

NASA
Technical
Paper
2536

January 1986

NASA-TP-2536 19860008590

Two Algorithms for Rational Spline Interpolation of Surfaces

James R. Schiess

RECEIVED

JAN 14 1986

LANGLEY RESEARCH CENTER
LIBRARY, 1986
HAMPTON, VIRGINIA

NASA

NASA
Technical
Paper
2536

1986

3 1176 01342 9403

Two Algorithms for Rational Spline Interpolation of Surfaces

James R. Schiess

Langley Research Center
Hampton, Virginia

NASA

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

INTRODUCTION

The development of high-speed computers and sophisticated display devices has encouraged the development of advanced algorithms for manipulating and displaying multidimensional data. The area of computer-aided geometric modeling (CAGM), in particular, has advanced significantly during the past few years. In CAGM, computational geometry and computer graphics have combined to give mathematical and graphical representations of curves, surfaces, and volumes. A wide variety of applications may be found in the design of automobiles and aircraft and in the mathematical representation of various physical phenomena, such as geophysical maps and meteorological data.

In CAGM it is often desirable to interpolate three-dimensional surface data consisting of two independent variables (x and y) and one dependent variable. A number of techniques have been developed for surface interpolation, including Coons and Bézier patches and tensor products of Bézier curves, cubic splines, and B-splines (ref. 1). A difficulty can arise with these methods, especially the spline methods: abrupt changes in the dependent variable of the data may induce artificial or exaggerated hills and valleys in the interpolating surface.

One way to reduce or eliminate the unwanted hills and valleys in an interpolating spline surface is to apply tension to the surface. Applying mathematical tension to a spline is analogous to grasping the opposite edges of a membrane and stretching the membrane to remove wrinkles.

This paper discusses two algorithms for interpolating surfaces with spline functions containing tension parameters. Both algorithms are based on the tensor products of univariate rational splines (ref. 2). The simplest algorithm, which uses a single tension parameter for the entire surface, is the birational spline algorithm developed by Späth (ref. 2). This algorithm is generalized in this paper to use a separate tension parameter for each subinterval along both the x - and y -axes. The new algorithm allows for local control of tension on the interpolating surface. Both algorithms are illustrated and the results are compared with the results of bicubic spline and linear interpolation of terrain elevation data.

SYMBOLS

A_{ij}	4 by 4 matrix of coefficients
a_{ijkl}	coefficients of multivariate rational spline in the subregion R_{ij} ($k, l = 1, 2, 3, 4$)
CX_i	coefficients in equation for FX_{ij}
CY_j	coefficients in equation for FY_{ij}
c_{ik}	coefficients in univariate rational spline on interval i ($k = 1, 2, 3, 4$)
dx_i	difference between consecutive values of x equal to $x_{i+1} - x_i$

dy_j difference between consecutive values of y equal to $y_{j+1} - y_j$
 F_{ij} data value at (x_i, y_j)
 $F_{X_{ij}}$ partial derivative of $f_{ij}(x, y)$ with respect to x evaluated at (x_i, y_j)
 $F_{Y_{ij}}$ partial derivative of $f_{ij}(x, y)$ with respect to y evaluated at (x_i, y_j)
 $F_{XY_{ij}}$ partial derivative of $f_{ij}(x, y)$ with respect to x and y evaluated at (x_i, y_j)
 $f_{ij}(x, y)$ rational spline on the subregion R_{ij}
 G_i 4 by 4 matrix of functions $g_{ik}(x)$
 $g_{ik}(x)$ functions of x used in rational spline representation on interval i ($k = 1, 2, 3, 4$)
 H_j 4 by 4 matrix of functions $h_{j\ell}(y)$
 $h_{j\ell}(y)$ functions of y used in rational spline representation on interval j ($\ell = 1, 2, 3, 4$)
 m number of data points along x -axis
 n number of data points along y -axis
 P tension factor for entire surface
 P_i tension factor for interval i
 Q_j tension factor for interval j
 R_{ij} rectangular subregion in xy -plane defined by $x_i \leq x < x_{i+1}$ and $y_j \leq y < y_{j+1}$
 S_{ij} 4 by 4 matrix of function and derivative values
 r, s, t, u variables used to define rational spline
 x independent variable
 y independent variable
 z dependent variable in univariate rational spline
 Δ_x, Δ_y differences with respect to x and y variables, respectively

Subscripts:

i index along x -axis
 j index along y -axis
 k, ℓ general indices

Superscripts:

T matrix transpose

-1 matrix inverse

A prime indicates first derivative with respect to the independent variable. A double prime indicates second derivative with respect to the independent variable.

RATIONAL SPLINES

In this section both the univariate and the bivariate rational splines are described. Because the bivariate rational spline is the tensor product of two univariate rational splines, the characteristics of the univariate rational spline are discussed first.

Univariate Rational Spline

Let a given set of n data points be represented by (x_i, z_i) , where $i = 1, 2, \dots, n$ and $x_1 < x_2 < \dots < x_n$. The values of the independent variable x_i need not be equally spaced. The rational spline on interval i ($i = 1, 2, \dots, n-1$) is defined in references 2 and 3 to be

$$z = \sum_{k=1}^4 c_{ik} g_{ik}(x) \quad (x_i < x < x_{i+1}) \quad (1)$$

where c_{ik} are unknown coefficients, p_i is the tension parameter for interval i , and

$$g_{i1}(x) = u \quad g_{i3}(x) = \frac{u^3}{p_i t + 1}$$

$$g_{i2}(x) = t \quad g_{i4}(x) = \frac{t^3}{p_i u + 1}$$

where

$$u = \frac{x_{i+1} - x}{dx_i}$$

$$t = \frac{x - x_i}{dx_i} = 1 - u$$

$$dx_i = x_{i+1} - x_i$$

Equation (1) is defined for all values of the independent variable x in the data range if the tension parameter p_i is restricted to $p_i > -1$. If p_i is set to zero, equation (1) reduces to a cubic spline function. As p_i increases from zero, the cubic terms decrease in magnitude and the function tends to the equation of the line joining the knots at x_i and x_{i+1} . Because a distinct, independent tension parameter is associated with each interval, the behavior of the function in each interval may be locally controlled.

Evaluation of equation (1) for each subinterval requires knowledge of the four coefficients c_{i1} , c_{i2} , c_{i3} , and c_{i4} . Thus, for n data points (equivalently, for $n - 1$ subintervals), $4n - 4$ coefficients must be determined. Späth (ref. 2) reduces the magnitude of this problem by writing the coefficients in terms of the values of the function and first derivative at the knots. End conditions are applied to the first derivative, and equations ensuring the continuity of the second derivative at the interior knots are derived. This derivation yields a system of $n - 2$ equations for the $n - 2$ unknown interior first derivatives. Frost and Kinzel (ref. 3) extend Späth's approach by allowing for three different end conditions and by developing an iterative method for determining the tension parameters. Tension parameters are found so that the rational spline deviates from the line joining knots by no more than a prescribed value.

Another approach to determining a good fit to the data was taken by Schiess and Kerr (ref. 4) in deriving a least-squares rational spline approximation. The rational spline is reformulated in terms of the unknown spline function and its second derivative at the knots. The conditions imposed at the interior knots are that the first derivatives be continuous. This leads to a constrained least-squares problem in the $2n$ values of the unknown function and its second derivative at the knots.

