Surface approximation with imposed conditions

Article · January 1983

CITATIONS
25

READS
41

2 authors, including:

Gregory M Nielson
Arizona State University
241 PUBLICATIONS **5,129** CITATIONS

SEE PROFILE

SURFACES IN CAGD
R.E. Barnhill, W. Boehm (eds.)

SURFACE APPROXIMATION WITH IMPOSED CONDITIONS

Richard Franke
Naval Postgraduate School
Monterey, CA, USA

Gregory M. Nielson
Arizona State University
Tempe, AZ, USA

1. INTRODUCTION

We discuss the problem of surface approximation when scattered data points are known and it is desired that the surface be smooth except for certain discontinuities of value or derivative. These are assumed to occur along line segments. The applications of such schemes in modelling terrain, underwater topography, and underground strata are immediately apparent. Some of our techniques have applicability in computer aided design when creases or holes are required in the surface. Our primary emphasis will be on discontinuities, or *faults*, but because some of our techniques extend readily, we also consider discontinuities of derivatives, or *creases*.

The possibility of having to model faults can occur in at least three ways, which we list by type.

(1) There is a known fault line segment, in a known location, with a known jump.

(2) There is a known fault line segment, but there is no definitive estimate of the jump at points along the segment. If there is a much greater than usual density of points along both sides of the fault, this type is similar to type 1.

(3) There is no known fault line, but the data indicates a probable fault because of steep gradients which must otherwise occur between data points.

The type 1 problem can be handled by the same kind of scheme as the type 2 problem, although we feel that better techniques could be devised for exclusively type 1 problems. Type 3 problems will require considerably more work. In the absence of any information about the location of the fault, its precise location is obviously difficult to pinpoint. The problem is related to that of edge detection in image processing. If an estimate is made of its location, the new problem is type 2. In the

interest of addressing the problem whose solution has the greatest applicability, we have chosen to look at the type 2 problem.

2. APPROACH

There is a rich variety of schemes for smooth interpolation of scattered data. Many of these have been studied extensively [2]. We wish to consider local methods applicable to large data sets. For these purposes the methods fall into two general classes: (i) weighted local approximations, and (ii) "patch", or "finite element" methods. While in certain instances patch schemes could be modified to solve the problem, weighted local approximations have more general applicability. We will digress for a discussion of these methods for the scattered data interpolation problem.

Assume that scattered data points, (x_k,y_k,f_k), $k=1,\ldots,N$ are given. It is convenient to think in terms of f_k being the value of some (unknown, in real applications) function, $f(x,y)$. Weighted local approximations are of the form

$$(1) \quad F(x,y) = \sum_{k=1}^{n} W_k(x,y)Q_k(x,y) \Big/ \sum_{k=1}^{n} W_k(x,y)$$

Here $W_k(x,y)$ is a *weight function*, usually taken as nonnegative, and $Q_k(x,y)$ is a local approximation to the underlying function over the support of W_k. Q_k is called a *nodal function* when it is associated with a particular point. Of course, most (perhaps all) of the functions W_k have finite support. There may (but need not) be a weight function associated with each data point. The W_k incorporate information concerning the disposition of the independent variable coordinates of the data points but are independent of the f_k values. The Q_k contain information concerning local behavior of the surface.

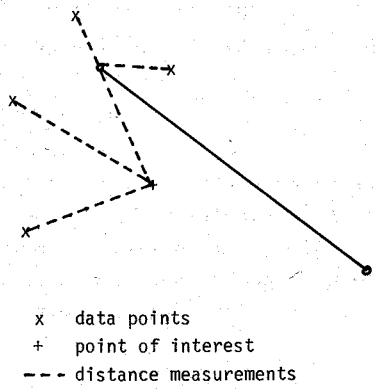
Recent tests [2] have shown that the Modified Quadratic Shepard's Method (MQS) performs very well for a variety of data sets, and we have chosen it as the basis for our investigation. In this scheme, the weight functions, $W_k(x,y)$ are of the form $W_k(x,y) = ((R - d_k)_+/Rd_k)^2$, where $d_k^2 = (x - x_k)^2 + (y - y_k)^2$. The $Q_k(x,y)$ are taken to be quadratic functions, the quadratic being constrained to pass through the point (x_k,y_k, f_k) and the remaining parameters being determined by a weighted least squares fit. For this process the weights are taken as $((r - d_k)_+/rd_k)^2$. Experience has shown that a reasonable value for the r is $\sqrt{2}$ R. R is generally chosen as a constant based on the density of the points and this works well for data sets which have a somewhat uniform density. See [3] for further discussion of the details. For point sets with greatly

varying density it would be useful to specify different values of R and r for each point.

