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

ON ESTIMATING PARTIAL DERIVATIVES FOR BIVARIATE INTERPOLATION OF SCATTERED DATA

HIROSHI AKIMA

ABSTRACT. The method for estimating partial derivatives in bivariate interpolation and smooth surface fitting for scattered data previously developed by the author (ACM Trans. Math. Software, June 1978) and its slightly modified version (IMSL, Edition 8) are reviewed from both theoretical and practical viewpoints. Theoretical aspects considered include (a) correspondingly smooth changes of the estimated results with changes in data-point configurations and (b) invariance of the results under certain types of linear coordinate transformation. The IMSL-8 version runs faster, but the ACM version generally performs better. Some modifications of the method are considered and tested with examples.

1. Introduction. Bivariate interpolation of a C^1 continuous function of scattered data (or fitting a smooth surface to scattered data) is an active problem of interest. (The terms "bivariate interpolation" and "surface fitting" are used synonymously in this paper.) Some time ago, the author proposed a method of bivariate interpolation and smooth surface fitting [**1**], and its associated algorithm was designated as ACM Algorithm 526 [**2**]. A slightly modified version of the algorithm has been included in Edition 8 of the IMSL (International Mathematical and Statistical Libraries, Inc.) library [**3**] as the IQHSCV routine.

The method of bivariate interpolation consists of the following three major steps: Step 1, Triangulation of the x-y plane (i.e., dividing the plane into a number of triangles) using the max-min angle criterion (that dictates maximizing the minimum angle of triangles) described by Lawson [**4**]; Step 2, Estimation of first and second partial derivatives at each data point; and Step 3, Fitting of a fifth-degree polynomial in x and y in each triangle. Step 2, estimation of partial derivatives, further consists of the following two substeps: Substep 2.1, Estimation of first partial derivatives, z_x and z_y; and Substep 2.2, Estimation of second partial derivatives, z_{xx}, z_{xy}, and z_{yy}, as the first derivatives of z_x and z_y with the same procedure as used in Substep 2.1, i.e., $z_{xx} = (z_x)_x$, $z_{xy} = [(z_x)_y + (z_y)_x]/2$, $z_{yy} = (z_y)_y$.

Key words and phrases: bivariate interpolation, partial derivative, scattered data, smooth surface fitting.

The ACM-526 and IMSL-8 versions are different only in the method of estimating first derivatives used in Substeps 2.1 and 2.2.

This paper reviews the two versions of the method of estimating first partial derivatives from both the theoretical and practical viewpoints. It considers and tests some modifications of the procedure with examples.

2. Outline of the method of estimating first partial derivatives. The method for estimating first partial derivatives at data point P_0, used in ACM Algorithm 526, consists of several procedures. It selects n_c data points, P_i, $i = 1, 2, \ldots, n_c$, that are closest to P_0 in the $x - y$ plane among all given data points. It next calculates, for every combination of i and j, a vector product $V_{ij} = \overline{P_0P_i} \times \overline{P_0P_j}$, $i, j = 1, 2, \ldots, n_c$, where P_0, P_i, and P_j are arranged to be counterclockwise in the $x - y$ plane so that the z component of V_{ij} is positive. It next calculates the vector sum, V, of all V_{ij}'s. Finally, it estimates first derivatives, z_x and z_y, as the slopes of a plane that is normal to the vector sum, or as

$$z_x = -(x \text{ component of } V)/(z \text{ component of } V),$$
$$z_y = -(y \text{ component of } V)/(z \text{ component of } V).$$

The method used in the IMSL-8 version calculates the vector product on each triangle determined by the triangulation of the $x - y$ plane. Thus, it spares the selection of n_c data points closest to the data point in question.

3. Disadvantages. The method for estimating first derivatives used in ACM Algorithm 526 has some disadvantages. First, there is no basis for the user to select n_c, the number of points to be used for calculating the vector products. It is painful for the user to have to select a number without a basis. Second, since the method uses closest data points, it is inconsistent with the max-min angle criterion used in the triangulation of the $x - y$ plane. Third, the method requires additional computation time for selecting n_c closest data points.

The method used in the IMSL-8 version is designed to eliminate all these disadvantages of the ACM-526 version. Unfortunately, however, the use of the IMSL-8 version sometimes produces excessive undulations (wiggles) in the resulting surfaces.

