15\frac{\partial f}{\partial x}\bigg|_{1,1} + 25\frac{\partial f}{\partial x}\bigg|_{1,2} - 13\frac{\partial f}{\partial x}_{1,3} + 3\frac{\partial f}{\partial x}\bigg|_{1,4}\right)\\
&= \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right) \text{at } (x_1, x_1).
\end{aligned}
$$

The above development shows that, regardless of the order in which the first partial derivatives are extrapolated, the twist which is determined is the same. It is also clear from this development that, even for unequal spacing of the data in the rectangular grid, the linearity of the extrapolation operator ensures that the twist which is obtained at the grid corners is independent of the order of extrapolation.

A similar development can be used to determine the twist at the other corners of the grid. We shall denote the computed values of the twist at (x_1, y_1) as τ_{11}, at (x_1, y_n) as τ_{1n}, etc.

From the form of the spline (*), it follows that

$$\frac{\partial^2 t}{\partial x \partial y} = \frac{1}{\Delta_x \Delta_y} \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} t_{ij}\, A'\left(\frac{x - x_i}{\Delta_x}\right) A'\left(\frac{y - y_j}{\Delta_y}\right)$$

and therefore

$$\begin{aligned}
\frac{\partial^2 t(x_r, y_s)}{\partial x \partial y} &= \frac{1}{\Delta_x \Delta_y} \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} t_{ij}\, A'\left(\frac{x_r - x_i}{\Delta_x}\right) A'\left(\frac{y_s - y_j}{\Delta_y}\right) \\
&= \frac{1}{\Delta_x \Delta_y} \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} t_{ij}\, A'(r - i)\, A'(s - j) \\
&= \frac{9}{\Delta_x \Delta_y} \Bigg(\sum_{i=0}^{r-1} \sum_{j=0}^{s-1} t_{ij} \lambda^{r+s-i-j} - \sum_{i=r+1}^{m+1} \sum_{j=0}^{s-1} t_{ij} \lambda^{i+s-r-j} \\
&\qquad - \sum_{i=0}^{r-1} \sum_{j=s+1}^{n+1} t_{ij} \lambda^{r+j-i-s} + \sum_{i=r+1}^{m+1} \sum_{j=s+1}^{n+1} t_{ij} \lambda^{i+j-r-s} \Bigg).
\end{aligned}$$

In the equation above, all but $t_{0,0}$, $t_{0,n+1}$, $t_{m+1,0}$, and $t_{m+1,\,n+1}$ are known; these are determined as the solution of the corner twist equations:

$$\begin{aligned}
\tau_{11} &= \frac{9}{\Delta_x \Delta_y} \left(t_{0,0} \lambda^2 - \sum_{i=2}^{m+1} t_{i0} \lambda^i - \sum_{j=2}^{n+1} t_{0j} \lambda^j + \sum_{i=2}^{m+1} \sum_{j=2}^{n+1} t_{ij} \lambda^{i+j-2} \right) \\
\tau_{1n} &= \frac{9}{\Delta_x \Delta_y} \left(\sum_{j=0}^{n-1} t_{0j} \lambda^{1+n-j} - \sum_{i=2}^{m+1} \sum_{j=0}^{n-1} t_{ij} \lambda^{i+n-1-j} - t_{0,n+1} \lambda^2 + \sum_{i=2}^{m+1} t_{i,n+1} \lambda^i \right) \\
\tau_{m1} &= \frac{9}{\Delta_x \Delta_y} \left(\sum_{i=0}^{m-1} t_{i0} \lambda^{m-i+1} - t_{m+1,0} \lambda^2 - \sum_{i=0}^{m-1} \sum_{j=2}^{n+1} t_{ij} \lambda^{m+j-i-1} + \sum_{j=2}^{n+1} t_{m+1,j} \lambda^j \right)
\end{aligned}$$

and

$$\tau_{mn}$$

$$= \frac{9}{\Delta_x \Delta_y} \left(\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} t_{ij} \lambda^{m+n-i-j} - \sum_{j=0}^{n-1} t_{m+1,j} \lambda^{1+n-j} - \sum_{i=0}^{m-1} t_{i,n+1}{}^{m+1-i} + t_{m+1,n+1} \lambda^2 \right).$$

8. B-Splines in two dimensions. The user-transparent end conditions developed for curve fitting provided a reference point for developing the user-transparent edge conditions for surface fitting described above. With the use of an augmented *B*-spline function set analogous to the one-dimensional case, we now develop a *B*-spline package which produces a computationally efficient fitting technique for surface representation.

