ROCKY MOUNTAIN
JOURNAL OF MATHEMATICS
Volume 14, Number 1, Winter 1984

THREE-STAGE INTERPOLATION TO SCATTERED DATA

THOMAS A. FOLEY

1. Introduction. A general three-stage procedure is presented here that solves the following multivariate interpolation problem: Let D be a subset of $\mathbf{R}^n$ that contains N distinct point v_i. Given N real values $f(v_i)$, construct a function $P[f]$ defined on D that satisfies $P[f]\,(v_i) = f(v_i)$ for $i = 1, \ldots, N$.

In the bivariate case with $n = 2$, this problem can be interpreted as fitting a surface through N points in three-dimensional space. Foley [7] used bivariate interpolation in the characterization of radionuclide activity resulting from nuclear tests in Nevada. The survey paper by Schumaker [14] gives applications in mineral exploration, medicine, computer aided design, and electronics.

Part of the motivation for three-stage interpolation is that some methods that apply directly to scattered data give undesirable results or they are inefficient when N is large. On the other hand, many methods that are accurate and efficient only apply to gridded data, which is a narrower class of data.

Three-stage interpolation is related to the following approaches. Schumaker [13] gives a two-stage approximation to the scattered data that does not generally solve the interpolation problem. The Boolean sum approach in Barnhill and Gregory [3] obtains a desired precision and retains the interpolation properties. The implementation of the Barnhill-Gregory Boolean sums by Poeppelmeier [12] is discussed in Barnhill [2]. The author in [6] and Foley and Nielson [8] used delta sums and delta iteration interpolants composed of a bicubic spline approximation and a correction term using Shepard's method. It will be shown that some of these approaches also can be classified as three-stage methods.

The next section describes general three-stage interpolation. §3 gives a bivariate example named BSPLASH that is globally defined and has continuous second order partial derivatives. This method is applied to several data sets in the final section, and the results compare favorably with the best methods tested in Franke [9].

2. General three-stage interpolation. The general three-stage process is defined by

$$P[f] = BL[f] + S[f - BL[f]] \tag{1}$$

where L, B, and S are any operators that satisfy
(2) $L[f]$ is defined for those v so that $BL[f]$ can be formed,
(3) $BL[f]$ and $S[f - BL[f]]$ are defined on D,
(4) $S[g](v_i) = g(v_i)$ for $i = 1, \ldots, N$ for all g defined on D, and
(5) if $g(v_i) = 0$ for $i = 1, \ldots, N$, then $S[g](v) = 0$ for all v in D.

A bivariate example is given in the next section where S is a modified Shepard's interpolant to scattered data, B is a bicubic spline interpolant to gridded data, and L uses several quadratic least squares approximations to generate the gridded data for B. The author has tried several other choices for L, B, and S. S can be any interpolant to scattered data. B could be a bicubic Hermite, a Berstein polynomial, a Bezier surface, any quasi-interpolant, or even a transfinite method. L can be any approximation or interpolation method that applies directly to the scattered data, and $L[f]$ does not even have to be continuous for it to be effective.

Interpolation, continuity, precision, and error properties of $P[f]$ are now given in the following theorem.

THEOREM 1. *Given N distinct points v_i in $\mathbf{R}^n$,*
a) $P[f](v_i) = f(v_i)$ *for* $i = 1, \ldots, N$,
b) *if* $S[f] \in C^t$ *on* D *for all* f, *and* $B[f] \in C^s$ *on* D *for all* f, *then* $P[f] \in C^u$ *where* $u = \min(s, t)$.
c) *if* $B[f] = f$ *for all* $f \in G$ *and* $L[f] = f$ for all $f \in H$, *then* $P[f] = f$ *for all* $f \in G \cap H$, *and*
d) *if B is a linear operator and I is the identity, then*

$$f - P[f] = (I - S)[(f - B[f]) + B(f - L[f])]. \tag{6}$$

PROOF. S is not assumed to be linear, but if it were linear, then parts a) and c) follow from the Barnhill-Gregory theorem [3] because P is the Boolean sum of S and BL. $P[f]$ interpolates the scattered data because by (1) and (4)

$$\begin{aligned} P[f](v_i) &= BL[f](v_i) + S[f - BL[f]](v_i) \\ &= BL[f](v_i) + f(v_i) - BL[f](v_i) = f(v_i). \end{aligned}$$

