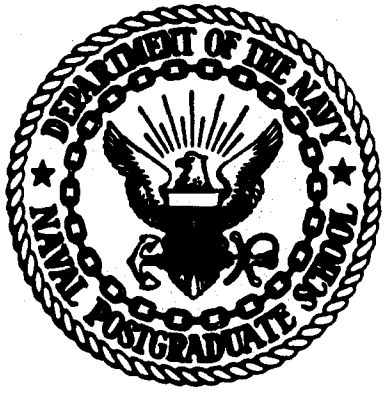
LEVEL II

NPS-53-79-003

NAVAL POSTGRADUATE SCHOOL

Monterey, California



A CRITICAL COMPARISON OF SOME METHODS FOR

INTERPOLATION OF SCATTERED DATA

Richard Franke

Final Report for Period January - March 1979

Approved for public release; distribution unlimited

Prepared for: Chief of Naval Research
Arlington, VA 22217

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Tyler F. Dedman
Superintendent

Jack Borsting
Provost

The work reported herein was supported by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

This report was prepared by:

Richard Franke
RICHARD FRANKE
Associate Professor
Department of Mathematics

Reviewed by:

Frank D. Faulkner
FRANK D. FAULKNER, Acting Chairman
Department of Mathematics

Released by:

William M. Tolles
W. M. TOLLES
Dean of Research

(9) Final rept. Jan-Mar 79

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

(14) 1. REPORT NUMBER: NPS-53-79-003

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

(6) 4. TITLE (and Subtitle): A Critical Comparison of Some Methods for Interpolation of Scattered Data

5. TYPE OF REPORT & PERIOD COVERED: Final January-March 1979

6. PERFORMING ORG. REPORT NUMBER

(10) 7. AUTHOR(s): Richard Franke

(12) 8. CONTRACT OR GRANT NUMBER(s): 378

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Mathematics
Naval Postgraduate School
Monterey, CA 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
61152N; RR 000-01-01
N0001479WR90027

11. CONTROLLING OFFICE NAME AND ADDRESS
Chief of Naval Research
Arlington, VA 22217

(11) 12. REPORT DATE: Dec[illegible]79

13. NUMBER OF PAGES: 378

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
(16) RR0001
(17) RR000101

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Scattered data interpolation
Local interpolation
Bivariate interpolation
Smooth interpolation
Multivariate approximation
Interpolation
x sub k
y sub k
f sub k

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is concerned with methods for solving the scattered data interpolation problem: Given points (x_k, y_k, f_k), $k = 1, \ldots, N$, construct a smooth function, $F(x, y)$, so that $F(x_k, y_k) = f_k$, $k = 1, \ldots, N$. A comparison of 29 methods for solution of this problem has been made. Each of the methods is discussed and the results of extensive testing for their properties and appropriate values of their parameters is given. Both local and global methods are considered. Comparisons of timing, storage, accuracy, visual pleasantness

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

251450

of the surface, and ease of implementation are made. A large number (over 200) of pages of perspective plots of surfaces are given. Suggestions for improvement of some methods are made, and methods which have poor approximation properties are identified.

TABLE OF CONTENTS

1.0.0. Introduction

The basic problem which is being addressed here is that of constructing a smooth (at least continuous first partial derivatives) bivariate function, $F(x, y)$, which takes on certain prescribed values, $F(x_k, y_k) = f_k$, $k = 1, \ldots, N$. The points (x_k, y_k) are not assumed to satisfy any particular conditions as to spacing or density, hence the term "scattered". It is usually convenient to think of the values f_k as arising from some underlying (not necessarily known) function $f(x, y)$, so that $f_k = f(x_k, y_k)$, $k = 1, \ldots, N$.

The problem of interpolation of scattered data in two or more independent variables has been addressed by numerous authors, as can be seen by the bibliography. Many of the basic ideas involved are discussed in two survey papers (both over a wider class than we consider here) due to Schumaker [49] and Barnhill [4]. Some of the ideas seem to be mainly that, ideas, with only a few numerical examples given, often not well thought out or very definitive in terms of the actual capabilities of the method. In addition, most of the methods involve one or more ad hoc assumptions requiring a user to specify parameters (one or more). Generally only cursory attention has been paid to appropriate choice of these parameters and their overall effect on the interpolant has usually not been determined.