In addition to the above disadvantages, there is another disadvantage that is common to the above two versions of the method. Both the ACM-526 and IMSL-8 versions of the method sometimes lack continuity in the change of the resulting surface with the change of the data, as exemplified by the following example. Consider a configuration of three data points, A, B, and C, where the projections in the $x - y$ plane of two data points, A and B, are on the convex hull of the data range, and the projection of another data point, C, is inside the convex hull and near the midpoint of the projection of line segment $\overline{AB}$. The projections of these three points form a triangle when the $x - y$ plane is triangulated. Consider further that the projection of C moves toward the projection of $\overline{AB}$. The contribution of the triangle (i.e., the vector product of $\overline{AB}$ and $\overline{AC}$) remains nonzero, unless the three points are collinear in the three-dimensional space, as long as the projection of C is not on the projection of $\overline{AB}$. When the projection of C is on the projection of $\overline{AB}$, however, the contribution is zero since triangle ABC does not exist any more. Thus, the fitted surface changes discontinuously against the changes in the data point configuration.

4. Theoretical aspects. The discontinuous behavior of the interpolated result (or the fitted surface) against the change in the data point configuration, just described above, is undesirable theoretically as well as in practice. This disadvantage must be corrected.

For some applications of a method of bivariate interpolation, invariance of the method under certain types of coordinate transformation is desired. (Invariance of the method under coordinate transformation here means the property that the first interpolated and then transformed result coincides with the first transformed and then interpolated result.) Both the ACM-526 and IMSL-8 versions of the method of bivariate interpolation including the method for estimating partial derivatives are invariant under the following types of linear coordinate transformation:

- rotation of the $x - y$ coordinate system;
- linear scale transformation of the z coordinate;
- tilting of the $x - y$ plane, i.e.,

$$x^{(2)} = x^{(1)},$$

$$y^{(2)} = y^{(1)},$$

$$z^{(2)} = z^{(1)} + ax^{(1)} + by^{(1)}, \text{ where } a \text{ and } b \text{ are arbitrary constants.}$$

In modifying the method, we intend to retain the desirable property of invariance under these types of linear coordinate transformation. This theoreical aspect imposes constraints on the modification of the method rather than providing guidelines for the modification. Retaining the property of invariance under these types of linear coordinate transformation dictates that geometric quantities such as the length or angle in the $x - y$ plane be used instead of the geometric quantities in the three-dimensional space.

5. Improvement. Examination of the method outlined in §2 reveals that far away points are given greater weights in estimating the partial derivatives. Examination of partial derivative values estimated by the IMSL-8 version has also indicated that poor estimation of partial derivatives usually occurs when a thin (or slim) triangle is involved. These observations suggest that a small weight be given to the contribution of a large triangle or a thin triangle (or the vector product of two vectors separated by a narrow angle) when the vector sum is calculated. In the rest of this paper, we examine whether or not the IMSL-8 version of the method for estimating partial derivatives can be improved by weighing the contribution of each triangle.

To get a small weight for a large triangle, we may wish to take the reciprocal of the product of the lengths (measured in the $x - y$ plane) of the two sides meeting at the projection of the data point in question as a possible weight for the triangle. We denote the reciprocal by w_1. To get a small weight for a triangle when the angle of the triangle at the data point in question is either small or close to π (or 180°), we may wish to take the sine value of the angle (measured in the $x - y$ plane) at the data point in question as a possible weight for the triangle. We denote the sine value by w_2. (We may wish to take the minimum of the three sine values at the three vertices as another possible weight, but the test results have indicated that the use of this weight has no advantage over the use of w_2. We will not consider this weight in the rest of this paper.) Combining the above two weights, w_1 and w_2, we may wish to take the product of the powers of w_1 and w_2 as the weight for the triangle; we consider the weight, w, expressed by $w = (w_1{**}n_1) * (w_2{**}n_2)$, where the symbol of single asterisk implies multiplication and the symbol of double asterisks implies exponentiation as in the Fortran language . (Obviously, modifications with these weights will reduce to the IMSL-8 version when one sets $n_1 = 0$ and $n_2 = 0$.)