Property b) holds because $P[f]$ is the sum of $B[g]$ and $S[h]$, where $g = L[f]$ and $h = f - BL[f]$. Property c) follows from (1) and (5), while d) holds because

$$\begin{aligned} f - P[f] &= f - BL[f] - S[f - BL[f]] \\ &= f - B[f] + B[f] - BL[f] - S(f - B[f] + B[f] - BL[f]) \\ &= (f - B[f]) + B(f - L[f]) - S[(f - B[f]) + B(f - L[f])]. \end{aligned}$$

If S is a linear operator, then Boolean sum and delta iteration interpolants can also be considered as three-stage interpolants. The Boolean sum $S \oplus L[f] = L[f] + S[f] - SL[f]$ is the three-stage interpolant (1) if B is the identity. Delta iteration is defined recursively by letting $\Delta_0 = S$ and

$$\Delta_{m+1}[f] = B\Delta_m[f] + S[f - B\Delta_m[f]].$$

Δ_0 is trivially a three-stage method, and if $L = \Delta_m$, then $\Delta_{m+1}[f] = P[f]$.

3. BSPLASH. This section presents a bivariate example of three-stage interpolation. In this context, $n = 2$, $v_i = (x_i, y_i)$, and $f(v_i) = z_i$. This implementation is called BSPLASH because the operator B produces a bicubic spline approximation, which is followed by a modified Shepard's interpolant S. This method has been applied to the data sets in Franke **[9]**, and some of these results are given in the next section.

To evaluate $L[f]$ at a point (x, y), find the seven nearest data points (x_i, y_i) to (x, y). A weighted least squares fit to the seven points (x_i, y_i, z_i) by a quadratic is formed, the weights being $((x - x_i)^2 + (y - y_i)^2)^{-1}$. $L[f](x, y)$ is the value of this quadratic evaluated at (x, y). If the least squares problem has many solutions, we will use one of lowest degree.

$L[f]$ is similar to the method is McLain **[11]**, but it uses only seven points instead of all the points. A function evaluation for McLain's method is time consuming, but evaluating $L[f]$ is much faster. Furthermore, $L[f]$ is only evaluated at the rectangular grid points used in the bicubic spline B. $L[f]$ is generally not continuous, but the continuity of $P[f]$ does not depend on $L[f]$ as seen in part b) of Theorem 1. What is important is that $L[f]$ gives a good approximation to f at the bicubic grid points because the error term (6) depends on $B(f - L[f])$.

The operator B is the natural bicubic spline that solves the gridded interpolation problem

$$B[g](XG_i, YG_j) = g(XG_i, YG_j) \qquad i = 1, \ldots, NXG, j = 1, \ldots, NYG.$$

See de Boor **[4]** for a detailed description. The bicubic grid points (XG_i, YG_j) can be input by the user or they can be computed by BSPLASH using an averaging process. Let $M = \text{IROUND}\ (\sqrt{N})$, $k = \text{I ROUND}\ (N/M)$, $NXG = M + 2$ and $NYG = M + 2$. Sort the x-coordinates into increasing order. Set XG_2 to the average of the first k x-coordinates, XG_3 to the average of the next k smallest x-coordinates, ..., and XG_{M+1} to the average of the k largest x-coordinates. Let $U = (XG_{M+1} - XG_2)/(M - 1)$. If the M grid points were equally spaced, then U would be the difference between two consecutive grid points. Set $XG_1 = x_1 - U$ and $XG_{NXG} = x_N + U$ so that all of the data points fall inside the grid. While the interior grid points are being computed, consecutive grid points are compared to see if their difference is between $U/2$ and $3*U$. If their difference is less than $U/2$, they are considered to be too close and they are averaged together thus reducing NXG by one. If their difference is greater than $3*U$, a new grid point is inserted at their midpoint and NXG is increased by one. The y-coordinates of the rectangular grid are defined in the same manner.

The operator S is the modified Shepard's method described in Foley [6].

$$S[f](x, y) = \frac{\sum_{i=1}^{N} f(x_i, y_i) \prod_{j \neq i} p_j(x, y)}{\sum_{i=1}^{N} \prod_{j \neq i} p_j(x, y)}$$

where $p_j(x, y) = d_j(r_j + d_j)/r_j$, $d_j = (x - x_j)^2 + (y - y_j)^2$, and r_j is the distance squared from (x_j, y_j) to its fifth nearest data point divided by four. $S[f]$ has continuous partial derivatives of all orders for all (x, y) and it satisfies (4) and (5). S has many interesting properties that are given in Foley [6] and in Gordon and Wixom [10]. This method gives results that are very similar to the localized Shepard's method used in Franke [9]. The primary difference between the two is that $S[f] \in C^\infty(R^2)$. It should be noted that S is not sensitive to the selection of r_j and that no square roots are needed. This interpolant is computationally fast, requires very little storage, and it is easily generalized to functions of several variables. Unfortunately, it is not visually smooth, nor is it very accurate.