Out of this situation arose a desire to attempt to answer a number of questions, basically all related to the question: Which of these many methods deserve further study and development, and which should be discarded? Included here is the determination of some default values for ad hoc parameters in methods which require them. The default values should give reasonably good results over a number of different sets of data, and preferably the interpolant should be rather stable with respect to changes in the parameter. Additionally,

it is convenient for the user if the parameter is related to something about the data which can be easily estimated. In many cases (perhaps all), subjective judgements must be made about these matters, although some hard information can be obtained.

Some previous fairly estensive work had been done by McLain [39] which inspired a somewhat similar study of another class of ideas by the current investigator [16]. The initial thrust of the investigation was to compare a few "local" methods to determine which seem to work reasonably well. As the investigation proceeded, more ideas were supplied by colleagues and others, so that in the end, more than a few methods are tested and compared here, including "global" methods. The total number of programs involved in this study is 29, some of which are fairly minor variations of others.

The concept of a "global" method is easily understood. The interpolant is dependent on all data points, and addition or deletion of a data point, or a change of one of the coordinates of a data point will propagate throughout the domain of definition. The idea of a "local" method is not so clear. Typically one thinks of it meaning that addition or deletion of a point, or a change of one of the coordinates of a data point will affect the interpolant only at nearby points, that is, the interpolant will be unchanged at distances greater than some given distance. There are some difficulties here. If the data (the (x_k, y_k) points) are "random", one must inspect (in some way) all the data to determine which are "nearby". Does this mean there is no such thing as a "local" method? (Rosemary Chang first mentioned this idea). We have taken a somewhat more liberal view of "local" and take it to mean that the interpolant involves only "nearby" points and one or more parameters. We allow the parameters to have been globally determined as a matter of user

convenience, even though a (successful) argument can be made that then the method is not local. Thus, we classify methods as local or global without regard to how parameters are chosen or computed.

The use of global methods is not feasible for very large N since they often involve the solution of a system of O(N) equations (often exactly N) and in any case involve processing all points. When systems of equations must be solved, the systems are often full and not well conditioned. While our primary aim was to investigate local methods suitable for very large data sets (several hundred points up to some millions, say), in many instances local methods involve the use of global methods on smaller sets which are then "blended" together to obtain a locally defined global interpolant. Thus it makes sense to test global methods on moderately sized sets of data. By the same token, it is not necessary to test local methods on sets of 10000 points (say) by virtue of the fact that they are local. If very large sets of data were to be considered, it is clear that a different implementation approach might be necessary, one which would involve a larger amount of pre-processing and perhaps additional storage.

1.1.0. Tested Characteristics of Methods

The characteristics on which various methods are to be compared, and how they are to be weighted in the final analysis, are somewhat subjective. While no representation is made that the list is exhaustive (or even close to it), nor that everyone will be in agreement on it, the following items are the ones considered here. We give them and discuss them in order of decreasing importance. In the presentation of information in the summary (tables and perspective plots) each reader may weight various aspects to suit his own needs.

1.1.1. Accuracy

Accuracy in reproducing a known surface is certainly one important aspect of comparison. In the usual application no representation of the underlying surface $z = f(x, y)$ is known, however, if the method approximates a variety of surface behavior faithfully we can expect it to give reasonable results in other instances. Quantitative numbers can be put on the performance of a method tested in this fashion, and we have used this idea extensively.

1.1.2. Visual Aspects

It has developed during the course of this project that the appearance of the interpolant when viewed in a perspective plot is very important. Visual ratings are often closely related to the accuracy with which an interpolant reproduces test surfaces. There seems to be a closer relationship when accuracy is high since there is less chance for the interpolant to misbehave. At moderate accuracies one interpolant may be visually pleasing while another with similar accuracy is not.