When the triangle is getting thinner and reduces to a line segment as in the example discussed in §3 above, the second weight described here converges to zero and, therefore, the contribution of the triangle also tends to zero if the second weight is used. The discontinuous behaviors of both the ACM-526 and IMSL-8 versions discussed there will be eliminated by the use of the second weight.

We have done some sample calculations to test the effectiveness of the use of these weights. As the first example, we have generated 30 data points randomly in the $x - y$ plane and generated also randomly a set of coefficients of a second-degree polynomial in x and y. We have estimated the five partial derivatives at each data point and calculated the rms (root-mean-square) deviation for each of the partial derivatives over all the 30 data points. The result is shown in Table 1. The rms deviations resulting from the ACM-526 version generally decrease as the number of closest points increases in this example. The IMSL-8 version results in a large rms deviation for z_{xx}. Even a modified version that uses w_2 only (i.e., $n_1 = 0$) outperforms not only the IMSL-8 verision but also the ACM-526

Table 1. The rms deviations of the estimated partial derivatives, obtained from an example of 30 data points and a second-degree polynomial.

Method and Parameter		rms deviations for z_x	z_y	z_{xx}	z_{xy}	z_{yy}
ACM-526	$n_c = 3$	0.278	0.289	1.048	0.732	2.154
	4	0.264	0.144	0.722	0.669	0.570
	5	0.226	0.149	0.841	0.487	0.521
	6	0.175	0.110	0.500	0.136	0.270
	7	0.139	0.124	0.418	0.118	0.279
	8	0.148	0.132	0.413	0.097	0.263
IMSL-8 ($n_1 = 0$, $n_2 = 0$)		0.180	0.118	2.854	0.425	0.494
$n_1 = 0$	$n_2 = 1$	0.128	0.089	2.706	0.208	0.219
	2	0.112	0.090	1.270	0.105	0.227
	3	0.119	0.092	0.681	0.105	0.236
	4	0.125	0.093	0.579	0.113	0.249
$n_1 = 1$	$n_2 = 0$	0.072	0.049	1.148	0.431	0.492
	1	0.092	0.059	0.504	0.165	0.433
	2	0.106	0.064	0.393	0.099	0.418
$n_1 = 2$	$n_2 = 0$	0.053	0.035	0.239	0.081	0.171
	1	0.092	0.049	0.209	0.059	0.147
	2	0.112	0.056	0.220	0.065	0.174
$n_1 = 3$	$n_2 = 0$	0.068	0.046	1.356	0.399	0.463
	1	0.086	0.053	0.858	0.384	0.397
	2	0.109	0.059	0.346	0.213	0.383

version. A modified version that uses both w_1 and w_2 performs even better.

Table 2 shows the result of a similar calculation with a randomly generated third-degree polynomial. The same set of 30 data points is used. In this example, the rms deviations resulting from the ACM-526 version do not decrease as the number of closest points increases. The IMSL-8 version results in a large rms deviation particularly for z_{xx}. A modified version that uses w_2 only outperforms the IMSL-8 version and performs at least as good as the ACM-526 version. As in the previous example, a modified version that uses both w_1 and w_2 outperforms the ACM-526 version.

As the third example, we compare, with each other, six versions of the method: (A) ACM-526 version ($n_c = 5$), (B) IMSL-8 version, (C) modified version with $n_1 = 0$ and $n_2 = 2$, (D) modified version with $n_1 = 1$ and $n_2 = 1$, (E) modified version with $n_1 = 2$ and $n_2 = 1$, and (F) modified

Table 2. The rms deviations of the estimated partial derivatives, obtained from an example of 30 data points and a third-degree polynomial.