Bivariate Rational Spline

Let a given set of m by n data points in three dimensions be represented by (x_i, y_j, F_{ij}) , where $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$. It is assumed that the independent variables are ordered ($x_1 < x_2 < \dots < x_n$ and $y_1 < y_2 < \dots < y_m$) and form a rectangular grid, but are not necessarily equally spaced. An interpolating surface through the given points is desired.

The bivariate rational spline on the subregion R_{ij} defined by $x_i < x < x_{i+1}$ ($i = 1, 2, \dots, n-1$) and $y_j < y < y_{j+1}$ ($j = 1, 2, \dots, m-1$) is defined in reference 2 by

$$f_{ij}(x, y) = \sum_{k=1}^4 \sum_{\ell=1}^4 a_{ijk\ell} g_{ik}(x) h_{j\ell}(y) \quad (2)$$

where $g_{ik}(x)$ and p_i are the same as for the univariate spline, $a_{ijk\ell}$ are unknown coefficients, q_j is the tension parameter for interval j , and

$$h_{j1}(y) = s \quad h_{j3}(y) = \frac{s^3}{q_j r + 1}$$

$$h_{j2}(y) = r \quad h_{j4}(y) = \frac{r^3}{q_j s + 1}$$

where

$$s = \frac{Y_{j+1} - Y}{dy_j}$$

$$r = \frac{Y - Y_j}{dy_j} = 1 - s$$

$$dy_j = Y_{j+1} - Y_j$$

As defined by equation (2), the bivariate rational spline on each subregion R_{ij} is a function of 2 tension parameters and 16 coefficients. The coefficients are determined so that the rational spline and its first and second derivatives are continuous over the entire region. The $m + n - 2$ tension parameters may be adjusted individually; each parameter affects the behavior of the rational spline in a strip parallel to either the x-axis (for q_j) or the y-axis (for p_i). In this paper, this form of rational spline is called the multiple-parameter rational spline.

Since 16 coefficients are needed on each subregion, a total of $16(m - 1)(n - 1)$ coefficients must be determined to define the entire rational spline. For example, for a 30 by 30 grid ($m = n = 30$), 13 456 coefficients are needed. Späth (ref. 2) reduces the actual number of unknown quantities by writing the coefficients as linear combinations of the values of the function and its derivatives at the grid points.

Let FX_{ij} and FY_{ij} be the first derivatives of $f_{ij}(x,y)$ with respect to x and y , respectively, and FX_{ij} the cross derivative, all evaluated at the point (x_i, y_j) . For the subregion R_{ij} , define the following 4 by 4 matrices:

$$S_{ij} = \begin{bmatrix} F_{ij} & FY_{ij} & F_{i(j+1)} & FY_{i(j+1)} \\ FX_{ij} & FXY_{ij} & FX_{i(j+1)} & FXY_{i(j+1)} \\ F_{(i+1)j} & FY_{(i+1)j} & F_{(i+1)(j+1)} & FY_{(i+1)(j+1)} \\ FX_{(i+1)j} & FXY_{(i+1)j} & FX_{(i+1)(j+1)} & FXY_{(i+1)(j+1)} \end{bmatrix}$$

$$A_{ij} = \begin{bmatrix} a_{ij11} & a_{ij12} & a_{ij13} & a_{ij14} \\ a_{ij21} & a_{ij22} & a_{ij23} & a_{ij24} \\ a_{ij31} & a_{ij32} & a_{ij33} & a_{ij34} \\ a_{ij41} & a_{ij42} & a_{ij43} & a_{ij44} \end{bmatrix}$$

$$G_i = \begin{bmatrix} g_{i1}(x_i) & g_{i2}(x_i) & g_{i3}(x_i) & g_{i4}(x_i) \\ g'_{i1}(x_i) & g'_{i2}(x_i) & g'_{i3}(x_i) & g'_{i4}(x_i) \\ g_{i1}(x_{i+1}) & g_{i2}(x_{i+1}) & g_{i3}(x_{i+1}) & g_{i4}(x_{i+1}) \\ g'_{i1}(x_{i+1}) & g'_{i2}(x_{i+1}) & g'_{i3}(x_{i+1}) & g'_{i4}(x_{i+1}) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 \\ -\frac{1}{dx_i} & \frac{1}{dx_i} & -\frac{3+p_i}{dx_i} & 0 \\ 0 & 1 & 0 & 1 \\ -\frac{1}{dx_i} & \frac{1}{dx_i} & 0 & \frac{3+p_i}{dx_i} \end{bmatrix}$$

$$H_j = \begin{bmatrix} h_{j1}(y_j) & h_{j2}(y_j) & h_{j3}(y_j) & h_{j4}(y_j) \\ h'_{j1}(y_j) & h'_{j2}(y_j) & h'_{j3}(y_j) & h'_{j4}(y_j) \\ h_{j1}(y_{j+1}) & h_{j2}(y_{j+1}) & h_{j3}(y_{j+1}) & h_{j4}(y_{j+1}) \\ h'_{j1}(y_{j+1}) & h'_{j2}(y_{j+1}) & h'_{j3}(y_{j+1}) & h'_{j4}(y_{j+1}) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 \\ -\frac{1}{dy_j} & \frac{1}{dy_j} & -\frac{3+q_j}{dy_j} & 0 \\ 0 & 1 & 0 & 1 \\ -\frac{1}{dy_j} & \frac{1}{dy_j} & 0 & \frac{3+q_j}{dy_j} \end{bmatrix}$$

Then in matrix notation,

$$S_{ij} = G_i A_{ij} H_j^T \quad (3)$$

Equation (3) can be verified by differentiating equation (2), as appropriate, and evaluating the results at the corners of the subregion. Note that G_i and H_j depend on the grid spacing and tension parameters but not on the function values.

Since the matrices G_i and H_j are nonsingular, equation (3) can be solved for the matrix of coefficients:

$$A_{ij} = G_i^{-1} S_{ij} (H_j^T)^{-1} \quad (4)$$

Therefore, for any subregion the 16 coefficients can be determined from the values of the function, its first derivatives with respect to x and y , and its cross derivatives at the four corners of the subregion. Therefore a total of $4mn$ function and derivative values are needed to calculate the coefficients. In addition to the given function values, the algorithms to be described require values for the derivatives FX_{ij} on the boundaries $x = x_1$ and $x = x_n$, for the derivatives FY_{ij} on the boundaries $y = y_1$ and $y = y_m$, and for the cross derivatives at the four corners of the region. This reduces the problem to one of solving for $3mn - 2(m + n) - 4$ derivative values. The magnitude of this problem is further reduced by decomposing it into four smaller subproblems, each of which consists of solving tridiagonal systems of equations.

RATIONAL SPLINE ALGORITHMS

In this section algorithms for finding the surface-interpolating rational spline in terms of the function and its derivatives are presented. The first algorithm presented is for the general case of $n - 1$ values of the tension parameters p_i ($i = 1, 2, \dots, n-1$) and $m - 1$ values of the tension parameters q_j ($j = 1, 2, \dots, m-1$). This is a new algorithm not presented elsewhere; the algorithm is derived in the appendix. The second algorithm is for the special case of a single tension parameter P for the entire surface. This algorithm was originally developed by Späth (ref. 2).