The visual aspect is quite subjective and ratings by different persons will give somewhat different results, although probably not contradictory ones. While it is felt that the visual aspect is quite important, exactly how this information is integrated into the overall assessment of a method is also a subjective matter, however it is rare that a dilemma occurs in this study.

1.1.3. Sensitivity to Parameters

Many of the tested methods involve the choice of one or more parameters. These choices have generally been converted to ones which are related to mean distances to nearest neighbor, although precisely that idea is never directly used. Here we are talking of nearest neighbor in the set of points $\{(x_k, y_k)\}$. Sometimes the parameter takes the form of an anticipated number of points in the region which defines a local interpolant.

Methods which involve parameters underwent informal testing for suitable values of the parameters. Methods which survived this and other tests have parameter variation tests tabulated in the results. Some methods were found to be capable of generating creditable results for an appropriate value of the parameter, but were sensitive to it, or gave poor results on similar data when the same value was used. These results are mentioned in Section 3. It is desirable to have a method which is stable with respect to changes in the parameter, and such methods were found, as we note later.

1.1.4. Timing

The computational effort required is generally not of great interest, unless it is very high. In these respects, only one method was tested which was discounted for this reason. Some methods are quite efficient in terms of time required for the calculations. These methods have generally been found deficient in other categories, unfortunately. For methods which involve a preprocessing phase, distinct from an evaluation (of the interpolant) phase, the two times for standard problems are given separately. Execution times were taken from the multiprogramming environment on the IBM 360/67 and as such may vary considerably with exactly the same data. More is said of this later.

1.1.5. Storage Requirements

As with computational effort, storage requirements are not crucial, unless they are very high. For very large problems this may be altered, of course. We count storage requirements only in terms of additional arrays needed to store data beyond the (x_k, y_k, f_k) points. No account is taken of simple variables or program length.

1.1.6. Ease of Implementation

Ease of implementation is of no great concern if one obtains a working program. In other instances it may be of considerable importance. The judgement is again subjective. Further, it could be different depending on the philosophy behind the implementation. The form of the implementation could involve trade-offs between timing and storage and would doubtlessly alter the ease of implementation.

Implementation of programs specifically for this project generally was done with a lack of frills. Reasonable care was taken to assure that a grossly inefficient algorithm was not coded, but no doubt it is possible to improve on most of them. In particular, use of some preprocessing and additional storage was not used to increase efficiency during the evaluation phase. For a general purpose program this should probably be done, in many instances. Some of the documented programs did use these devices. Ease of implementation is generally meant to take into account the complexity of the ideas involved in the method and the amount of code required.

1.2.0. The Testing Process

The initial tests performed on a few methods eventually gave rise to a standard set of test problems and a set of supporting subprograms to generate statistics from the tests and generate perspective plots of surfaces. Due to the evolution of ideas as the study progressed, some aspects of the process are not as simple as they might have been. This is particularly true of some of the test functions, but this has no bearing on the validity of the tests.

1.2.1. The Test Program

To enable testing many different methods in a consistent manner, and with a minimum of effort, a set of standard subprograms was developed which generate the test cases, compute deviation statistics for known test surfaces, obtain timing statistics, and generate and label perspective plots of the surfaces. With the current set of supporting subprograms it is generally quite easy to test a new method which is typically supplied as a subprogram (or several) which generates the values of the interpolant at a grid of x-y points. Typically all that is required is to set certain parameters, reserve any required workspace, and call the subroutine, all of which can be done with a few statements added to the prototype driver program.

1.2.2. The Test Problems

The basic set of test problems consisted of six different test functions over three different x-y point sets, and two x-y-z point sets from the literature, one of those used in a second version with one of the coordinates scaled. Another interesting test was the computation of a "cardinal" function obtained by setting all function values on a point set to zero, save one.

The six test functions were all to be approximated on $[0, 1]^2$. Four of them were basically obtained from McLain's paper [39], but were translated to $[0, 1]^2$ from $[1, 10]^2$ and some modified slightly to enhance the visual aspects of the surface. The other two were generated by the author to provide a fundamentally different shape in one case (saddle), and to provide a surface with a variety of behavior on one surface to serve as a principal test function.