Method and Parameter		rms deviations for				
		z_x	z_y	z_{xx}	z_{xy}	z_{yy}
ACM-526	$n_c = 3$	0.420	0.461	5.637	2.963	8.264
	4	0.307	0.375	1.649	1.141	1.559
	5	0.466	0.438	1.553	0.979	1.461
	6	0.486	0.450	1.600	0.826	1.315
	7	0.398	0.414	1.127	0.703	1.171
	8	0.439	0.463	1.085	0.619	1.290
IMSL-8 ($n_1 = 0, n_2 = 0$)		1.305	0.540	21.106	2.640	2.612
$n_1 = 0$	$n_2 = 1$	0.766	0.357	17.822	1.063	1.496
	2	0.441	0.351	4.738	1.174	1.481
	3	0.418	0.351	0.994	1.342	1.528
	4	0.420	0.352	1.671	1.360	1.590
$n_1 = 1$	$n_2 = 1$	0.412	0.322	5.642	1.325	1.880
	1	0.386	0.275	0.806	0.993	1.882
	2	0.396	0.281	1.127	1.081	1.887
$n_1 = 2$	$n_2 = 0$	0.321	0.306	0.615	0.609	1.081
	1	0.376	0.255	0.685	0.511	1.208
	2	0.404	0.267	0.754	0.485	1.304
$n_1 = 3$	$n_2 = 0$	0.274	0.347	2.165	0.946	1.418
	1	0.355	0.267	1.424	0.730	1.257
	2	0.402	0.277	0.812	0.493	1.277

version with $n_1 = 3$ and $n_2 = 1$. A set of randomly generated 20 data points is used in this example. The data values at these data points are taken from the original surface that is a randomly generated second-degree polynomial. We have tried to reproduce the original surface by fitting surfaces to the data points and values with the above six versions of the method. The results are shown in Figures 1 and 2. Figure 1 depicts the contour maps of six surfaces. In each contour map in Figure 1, heavy contours are for the fitted surface, while light contours superimposed are for the original surface. (The difference between a heavy line and its corresponding light line represents the error.) As another way of looking at the performances of various versions of the method, Figure 2 depicts the contour maps of the differences between the interpolated (fitted) and expected (original) values. Figure 2 indicates that the modified version,