Multiple-Tension-Parameter Algorithm

Let the data (x_i, y_j, F_{ij}) and tension parameters p_i and q_j (for $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$) be given. In addition, $2m + 2n + 4$ boundary conditions must be given: the derivatives with respect to x along the boundaries $x = x_1$ and $x = x_n$ (FX_{1j} and FX_{nj} , $j = 1, 2, \dots, m$), the derivatives with respect to y along the boundaries $y = y_1$ and $y = y_m$ (FY_{i1} and FY_{im} , $i = 1, 2, \dots, n$), and the cross derivatives at the corners (FXY_{11} , FXY_{1m} , FXY_{n1} , and FXY_{nm}).

The algorithm for finding the remaining $3mn - 2(m + n) - 4$ derivatives proceeds in four stages:

1. Find the derivative with respect to x at each interior point; for each value of j ($j = 1, 2, \dots, m$), solve for $n - 2$ values of FX_{ij} in the equations

$$\begin{aligned}
 & CX_{i-1} FX_{(i-1)j} + [(2 + p_{i-1})CX_{i-1} + (2 + p_i)CX_i]FX_{ij} + CX_i FX_{(i+1)j} \\
 & = \frac{3 + p_{i-1}}{dx_{i-1}} CX_{i-1} \Delta_x F_{(i-1)j} + \frac{3 + p_i}{dx_i} CX_i \Delta_x F_{ij} \quad (i = 2, 3, \dots, n-1) \quad (5)
 \end{aligned}$$

where

$$CX_i = \frac{p_i^2 + 3p_i + 3}{[(2 + p_i)^2 - 1]dx_i}$$

$$\Delta_x F_{ij} = F_{(i+1)j} - F_{ij}$$

2. Find the derivative with respect to y at each interior point; for each value of i ($i = 1, 2, \dots, n$), solve for $m - 2$ values of FY_{ij} in the equations

$$\begin{aligned}
 & CY_{j-1} FY_{i(j-1)} + [(2 + q_{j-1})CY_{j-1} + (2 + q_j)CY_j]FY_{ij} + CY_j FY_{i(j+1)} \\
 & = \frac{3 + q_{j-1}}{dy_{j-1}} CY_{j-1} \Delta_y F_{i(j-1)} + \frac{3 + q_j}{dy_j} CY_j \Delta_y F_{ij} \quad (j = 2, 3, \dots, m-1) \quad (6)
 \end{aligned}$$

where

$$CY_j = \frac{q_j^2 + 3q_j + 3}{[(2 + q_j)^2 - 1]dy_j}$$

$$\Delta_y F_{ij} = F_{i(j+1)} - F_{ij}$$

3. Find the cross derivatives along the boundaries $y = y_1$ and y_m ; for $j = 1$ and m , solve for $n - 2$ values of FX_{ij} in the equations

$$\begin{aligned}
 & CX_{i-1} FXY_{(i-1)j} + [(2 + p_{i-1})CX_{i-1} + (2 + p_i)CX_i] FXY_{ij} + CX_i FXY_{(i+1)j} \\
 &= \frac{3 + p_{i-1}}{dx_{i-1}} CX_{i-1} \Delta_x FY_{(i-1)j} + \frac{3 + p_i}{dx_i} CX_i \Delta_x FY_{ij} \quad (i = 2, 3, \dots, n-1) \quad (7)
 \end{aligned}$$

where

$$\Delta_x FY_{ij} = FY_{(i+1)j} - FY_{ij}$$

4. Find the cross derivatives at the interior points; for each value of i ($i = 1, 2, \dots, n$), solve for $m - 2$ values of FX_{ij} in the equations

$$\begin{aligned}
 & CY_{j-1} FXY_{i(j-1)} + [(2 + q_{j-1})CY_{j-1} + (2 + q_j)CY_j] FXY_{ij} + CY_j FXY_{i(j+1)} \\
 &= \frac{3 + q_{j-1}}{dy_{j-1}} CY_{j-1} \Delta_y FX_{i(j-1)} + \frac{3 + q_j}{dy_j} CY_j \Delta_y FX_{ij} \quad (j = 2, 3, \dots, m-1) \quad (8)
 \end{aligned}$$

where

$$\Delta_y FX_{ij} = FX_{i(j+1)} - FX_{ij}$$

Equations (5) to (8) have a similar form; in fact the coefficients of equations (5) and (6) are identical to those of equations (7) and (8), respectively. The similarity of the equations can be exploited to reduce computational requirements since only two distinct matrices must be inverted.

The final step of the procedure is the application of equation (4) to the derivatives in order to calculate the coefficients needed for equation (2). Depending on the availability of computer time and memory, the user may prefer to store the derivatives and to calculate coefficients only as they are needed.

Single-Tension-Parameter Algorithm

The algorithm requiring only one tension parameter value P uses the same data and derivative values that are required for the multiple-tension-parameter algorithm. However, the equations for this algorithm can be obtained from the equations for the preceding algorithm by setting all the tension parameters equal to P ($p_i = P$ for $i = 1, 2, \dots, n-1$ and $q_j = P$ for $j = 1, 2, \dots, m-1$). Using a single tension parameter simplifies the coefficients of equations (5) to (8) considerably.

The four stages of the algorithm are as follows:

1. Find the derivative with respect to x at each interior point; for each value of j ($j = 1, 2, \dots, m$), solve for $n - 2$ values of FX_{ij} in the equations

$$\begin{aligned} \frac{1}{dx_{i-1}} FX_{(i-1)j} + (2 + P) \left(\frac{1}{dx_{i-1}} + \frac{1}{dx_i} \right) FX_{ij} + \frac{1}{dx_i} FX_{(i+1)j} \\ = (3 + P) \left(\frac{\Delta_x F_{i-1j}}{dx_{i-1}^2} + \frac{\Delta_x F_{ij}}{dx_i^2} \right) \quad (i = 2, 3, \dots, n-1) \end{aligned} \quad (9)$$

2. Find the derivative with respect to y at each interior point; for each value of i ($i = 1, 2, \dots, n$), solve for $m - 2$ values of FY_{ij} in the equations

$$\begin{aligned} \frac{1}{dy_{j-1}} FY_{i(j-1)} + (2 + P) \left(\frac{1}{dy_{j-1}} + \frac{1}{dy_j} \right) FY_{ij} + \frac{1}{dy_j} FY_{i(j+1)} \\ = (3 + P) \left(\frac{\Delta_y F_{i(j-1)}}{dy_{j-1}^2} + \frac{\Delta_y F_{ij}}{dy_j^2} \right) \quad (j = 2, 3, \dots, m-1) \end{aligned} \quad (10)$$

3. Find the cross derivatives along the boundaries $y = y_1$ and y_m ; for $j = 1$ and m , solve for $n - 2$ values of FX_{ij} in the equations

$$\begin{aligned} \frac{1}{dx_{i-1}} FXY_{(i-1)j} + (2 + P) \left(\frac{1}{dx_{i-1}} + \frac{1}{dx_i} \right) FXY_{ij} + \frac{1}{dx_i} FXY_{(i+1)j} \\ = (3 + P) \left(\frac{\Delta_x F_{Y(i-1)j}}{dx_{i-1}^2} + \frac{\Delta_x F_{Yij}}{dx_i^2} \right) \quad (i = 2, 3, \dots, n-1) \end{aligned} \quad (11)$$

4. Find the cross derivatives at the interior points; for each value of i ($i = 1, 2, \dots, n$), solve for $m - 2$ values of FXY_{ij} in the equations

$$\frac{1}{dy_{j-1}} FXY_{i(j-1)} + (2 + P) \left(\frac{1}{dy_{j-1}} + \frac{1}{dy_j} \right) FXY_{ij} + \frac{1}{dy_j} FXY_{i(j+1)}$$

$$= (3 + P) \left(\frac{\Delta_y FX_{i(j-1)}}{dy_{j-1}^2} + \frac{\Delta_y FX_{ij}}{dy_j^2} \right) \quad (j = 2, 3, \dots, m-1) \quad (12)$$

Equations (9) to (12) are identical to equations (8.25) to (8.28), respectively, of reference 2. All the equations have the same general form and the coefficients of equations (9) and (10) are identical to the coefficients of equations (11) and (12), respectively.