Having settled upon a variation of the MQS method, we note that there are two possible ways to proceed. The expression in Equation (1) can incorporate nonsmooth behavior into either the weight functions or the nodal functions. It is possible that the choice can be made on philosophical grounds: If one views the fault as having something to do with the domain of the function (for which an argument can be made) the behavior is properly built into the weight function. On the other hand, if the behavior is viewed as pertaining to the surface, then the nodal functions should incorporate the behavior as part of the local properties of the surface. Only those nodal functions whose domain intersects the fault (or crease) line need to provide for the anomalous behavior.

Previous work in simulating faults has been done by Bolondi, et.al. [1], and Pouzet [4]. Each of these papers considered methods involving distance, although not necessarily in a weighting scheme. Bolondi considered several schemes, including some based on Kriging, and others based on minimization of sums of squares of difference quotients. In the part of Bolondi's work which is similar to ours, basis functions which were functions of distance were used, and a distance penalty identical to that we consider was used. Pouzet used Kriging to obtain a grid of points, then a Shepard type of scheme for the fine details and contouring. The penalty scheme used by Pouzet was to penalize the straight line distance by adding twice the distance to the nearest end of the fault. It is easy to see that this results in a distance penalty which is continuous in value, but not in slope. Whether the resulting derivative discontinuity would be visually noticeable is probably dependent on the particular data set.

Measuring distances around the end of the fault is relatively easy to implement and results in a scheme which yields good results. The sketch at right illustrates the measurement. An implicit assumption is that the fault is long compared to the radius R (practically speaking, at least as long as R). The boundary of the resulting region of influence for a point is that traced by the end of a string of length R, one end held at the point, and the other of which



wraps around the end of a "fence" along the fault line. We refer to this method as the Distance Penalty Fault Method (DPF) in the examples.

The generalization of this idea for creases is not clear. The penalty must be continuous in value across the fault, while discontinuous in slope. It is desirable, but not mandatory, that the penalty correspond to a distance which is the same regardless of the direction in which distance is measured between the two points. We tried a number of schemes to impose such a penalty, but none were as natural as the scheme for faults, and none we tried gave particularly good results.

In our opinion, a more natural idea, and one which also generalizes easily to creases, is to allow the nodal functions to incorporate the discontinuity in value or derivative. This can be achieved by including a suitable term along with the quadratic approximation, the coefficient being determined by least squares at the same time as those of the quadratic. The purpose of this term is to remove the anomalous behavior so that the remaining function data can be well approximated by a quadratic function. Consider the following coordinate system for that term: Let t represent the parameterized distance along the fault line segment, $0 \leq t \leq 1$, and let s represent the coordinate in a perpendicular direction. For convenience, take the units the same for the s direction as for the t direction. Of course, this term only needs to be included for those nodal functions associated with points that are within a distance r of the fault or crease line.

We desire a smooth function, except along the fault/crease line, where it exhbits the appropriate behavior. Our first attempt was to use the following.

$$(2) \quad \phi_f(s,t) = H_3(s)\, H_3(2t-1)\, \mathrm{sgn}(s), \text{ and}$$

$$(3) \quad \phi_c(s,t) = H_3(s)\, H_3(2t-1)\, |s|.$$

Here $H_3(s)$ is the piecewise cubic

$$H_3(s) = \begin{cases} 1 - s^2(3 - 2|s|), & |s| \leq 1 \\ 0, & |s| > 1 \end{cases}$$

We refer to this method as the Nodal Function Fault Method One (NFF1) in the examples.