As before, we assume data f_{ij} to be given at points (x_i, y_j) on a uniform rectangular mesh ($i = 1, \ldots, m; j = 1, \ldots, n$).

If we write

$$s(x, y) = \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} a_{ij}\, B\left(\frac{x - x_i}{\Delta_x}\right) B\left(\frac{y - y_j}{\Delta_y}\right), \tag{3}$$

where $B(x)$ is the *B*-spline on a uniform mesh defined in the Appendix, then

$$
\begin{aligned}
s(x_r, y_t) &= \sum_{i=0}^{m+1}\sum_{j=0}^{n+1} a_{ij}\, B\left(\frac{x_r - x_i}{\Delta_x}\right) B\left(\frac{y_t - y_j}{\Delta_y}\right) \\
&= \sum_{i=0}^{m+1}\sum_{j=0}^{n+1} a_{ij}\, B(r-i)\, B(t-j) \\
&= (1/16)q_{r-1,t-1} + (1/4)a_{r-1,t} + (1/16)a_{r-1,t+1} \\
&\quad + (1/4)a_{r,t-1} + a_{r,t} + (1/4)a_{r,t+1} \\
&\quad + (1/16)a_{r+1,t-1} + (1/4)a_{r+1t} + (1/16)a_{r+1,t+1}
\end{aligned}
$$

or the interpolation requirement ($s(x_r, y_t) = f(x_r, y_t)$) can be written as the constraint

$$
\begin{aligned}
a_{r-1,t-1} + 4a_{r-1,t} + a_{r-1,t+1} + 4(a_{r,t-1} + 4a_{r,t} + a_{r,t+1}) \\
+ a_{r+1,t-1} + 4a_{r+1,t} + a_{r+1,t+1} = 16f(x_r, y_t).
\end{aligned}
\tag{4}
$$

From (3), it follows that

$$
\frac{\partial s(x, y)}{\partial x} = \frac{1}{\Delta_x} \sum_{i=0}^{m+1}\sum_{j=0}^{n+1} a_{ij}\, B'\left(\frac{x - x_i}{\Delta_x}\right) B\left(\frac{y - y_j}{\Delta_y}\right)
$$

so

$$
\begin{aligned}
\frac{\partial s(x_r, y_t)}{\partial x} &= \frac{1}{\Delta_x} \sum_{i=0}^{m+1}\sum_{j=0}^{n+1} a_{ij}\, B'\left(\frac{x_r - x_i}{\Delta_x}\right) B\left(\frac{y_t - y_j}{\Delta_y}\right) \\
&= \frac{1}{\Delta_x} \sum_{i=0}^{m+1}\sum_{j=0}^{n+1} a_{ij}\, B'(r-i)\, B(t-j) \\
&= \frac{1}{\Delta_x} [a_{r-1,t-1}\, B'(1)B(1) + a_{r-1,t}\, B'(1)B(0) \\
&\quad + a_{r-1,t+1}\, B'(1)B(-1) + a_{r++1,t-1}\, B'(-1)B(1) \\
&\quad + a_{r+1,t}\, B'(-1)B(0) + a_{r+1,t+1}\, B'(-1)B(-1)] \\
&= \frac{1}{\Delta_x}\Big[-\frac{3}{16}a_{r-1,t-1} - \frac{3}{4}a_{r-1,t} - \frac{3}{16}a_{r-1,t+1} \\
&\quad + \frac{3}{16}a_{r+1,t-1} + \frac{3}{4}a_{r+1,t} + \frac{3}{16}a_{r+1,t+1}\Big] \\
&= \frac{3}{16\Delta_x}[-a_{r-1,t-1} - 4a_{r-1,t} - a_{r-1,t+1} \\
&\quad + a_{r+1,t-1} + 4a_{r+1,t} + a_{r+1,t+1}],
\end{aligned}
\tag{5}
$$