Selection of Tension Parameters

Although an automated procedure for adjusting tension parameters is available for univariate rational splines (refs. 3 and 4), no such procedure has been devised for bivariate rational splines. Instead, it is recommended that the user visually examine three-dimensional or contour plots of the data and the rational spline interpolating surface and then use engineering judgment to select tension-parameter values. This approach is outlined here.

First the user should obtain a plot of the original data. This plot will show both general trends and local anomalies of the data. Second the user should compute the interpolating cubic spline surface by calculating the interpolating rational spline with all tension parameters set to zero. Comparison of the data and the cubic spline plots will indicate regions where the cubic spline exhibits undesirable or exaggerated hills and valleys. Using small to moderate tension-parameter values (say, from 1 to 10) for those regions, the user can recalculate an interpolating rational spline. In this way, after a few trials with different tension values, a rational spline surface can be obtained which the user considers representative of the data.

EXAMPLE

For this example a set of data was chosen to show the flexibility and capabilities of the interpolating rational spline. The data consist of terrain elevation measurements on a rectangular area of size 2600 by 2000 feet. The measurements were taken at 100-foot intervals in both the x- and the y-direction. Elevations were measured to an accuracy of 0.1 foot.

Figure 1 shows a surface plot of the terrain elevation data. Note the cliff along the line $x = 1800$ feet. Because of this cliff, extra measurements were taken at the base of the cliff along the line $x = 1795$ feet. Including this set of measurements, the size of the data grid is 28 by 21. The elevations range from 7.9 feet (at $x = 2300$ ft, $y = 100$ ft) to 42.6 feet (on the cliff at $x = 1800$ ft, $y = 800$ ft). The maximum rate of change in elevation also occurs at the point of maximum elevation, where the elevation increases 11.2 feet in a distance of 5 feet (from $x = 1795$ ft to $x = 1800$ ft).

It is desired to produce a finer grid by interpolating the data every 25 feet in both the x- and the y-direction. As a basis of comparison with the rational splines, a bilinear interpolation was first performed. The bilinear interpolation consists of a linear Lagrange interpolation in each of the x and y variables (ref. 5). Figure 2 shows a surface plot of the bilinear interpolation of the terrain data.

Computer software used to determine the interpolating bivariate splines in this study is based on the software published by Späth (ref. 2). For the single-parameter rational spline the FORTRAN code given in reference 2 was used. This code solves equations (9) to (12). For the multiple-parameter bivariate rational spline the Späth software was modified to solve the multiple-tension-parameter equations (eqs. (5) to (8)). Neither the original Späth software nor the modified software has been optimized to take advantage of the similarity in the respective sets of equations. Rather, each of the equations is re-solved as many times as are dictated by the algorithms. Timing comparisons were made on a Control Data Corporation CYBER 175 computer at Langley Research Center. For the example given here the required central processor times were 4.7 seconds for the bilinear interpolation, 9.2 seconds for the single-parameter rational spline, and 9.4 seconds for the multiple-parameter rational spline. Both rational spline algorithms therefore require approximately the same amount of central processor time. Times for both rational spline algorithms can be reduced considerably by taking advantage of the similarity of the equations solved.

Derivative information along the surface boundaries is needed for both rational spline algorithms. A simple way to estimate these derivatives, which is also the method used in the example, is to calculate one-sided finite differences. These differences are calculated from data points on the boundaries and immediately interior to the boundaries.

With the tension parameters set to zero in either bivariate rational spline function, a bicubic spline function results. Figure 3 illustrates the interpolating bicubic spline surface for the terrain data interpolated at 25-foot increments in each direction. For univariate data having a drastic change in slope, the cubic spline typically deviates extensively from the desired trend between data points (ref. 3). Figure 3 shows that the same phenomenon occurs with the bicubic spline. The interpolating bicubic spline has exaggerated the height of the cliff between data points to an elevation of approximately 75 feet (at $y = 925$ ft). At the same time another oscillation in the surface has created a valley and hill in front of the cliff. In some areas the bicubic spline surface (fig. 3) appears to be slightly more rounded with low hills where the bilinear surface (fig. 2) is essentially flat.

The single-parameter rational spline was applied first in order to reduce the artificial hills and valleys created by the bicubic spline. Recall that this type of rational spline is analogous to grasping the surface at all the edges and pulling to stretch the surface. Figure 4 shows the result of applying a tension of 40. Most of the exaggerated cliff height and the artificial valley and hill in front of the cliff have been eliminated. The surface in this figure closely resembles the original surface in figure 1.

The advantage of the multiple-parameter rational spline is that it allows local control of surface undulations through selective assignment of nonzero tension parameters. In this example it is especially desirable to eliminate surface oscillations in the neighborhood of the cliff. This can be accomplished by applying nonzero tension in the neighboring intervals parallel to the cliff. A tension value of 40

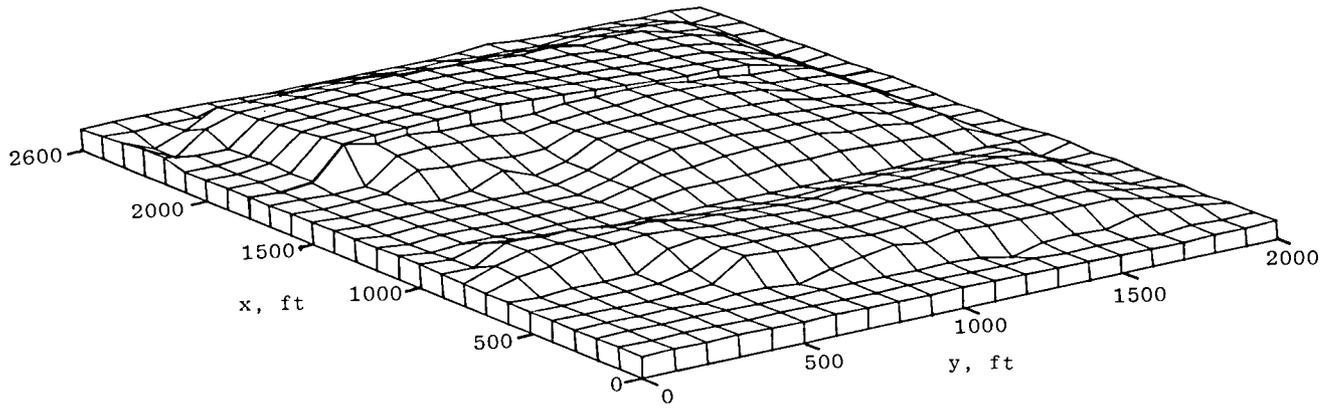


Figure 1.- Terrain elevation data.