The principal test function is given by

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

$$+ .5 \exp\left[- \frac{(9x-7)^2 + (9y-3)^2}{4}\right] - .2 \exp\left[- (9x-4)^2 - (9y-7)^2\right].$$

This surface consists of two Gaussian peaks and a sharper Gaussian dip superimposed on a surface sloping toward the first quadrant. The latter was included mainly to enhance the visual aspects of the surface, which is shown in Figure 4.0.1.0.

The second test function, essentially obtained from McLain is

$$f_2(x, y) = \frac{1}{9}[\tanh(9y - 9x) + 1].$$

This surface consists of two nearly flat regions of height 0 and $\frac{2}{9}$, joined

by a sharp rise, almost a cliff, running diagonally from (0, 0) to (1, 1). The test surface is shown in Figure 4.0.2.0.

The third test function was generated by the investigator and is

$$f_3(x, y) = \frac{1.25 + \cos(5.4y)}{6[1 + (3x - 1)^2]}.$$

This surface is saddle shaped and is shown in Figure 4.0.3.0.

The fourth test function, essentially obtained from McLain, is

$$f_4(x, y) = \frac{1}{3} \exp[- \frac{81}{16}((x-\frac{1}{2})^2 + (y-\frac{1}{2})^2)].$$

This surface is a Gaussian hill which slopes off in rather gentle fashion in $[0, 1]^2$. It can be seen in Figure 4.0.4.0.

The fifth test function was also essentially obtained from McLain and is

$$f_5(x, y) = \frac{1}{3} \exp[- \frac{81}{4}((x-\frac{1}{2})^2 + (y-\frac{1}{2})^2)].$$

This surface is a steep Gaussian hill which becomes almost zero at the boundaries of the unit square. It can be seen in Figure 4.0.5.0.

The sixth test function is also essentially from McLain, and is

$$f_6(x, y) = \frac{1}{9}[64 - 81((x-\frac{1}{2})^2 + (y-\frac{1}{2})^2)]^{\frac{1}{2}} - \frac{1}{2}.$$

This surface represents the part of a sphere above the unit square. The sphere is of radius $\frac{8}{9}$ with center at $(\frac{1}{2}, \frac{1}{2}, -\frac{1}{2})$. The surface is shown in Figure 4.0.6.0.

There were three different sets of points over $[0, 1]^2$ used in the tests. The first set consisted of 100 points generated by a pseudorandom number generator, one point in each square of side $\frac{1}{9}$ centered at $(\frac{i}{9}, \frac{j}{9})$ for $i, j = 1, \ldots, 10$. This yields a set of scattered points forced to have

somewhat uniform density, although as can be seen in Figure 0.1.0.0. there are locally large variations in density. The triangulated set of points is also shown in Figure 0.1.0.0. Part of the unit square is outside of the convex hull. The points are listed in Table 1.

The second set of data consists of 33 points and was generated by the investigator to purposely have some areas sparsely populated by points while other areas are not. This set of points is shown in Figure 0.2.0.0. The points are listed in Table 2.

The third set of points was digitized by Gregory M. Nielson and is similar in disposition to a set of points appearing in McLain [40]. This set of points is shown in Figure 0.3.0.0. Part of the unit square is outside the convex hull. The points are listed in Table 3.

Two sets of data were obtained from the literature, and one of these was scaled in one variable to obtain another. A fourth set was used to generate a "Cardinal Function". The data given in Table 3, and shown in Figure 0.3.0.0. was given the following function values: $f(x_k, y_k) = 0$ except $f(.1875, .2625) = .2$. Here .2 was used for visual purposes rather than 1 as would ordinarily be done for a true cardinal function. This gives some information about the influence of one point on the surface for moderate sized point sets. Of the two sets of points from the literature, one is from Akima [1] and was obtained during a study of waveform distortion. It is repeated here in Table 5, and shown in Figure 0.5.0.0. The second was obtained from Ferguson [14] and is repeated here in Table 6, and shown in Figure 0.6.0.0. The same set of data, but with the y coordinate multiplied by three was also used to show effects of scaling only one variable, and is shown in Figure 0.7.0.0. For visual purposes, the function values given in Table 2 are actually .5 more than given by Ferguson. As can be seen from Figure

0.6.0.0. the convex hull of the data is trapezoidal shaped. Since the plotting routines expect function values on a rectangular grid it was decided to evaluate the interpolating surfaces on a rectangle which contained most of the convex hull, but also included a mild amount of extrapolation. The rectangle was [2, 18] x [-3, 3.4] on the original data and [2, 18] x [-9, 10.2] on the modified data. The convex hull of Akima's data is rectangular and this rectangle was used for evaluating the surface points.