for $1 \leqq r \leqq m$, $1 \leqq t \leqq n$. Similarly,

$$
\begin{aligned}
\frac{\partial s(x_r, y_t)}{\partial y} &= \frac{3}{16\Delta_y}[-a_{r-1,t-1} - 4a_{r,t-1} - a_{r+1,t-1} \\
&\quad + a_{r-1,t+1} + 4a_{r,t+1} + a_{r+1,t+1}],
\end{aligned}
\tag{6}
$$

for $1 \leqq r \leqq m$, $1 \leqq t \leqq n$. Finally,

$$\frac{\partial^2 s(x, y)}{\partial x \partial y} = \frac{1}{\Delta_x \Delta_y} \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} a_{ij}\, B'\left(\frac{x - x_i}{\Delta_x}\right) B'\left(\frac{y - y_j}{\Delta_y}\right)$$

so

$$\begin{aligned}\frac{\partial^2 s(x_r, y_t)}{\partial x \partial y} &= \frac{1}{\Delta_x \Delta_y} \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} a_{ij}\, B'\left(\frac{x_r - x_i}{\Delta_x}\right) B'\left(\frac{y_t - y_j}{\Delta_y}\right) \\ &= \frac{1}{\Delta_x \Delta_y} \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} a_{ij}\, B'(r - i)\, B'(t - j) \\ &= \frac{9}{16\Delta_x \Delta_y}[a_{r-1,\, t-1} - a_{r-1,\, t+1} - a_{r+1,\, t-1} + a_{r+1,\, t+1}],\end{aligned}$$

for $1 \leqq r \leqq m$, $1 \leqq t \leqq n$.

9. Formulation of the two-dimensional equation system. We now examine the system of equations which determines the coefficients a_{ij} in (3) when user-transparent edge conditions are assumed for the rectangular fit.

In this case, we extrapolate the given function data to estimate $\partial s/\partial x$ at $x = x_1$ and $x = x_m$, $\partial s/\partial y$ at $y = y_1$, and $y = y_n$, and $\partial^2 s/\partial x \partial y$ at each of the four corners of the data grid. To be precise, we set

$$s_x(1, t) = \left.\frac{\partial s}{\partial x}\right|_{(x_1, y_t)} = \frac{1}{8\Delta_x}(-15f_{1,t} + 25f_{2,t} - 13f_{3,t} + 3f_{4,t}) \tag{8}$$

$$s_x(m, t) = \left.\frac{\partial s}{\partial x}\right|_{(x_m, y_t)} = \frac{1}{8\Delta_x}(15f_{m,t} - 25f_{m-1,t} + 13f_{m-2,t} - 3f_{m-3,t}) \tag{9}$$

$$s_y(r, 1) = \left.\frac{\partial s}{\partial y}\right|_{(x_r, y_1)} = \frac{1}{8\Delta_y}(-15f_{r,1} + 25f_{r,2} - 13f_{r,3} + 3f_{r,4}) \tag{10}$$

$$s_y(r, n) = \left.\frac{\partial s}{\partial y}\right|_{(x_r, y_n)} = \frac{1}{8\Delta_y}(15f_{r,n} - 25f_{r,n-1} + 13f_{r,n-2} - 3f_{r,n-3}) \tag{11}$$

$$\begin{aligned}&s_{xy}(1, 1) \\ &= \left.\frac{\partial^2 s}{\partial x \partial y}\right|_{(x_1, x_1)} = \frac{1}{8\Delta_x}(-15s_y(1, 1) + 25s_y(2, 1) - 13s_y(3, 1) + 3s_y(4, 1))\end{aligned} \tag{12}$$

$$\begin{aligned}s_{xy}(m, n) &= \left.\frac{\partial^2 s}{\partial x \partial y}\right|_{(x_m, y_n)} \\ &= \frac{1}{8\Delta_x}(15s_y(m,n) - 25s_y(m - 1,n) + 13s_y(m - 2,n) - 3s_y(m - 3,n))\end{aligned} \tag{13}$$