After our initial attempts, it became apparent that these functions do not incorporate enough flexibility. We then added three terms to the quadratic, each of the form

$$(4)\quad \phi_f^i(s,t) = H_3(s)\,[t^{i+1}(1-t)^{5-i}]\ \mathrm{sgn}(s), \text{ and}$$

$$(5)\quad \phi_c^i(s,t) = H_3(s)\,[t^{i+1}(1-t)^{5-i}]\ |s|,$$

for $i = 1,2,3$.

We refer to this scheme as the Nodal Function Fault Method Two (NFF2) in the examples.

As our examples show, this strategy yields much better results due to the improved local fitting ability of the nodal functions. The polynomial, as opposed to piecewise cubic, form is probably unimportant in this regard. We note that the polynomial part of the above terms are those of the Bernstein polynomial of degree six having zero value and derivative at the endpoints. This is necessary to avoid propogation of nonsmooth behavior in the s direction.

3. EXAMPLES

We present a number of examples to illustrate the abilities of the above schemes to model faults and creases. Since this work is relatively new in the mathematical literature, at least, we have been especially careful to document our examples to that they are reproducible and so that they may be used by others to compare in a qualitative way with new schemes which may be developed. Our examples are based mostly on two sets of data, given in Tables 1 and 2. The function values were computed from a piecewise polynomial surface, the surface having a fault line segment specified between two pieces, and some creases which were not specified, but the data being sparse there to allow the method to smoothly join them. Some perturbations can be seen at the right rear of Figures 3 - 8 where this occurs. The precise definition of the surface being modelled by those figures is

$$f(x,y) = \begin{cases} .5 & , \ y \le .4 \\ .5(1-((y-.4)/.6)^2) & , \ x \le .1,\ y > .4 \\ .5((y-1)/.6)^2(1-x)/.8 & , \ x > .2,\ y > .4 \\ 5((y-1)/.6)^2(x-.1)+(1-((y-.4)/.6)^2(.2-x)) & , \ .1 < x \le .2,\ y > .4 \end{cases}$$

The above function was sampled over two different data sets, of 33 and 130 points. These points are shown in Figures 1 and 2, and are listed along with the function values of Tables 1 and 2. The disposition in the x-y points and the fault line segment are shown in Figures 1 and 2.

respectively. The unit square, over which the functions are plotted, is superimposed. Table 1 also lists data for an example illustrating crease lines which is based on the same set of x-y points. We note that the viewpoint for all of our plots is in the first quadrant at spherical coordinate angles of 30° and 60°.

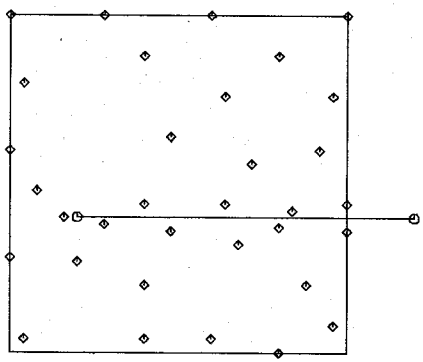


Figure 1: 33 Point F/C Data

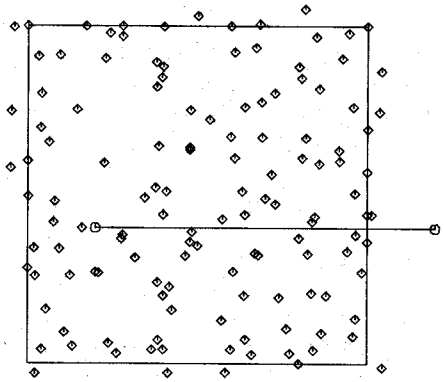


Figure 2: 130 Point Data

Figure 3, showing the results of applying the DPF method to the data of Table 1, shows a very reasonable surface. The behavior of the surface near (but not on) the fault is not plotted quite accurately because of the resolution of the plot. The cross-section exactly on the fault shows a wavy character because the program allowed points on both sides of the fault to influence the value when this occurred. Figures 4 and 5 show the results of applying the two nodal function methods, NFF1 and NFF2, respectively, for the same data. The lack of sufficient flexibility to fit the data locally in the vicinity of the fault is apparent in Figure 4, resulting in somewhat wavy behavior near the fault. This behavior is almost completely done away with in Figure 5, resulting in a much more pleasing surface. All three of the figures show some "edge effects" near the front edge of the surface, on both sides of the fault.