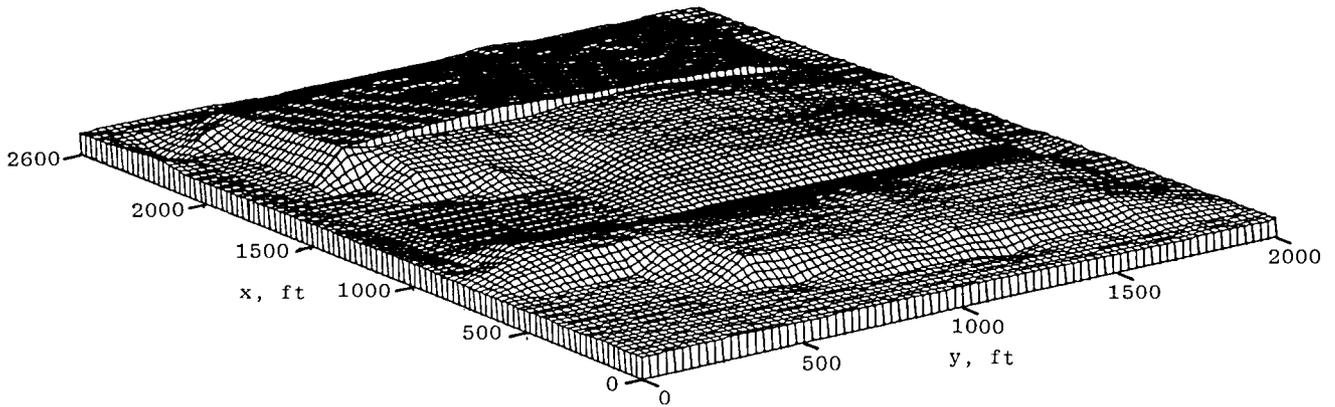


Figure 2.- Bilinear interpolation of terrain elevation data.

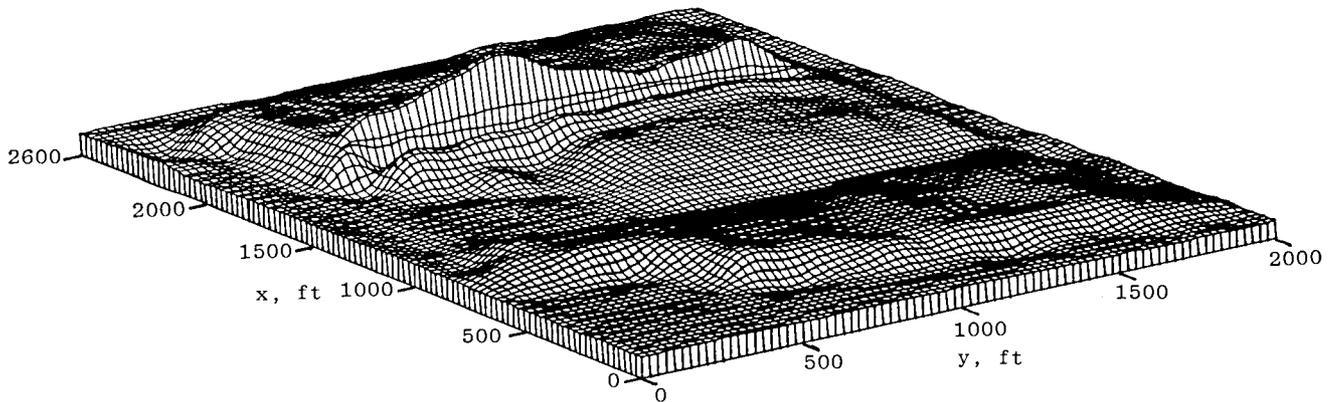


Figure 3.- Bicubic spline interpolation of terrain elevation data.

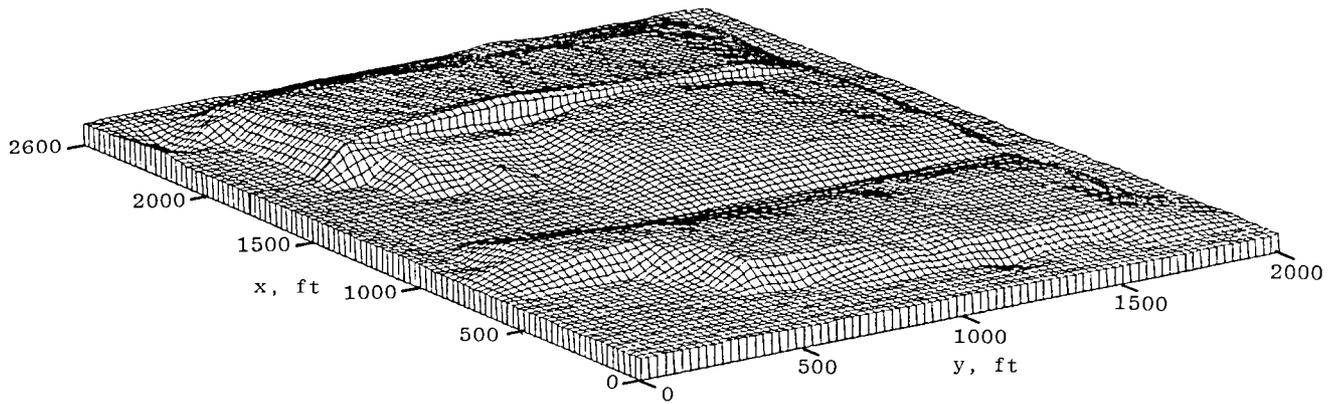


Figure 4.- Single-parameter rational spline interpolation of terrain elevation data with tension of 40.

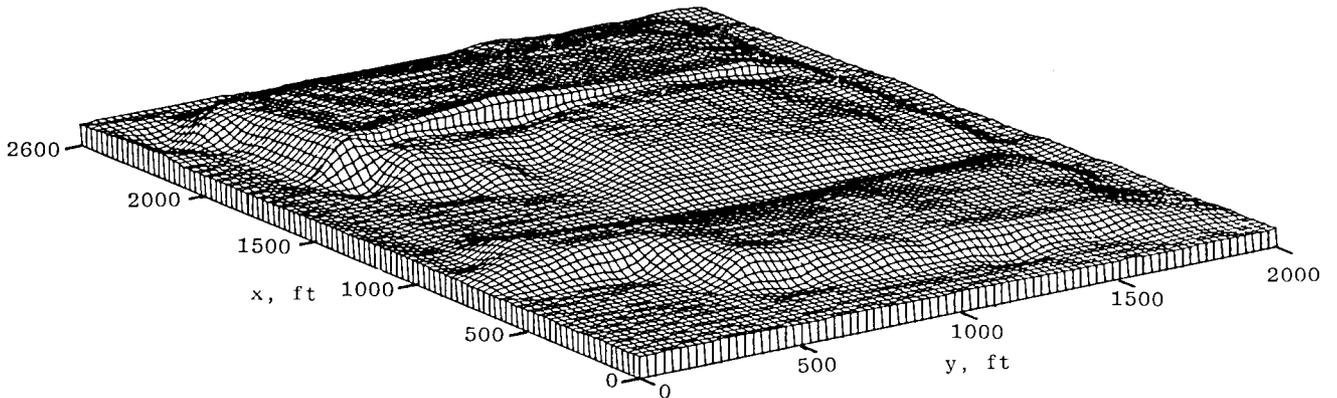


Figure 5.- Multiple-parameter rational spline interpolation of terrain elevation data with tension of 40 for $1700 \text{ ft} < x < 1900 \text{ ft}$; i.e., $P_{18} = P_{19} = P_{20} = 40$.