$$\begin{aligned}s_{xy}(m, 1) &= \left.\frac{\partial^2 s}{\partial x \partial y}\right|_{(x_m, y_1)} \\ &= \frac{1}{8\Delta_y}(15s_y(m, 1) - 25s_y(m - 1, 1) + 13s_y(m - 2, 1) - 3s_y(m - 3. 1))\end{aligned} \tag{14}$$

$$\begin{aligned}s_{xy}(1, n) &= \left.\frac{\partial^2 s}{\partial x \partial y}\right|_{(x_1, y_n)} \\ &= \frac{1}{8\Delta_y}(-15s_y(1,n) + 25s_y(2, n) - 13s_y(3, n) + 3s_y(r, n)).\end{aligned} \tag{15}$$

The matrix formulation for his problem is as follows: Let

$$b_i = (a_{0,i}\, a_{1,i}\, a_{2,i} \cdots a_{m,i}\, a_{m+1,i})^T,\ i = 0, 1, \ldots, n+1,$$

and let the $(m+2) \times (m+2)$ matrix $C_{m+2} = (c_{ij}^{(m+2)})$ be defined by

$$c_{ij}^{(m+2)} = \begin{cases} -1 & \text{if } (i, j) = (1, 1) \text{ or } (m+2, m), \\ 1 & \text{if } (i\ j) = (1, 3) \text{ or } (m+2, m+2), \\ 4 & \text{if } 1 < i = j < m+2, \\ 1 & \text{if } |i-j| = 1 \text{ and } 1 < i < m+2, \\ 0 & \text{otherwise.} \end{cases}$$

Then the spline equations at $y = y_1$, for $\partial^2 s/\partial x \partial y$ at $x = x_1$, for $\partial s/\partial y$ at $x = x_1, x_2, \ldots, x_m$, and for $\partial^2 s/\partial x \partial y$ at $x = x_m$ are given by

$$-C_{m+2}b_0 + C_{m+2}b_2 = p_0, \tag{16}$$

where p_0 is the (appropriately scaled) vector of (corresponding) extrapolated first and second partial derivative values given by (10), (12), and (14),

$$p_0 = \left(\frac{16hk}{9} s_{xy}(1, 1), \frac{16k}{3} s_y(1, 1), \frac{16k}{3} s_y(2, 1), \ldots, \frac{16k}{3} s_y(m, 1), \frac{16hk}{9} s_{xy}(m, 1)\right)^T.$$

The spline equations for $\partial s/\partial x$ at $x = x_1, y = y_j$, for s at $x = x_i, y = y_j$, $i = 1, 2, \ldots, m$, and for $\partial s/\partial x$ at $x = x_m, y = y_j$ are given by

$$C_{m+2}b_{j-1} + 4C_{m+2}b_j + C_{m+2}b_{j+1} = q_j, \tag{17}$$

where

$$q_j = \left(\frac{16h}{3} s_x(1, j),\ 16f_{1,j},\ 16f_{2,j} \ldots,\ 16f_{m,j},\ \frac{16h}{3} s_x(m, j)\right)$$

as given by (4), (8), and (9).

Finally, the spline equations at $y = y_n$ for $\partial^2 s/\partial x \partial y$ at $x = x_1$, for $\partial s/\partial y$ at $x = x_1, x_2, \ldots, x_m$, and for $\partial^2 s/\partial x \partial y$ at $x = x_m$ are given by

$$-C_{m+2}b_{n-1} + C_{m+2}b_{n+1} = p_{n+1}, \tag{18}$$

where p_{n+1} is the (appropriately scaled) vector of (corresponding) extrapolated first and second partial derivative values given by (11), (13), and (15),

$$p_{n+1} = \left(\frac{16hk}{9} s_{xy}(1, n), \frac{16k}{3} s_y(1, n), \frac{16k}{3} s_y(2, n), \ldots, \frac{16k}{3} s_y(m, n), \frac{16hk}{9} s_{xy}(m, n)\right)^T.$$