Figures 6, 7, and 8 are the analogues of the previous for the data given in Table 2. The decreased radius of influence of each point due to the increased density of points is apparent, especially in Figure 7, where once more the lack of adequate fitting ability of a single discontinuous term is apparent. Figure 8 shows once more that three terms is adequate (and probably necessary as well) to give a good representation

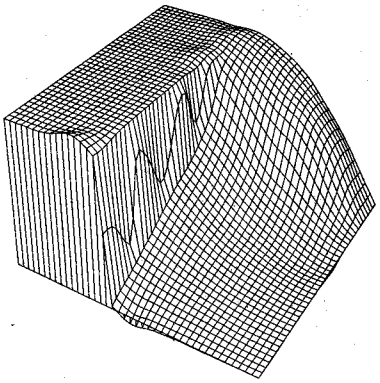


Figure 3: DPF, 33 Points

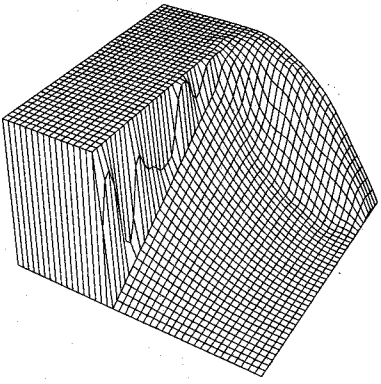


Figure 6: DPF, 130 Points

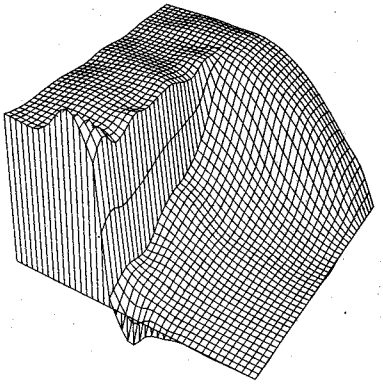


Figure 4: NFF1, 33 Points

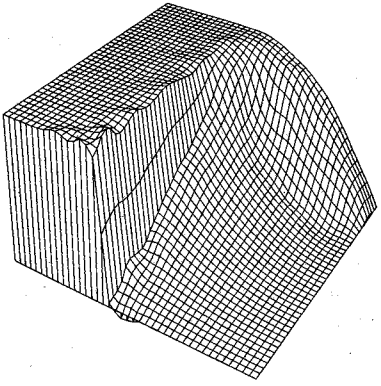


Figure 7: NFF1, 130 Points

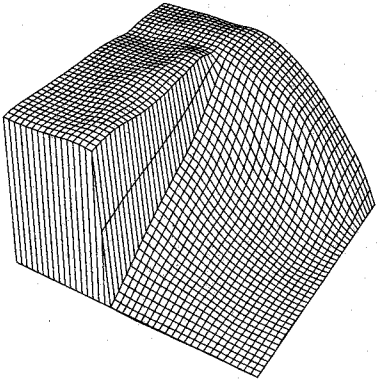


Figure 5: NFF2, 33 Points

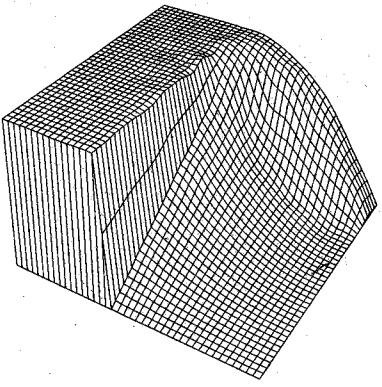


Figure 8: NFF2, 130 Points

of the fault edge. Figure 6 also shows a very pleasant surface resulting from the DPF method. The local quadratic precision of the methods is apparent, especially on the flat part at the left edge and the "nearly quadratic" part at the front corner. The part at the right rear is quadratic, but with too few points on it to allow it to be represented exactly at most points.