The problem of extrapolation outside the convex hull has been addressed by taking the attitude that while it is undesirable to have to do so, it is likely possible to do it in a "reasonable" fashion. Certainly in many instances (our cases mostly among them) one may have better information for mild extrapolation than for some points within the convex hull. The final result is that some programs were modified to extrapolate in a "reasonable" manner, some were implemented that way to begin with, and with others the problem does not arise. Basically only triangle based programs need to address the problem, and among those, only Lawson's [33] program does no extrapolation. Points outside the convex hull were omitted from the deviation statistics in Lawson's method. For the 100 point data set, only 13 points of the 1089 evaluation points were outside the convex hull, and for the 25 point data set 54 points of the 1089 evaluation points were outside the convex hull.

1.3.0. Plot and Table Identification Scheme

The output of this study consists in part of a large number of perspective plots of surfaces and extensive tables. For ease in referencing them they have been gathered at the end of the report, and the entire report has been published in loose leaf form to facilitate reader comparisons of corresponding plots.

The plots have been arranged according to a scheme involving 4 numbers, each of which identifies a particular aspect of the plot. The plot identification is of the form Figure $N_1.N_2.N_3.N_4$, where the N_i are used to identify the characteristics listed below.

N_1 - Type of plot

0 - plot of (x, y) point set

1 - indicates plot has four 3" plots as arranged in Figure 1

2 - indicates plot has four 3" plots as arranged in Figure 2

3 - indicates plot has four 3" plots as arranged in Figure 3

4 - one 6" plot per page

N_2 - Indicates (x, y) or (x, y, z) point set used

0 - does not apply

1 - 100 points as described in Section 1.2.2

2 - 33 points as described in Section 1.2.2

3 - 25 points as described in Section 1.2.2

4 - all of 1, 2, 3 were used as indicated in Figure 1

5 - 50 points from Akima [1], given in Table 7

6 - 25 points from Ferguson [13], given in Table 5

7 - 25 points, obtained as Ferguson's points with y coordinate x 3.

N_3 - Test surface

0 - does not apply

1 - f_1 as defined in Section 1.2.2., $1 \leq i \leq 6$.

N_4 - Program Number, as identified in Section 3, and given in Table S.

a Test Surface	b 100 point interpolant
c 33 point interpolant	d 25 point interpolant

Figure 1

a Cardinal Function	b Akima Surface
c Ferguson Surface	d Modified Ferguson Surface

Figure 2

a Test Surface	b Parameter < Nominal Value
c Parameter = Nominal Value	d Parameter > Nominal Value

Figure 3

The plots all involved evaluation of the interpolant on a 33 x 33 grid of equally spaced points. Generally this grid is fine enough so that piecewise linear plotting of the cross sections, which is the process used by the plot program, yields sufficiently smooth looking results. In some instances this is not really fine enough to show the true character of the surface, but in these cases the surface is not a good approximation to the test surface

and the plot is considered sufficiently accurate to evaluate the visual aspect of the surface anyway.