10. Solution of the equation system. The system of equations $Sb = r$

derived above (in (16), (17), and (18) can be rewritten in banded partitioned matrix form as ($C = C_{m+2}$)

$$Sb = \begin{pmatrix} -C & 0 & C & & & & \\ C & 4C & C & & & & \\ & C & 4C & C & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot \\ & & & & C & 4C & C \\ & & & & & C \;\; 4C & C \\ & & & & & -C \;\; 0 & C \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ b_{n-1} \\ b_n \\ b_{n+1} \end{pmatrix} = \begin{pmatrix} p_0 \\ q_1 \\ q_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ q_{n-1} \\ q_n \\ q_{n+1} \end{pmatrix}.$$

The system matrix can be rewritten as

$$\begin{aligned} S = [N_{m+2}] \times \\ \operatorname{diag}([C_{m+2}]) = \end{aligned} \begin{pmatrix} [-1] & [0] & [1] & & & \\ [\;\;1] & [4] & [1] & & & \\ & [1] & [4] & [1] & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & [1] & [\;\;4] & [1] \\ & & & & [\;\;1] \;\; [0] & [1] \\ & & & & [-1] \;\; [0] & [1] \end{pmatrix} \begin{pmatrix} [C] & & & & & & & & \\ & [C] & & & & & & & \\ & & [C] & & & & & & \\ & & & \cdot & & & & & \\ & & & & \cdot & & & & \\ & & & & & \cdot & & & \\ & & & & & & [C] & & \\ & & & & & & & [C] & \\ & & & & & & & & [C] \end{pmatrix}$$

where each of the bracketed numbers in the left-hand band matrix $[N_{m+2}]$ above represents that multiple of an $(m + 2) \times (m + 2)$ identity matrix.

Let N_{m+2} be the scalar matrix corresponding to $[N]$, i.e., let N_{n+2} be the $(n + 2) \times (n + 2)$ matrix

$$N_{n+2} = \begin{pmatrix} 1 & 0 & 1 & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot \\ & & & & 1 & 4 & 1 \\ & & & & & 1 \;\; 4 & 1 \\ & & & & & -1 \;\; 0 & 1 \end{pmatrix}.$$

If we factor N_{n+2} into lower and upper triangular factors, say $N_{n+2} = L_{n+2} U_{n+2}$ with

$$L_{n+2} = \begin{pmatrix} 1 & & & & & & \\ \ell_{10} & 1 & & & & & \\ & \ell_{21} & 1 & & & & \\ & & \ell_{32} & 1 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \ell_{n-1,\, n-2} & 1 & & \\ & & & & \ell_{n,\, n-1} & 1 & \\ & & & & \ell_{n+1,\, n-1} & \ell_{n+1,\, n} & 1 \end{pmatrix}$$

and

$$U_{n+2} = \begin{pmatrix} \mu_{00} & 0 & \mu_{02} & & & & & \\ & \mu_{11} & \mu_{12} & & & & & \\ & & \mu_{22} & \mu_{23} & & & & \\ & & & \ddots & \ddots & & & \\ & & & & \mu_{n-1,n-1} & \mu_{n-1,n} & \\ & & & & & \mu_{n,n} & \mu_{n,n+1} \\ & & & & & & \mu_{n+1,n+1} \end{pmatrix}$$

and define matrices $[L_{n+2}]$ and $[U_{n+2}]$ by their partitioned matrix representations,

$$[L_{n+2} = \begin{pmatrix} [1] & & & & & & \\ [\ell_{10}] & [1] & & & & & \\ & [\ell_{21}] & [1] & & & & \\ & & [\ell_{32}] & [1] & & & \\ & & & \ddots & \ddots & & \\ & & & & [\ell_{n-1,n-2}] & [1] & & \\ & & & & & [\ell_{n,n-1}] & [1] & \\ & & & & & [\ell_{n+1,n-1}] & [\ell_{n+1,n}] & [1] \end{pmatrix}$$