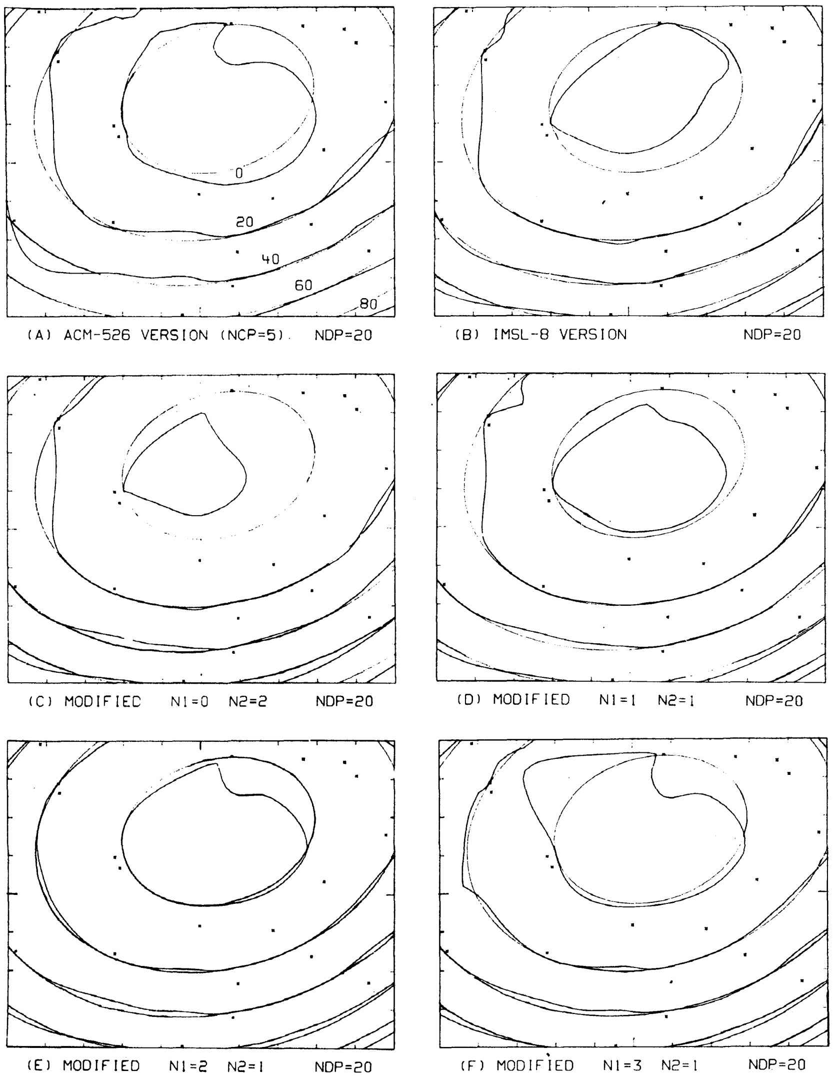

FIGURE 1. Contour maps of the surfaces fitted by six versions of the method to a set of 20 data points and a second-degree polynomial. (Heavy contours are for the fitted surfaces, and light contours are for the original surface.)

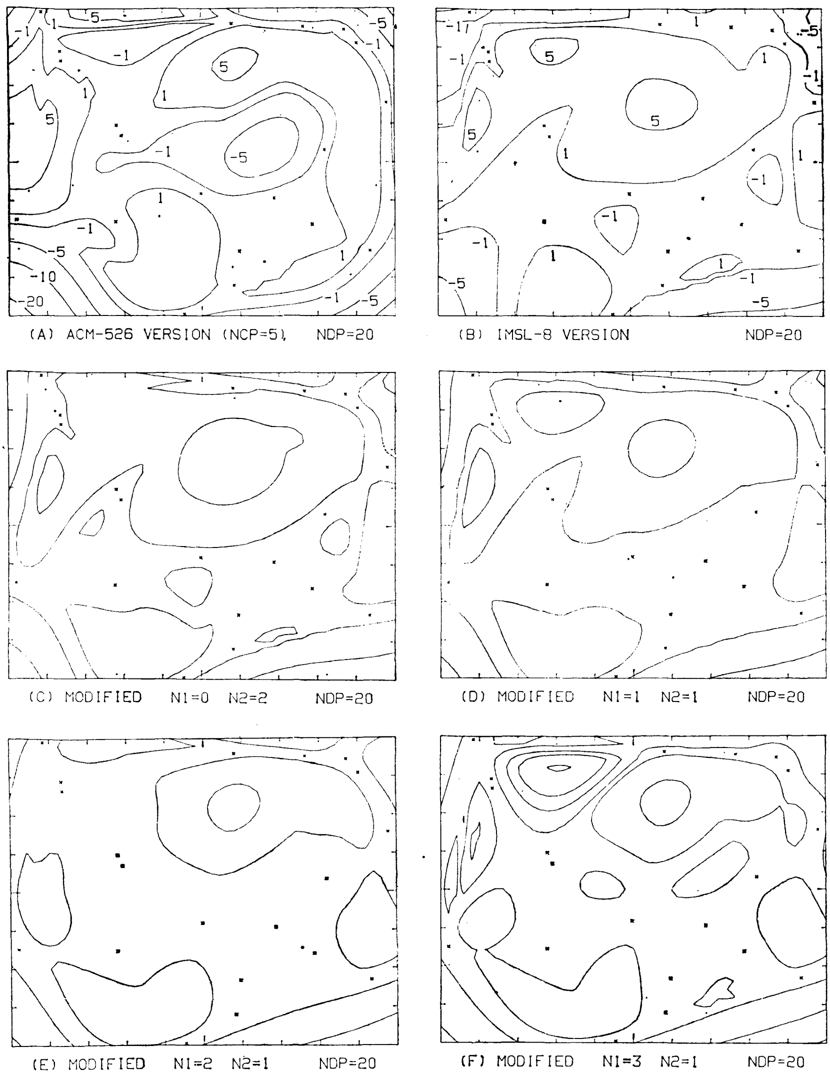

FIGURE 2. Contour maps of the differences between the fitted and original surfaces depicted in Figure 1.