Tables of comparative results are arranged and labeled according to information contained and test function and data set from which it arose, if pertinent. There are several kinds of tables: (1) Deviation tables, giving the maximum, mean, and root-mean-square deviations over the set of evaluation points used for plotting. These are labeled Table $D.M_1.M_2$, where D indicates "deviation table", $M_1 = i$ means (x, y) data set i, as described for $N_2 = i$, above; and $M_2 = i$ indicates the test function f_i as described for N_3 above. (2) Timing tables, giving the execution times in seconds on the IBM 360/67. These times are divided into preprocessing (for methods for which there is preprocessing), evaluation, and total. All programs were compiled using the Fortran H (optimizing) compiler. Since the configuration of the machine involves multiprogramming, these times are dependent on external factors, and may vary 10% or more, in either direction, on otherwise identical runs. Therefore, times are given to two digits, the second probably not being significant. The tables are labeled Table T.M, where T indicates "timing table" and $M = i$ means for the (x, y) data set as described for $N_2 = i$, above. (3) Parameter variation tables give the deviations for the nominal value of the parameter (for methods involving a parameter), and for values larger and smaller than the nominal value. The tables are labeled Table P.M, where P indicates "parameter variation" and $M = i$ means for the (x, y) data set as described for $N_2 = i$, above. (4) Summary table, Table S summarizes the pertinent information about all tested methods. (5) Two tables compact the information in the deviations tables, indicating only which method (by number) has the smallest deviations for each test

surface and point set. The two tables are for local methods (Table E.1) and all methods (Table E.2).

All tables listing results for various methods are grouped into two or three separate groups. The first group contains extensively tested local methods, the second contains extensively tested global methods, and third (when it appears) contains all other methods.

Certain information in the summary table, Table S needs additional explanation, in particular those given letter grades. Sensitivity to parameters is a purely subjective score, based on informal testing of the scheme. Included were whether some value of the parameter worked well for a variety of surfaces for a given set of (x, y) points, and whether the interpolant was stable with respect to changes in the parameter from that value. Complexity simply reflects the investigator's perception as to the complexity of ideas involved and the ease of implementation into a computer program. Accuracy is again subjective and is based on the relative amount of deviation one might expect from the true surface for a given method. Of course, perusal of the deviations tables will reveal some methods do well on some surfaces and not so well (relatively speaking) on others. Timing is relatively well defined. The first letter represents the sum of the evaluation times, given in Tables T.1, T.2, and T.3. Ranges for A, B, C, D, and F, respectively, are (0, 7], (7, 21], (21, 30], (30, 50], and (50, ∞). The second letter represents the total time for 100 data points and 1089 evaluation points, the time given in Table T.1. Ranges are (0, 4], (4, 12], (12, 20], (20, 30], and (30, ∞)

2.0.0. Descriptions of Tested Methods

For description purposes the methods are classed into six groups: (1) Inverse distance weighted methods, (2) Franke's method, (3) Triangle based blending methods, (4) Finite element based methods, (5) Foley's methods and (6) Nodal basis function methods. While there is necessarily a blurring of distinctions across these group lines, they constitute fairly distinct ideas and it is convenient to group them this way. In the Section headings, the number appearing is the number assigned to the program implementing that scheme. This number has no significance except that it gives the approximate order in which the programs were implemented or obtained. Not all numbers appear because certain ideas were discarded as not within the context of the study (in one case), or as extremely deficient (one case). The programs included in the test and a few words describing it (also used in Section headings in this chapter) are given in Table S to have them available for easy reference.

2.1.0. Inverse Distance Weighted Methods

The original inverse distance weighted interpolation method is due to Shepard [50]. All methods of this type which we consider may be viewed as generalizations of Shepard's method, or variations of such generalizations. The basic Shepard's method is

$$(1) \qquad F(x, y) = \sum_{k=1}^{N} w_k(x, y) f_k \Big/ \sum_{k=1}^{N} w_k(x, y),$$

where $w_k(x, y) = d_k^{-\mu}$, and typically $\mu = 2$, although other values may be used. μ may be replaced by μ_k and could possibly be different for each k. Several authors have considered various aspects of Shepard's method [4], [5], [20], [49].

