Algorithm 761: Scattered-Data Surface Fitting that Has the Accuracy of a Cubic Polynomial

HIROSHI AKIMA
U.S. Department of Commerce, NTIA/ITS

An algorithm for smooth surface fitting for scattered data has been presented. It has the accuracy of a cubic polynomial in most cases and is a local, triangle-based algorithm.

Categories and Subject Descriptors: D.3.2 [**Programming Languages**]: Language Classifications—*Fortran*; G.1.1 [**Numerical Analysis**]: Interpolation; G.4 [**Mathematics of Computing**]: Mathematical Software

General Terms: Algorithm

Additional Key Words and Phrases: Bivariate interpolation, interpolation, local interpolation

1. DESCRIPTION

This algorithm performs bivariate interpolation and surface fitting for scattered data and is presented as an improvement of Algorithm 526 [Akima 1978].

In most cases, this algorithm has the accuracy of a cubic (third-degree) polynomial, i.e., it correctly interpolates data taken from a surface that represents a cubic polynomial. The accuracy of a cubic polynomial is considered desirable for an interpolation algorithm, as demonstrated in univariate interpolation by Akima [1991].

This algorithm is a local, triangle-based algorithm, as is Algorithm 526. The method used by this algorithm consists of the following steps:

(1) Estimation of five partial derivatives, z_x, z_y, z_{xx}, z_{xy}, and z_{yy}, at each input data point;
(2) triangulation of the data area in the x-y plane;
(3) locating, in which triangle, the point at which interpolation is to be performed;

Author's address: U.S. Department of Commerce, NTIA/ITS.N4, 325 Broadway, Boulder, CO 80303-3328; email: akima@ntia.its.bldrdoc.gov.

(4) determining a fifth-degree polynomial in x and y for the triangle; and
(5) calculation of the z value by evaluating the polynomial.

Note that, steps (1) and (2) are mutually independent, and either step can be performed first. In the SDBI3P/SDSF3P subroutine package that implements this algorithm and is presented here, step (2) is performed before step (1) for computational convenience.

Step (1) of the method is new and is described in Appendix A. Step (2) is a slight modification of (or small addition to) the convex-hull triangulation scheme included in the TRIPACK package of Algorithm 751 by Renka [1996]. Steps (3)–(5) are slight modifications of the corresponding steps in Algorithm 526; modifications in these steps are necessitated by the modification of step (2). The modification of the triangulation and related topics are described in Appendix B.

The SDBI3P/SDSF3P subroutine package consists of two master subroutines that interface the user and 10 supporting subroutines. The subroutines are:

SDBI3P:	a master subroutine that performs bivariate interpolation, i.e., estimates the z values at the specified points in the x-y plane;
SDSF3P:	a master subroutine that performs surface fitting, i.e., estimates the z values at the specified rectangular grid points in the x-y plane and generates a two-dimensional array containing these estimated z values;
SDTRAN:	a first-level supporting subroutine that triangulates the data area in the x-y plane with a scattered data point set (for SDBI3P and SDSF3P);
SDTRCH:	a second-level supporting subroutine that triangulates the convex hull of the data points (for SDTRAN);
SDTRTT:	a second-level supporting subroutine that removes thin triangles along the borderline of the data area (for SDTRAN);
SDPD3P:	a first-level supporting subroutine that estimates partial derivatives at the data points for scattered data (for SDBI3P and SDSF3P);
SDCLDP:	a second-level supporting subroutine that selects, at each of the data points, nine data points closest to it (for SDPD3P);
SDCF3P:	a second-level supporting subroutine that calculates, at each data point, coefficients of the cubic polynomial fitted to the set of 10 data points consisting of the data point in question and nine data points closest to it (for SDPD3P);
SDLEQN:	a third-level supporting subroutine that solves a set of linear equations (for SDCF3P);

SDLS1P: a second-level supporting subroutine that performs, at each of the data points, the least-squares fit of a plane to z values at the 10 data points (for SDPD3P);

SDLCTN: a first-level supporting subroutine that locates points in a scattered data point set in the x-y plane, i.e., determines to which triangle or rectangle each of the points to be located belongs (for SDBI3P and SDSF3P); and

SDPLNL: a first-level supporting subroutine that determines a polynomial in x and y for each triangle or rectangle in the x-y plane and calculates the z value by evaluating the polynomial for each of the desired points (for SDBI3P and SDSF3P).