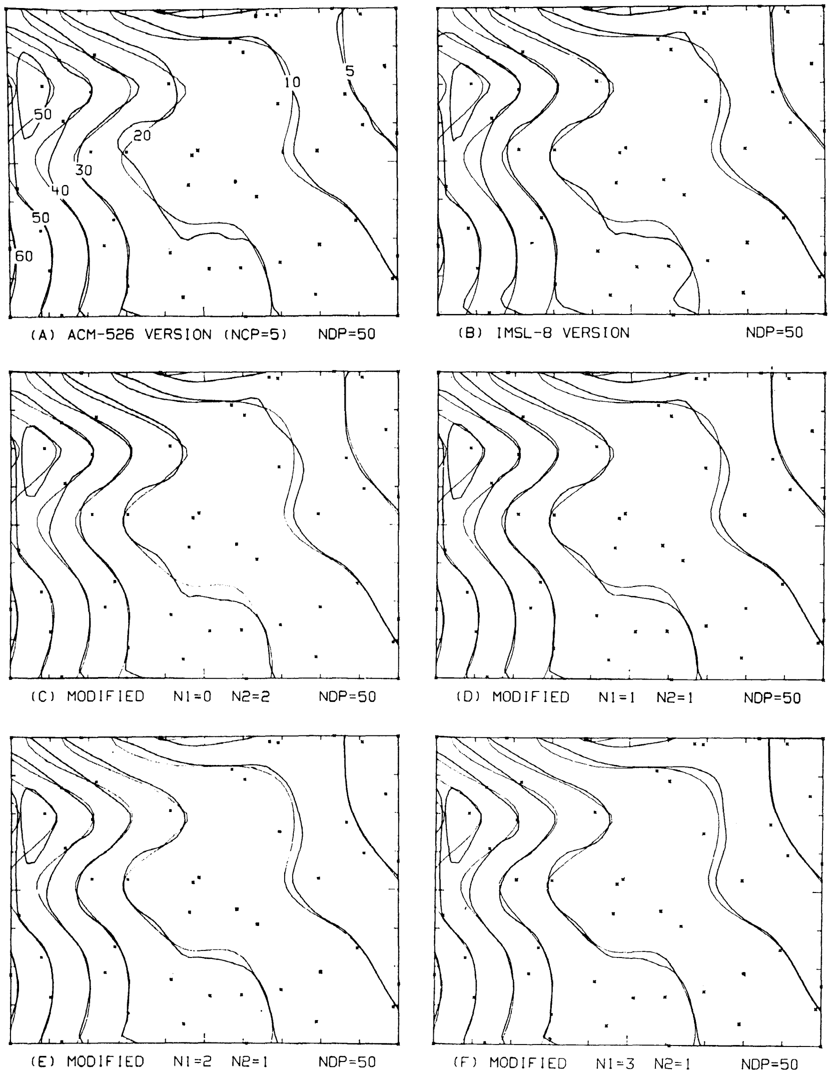

FIGURE 3. Contour maps of the surfaces fitted by six versions of the method to the data taken from the previous paper [1] consisting of 50 data points. (Heavy contours are for the fitted surfaces, and light contours are for the original surface.)