Shepard's method is a global method, and the original paper suggested a scheme for localizing it by piecing together a parabolic segment with d_k^{-2} in such a way as to obtain a w_k which is zero outside some disk, say of given radius R, centered at (x_k, y_k), and which is still C^1. A simpler and more natural scheme suggested by Franke and Little [4, p. 112] is used in much of this work, that is,

$$(2) \qquad w_k(x, y) = \left[\frac{(R - d_k)_+}{R d_k} \right]^2 .$$

Shepard's method has an undesirable property for general use in that a flat spot occurs at each data point. Use of information about derivatives, either given or generated from the data was suggested by Shepard, and resulted in an approximation of the form

$$(3) \quad F(x, y) = \sum_{k=1}^{N} w_k(x, y)\left[f_k + \left(\frac{\partial f}{\partial x}\right)_k (x - x_k) + \left(\frac{\partial f}{\partial y}\right)_k (y - y_k)\right] \Big/ \sum_{k=1}^{N} w_k(x, y).$$

More generally, one may consider approximations of the form

$$(4) \qquad F(x, y) = \sum_{k=1}^{N} w_k(x, y) L_k f(x, y) / \sum_{k=1}^{N} w_k(x, y),$$

where $L_k f$ is an approximation to f such that $L_k f(x_k, y_k) = f_k$. This is the basis for several of our methods. In this context we refer to the $L_k f$ as nodal functions.

Another way in which Shepard's method can be generalized is to view the method as an inverse distance weighted least squares approximation to $f(x, y)$ by a constant. One can then generalize to an approximation taking the form

$$(5) \qquad F(x, y) = \tilde{F}(a_0, a_1, \ldots, a_n; x, y),$$

where $a_0, \ldots, a_n$ are parameters chosen by taking them to minimize (for a given (x, y)) the expression

$$\sum_{k=1}^{N} [f_k - F(a_0, a_1 \ldots, a_n; x_k, y_k)]^2 w_k(x, y).$$

This approach was taken by McLain [39] in evaluating a number of methods where $\tilde{F}$ was taken as a linear combination of low order monomials and $w_k(x, y)$ as d_k^{-2} or $\exp(-\alpha d_k^2) d_k^{-2}$. McLain also considered some approximations where f entered nonlinearly. We have considered one of McLain's methods and a variation of another. All of the methods of this class may be derived as variations of the above formula for $\tilde{F}$[18].

2.1.1. Shepard's Method (18)

We consider Shepard's method mainly to show how the original method performs in comparison with variations. The formula is described by Equation (1), but was achieved computationally as a special case of the Modified Shepard's Method by taking R very large.

2.1.2. Modified Shepard's Method (7)

This variation is obtained by using the weight-function

$$w_k(x, y) = \left[\frac{(R - d_k)+}{Rd_k}\right]^{-2}$$

in place of d_k^{-2}. In general R can be different for each k, but we have not done this. In order to simplify the choice of R and to remove effects of scaling from the procedure, R is actually computed from the expression

$$R = \frac{1}{2}\sqrt{\frac{N_w}{N}}\, D, \tag{6}$$

where D is the diameter of the point set $\{(x_k, y_k)\}$ and N_w is a new parameter to be specified. Geometrically N_w represents the anticipated number of points which will be in a disk of radius R. Computational experiments have led to a nominal (or default) value of $N_w = 12$. For point sets of widely varying density this is probably not an appropriate value, since the use of constant R for all k assumes a somewhat uniform distribution.

2.1.3. Modified Linear Shepard's Method (3)

This variation is obtained by taking $L_k f$ in Eq. (4) to be the inverse distance weighted least squares approximation to the (x_j, y_j, f_j) by a plane with weight given by $\left[\frac{(R - d_k)_+}{Rd_k}\right]^2$, and the weight function is that given by (2). The comments regarding the choice of R in the previous method apply here as well, including the nominal (or default) value of $N_w = 12$. The coefficients of the plane are obtained in a preprocessing phase.