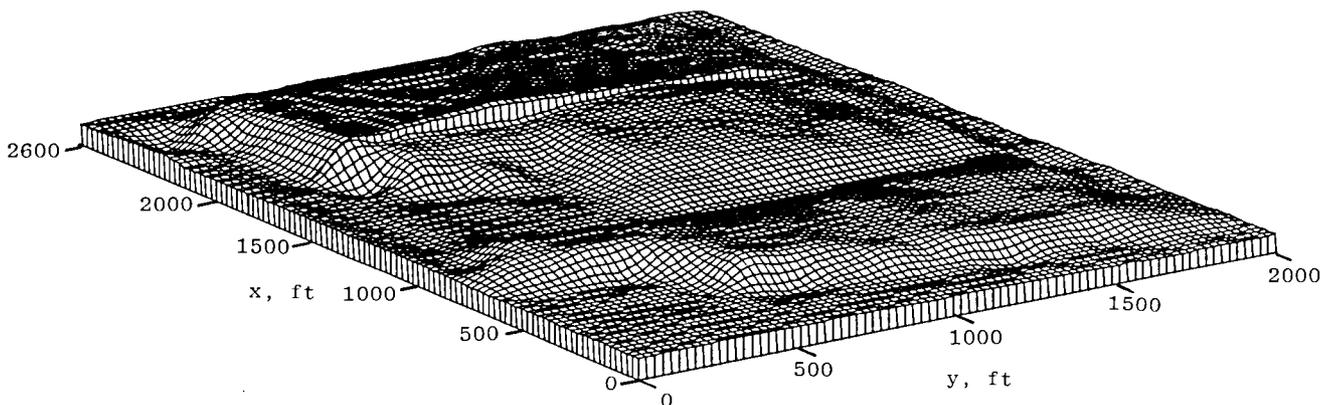


Figure 6.- Multiple-parameter rational spline interpolation of terrain elevation data with tension of 100 for $1700 \text{ ft} < x < 1900 \text{ ft}$; i.e., $P_{18} = P_{19} = P_{20} = 100$.

(which was used for the single-parameter rational spline) was assigned to parameters p_{18} , p_{19} , and p_{20} . The nonzero tension parameters are in the range $1700 \text{ ft} < x < 1900 \text{ ft}$, which is centered about the cliff ridge. All other tension parameters are zero. The result of this rational spline interpolation is shown in figure 5. This figure shows that the cliff elevation has been reduced considerably and the valley and hill in front of the cliff eliminated. The remaining areas of this surface are the same as the bicubic spline surface.

As a further test of the multiple-parameter rational spline, the values of the same three nonzero tension parameters were increased to 100. Figure 6 shows that the larger tension reduces the cliff height even further. Note, however, that most of the excessive height was reduced by increasing the tension from zero to 40. The result of the tension value of 100 is to produce an interpolating surface that is essentially linear along the x-direction near the cliff and cubic in x and y otherwise.

CONCLUDING REMARKS

Two algorithms for surface interpolation with bivariate rational spline functions on a rectangular grid have been presented. The rational spline functions combine the advantages of a cubic function having continuous first and second derivatives throughout the interpolatory region with the advantage of a function having variable tension. Adjustment of one tension parameter in the single-parameter rational spline allows the user to reduce unwanted oscillations to any desired extent across the entire surface. The multiple-parameter rational spline provides adjustable parameters for each rectangular subregion and thus gives the user control over local behavior of the interpolatory surface. A new algorithm for finding the coefficients of the multiple-parameter rational spline has been derived and presented.

The terrain elevation example presented illustrates the reduction in undesirable oscillations that is possible with a bivariate rational spline. The example demonstrates that the single-parameter rational spline can be adjusted to behave like any interpolatory surface ranging from a bilinear to a bicubic spline surface. The example also illustrates that the multiple-parameter rational spline provides the capability of controlling local oscillations in the surface caused by abrupt changes in trends in the data.

There are two major disadvantages to using the bivariate rational spline. First, the algorithms for finding the rational spline coefficients and interpolating points may require significantly more central processor time than does a bilinear interpolation. However optimizing the computer code for these algorithms can reduce the necessary central processor time. Second, several computer runs with different tension values may be necessary to find an interpolatory surface satisfactory to the user. The single-parameter rational spline in the terrain example required four computer runs to attain the results presented. These results, however, immediately led to the first results of the multiple-parameter rational spline interpolation. The second disadvantage can be overcome with experience in applying the rational splines. An alternative approach would be to apply an expert system to selection of the tension parameters.

APPENDIX

DERIVATION OF CONTINUITY EQUATIONS

In this appendix equations (5) to (8), which are solved for the unknown first derivatives and cross derivatives at interior grid points, are derived. All four sets of equations have comparable derivations.

Derivation of Equation (5)

Using the definition of the rational spline in equation (2), consider the interval $x_i < x < x_{i+1}$, fix j , and evaluate $f_{ij}(x,y)$ at y_j . Then $h_{j1}(y_j) = h_{j3}(y_j) = 1$, $h_{j2}(y_j) = h_{j4}(y_j) = 0$, and

$$f_{ij}(x,y_j) = \sum_{k=1}^4 g_{ik}(x)(a_{ijk1} + a_{ijk3}) \quad (A1)$$

For simplicity, define $b_{ijk} = a_{ijk1} + a_{ijk3}$. Then from equation (A1) and the definitions of $g_{ik}(x)$,

$$\left. \begin{aligned} f_{ij}(x_i, y_j) &= b_{ij1} + b_{ij3} = F_{ij} \\ f_{ij}(x_{i+1}, y_j) &= b_{ij2} + b_{ij4} = F_{(i+1)j} \end{aligned} \right\} \quad (A2)$$

where F_{ij} and $F_{(i+1)j}$ are known function values at the grid points.

The first derivatives of $g_{ik}(x)$ are

$$\left. \begin{aligned} g'_{i1}(x) &= -\frac{1}{dx_i} & g'_{i3}(x) &= \frac{-3u^2(p_i t + 1) - u^3 p_i}{dx_i (p_i t + 1)^2} \\ g'_{i2}(x) &= \frac{1}{dx_i} & g'_{i4}(x) &= \frac{3t^2(p_i u + 1) + t^3 p_i}{dx_i (p_i u + 1)^2} \end{aligned} \right\} \quad (A3)$$

Differentiating equation (A1) with respect to x , applying equations (A3), and evaluating at x_i and x_{i+1} yields

$$\left. \begin{aligned} \frac{\partial f_{ij}}{\partial x}(x_i, y_j) &= -\frac{b_{ij1}}{dx_i} + \frac{b_{ij2}}{dx_i} - \frac{3 + p_i}{dx_i} b_{ij3} = FX_{ij} \\ \frac{\partial f_{ij}}{\partial x}(x_{i+1}, y_j) &= -\frac{b_{ij1}}{dx_i} + \frac{b_{ij2}}{dx_i} + \frac{3 + p_i}{dx_i} b_{ij4} = FX_{(i+1)j} \end{aligned} \right\} \quad (A4)$$

where FX_{ij} and $FX_{(i+1)j}$ are the unknown first derivatives with respect to x at interior grid points.