All the subroutines are written in the 1977 ANSI Standard Fortran.

The SDBI3P/SDSF3P package requires 14 subprograms included in TRIPACK [Renka 1996]. The 14 subprograms are ADDNOD, BDYADD, CRTRI, INDXCC, INSERT, INTADD, LEFT, LSTPTR, STORE, SWAP, SWPTST, TRFIND, TRLIST, and TRMESH.

Mutual dependencies of the subroutines of the SDBI3P/SDSF3P subroutine package are described in the following. Those followed by (T) are the TRIPACK subprograms.

The user	calls SDBI3P or SDSF3P.
SDBI3P	calls SDTRAN, SDPD3P, SDLCTN, and SDPLNL.
SDSF3P	calls SDTRAN, SDPD3P, SDLCTN, and SDPLNL.
SDTRAN	calls SDTRCH and SDTRTT.
SDTRCH	calls TRMESH (T) and TRLIST (T).
TRMESH (T)	calls 12 other TRIPACK subprograms.
TRLIST (T)	calls none.
SDTRTT	calls none.
SDPD3P	calls SDCLDP, SDCF3P, and SDLS1P.
SDCLDP	calls none.
SDCF3P	calls SDLEQN.
SDLEQN	calls none.
SDLS1P	calls none.
SDLCTN	calls none.
SDPLNL	calls none.

2. TESTS

Accuracy of the algorithm has been tested with the test procedure devised by Franke [1979], where six test functions are used on three data point sets (DPSs). Since Franke has given the test results on many existing methods, use of Franke's test procedure has an advantage in that one can compare an algorithm with those existing methods without acquiring and running those methods. The test procedure is reproduced in Appendix C to this algorithm.

The SDSF3P subroutine has been run for a uniformly spaced 33-by-33 output grid for 18 combinations of the six functions times the three data point sets. For each run, the mean and maximum (absolute) errors (deviations from the expected values) have been calculated. Table I shows the errors resulting from these test runs in the "SDSF3P" rows. The table also lists the best (minimum) error values for all local methods taken from Franke's [1979] report and Renka's [1988] paper for the first and second data point sets in the "Best from FR and RE" rows, and from Franke's [1979] report for the third data point set in the "Best from FR" rows. (Renka's paper does not include the results on the third data point set.) The ratios of errors resulting from SDSF3P to the corresponding best errors in Franke [1979] and Renka [1988] or in Franke [1979] alone are listed in the "Ratio" rows. We consider that SDSF3P is better when the product of the two ratios for the mean and maximum is less than 1.0 for each combination of test function and data point set. In the "Is SDSF3P better?" rows, the symbol "O" signifies the case and "X" otherwise. Table I indicates that SDSF3P outperforms other algorithms tested by Franke and Renka in 14 combinations out of a total of 18 combinations of six functions and three data point sets.

In Table II, SDSF3P is compared with each local method selected by Franke [1979] and two others described by Renka [1988]. This table indicates that none of the methods outperforms SDSF3P in more than four combinations out of a total of 18 combinations of six functions and three data point sets.

APPENDIX

A. ESTIMATION OF PARTIAL DERIVATIVES

Estimation of five partial derivatives, z_x, z_y, z_{xx}, z_{xy}, and z_{yy}, at the data points consists of the following substeps:

(1.1) Selection, at each of the data points, of nine data points closest to the data point in question to form a 10-point data point set;

(1.2) fitting, at each of the data points, a cubic polynomial to z values at 10 data points of the 10-point data point set;

(1.3) calculation, at each of the data points, of sets of the five partial derivatives from the cubic polynomials fitted to the 10-point data point sets that include the data point in question, as the primary estimates of the partial derivatives; and

(1.4) calculation, as the final estimates of partial derivatives at each of the data points, of the weighted means of their primary estimates.