The first stage of BSPLASH computes

$$ZG_{ij} = L[f](XG_i, YG_j) \qquad i = 1, \ldots, NXG, j = 1, \ldots, \text{NYG}.$$

The second stage forms the bicubic spline $BL[f]$ through the points (XG_i, YG_j, ZG_{ij}). This generally yields a smooth approximation to $f(x, y)$, but it does not interpolate the original scattered data (x_i, y_i, z_i). The final stage adds to $BL[f]$ the correction term $S[f - BL[f]]$ which uses S to interpolate the residuals $z_i - BL[f](x_i, y_i)$.

By Theorem 1, this interpolant has continuous second order partial derivatives for all (x, y) because $S[f] \in C^\infty$ and $B[f] \in C^2$.

4. Results. Figure 1 is a plot of the function

$$f_1(x, y) = .75 \exp\left(-\frac{(9x-2)^2 + (9y-2)^2}{4}\right) + .75 \exp\left(-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}\right)$$
$$+ .5 \exp\left(-\frac{(9x-7)^2 + (9y-3)^2}{4}\right) - .2 \exp(-(9x-4)^2 - (9y-7)^2).$$

Data sets were generated by evaluating this function at the $N = 100$, 33, and 25 (x, y) coordinates displayed in Figure 2. This function and data were used by Franke [9] in the comparison of several interpolants. The

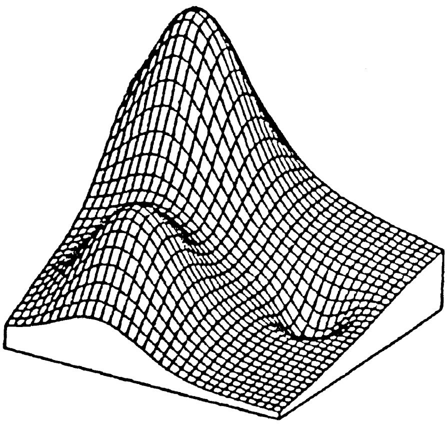

FIGURE 1. $f_1(x, y)$

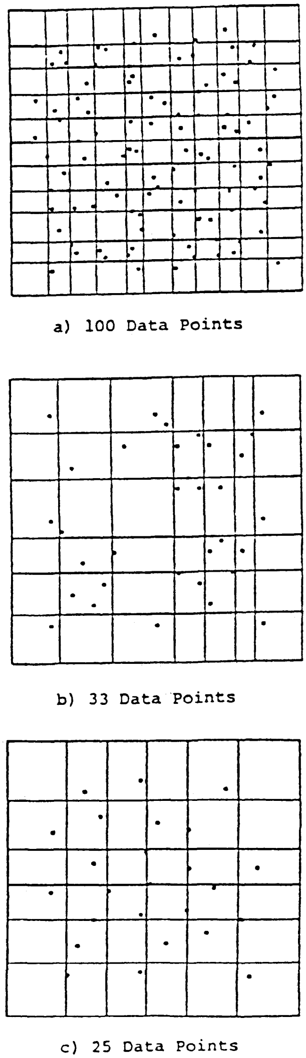

FIGURE 2. (x, y)
data points with grid lines

intersections of the orthogonal lines in Figure 2 are the bicubic grid points that were computed by BSPLASH. Figure 3 shows the results of BSPLASH applied to these three data sets. Table 1 gives the maximum absolute errors and the average absolute errors for these results as well as the errors for BSPLASH applied to the functions f_2, f_3, f_4 and f_5 defined in Franke