The second derivatives of $g_{ik}(x)$ are

$$\left. \begin{aligned} g''_{i1}(x) &= g''_{i2}(x) = 0 \\ g''_{i3}(x) &= \frac{6u(p_i t + 1)^2 + 6u^2 p_i (p_i t + 1) + 2u^3 p_i^2}{dx_i^2 (p_i t + 1)^3} \\ g''_{i4}(x) &= \frac{6t(p_i u + 1)^2 + 6t^2 p_i (p_i u + 1) + 2t^3 p_i^2}{dx_i^2 (p_i u + 1)^3} \end{aligned} \right\} \quad (A5)$$

Differentiating equation (A1) twice with respect to x , applying equations (A5), and evaluating the results at x_i and x_{i+1} leads to

$$\left. \begin{aligned} \frac{\partial^2 f_{ij}}{\partial x^2}(x_i, y_j) &= \frac{2p_i^2 + 6p_i + 6}{dx_i^2} b_{ij3} \\ \frac{\partial^2 f_{ij}}{\partial x^2}(x_{i+1}, y_j) &= \frac{2p_i^2 + 6p_i + 6}{dx_i^2} b_{ij4} \end{aligned} \right\} \quad (A6)$$

Solve equation (A2) for b_{ij1} and b_{ij2} , substitute into equations (A4), and rearrange the results to obtain

$$-(2 + p_i)b_{ij3} - b_{ij4} = dx_i FX_{ij} + F_{ij} - F_{(i+1)j} \quad (A7)$$

$$b_{ij3} + (2 + p_i)b_{ij4} = dx_i FX_{(i+1)j} + F_{ij} - F_{(i+1)j} \quad (A8)$$

Define $\Delta_x F_{ij} = F_{(i+1)j} - F_{ij}$ and solve equation (A7) for b_{ij4} to get

$$b_{ij4} = -(2 + p_i)b_{ij3} - dx_i FX_{ij} + \Delta_x F_{ij} \quad (A9)$$

Substituting equation (A9) into equation (A8) and solving for b_{ij3} yields

$$b_{ij3} = \frac{(3 + p_i)\Delta_x F_{ij} - dx_i FX_{(i+1)j} - (2 + p_i)dx_i FX_{ij}}{(2 + p_i)^2 - 1} \quad (A10)$$

Substituting equation (A10) into equation (A9) gives

$$b_{ij4} = \frac{-(3 + p_i)\Delta_x F_{ij} + (2 + p_i)dx_i FX_{(i+1)j} + dx_i FX_{ij}}{(2 + p_i)^2 - 1} \quad (A11)$$

The continuity condition to be imposed at x_i is that the second derivative is continuous:

$$f''_{(i-1)j}(x_i, y_j) = f''_{ij}(x_i, y_j)$$

or from equations (A6),

$$\frac{p_{i-1}^2 + 3p_{i-1} + 3}{dx_{i-1}^2} b_{(i-1)j4} = \frac{p_i^2 + 3p_i + 3}{dx_i^2} b_{ij3}$$

Substituting equation (A10) and equation (A11) with i replaced by $i - 1$ and rearranging yields

$$\begin{aligned}
& \frac{p_{i-1}^2 + 3p_{i-1} + 3}{[(2 + p_{i-1})^2 - 1]dx_{i-1}} FX^{(i-1)j} + \left\{ \frac{(p_{i-1}^2 + 3p_{i-1} + 3)(2 + p_{i-1})}{[(2 + p_{i-1})^2 - 1]dx_{i-1}} \right. \\
& \left. + \frac{(p_i^2 + 3p_i + 3)(2 + p_i)}{[(2 + p_i)^2 - 1]dx_i} \right\} FX_{ij} + \frac{p_i^2 + 3p_i + 3}{[(2 + p_i)^2 - 1]dx_i} FX^{(i+1)j} \\
& = \frac{(p_{i-1}^2 + 3p_{i-1} + 3)(3 + p_{i-1})}{[(2 + p_{i-1})^2 - 1]dx_{i-1}^2} \Delta_x F^{(i-1)j} + \frac{(p_i^2 + 3p_i + 3)(3 + p_i)}{[(2 + p_i)^2 - 1]dx_i^2} \Delta_x F_{ij} \quad (A12)
\end{aligned}$$

Equation (A12) is identical to equation (5), which holds for $j = 1, 2, \dots, m$ and $i = 2, 3, \dots, n-1$.

Derivation of Equation (6)

The derivation of equation (6) is very similar to the derivation of equation (5). For this derivation, fix i and consider the interval $y_j < y < y_{j+1}$. Proceed through the same steps as before, but consider the first and second derivatives with respect to y of $f_{ij}(x_i, y)$ evaluated at y_j and y_{j+1} . Equations (5) and (6) have a similar form because the derivations are entirely analogous.

Derivation of Equation (7)

The derivations of equations (7) and (8) are very similar to the derivations of equations (5) and (6). The major difference is that the conditions imposed in this derivation and the next are the continuity of the mixed third-order derivatives at the grid points.

First consider the first derivative of equation (2) with respect to y evaluated at $y = y_j$, which will remain fixed. This derivative can be written

$$\begin{aligned}
 \frac{\partial f_{ij}(x, y_j)}{\partial y} &= \sum_{k=1}^4 \sum_{\ell=1}^4 a_{ijk\ell} g_{ik}(x) h'_{j\ell}(y_j) \\
 &= \sum_{k=1}^4 \left(\frac{-a_{ijk1}}{dy_j} + \frac{a_{ijk2}}{dy_j} - \frac{3 + q_j}{dy_j} a_{ijk3} \right) g_{ik}(x) \\
 &= \sum_{k=1}^4 d_{ijk} g_{ik}(x)
 \end{aligned} \tag{A13}$$

where d_{ijk} has been defined for convenience. Differentiate equation (A13) twice with respect to x and evaluate at x_i and x_{i+1} to get

$$\left. \begin{aligned}
 \frac{\partial^3 f_{ij}(x_i, y_j)}{\partial^2 x \partial y} &= d_{ij3} \frac{2p_i^2 + 6p_i + 6}{dx_i^2} \\
 \frac{\partial^3 f_{ij}(x_{i+1}, y_j)}{\partial^2 x \partial y} &= d_{ij4} \frac{2p_i^2 + 6p_i + 6}{dx_i^2}
 \end{aligned} \right\} \tag{A14}$$

The continuity condition to be fulfilled is that the third-order mixed derivatives are continuous at x_i :

$$\frac{\partial^3 f_{(i-1)j}(x_i, y_j)}{\partial^2 x \partial y} = \frac{\partial^3 f_{ij}(x_i, y_j)}{\partial^2 x \partial y}$$

or from equations (A14),

$$d_{(i-1)j4} \frac{p_{i-1}^2 + 3p_{i-1} + 3}{dx_{i-1}^2} = d_{ij3} \frac{p_i^2 + 3p_i + 3}{dx_i^2} \tag{A15}$$

Equation (A15) may be used to solve for FXY_{ij} ($i = 1, 2, \dots, m$). In order to do so, the derivative with respect to x of equation (A13) and equation (A13) itself must be evaluated at x_i and x_{i+1} . Evaluating equation (A13) at x_i and x_{i+1} gives

$$\left. \begin{aligned} \frac{\partial f_{ij}(x_i, y_j)}{\partial y} &= d_{ij1} + d_{ij3} = FY_{ij} \\ \frac{\partial f_{ij}(x_{i+1}, y_j)}{\partial y} &= d_{ij2} + d_{ij4} = FY_{(i+1)j} \end{aligned} \right\} \quad (A16)$$