and

$$[U_{n+2}] = \begin{pmatrix} [\mu_{00}] & [0] & [\mu_{02}] & & & & \\ & [\mu_{11}] & [\mu_{12}] & & & & \\ & & [\mu_{22}] & [\mu_{23}] & & & \\ & & & \ddots & \ddots & & \\ & & & & [\mu_{n-1,n-1}] & [\mu_{n-1,n}] & \\ & & & & & [\mu_{n,n}] & [\mu_{n,n+1}] \\ & & & & & & [\mu_{n+1,n+1}] \end{pmatrix}$$

where each bracketed number represents the appropriate multiple of the $(m+2) \times (m+2)$ identity matrix, i.e., $[\ell_{10}] = \ell_{10} I_{m+2}$, etc, then the system of equations can be written $[L_{n+2}][U_{n+2}]\ \mathrm{diag}([C_{n+2}])\, b = r$. Since all "elements" in $[U_{n+2}]$ are multiples of the $(m+2) \times (m+2)$ identity matrix, $[U_{n+2}]$ commutes with diag($[C]$), and $[L_{n+2}]\ \mathrm{diag}([C_{n+2}])\ [U_{n+2}]\, b = r$. But the matrix C_{m+2} is precisely equal to N_{m+2}, so we have immediately

$$[L_{n+2}]\ \mathrm{diag}([L_{m+2}])\ \mathrm{diag}([U_{m+2}])[U_{n+2}]\, b = r.$$

We can now solve for the vector b by solving the succession of problems

(a) Solve $[L_{n+2}]\, b_1 = r$ by forward block substitution.

(b) Solve $\operatorname{diag}([L_{m+2}])b_2 = b_1$ by forward substitution within each block.
(c) Solve $\operatorname{diag}([U_{m+2}])\, b_3 = b_2$ by backward substitution within each block.
(d) Solve $[U_{n+2}]\, b = b_3$ by backward block substitution.

Note: It is clear that because of the band matrix formulation of the system problem, all solution and substitution operations are proportional to the number of extended grid points, i.e., to $(m + 2) \times (n + 2)$. This presents an extremely efficient determination of the B-spline surface fit. In addition, when the number of data points is identical in the two coordinate directions, only one LU factorization must be performed.

11. Evaluation of the surface fit. To evaluate the spline determined above at any point (x, y) with $x_1 \leqq x < x_m$, $y_1 \leqq y < y_n$, let u, v be integers such that $x_u \leqq x < x_{u+1}$ and $y_v \leqq x < y_{v+1}$. Then

$$\begin{aligned} s(x, y) &= \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} a_{ij}\, B\left(\frac{x - x_i}{\Delta_x}\right) B\left(\frac{y - y_i}{\Delta_y}\right) \\ &= \sum_{i=u-1}^{u+2} \sum_{j=v-1}^{v+2} a_{ij}\, B\left(\frac{x - x_i}{\Delta_x}\right) B\left(\frac{y - y_j}{x_y}\right) \end{aligned}$$

or a total of 16 terms are required for this sum. If $x = x_u + \delta_x \Delta_x$, $\delta_x < 1$, $y = y_v + \delta_y \Delta_y$, $\delta_y < 1$, then

$$\begin{aligned} s(x, y) = {} & (a_{u-1,v-1}B(1 + \delta_x) + a_{u,v-1}B(\delta_x) + a_{u+1,v-1}B(\delta_x - 1) \\ & + a_{u+2,v-1}B(\delta_x - 2))B(1 + \delta_y) + (a_{u-1,v}B(1 + \delta_x) \\ & + a_{u,v}B(\delta_x) + a_{u+1,v}B(\delta_x - 1) + a_{u+2,v}B(\delta_x - 2))B(\delta_y) \\ & + (a_{u-1,v+1}B(1 + \delta_x) + a_{u,v+1}B(\delta_x) + a_{u+1,v+1}B(\delta_x - 1) \\ & + a_{u+2,v+1}B(\delta_x - 2))B(\delta_y - 1) + (a_{u-1,v+2}B(1 + \delta_x) \\ & + a_{u,v+2}B(\delta_x) + a_{u+1,v+2}B(\delta_x - 1) \\ & + a_{u+2,v+2}B(\delta_x - 2))B(\delta_y - 2). \end{aligned}$$

In fact, for $0 \leqq \delta \leqq 1$ $B(1 + \delta) = (1/4)(1 - \delta)^3$, $B(\delta) = 1 - (3/2)\delta^2 + (3/4)\delta^3$, $B(1 - \delta) = B(\delta - 1) = 1 - (3/2)(1 - \delta)^2 + (3/4)(1 - \delta)^3$, $B(2 - \delta) = B(\delta - 2) = (1/4)\delta^3$.