If, in substep (1.2), the condition number of the matrix associated with the set of linear equations for the coefficients of the cubic polynomial is greater than a specified maximum value, a quadratic (second-degree) polynomial is fitted to a set of six data points that consists of the data point in question and five other data points closest to it. If the condition number

Table I. Comparison of SDSF3P with the Best from All Methods

Input			Function					
DPS	Error	Method	F1	F2	F3	F4	F5	F6
1	Mean	SDSF3P	0.00455	0.00166	0.00037	0.00026	0.00094	0.00013
		Best from FR and RE	0.00545	0.00199	0.00076	0.00035	0.00119	0.00022
		Ratio	0.83	0.83	0.48	0.74	0.79	0.58
	Max	SDSF3P	0.04522	0.02570	0.00579	0.00261	0.00823	0.00302
		Best from FR and RE	0.05050	0.02490	0.01080	0.00200	0.00990	0.00343
		Ratio	0.90	1.03	0.54	1.31	0.83	0.88
	—	Is SDSF3P better?	O	O	O	O	O	O
2	Mean	SDSF3P	0.03785	0.01018	0.00578	0.00230	0.00933	0.00134
		Best from FR and RE	0.03201	0.00747	0.00776	0.00353	0.00974	0.00167
		Ratio	1.18	1.36	0.74	0.65	0.96	0.80
	Max	SDSF3P	0.18298	0.07178	0.03344	0.01562	0.07095	0.00885
		Best from FR and RE	0.14640	0.05180	0.03670	0.01600	0.06950	0.01020
		Ratio	1.25	1.39	0.91	0.98	1.02	0.87
	—	Is SDSF3P better?	X	X	O	O	O	O
3	Mean	SDSF3P	0.03495	0.00790	0.00716	0.00266	0.01310	0.00130
		Best from FR	0.02670	0.01430	0.00983	0.00399	0.00756	0.00190
		Ratio	1.31	0.55	0.73	0.67	1.73	0.69
	Max	SDSF3P	0.14773	0.06125	0.04943	0.01940	0.06615	0.01330
		Best from FR	0.12900	0.09870	0.06880	0.02270	0.03170	0.01740
		Ratio	1.15	0.62	0.72	0.85	2.09	0.76
	—	Is SDSF3P better?	X	O	O	O	X	O

of the matrix associated with the set of linear equations for the coefficients of the quadratic polynomial is greater than a specified value, a linear (first-degree) polynomial (or a plane) is fitted to a set of three data points that are closest to the data point in question. If the condition number of the matrix associated with the set of linear equations for the coefficients of the linear polynomial is greater than a specified value, a plane is fitted to the data point in question and another data point closest to it in such a way that the plane passes the two points and is horizontal in the direction perpendicular to the line connecting the two data points. In the SDBI3P/SDSF3P package, the ratio of this maximum value to the number of data points (either 10, 6, or 3) is set, as CNRMX in the PARAMETER statement in the SDCF3P subroutine, at a value of 15,000, selected empirically from test runs.

The weight for a set of primary estimates of partial derivatives used in substep (1.4) is the product of two weights, i.e., the probability weight and the volatility weight. These weights have been selected after various possible weights were tested.