We have made the a priori assumption for the nodal function fault methods that the effects of the fault die out at a distance equal to the length of the fault. In addition, the form of the term(s) we give results in the right and left derivatives at the fault having the same value. This is, of course, not the case for the underlying function in our example, and results in a somewhat poorer surface near the fault than might be achieved otherwise. However, to correct for the above two defects of the method would seem to require the use of nonlinear terms in the least squares process, a complication which is certainly tractable, but one which we wish to avoid. It seems apparent from our examples that the fact of convergence is not in doubt. While no convergence proof has been carried out (indeed, we know of no published convergence results for any version of Shepard's method), this can be done using a technique similar to that used to prove convergence of Bernstein polynomials. Properly allowing for the slope differences across the boundary of the fault should result in a faster rate of convergence, and in specific instances, could result in a more pleasant appearing surface.

The capability of the nodal function methods to model creases as well as faults is shown in Figures 9 and 10. The data is given in Table 1, and the figures show the results of applying NFF1 and NFF2, respectively.

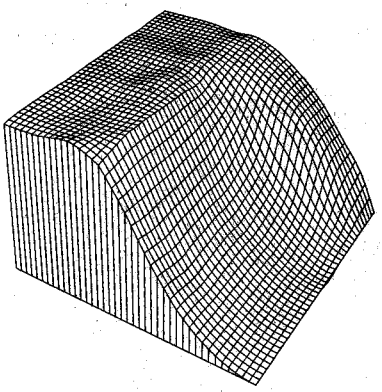


Figure 9: NFF1, 33 Pts

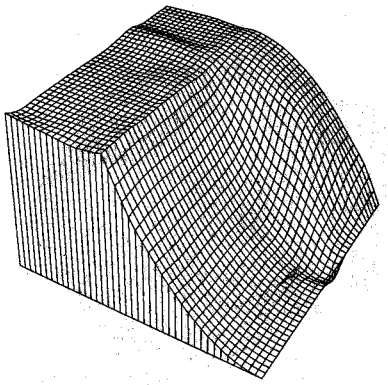


Figure 10: NFF2, 33 Pts

Both methods show a definite crease, but the increased flexibility of the second method is apparent in the plots. The slight perturbation seen at the rear end of the crease and extending out in both directions from it is apparently due to the lack of nearby data points and is the only real defect which is noticeable in Figure 10. Figure 9 shows the usual variations along the crease due to not being able to model the crease with great enough accuracy.

As a final example, to illustrate the ease with which the distance penalty method can handle multiple connected line segments to specify the faults, consider the data in Table 3. The x-y points and the location of the fault lines is shown in Figure 11. This data essentially represents information from two planes, one inside a "box", although the left front corner involves some points above the plane. The resulting surface shows essentially a "box canyon", as is seen in Figure 12. The fault line extends far enough out at the right to avoid the relatively far away points outside the fault lines from affecting the portion inside. On the other hand, the relationship between fault line at the left and the data points allows the outside points to influence the part near the end of the fault line and inside of it. This could be avoided by extending the fault line at the left. In fact, one can achieve two independent surfaces by this device, separated by the fault line.