12. Remarks. The B-spline formulation of surface fitting to data defined on a uniform rectangular grid has been derived. The user-transparent edge conditions incorporated in the fit have been derived as a natural generalization of the one-dimensional results.

The development has been pursued to its conclusion for the uniform spacing case but the important elements, viz., the existence of natural extrapolation procedures, the uniqueness of the corner twist predictions due to the linearity of the extrapolation, and the role of the extended basis set, are independent of the grid regularity.

13. Appendix: B-spline curve fitting with user-transparent end conditions. We consider the case of equally spaced data. The cubic B-spline with compact support on $[-2, 2]$ is the C^2 twice continuously differentiable function $B(x)$, defined by

$$B(x) = \begin{cases} -(1/4)(x-2)^3 & , \quad 1 \leqq x \leqq 2 \quad , \\ 1-(3/2)x^2 + (3/4)x^3 & , \quad 0 \leqq x \leqq 1 \quad , \\ 1-(3/2)x^2 - (3/4)x^3 & , \quad -1 \leqq x \leqq 0 \quad , \\ (1/4)(x+2)^3 & , \quad -2 \leqq x \leqq -1 , \\ 0 & , \quad \text{otherwise.} \end{cases}$$

We note that

$$\begin{aligned} B(0) &= 1, & B(\pm 1) &= 1/4, & B(\pm n) &= 0, \quad n > 1, \\ B'(0) &= 0, & B'(\pm 1) &= \mp 3/4, & B'(\pm n) &= 0, \quad n > 1, \\ B''(0) &= -3, & B''(\pm 1) &= 3/2, & B''(\pm n) &= 0, \quad n > 1, \end{aligned}$$

for all integers n.

One-dimensional curve fitting. Let $y(x_i)$, $i = 1, 2, \ldots, m$ be given data points. We determine the cubic spline which interpolates to these data and given end conditions, using the extended basis set $B(x - x_i)/h$, $i = 0, 1, 2, \ldots, m + 1$.

If we write $(x_i = x_0 + ih, \quad i = 0,1, \ldots, m + 1)$ $s(x) = \sum_{i=0}^{m+1} a_i B(x - x_i)/h$, then

$$\begin{aligned} s(x_r) &= \sum_{i=0}^{m+1} a_i B\left(\frac{x_r - x_i}{h}\right) \\ &= \sum_{i=0}^{m+1} a_i B(r - i) \\ &= (1/4)a_{r-1} + a_r + (1/4)a_{r+1}, \text{ if } 1 \leqq r \leqq m. \end{aligned}$$

Also $s'(x) = \sum_{i=0}^{m+1}(1/h)a_i B'(x - x_i)/h$, so

$$\begin{aligned} s'(x_r) &= \frac{1}{h}\sum_{i=0}^{m+1} a_i\, B'\left(\frac{x_r - x_i}{h}\right) \\ &= \frac{1}{h}\sum_{i=0}^{m+1} a_i\, B'(r - i) \\ &= \frac{1}{h}\left[-\frac{3}{4}a_{r-1} + \frac{3}{4}a_{r+1}\right], \end{aligned}$$

but then the cubic spline with user-transparent end conditions is given by the solution of the tridiagonal system,

$$\begin{pmatrix} -1 & 0 & 1 & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & \cdot & \cdot & & & \\ & & & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 4 & 1 \\ & & & & & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_{m-1} \\ a_m \\ a_{m+1} \end{pmatrix} = \begin{pmatrix} \frac{4h}{3}\bar{s}(x_1) \\ 4f(x_1) \\ 4f(x_2) \\ \cdot \\ \cdot \\ \cdot \\ 4f(x_{m-1}) \\ 4f(x_m) \\ \frac{4h}{3}\bar{s}(x_m) \end{pmatrix}$$