The idea behind the probability weight is that we give a small weight to a set of primary estimates of partial derivatives that is far apart from the average of all sets of primary estimates. The probability weight adopted here is the product of five Gaussian probability density functions, each corresponding to a partial derivative. In each function, the mean and the

Table II. Comparison of SDSF3P with Each Method

Method	Input DPS	Function F1	F2	F3	F4	F5	F6
(1) Franke-3	1	O	O	O	O	O	O
	2	O	O	O	O	O	O
	3	O	O	O	O	O	O
(4) Akima	1	O	O	O	O	O	O
	2	X	X	O	O	O	O
	3	X	O	O	O	X	O
(10) Akima Mod. I	1	O	O	O	O	O	O
	2	O	X	O	O	O	O
	3	X	O	O	O	X	O
(13) Nielson-Franke Q	1	O	O	O	O	O	O
	2	X	O	O	O	O	O
	3	O	O	O	O	X	O
(14) Mod Quad Shepard	1	O	O	O	O	O	O
	2	X	O	O	O	O	O
	3	O	O	O	O	X	O
(16) Akima Mod. III	1	O	O	O	O	O	O
	2	X	X	O	O	O	O
	3	O	O	O	O	X	O
(24) Franke-TPS	1	O	O	O	O	O	O
	2	O	X	O	O	O	O
	3	X	O	O	O	X	O
(28) Lawson	1	O	O	O	O	O	O
	2	O	O	O	O	O	O
	3	O	O	O	O	X	O
Renka-Cline Local	1	O	O	O	O	O	O
	2	X	X	O	O	O	O
Renka QSHEP2D	1	O	O	O	O	O	O
	2	X	O	O	O	O	O

variance are their unbiased estimates of the partial derivative calculated from all its primary estimates.

The volatility weight is the reciprocal of a measure of "distance" between the set of primary estimates of partial derivatives and the set of partial derivatives of the least-squares-fit plane fitted to the data point set from which the set of primary estimates is obtained. Use of this weight ensures that, if a data point and its nine closest data points fall in a plane in the three-dimensional space, the final estimates of partial derivatives at any of the 10 data points are those of the plane.

B. MODIFICATION OF TRIANGULATION

In triangulation in this algorithm, thin triangles along the border line of the data area are removed after the triangulation in the "convex hull" of the data area is performed.

The triangulation in the convex hull is performed by the TRMESH and TRLIST subroutines in TRIPACK of Algorithm 751 [Renka 1996]. Use of TRMESH and TRLIST is faster than the use of Algorithm 526 [Akima 1978] and results in the same triangulation.

After the triangulation in the convex hull is performed, thin triangles with the height-to-bottom ratios of less than a specified minimum value are removed. In the SDBI3P/SDSF3P package, this minimum value is set, as HBRMN in the PARAMETER statement in the SDTRTT subroutine, at a value of 0.1, selected empirically from test runs. For some data point sets, removal of thin triangles along the borderline only once is not sufficient, and repetition of removal is required. For the data point sets described in Appendix C, three repetitions at most are sufficient, but there seems to exist no theoretically sufficient upper bound. In the SDBI3P/SDSF3P package, the maximum number of repetitions is set, as NRRTT in the PARAMETER statement in the SDTRTT subroutine, at a value of five. By limiting the maximum number of repetitions, any possible complications and danger of infinite loop can be avoided.

In the triangulation of the convex hull, the plane is divided into several categories of pieces of plane: (1) a triangle inside the data area, (2) a semiinfinite rectangle outside a borderline segment, and (3) a semiinfinite triangle that does not belong to a rectangle outside the data area. After removal of thin triangles, there is another category: (4) a semiinfinite triangle, which is an overlap of two semiinfinite rectangles outside two consecutive borderline segments.

In accord with the removal of thin triangles in triangulation, the schemes of locating the points and calculating z values in Algorithm 526 have been modified in such a way that category (4) is included in this algorithm. In the semiinfinite triangle of category (4), the z value is calculated as the weighted mean of two z values calculated in the two mutually overlapping semiinfinite rectangles.

In addition to the preceding four categories, there exist more categories in parts of the plane that are far from the data area. The whole picture seems so complicated that we cannot manage it. Fortunately, however, since this algorithm is intended for "interpolation" rather than "extrapolation," we can avoid such complexity by restricting the use of this algorithm to interpolation inside the data area and extrapolation in the vicinity of it.

C. TEST PROCEDURE

This section describes the procedure for testing methods for scattered-data surface fitting used in the text of this algorithm. The test procedure was originally devised and used extensively by Franke [1979].