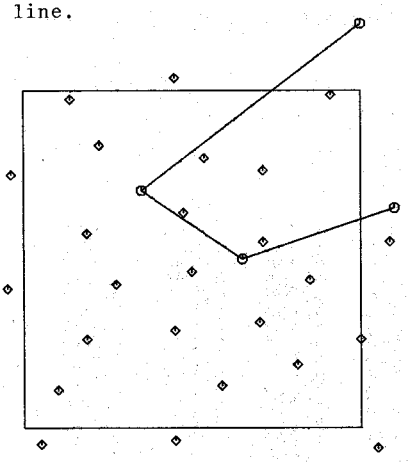


Figure 11: Box Canyon Data

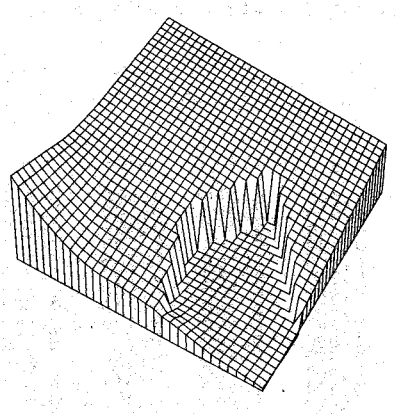


Figure 12: DPF, Box Canyon

4. CONCLUSIONS

In this paper we have developed some methods for modelling faults and creases. For faults the simpler of the schemes, the distance penalty method (DPF) works quite well, has more general applicability, and requires less computation. Its use is recommended for situations where it is applicable. The use of the nodal function fault method (NFF2) is theoretically more interesting and generalizes readily to include creases. With the discontinuous terms we have used (and with many others, we expect) the method performs quite well. It also has the capability to model some more specialized behavior as well if it is specified in the discontinuous terms. The method also performs very well. One disadvantage is that the method has not yet been extended to allow for the possibility of a fault being specified by several connecting line segments. Use of only one discontinuous term in the nodal function is not adequate to properly represent the local behavior of the surface and is not recommended.

In addition to proposing some ideas for solution of the problem, we have also given a number of examples with sufficient documentation to allow them to serve as initial test problems for any new schemes which may be developed. Inclusion of sufficient information about examples to allow close scrutiny of them has often been lacking in the past. We feel it has become increasingly important to do so and hope that other authors will concur.

X	Y	F/F	F/C	X	Y	F/F	F/C
0.040	0.040	0.500	0.500	0.400	0.040	0.500	0.500
0.600	0.040	0.500	0.500	0.800	0.0	0.500	0.500
0.960	0.080	0.500	0.500	0.0	0.280	0.500	0.500
0.200	0.270	0.500	0.500	0.400	0.200	0.500	0.500
0.880	0.200	0.500	0.500	0.160	0.400	0.500	0.500
0.280	0.380	0.500	0.500	0.480	0.360	0.500	0.500
0.680	0.320	0.500	0.500	0.800	0.370	0.500	0.500
1.000	0.360	0.500	0.500	0.400	0.440	0.327	0.440
0.640	0.440	0.196	0.440	0.840	0.420	0.093	0.480
1.000	0.440	0.0	0.440	0.720	0.560	0.094	0.270
0.920	0.600	0.022	0.220	0.480	0.640	0.117	0.180
0.960	0.760	0.004	0.080	0.640	0.760	0.036	0.080
0.400	0.880	0.015	0.020	0.800	0.880	0.005	0.020
0.600	1.000	0.0	0.0	1.000	1.000	0.0	0.0
0.080	0.480	0.491	0.491	0.0	0.600	0.444	0.444
0.040	0.800	0.278	0.278	0.0	1.000	0.0	0.0
0.280	1.000	0.0	0.0				

FAULT/CREASE: (0.20,0.40) - (1.20,0.40)