Solving equations (A16) for d_{ij1} and d_{ij2} in terms of d_{ij3} , d_{ij4} , FY_{ij} , and $FY_{(i+1)j}$ yields

$$\left. \begin{aligned} d_{ij1} &= FY_{ij} - d_{ij3} \\ d_{ij2} &= FY_{(i+1)j} - d_{ij4} \end{aligned} \right\} \quad (A17)$$

Since equation (6) was previously solved for FY_{ij} ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$), these quantities are now known. Differentiating equation (A13) with respect to x and evaluating at x_i and x_{i+1} gives

$$\left. \begin{aligned} \frac{\partial^2 f_{ij}(x_i, y_j)}{\partial x \partial y} &= \frac{1}{dx_i} [-d_{ij1} + d_{ij2} - (3 + p_i)d_{ij3}] = FXY_{ij} \\ \frac{\partial^2 f_{ij}(x_{i+1}, y_j)}{\partial x \partial y} &= \frac{1}{dx_i} [-d_{ij1} + d_{ij2} + (3 + p_i)d_{ij4}] = FXY_{(i+1)j} \end{aligned} \right\} \quad (A18)$$

Substitute in equations (A18) for d_{ij1} and d_{ij2} from equations (A17) and rearrange to get

$$\left. \begin{aligned} -(2 + p_i)d_{ij3} - d_{ij4} &= dx_i FXY_{ij} + FY_{ij} - FY_{(i+1)j} \\ d_{ij3} + (2 + p_i)d_{ij4} &= dx_i FXY_{(i+1)j} + FY_{ij} - FY_{(i+1)j} \end{aligned} \right\} \quad (A19)$$

Using the notation $\Delta_x F_{Y_{ij}} = F_{Y_{(i+1)j}} - F_{Y_{ij}}$, equations (A19) can be solved for d_{ij3} and d_{ij4} to yield

$$\left. \begin{aligned} d_{ij3} &= \frac{(2 + p_i) dx_i F_{XY_{ij}} + dx_i F_{XY_{(i+1)j}} - (3 + p_i) \Delta_x F_{Y_{ij}}}{1 - (2 + p_i)^2} \\ d_{ij4} &= \frac{-dx_i F_{XY_{ij}} - (2 + p_i) dx_i F_{XY_{(i+1)j}} + (3 + p_i) \Delta_x F_{Y_{ij}}}{1 - (2 + p_i)^2} \end{aligned} \right\} \quad (A20)$$

Use equations (A20) to substitute for d_{ij3} and $d_{(i-1)j4}$ in equation (A15); then combine and rearrange terms so that the unknown cross derivatives are on the left side of the equation. The final result is

$$\begin{aligned} & \frac{p_{i-1}^2 + 3p_{i-1} + 3}{dx_{i-1} [1 - (2 + p_{i-1})^2]} F_{XY_{(i-1)j}} + \left\{ \frac{(p_{i-1}^2 + 3p_{i-1} + 3)(2 + p_{i-1})}{dx_{i-1} [1 - (2 + p_{i-1})^2]} \right. \\ & \left. + \frac{(p_i^2 + 3p_i + 3)(2 + p_i)}{dx_i [1 - (2 + p_i)^2]} \right\} F_{XY_{ij}} + \frac{p_i^2 + 3p_i + 3}{dx_i [1 - (2 + p_i)^2]} F_{XY_{(i+1)j}} \\ & = \frac{(p_{i-1}^2 + 3p_{i-1} + 3)(3 + p_{i-1})}{dx_{i-1}^2 [1 - (2 + p_{i-1})^2]} \Delta_x F_{Y_{(i-1)j}} + \frac{(p_i^2 + 3p_i + 3)(3 + p_i)}{dx_i^2 [1 - (2 + p_i)^2]} \Delta_x F_{Y_{ij}} \quad (A21) \end{aligned}$$

Equation (A21) is solved for $F_{XY_{ij}}$ for $i = 2, 3, \dots, n-1$ and for $j = 1$ and m ; this gives the cross derivatives along the boundaries $y = y_1$ and y_m .

Derivation of Equation (8)

Equation (8) is derived in a manner similar to the way equation (7) is derived. However for this derivation the continuity condition

$$\frac{\partial^3 f_{i(j-1)}}{\partial x \partial^2 y} (x_i, y_j) = \frac{\partial^3 f_{ij}}{\partial x \partial^2 y} (x_i, y_j)$$

is imposed. This is accomplished by fixing i and using an expression analogous to equation (A13) for the first derivative with respect to x . The derivation then proceeds comparably to the derivation of equation (7).

REFERENCES

1. Böhm, Wolfgang; Farin, Gerald; and Kahmann, Jürgen: A Survey of Curve and Surface Methods in CAGD. *Comput. Aided Geom. Design*, vol. 1, no. 1, July 1984, pp. 1-60.
2. Späth, Helmuth (W. D. Hoskins and H. W. Sager, transl.): *Spline Algorithms for Curves and Surfaces*. Utilitas Mathematica Publ. Inc., 1974.
3. Frost, Charles E.; and Kinzel, Gary L.: An Automatic Adjustment Procedure for Rational Splines. *Comput. & Graphics*, vol. 6, no. 4, 1982, pp. 171-176.
4. Schiess, James R.; and Kerr, Patricia A.: *Rational-Spline Approximation With Automatic Tension Adjustment*. NASA TP-2366, 1984.
5. Isaacson, Eugene; and Keller, Herbert Bishop: *Analysis of Numerical Methods*. John Wiley & Sons, Inc., c.1966.

1. Report No. NASA TP-2536		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Two Algorithms for Rational Spline Interpolation of Surfaces				5. Report Date January 1986	
				6. Performing Organization Code 505-31-83-02	
7. Author(s) James R. Schiess				8. Performing Organization Report No. L-16045	
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665-5225				10. Work Unit No.	
				11. Contract or Grant No.	
				13. Type of Report and Period Covered Technical Paper	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract This paper discusses two algorithms for interpolating surfaces with spline functions containing tension parameters. Both algorithms are based on the tensor products of univariate rational spline functions. The simpler algorithm uses a single tension parameter for the entire surface. This algorithm is generalized in this paper to use separate tension parameters for each rectangular subregion. The new algorithm allows for local control of tension on the interpolating surface. Both algorithms are illustrated and the results are compared with the results of bicubic spline and bilinear interpolation of terrain elevation data.					
17. Key Words (Suggested by Author(s)) Tension parameter Bicubic rational spline			18. Distribution Statement Unclassified - Unlimited Subject Category 64		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 25	22. Price A02

National Aeronautics and
Space Administration
Code NIT-4

Washington, D.C.
20546-0001

Official Business
Penalty for Private Use, \$300



**BULK RATE
POSTAGE & FEES PAID**
NASA Washington, DC
Permit No. G-27



If Undeliverable (Section 158
Postal Manual) Do Not Return

DO NOT REMOVE SLIP FROM MATERIAL

Delete your name from this slip when returning material to the library.

NAME	MS
Hemora, Chandra	261
Library	185
Tatum	413

NASA Langley (Rev. May 1988) RIAD N-75