Franke tested a number of methods for scattered-data surface fitting with the following test functions:

(1) a combination of four Gaussian functions;

(2) a hyperbolic tangent function (almost a cliff);

Table C.I. First Data Point Set (NDP = 100)

X	Y	X	Y	X	Y	X	Y
0.022703	−0.031021	0.053989	0.158674	0.584869	−0.027195	0.573008	0.127243
0.021701	257692	0.017513	0.341401	0.606389	0.270927	0.501389	0.347773
0.001903	494360	−0.050968	0.578285	0.574131	425942	0.610695	0.608471
0.039541	699342	−0.048706	0.747019	0.599010	673378	0.538062	0.723524
0.031583	910765	−0.041878	0.996289	0.609697	924241	0.502619	1.030876
0.132419	050133	0.109027	0.091855	0.661693	025596	0.642784	0.070783
0.125444	259297	0.093454	0.338159	0.639647	200834	0.670396	0.325984
0.076758	417112	0.145187	0.561556	0.700118	489070	0.633359	0.509632
0.062649	655223	0.145273	0.752407	0.690895	0.669783	0.689564	0.775957
0.095867	914652	0.069556	0.963242	0.671889	936610	0.683767	1.006451
0.264560	029294	0.239164	0.060230	0.773694	028537	0.763533	0.102140
0.208899	266878	0.276733	0.369604	0.741042	193658	0.825898	0.323577
0.171473	480174	0.226678	0.594059	0.730603	471423	0.808661	0.609159
0.190921	687880	0.186765	0.818558	0.821453	668505	0.729064	0.802281
0.230463	904651	0.242622	0.980541	0.807664	847679	0.817095	1.051236
0.366317	0.039695	0.385766	0.068448	0.842457	038050	0.868405	0.090205
0.383239	238955	0.317909	0.312413	0.836692	208309	0.941846	0.331849
0.346632	490299	0.377659	0.519930	0.847812	433563	0.859958	0.591014
0.387316	644523	0.381292	0.820379	0.917570	630738	0.859633	0.814484
0.379536	893803	0.280351	0.971172	0.927987	904231	0.851280	0.969603
0.414977	−0.028462	0.427768	0.156096	1.044982	−0.012090	0.967063	0.133411
0.420001	226247	0.466363	0.317509	0.985788	269584	0.967631	0.379528
0.485566	389142	0.409203	0.508495	1.012929	0.439605	0.965704	0.504442
0.479258	632425	0.481228	0.751101	1.001985	694152	1.035930	0.745992
0.397776	848971	0.402732	0.997873	1.041468	868208	0.947151	0.980141

(3) a saddle-shaped function;
(4) a Gaussian function (gentle slope);
(5) a Gaussian function (steep slope); and
(6) a part of a sphere.

In the Fortran notation, these six functions are written as

```
      F1(X, Y) = 0.75*EXP(−((9.0*X−2.0)**2+(9.0*Y−2.0)**2)/4.0)
     1         +0.75*EXP(−((9.0*X+1.0)**2)/49.0−(9.0*Y+1.0)/10.0)
     2         +0.50*EXP(−((9.0*X−7.0)**2+(9.0*Y−3.0)**2)/4.0)
     3         −0.20*EXP(−(9.0*X−4.0)**2−(9.0*Y−7.0)**2)
      F2(X, Y) = (TANH(9.0*Y−9.0*X)+1.0)/9.0
      F3(X, Y) = (1.25+COS*(5.4*Y))/(6.0*(1.0+(3.0*X−1.0)**2))
      F4(X, Y) = EXP(−81.0*((X−0.5)**2+(Y−0.5)**2)/16.0)/3.0
      F5(X, Y) = EXP(−81.0*((X−0.5)**2+(Y−0.5)**2)/4.0)/3.0
      F6(X, Y) = SQRT(64.0−81.0*((X−0.5)**2+(Y−0.5)**2))/9.0−0.5
```

Franke tested the methods on the following data point sets:

(1) a somewhat uniform 100-point set;
(2) a sparse 33-point set; and
(3) a 25-point set.