TABLE 1: DATA FOR FIGURES 3-5 AND 9-10

X	Y	F	X	Y	F	X	Y	F
0.040	0.040	0.500	0.400	0.040	0.500	0.600	0.040	0.500
0.800	0.0	0.500	0.960	0.080	0.500	0.0	0.280	0.500
0.200	0.270	0.500	0.400	0.200	0.500	0.880	0.200	0.500
0.160	0.400	0.500	0.280	0.380	0.500	0.480	0.360	0.500
0.680	0.320	0.500	0.800	0.370	0.500	1.000	0.360	0.500
0.400	0.440	0.327	0.640	0.440	0.196	0.840	0.420	0.093
1.000	0.440	0.0	0.720	0.560	0.094	0.920	0.600	0.022
0.480	0.640	0.117	0.960	0.760	0.004	0.640	0.760	0.036
0.400	0.880	0.015	0.800	0.880	0.005	0.600	1.000	0.0
1.000	1.000	0.0	0.080	0.480	0.491	0.0	0.600	0.444
0.040	0.800	0.278	0.0	1.000	0.0	0.280	1.000	0.0
0.023	-0.031	0.500	0.054	0.159	0.500	0.022	0.258	0.500
0.018	0.341	0.500	0.002	0.494	0.488	-0.051	0.578	0.456
0.040	0.699	0.376	-0.049	0.747	0.333	0.032	0.911	0.138
-0.042	0.996	0.006	0.132	0.050	0.500	0.109	0.092	0.500
0.125	0.259	0.500	0.093	0.338	0.500	0.077	0.417	0.500
0.145	0.752	0.207	0.096	0.915	0.132	0.063	0.655	0.410
0.265	0.029	0.500	0.239	0.060	0.500	0.209	0.267	0.500
0.277	0.370	0.500	0.171	1.000	0.0	0.227	0.594	0.221
0.230	0.905	0.012	0.243	0.981	0.001	0.366	0.040	0.500
0.386	0.068	0.500	0.383	0.239	0.500	0.318	0.312	0.500
0.347	0.490	0.295	0.378	0.520	0.249	0.387	0.645	0.134
0.381	0.820	0.035	0.380	0.894	0.012	0.280	0.971	0.001
0.415	-0.028	0.500	0.428	0.156	0.500	0.420	0.226	0.500
0.466	0.318	0.500	0.486	0.389	0.500	0.409	0.508	0.248
0.479	0.632	0.122	0.481	0.751	0.056	0.398	0.849	0.024
0.403	0.998	0.0	0.585	-0.027	0.500	0.573	0.127	0.500
0.606	0.271	0.500	0.501	0.348	0.500	0.574	0.426	0.244
0.611	0.608	0.104	0.599	0.673	0.074	0.538	0.724	0.061
0.610	0.924	0.004	0.503	1.031	0.001	0.662	0.026	0.500
0.643	0.071	0.500	0.640	0.201	0.500	0.670	0.326	0.500
0.700	0.489	0.136	0.633	0.510	0.153	0.691	0.670	0.059
0.690	0.776	0.027	0.672	0.937	0.002	0.684	1.006	0.000
0.774	0.029	0.500	0.764	0.102	0.500	0.741	0.194	0.500
0.826	0.324	0.500	0.731	0.471	0.131	0.809	0.609	0.051
0.821	0.669	0.034	0.729	0.802	0.018	0.808	0.848	0.008
0.817	1.051	0.001	0.842	0.038	0.500	0.868	0.090	0.500
0.837	0.208	0.500	0.942	0.332	0.500	0.848	0.434	0.085
0.860	0.591	0.041	0.918	0.631	0.020	0.860	0.814	0.008
0.928	0.904	0.001	0.851	0.970	0.000	1.045	-0.012	0.500
0.967	0.133	0.500	0.986	0.270	0.500	0.968	0.380	0.500
1.013	0.440	-0.007	0.966	0.504	0.015	1.002	0.694	-0.000
1.036	0.746	-0.004	1.041	0.868	-0.001	0.947	0.980	0.000
1.000	0.568	0.0						

FAULT: (0.20,0.40) - (1.20,0.40)

TABLE 2: DATA FOR FIGURES 6-8

X	Y	F	X	Y	F	X	Y	F
0.650	0.500	1.100	0.650	0.0	0.0	0.137	0.975	0.444
0.450	1.038	0.256	0.913	0.988	0.0	1.087	0.550	0.181
0.712	0.762	0.0	0.538	0.800	0.0	0.225	0.837	0.469
-0.038	0.750	0.644	-0.050	0.413	0.819	0.188	0.575	0.619
0.475	0.637	0.0	0.712	0.550	0.0	0.850	0.438	0.356
1.000	0.262	0.369	0.700	0.313	0.494	0.500	0.462	0.519
0.275	0.425	0.650	0.188	0.262	0.775	0.450	0.288	0.631
0.587	0.125	0.644	0.813	0.188	0.500	1.050	-0.061	0.989
0.450	-0.038	0.794						

FAULT: (1.00,1.20) - (0.35,0.70) - (0.65,0.50) - (1.10,0.65)

TABLE 3: DATA FOR FIGURE 12

5. ACKNOWLEDGEMENTS

The first author was supported in part by the Foundation Research Program at the Naval Postgraduate School. The second author was supported by the Office of Naval Research under contract NR 044-433.

6. REFERENCES

1. G. Bolondi, F. Rocca, and S. Zanoletti, "Automatic contouring of faulted subsurfaces", Geophysics 41(1976)1377-1393.

2. R. Franke, "Scattered data interpolation: Tests of some methods", Mathematics of Computation 38(1982)181-200.

3. R. Franke and G. Nielson, "Smooth interpolation of large sets of scattered data", International Journal for Numerical Methods in Engineering 15(1980)1691-1704.

4. J. Pouzet, "Estimation of a surface with known discontinuities for automatic contouring purposes", Mathematical Geology 12 (1980)559-575.