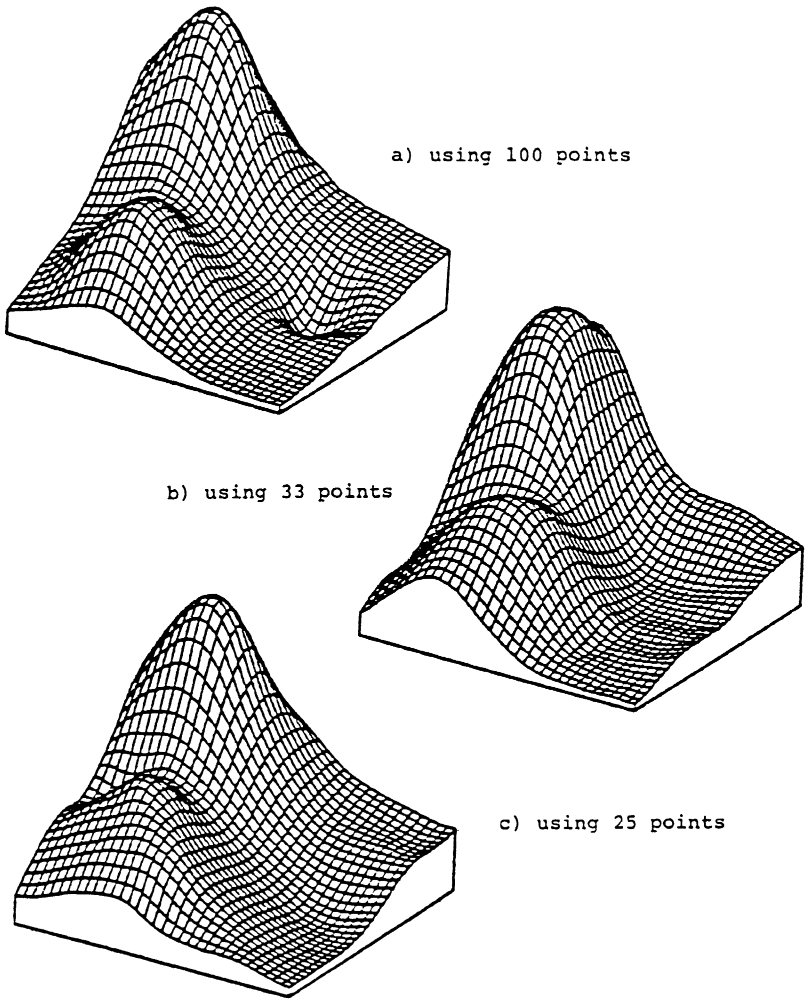

FIGURE 3. BSPLASH applied to $f_1(x, y)$

[9]. These discrete errors were computed using the differences at the 33 by 33 grid used to plot the surfaces. The errors and the visual smoothness of the plots compare favorably with the best methods tested in Franke [9].

Figures 4 & 5 show BSPLASH applied to the $N = 50$ data points in Akima [1] and the $N = 25$ points in Ferguson [5]. The function values were actually .5 more than those given by Ferguson, just as were those used by Franke. The evaluation bounds were also the same as those

TABLE 1

Discrete Errors Using BSPLASH

Function	Data set	Max Abs. Error	Mean Abs. Error
$f1$	100	.0443	.0060
$f1$	33	.2293	.0435
$f1$	25	.1220	.0277
$f2$	100	.0268	.0021
$f2$	33	.0493	.0090
$f2$	25	.0779	.0107
$f3$	100	.0195	.0010
$f3$	33	.0723	.0105
$f3$	25	.0397	.0065
$f4$	100	.0077	.0006
$f4$	33	.0319	.0047
$f4$	25	.0221	.0038
$f5$	100	.0265	.0016
$f5$	33	.1267	.0139
$f5$	25	.0402	.0066