Table C.II. Second Data Point Set (NDP = 33)

X	Y	X	Y	X	Y	X	Y
0.00	0.00	1.00	0.00	0.80	0.40	0.85	0.25
0.00	1.00	1.00	1.00	0.70	0.20	0.80	0.65
0.00	0.50	0.50	0.00	0.95	0.90	0.75	0.85
0.50	1.00	1.00	0.50	0.60	0.65	0.70	0.90
0.10	0.15	0.20	0.10	0.65	0.70	0.70	0.65
0.15	0.30	0.25	0.20	0.35	0.85	0.75	0.10
0.30	0.35	0.60	0.25	0.60	0.85	0.75	0.35
0.10	0.75	0.90	0.35	0.90	0.80	0.55	0.95
0.05	0.45						

Table C.III. Third Data Point Set (NDP = 25)

X	Y	X	Y	X	Y	X	Y
0.13750	0.97500	0.45000	1.03750	0.85000	0.43750	1.00000	0.26250
0.91250	0.98750	1.08750	0.55000	0.70000	0.31250	0.50000	0.46250
0.71250	0.76250	0.53750	0.80000	0.27500	0.42500	0.18750	0.26250
0.22500	0.83750	−0.03750	0.75000	0.45000	0.28750	0.58750	0.12500
−0.05000	0.41250	0.18750	0.57500	0.81250	0.18750	1.05000	−0.06125
0.47500	0.63750	0.71250	0.55000	0.45000	−0.03750	0.10000	0.11250
0.05000	−0.05000						

The x and y values of the data points in these data point sets are given in Tables C.I, C.II, and C.III.

Franke calculated the z values at the data points of each data point set with each test function and applied each surface-fitting method for the 33-by-33 output mesh in the unit square. He then selected the maximum value from, and calculated the mean and rms (root mean square) values of, the absolute errors (or deviations from the correct values) over the 1089 mesh points.

Franke tested a total of 29 methods. He listed, for each of selected eight local methods and five global methods, the maximum, mean, and rms values of errors for each of 18 combinations of six test functions and three data point sets.

Renka used this test procedure with the same six test functions on the first two data point sets [Renka 1988]. Since the patterns for rms errors are somewhat similar to those of mean errors, Renka dropped the rms errors, and we follow suit.

ACKNOWLEDGMENTS

The author is grateful to the anonymous *ACM TOMS* referee for the information on Renka's triangulation algorithm, and to Professor Renka of the University of North Texas for his kindness in providing the author with his triangulation algorithm.

REFERENCES

AKIMA, H. 1978. A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points. *ACM Trans. Math. Softw. 4*, 2 (June), 148–159. See also the companion algorithm. Algorithm 526. *ACM Trans. Math. Softw. 4*, 2, 160–164.

AKIMA, H. 1991. A method of univariate interpolation that has the accuracy of a third-degree polynomial. *ACM Trans. Math. Softw. 17*, 3 (Sept.), 341–366. See also the companion algorithm. Algorithm 697. *ACM Trans. Math. Softw. 17*, 3, 367.

FRANKE, R. 1979. A critical comparison of some methods for interpolation of scattered data. Tech. Rep. NPS-53-79-003, Dept. of Mathematics, Naval Postgraduate School, Monterey, Calif.

RENKA, R. J. 1988. Multivariate interpolation of large sets of scattered data. *ACM Trans. Math. Softw. 14*, 2 (June), 139–148. See also the companion algorithm. Algorithm 660. *ACM Trans. Math. Softw. 14*, 2, 149–150.

RENKA, R. J. 1996. Algorithm 751. TRIPACK: Constrained two-dimensional Delaunay triangulation package. *ACM Trans. Math. Softw. 22*, 1 (Mar.), 1–8.

Received August 1994; accepted August 1995