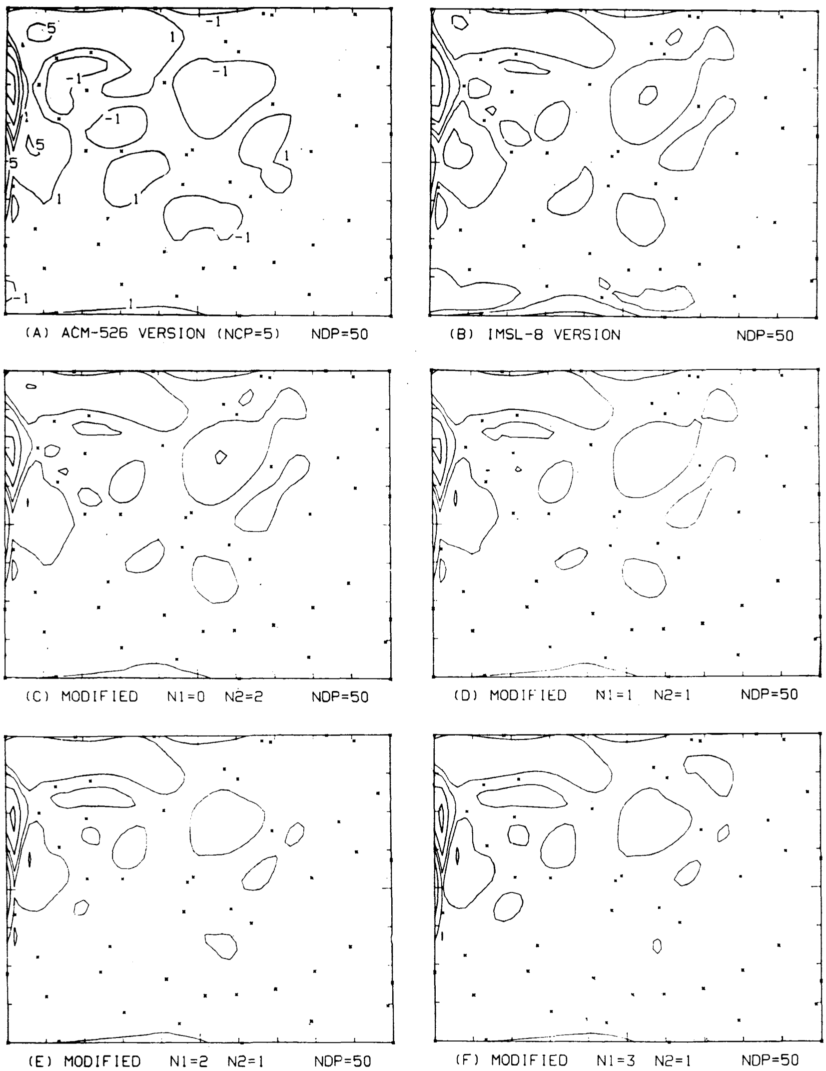

FIGURE 4. Controur maps of the differences between the fitted and original surfaces depicted in Figure 3.

(E) outperforms both the ACM-526 version, (A), and the IMSL-8 version, (B).

As the fourth (and the last) example, we compare with each other the same six versions of the method as in the third example but with a different data set. Used in this example is the same set of 50 data points and data values as used in Figure 2 of a previous paper by the author [**1**]. The results are shown in Figures 3 and 4 of this paper in the same format as in Figures 1 and 2 of this paper, respectively. (The set of light contours in each map of Figure 3 of this paper is the same as Figure 2 (a) of the previous paper. The set of heavy contours in Figure 3 (A) of this paper is the same as Figure 2 (f) of the previous paper.) It is observed from Figures 3 and 4 of this paper that any modified version outperforms the IMSL-8 version, (B), and that the modified version, either (E) or (F), outperforms the ACM-526 version, (A).

Summarizing the results of these tests, we may suggest that the combined use of the two weights, w_1 and w_2, with $n_1 = 2$ and $n_2 = 1$ is a good choice.

6. Conclusion. We have reviewed the method for estimating first partial derivatives used in the ACM-526 and IMSL-8 versions of the method of bivariate interpolation and smooth surface fitting for scattered data. In an attempt to improve the performance of the IMSL-8 version of the method, we have considered an idea of using weights in calculating the vector sum and tested the idea with examples. The test results have indicated that replacement of the weighted vector sum for the simple vector sum in the IMSL-8 version can improve its performance to the performance level of the ACM-526 version or better, retaining all the advantages of its own. Use of the weight that is based on the angle of the triangle in calculating the vector sum also eliminates the discontinuous behaviors that are observed sometimes in the applications of both the ACM-526 and IMSL-8 versions of the method. (It must be noted, however, that the discontinuous behavior cannot be eliminated completely since the overall interpolation scheme is based on a triangulation; the triangulation can change abruptly as the data points are moved.)

REFERENCES

1. Hiroshi Akima, *A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points*, ACM Trans. Math. Software **4** (1978), 148–159.

2. ———, *Algorithm 526, Bivariate interpolation and smooth surface fitting for irregulary distributed data points* [E1], ACM Trans, Math. Software **4** (1978), 160–164.

3. IMSL (International Mathematical and Statistical Libraries, Inc.), IMSL Reference Manual, Edition 8, 1980.

4. Charles L. Lawson, *Generation of a triangular grid with application to contour plotting*. Tech. Memo. 299, Jet Propulsion Lab., Calif, Inst. Tech., 1972.

INSTITUTE FOR TELECOMMUNICATION SCIENCES, NATIONAL TELECOMMUNICATIONS AND INFORMATION ADMINISTRATION, U.S. DEPARTMENT OF COMMERCE, BOULDER, CO 80303