where $\bar{s}(x_1)$, $\bar{s}(x_m)$ are given by the extrapolated slope end condition predictions, i.e.,

$$\bar{s}(x_1) = (1/8h)(-15f(x_1) + 25f(x_2) - 13f(x_3) + 3f(x_4))$$

and

$$\bar{s}(x_m) = (1/8h)(15f(x_m) - 25f(x_{m-1}) + 13f(x_{m-2}) - 3f(x_{m-3})).$$

Finally,

$$s''(x) = \sum_{i=0}^{m+1} \frac{1}{h^2} a_i B''\left(\frac{x - x_i}{h}\right)$$

so

$$\begin{aligned} s''(x_r) &= \frac{1}{h^2} \sum_{i=0}^{m+1} a_i B''\left(\frac{x_r - x_i}{h}\right) \\ &= \frac{1}{h^2} \sum_{i=0}^{m+1} a_i B''(r - i) \\ &= \frac{1}{h^2}\left[\frac{3}{2} a_{r-1} - 3a_r + \frac{3}{2} a_{r+1}\right], \text{ if } 1 \leqq r \leqq m, \end{aligned}$$

so that the cubic spline with "natural" end conditions ($s''(x) = 0$ at the ends) is given by the solution of the tridiagonal system

$$\begin{pmatrix} 1 & -2 & 1 & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & \cdot & \cdot & & & \\ & & & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \\ & & & & & \cdot & \cdot \\ & & & & 1 & 4 & 1 \\ & & & & & -2 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ a_m \\ a_{m+1} \end{pmatrix} = \begin{pmatrix} 0 \\ 4f(x_1) \\ 4f(x_2) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 4f(x_m) \\ 0 \end{pmatrix}.$$

(We note that by subtracting the first equation from the second equation, we obtain, for the natural spline, $6a_1 = 4f(x_1)$, or $a_1 = (2/3)f(x_1)$, and similarly, $a_m = (2/3)f(x_m)$.) Once the a_i values are determined, the spline $s(x)$ can be easily evaluated, using the fact that the function $B(x)$ is identically zero outside of the interval $[-2, 2]$. In fact, if $x_r \leqq x \leqq x_{r+1}$, with $1 \leqq r \leqq m$, let $\delta = (x - x_r)/h$; then

$$\begin{aligned} s(x) &= a_{r-1}B\left(\frac{x-x_{r-1}}{h}\right) + a_rB\left(\frac{x-x_r}{h}\right) + a_{r+1}B\left(\frac{x-x_{r+1}}{h}\right) + a_{r+2}B\left(\frac{x-x_{r+2}}{h}\right) \\ &= a_{r-1}B(1+\delta) + a_r B(\delta) + a_{r+1}B(\delta-1) + a_{r+2}B(\delta-2) \\ &= (1/4)a_{r-1}(1-\delta)^3 + a_r(1-3/2)\delta^2 + (3/4)\delta^3 + a_{r+1}(1-(3/2)(\delta-1)^2 \\ &\quad -(3/4)(\delta-1)^3 + (1/4)a_{r+2}\delta^3, \end{aligned}$$

which is a much simpler expression than the one obtained by use of the cardinal spline formulation.

References

1. J.H. Ahlberg, E.N. Nilson, and J.L. Walsh, *The Theory of Splines and Their Applications*, Academic Press, New York, 1967.
2. C.W deBoor, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.
3. ———, *Bicubic spline interpolation*, J. Math. Phys. **41** (1962), 212–218.
4. E.N. Nilson, *Cubic splines on uniform meshes*, CACM **13** (1970), 255–258.
5. L.H. Seitelman, *A New User-Transparent End Condition for Cubic Spline Data Fitting*, SIAM 1977 Fall Meeting, Albuguerque.
6. "*Natural cubic splines are unnatural*," in *Information Linkage between Applied Mathematics and Industry*, Wang, *et al*, ed., Academic Press, 1979.

UNITED TECHNOLOGIES,
PRATT AND WHITNEY AIRCRAFT, EAST HARTFORD, CT 06108