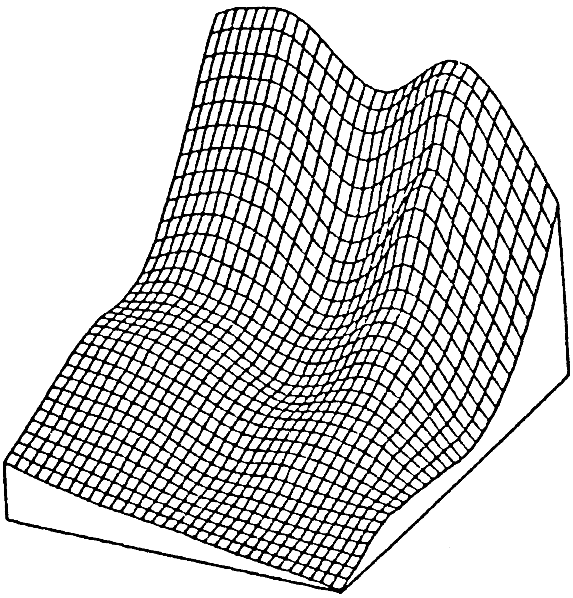

FIGURE 4. BSPLASH on Akima's data

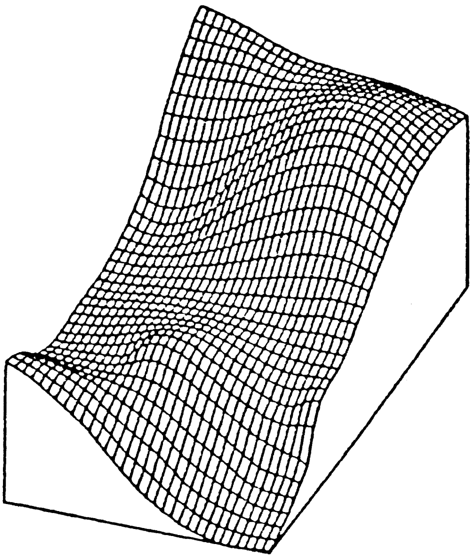

FIGURE 5. BSPLASH on Ferguson's data

used by Franke. The visual smoothness of these plots also compares favorably with the best methods tested in [**9**].

BSPLASH was efficient on large data sets. The execution times were tested on a Cyber 74/CDC6400 using default grid selections and evaluating the interpolant on a 33 by 33 grid. When $N = 25$, 50, 100, 200, 400, and 800 points were used, the execution time were 1.4, 2.4, 5.4, 12.4, 32.1, and 64.7 seconds respectively. These observed times are nearly linear in N.

With respect to storage requirements, most methods need to store the N data points (x_i, y_i, z_i) and an output array defining the surface. Other than that, approximately $3N$ locations are needed for storage.

Some additional comments are in order. In the cases tested by the author, the bicubic spline $BL[f]$ was the dominant part. That is, the correction term $S[f - BL[f]]$ was small relative to the magnitude of $BL[f]$. The interpolant was not sensitive to the grid selection on the data in Franke [**9**]. Extrapolation is not generally dependable, but in most of the cases tested, the interpolant behaved reasonably outside of the convex hull of the (x_i, y_i) points. Finally, this C^2 bivariate interpolant easily generalizes to functions of several variables because it depends on distances and tensor product methods, both of which are dimension-independent.

REFERENCES

1. H. Akima, *A method of bivariate interpolation and smooth surface fitting based on local procedures*, C. ACM **17** (1974), 18–20.

2. R. E. Barnhill, *Representation and approximation of surfaces*, Mathematical Software III, J. R. Rice, ed., Academic Press, New York, 1977, 69–120.

3. R. E. Barnhill and J. A. Gregory, *Compatible smooth interpolation in triangles*, J. Approx. Theory **15** (1975), 214–225.

4. C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.

5. J. C. Ferguson, *Multivariable curve interpolation*, J. ACM **11** (1964), 221–228.

6. T. A. Foley, *Off-site radiation exposure review project*: *Computer aided surface interpolation and graphical display*, DOE DP/01253–45021, Water Resources Center, Desert Research Institute, University of Nevada System, 1981.

7. T. A. Foley, *Surface interpolation using* DELTASH *and kriging*, TRAN-STAT: Statistics for Environmental Studies, Department of Energy Publication, to appear.

8. T. A. Foley and G. M. Nielson, *Multivariate interpolation to scattered data using delta iteration*, Approximation Theory III, E. W. Cheney, Ed., Academic Press, New York, 1980

9. R. Franke, *A critical comparison of some methods for interpolation of scattered data*, Technical Report No. NPS 53–79–003, Naval Postgraduate School, Monterey, CA 1979.

10. W. J. Gordon and J. A. Wixom, *Shepard's method of metric interpolation to bivariate and multivariate data*, Mathematics of Computation **32** (1978) 253–264.

11. D. H. McLain, *Drawing contours from arbitrary data points*, The Computer Journal **17** (1974), 318–324.

12. C. Poeppelmeier, *A boolean sum interpolation scheme to random data for computer aided geometric design*, M. S. thesis, Computer Science Department, University of Utah, 1975.

13. L. L. Schumaker, *Two-stage spline methods for fitting surfaces*, Approximation Theory Bonn, 1976, A. Dold and B. Eckmann, eds., Springer-Verlag, New York 1976, 378–389.

14. L. L. Schumaker, *Fitting surfaces to scattered data*, Approximation Theory II, G. G. Lorentz, C. K. Chui and L. L. Schumaker, eds., Academic Press, New York, 1976.

MATHEMATICAL SCIENCES DEPARTMENT, UNIVERSITY OF NEVADA, LAS VEGAS NV