2.1.4. Modified Shepard's Method Boolean Sum Plane (2)

Barnhill and Gregory [6] have shown that the operator $P \oplus Q = P + Q - PQ$ has the interpolation properties of the operator P and the precision of operator Q. A suggested scheme for obtaining polynomial precision for Shepard's method is to use an operator with linear precision for Q, while P is taken as Shepard's Method. We have used the following scheme. Let $L(x, y) = a(x, y)x + b(x, y)y + c(x, y)$ represent the approximation to $f(x, y)$ obtained by a least squares approximation with weight $(R - d_k)_+^2$ for the kth point, and let Sf represent the Modified Shepard's Method operator of Section 2.1.2. where R above is the same as in Section 2.1.2. Then the approximation is $F(x, y) = S \oplus Lf(x, y)$. Computationally this is achieved by $S \oplus Lf = S(I - L)f + Lf$, or

$$(7) \quad F(x, y) = \sum_{k=1}^{N} W_k(x, y)(f_k - L(x_k, y_k)) / \sum_{k=1}^{N} W_k(x, y) + L(x, y).$$

Thus the values $L(x_k, y_k)$ are computed as a preprocessing step, and the two terms in Eq. (7) are computed for the given (x, y).

2.1.5. Modified McLain Method $M_8(8)$

McLain's method M_8 [38] is of the form given in Eq. (5) with $F(a_0, a_1, a_2; x, y) = a_0(x, y) + a_1(x, y)x + a_2(x, y)y$ and inverse distance weighting d_k^{-2}. We have modified this by taking weighting given by $\left[\frac{(R - d_k)_+}{Rd_k}\right]^2$, where R is again computed from expression (6) with a nominal value of $N_w = 12$.

2.1.6. Quadratic Shepard's Method (17)

This method is of the form given by equation (4) where $w_k(x, y) = d_k^{-2}$ and $L_k(x, y)$ is the inverse distance weighted least square quadratic at the point (x_k, y_k) with weight d_k^{-2}. The coefficients of the quadratics are obtained in a preprocessing phase. This method is actually treated as a special case of the next method with R and R_q taken very large.

2.1.7. Modified Quadratic Shepard's Method (14)

In this method the weights for obtaining the nodal functions (quadratics) are taken as

$$\left[\frac{(R_q - d_k)_+}{R_q d_k}\right]^2 ,$$

where R_q bears the same relationship to N_q as R to N_w in Eq. (6), i.e., $R_q = \frac{1}{2}\sqrt{\frac{N_q}{N}}\, D$. The nominal value for N_q was determined by computational experiments and is $N_q = 18$. If fewer than 6 points lie in a disk of radius R_q at some (x_k, y_k), the approximation is taken to be linear. In any case nonuniqueness of the nodal functions is avoided by using the pseudo-inverse, obtaining the least squares approximation which has minimum ℓ_2 norm of the coefficients. The weight function is given by equation (2), and R obtained from N_w with the nominal value of $N_w = 9$. Complete details are given in [18].

2.1.8. McLain's Method M_{10} (5)

McLain's Method M_{10} [39] is of the form given by Eq. (5) where $F(a_0, \ldots, a_5; x, y) = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2$, and the weight function for the approximation is taken to be $\exp(-\alpha d_k^2)d_k^{-2}$. Here, in order to remove the effects of scaling we have taken

$$\alpha = \frac{1.62N}{D^2},$$

where D is the diameter of the point set $\{(x_k, y_k)\}$. This choice yields $\alpha = 1$ in McLain's original numerical experiments, where McLain suggested α should be something like the usual distance to the nearest neighbour. Experiments have confirmed that the above α is a reasonable choice in a variety of instances. This is a global method.

2.2.0. Franke's Method

The class of methods [16] was inspired by a short paper by Maude [37] which generalized the idea of deficient quintic splines to several variables. Unfortunately the original interpolation function exhibits rather poor behavior and has not even been included in our tests. The original idea was to represent the interpolation function as

$$F(x, y) = \sum_{k=1}^{N} W_k(x, y)Q_k(x, y) \Big/ \sum_{k=1}^{N} W_k(x, y), \tag{8}$$

where $Q_k(x, y)$ is the quadratic polynomial interpolating $f(x, y)$ at (x_k, y_k) and the five nearest neighbors to (x_k, y_k) from the set $\{(x_j, y_j)\}$, and

$$W_k(x, y) = \begin{cases} 1 - \left(\frac{d_k}{R_k}\right)^2 \left(3 - 2\frac{d_k}{R_k}\right) & d_k \le R_k \\ 0 & d_k > R_k \end{cases},$$

where R_k is the distance between (x_k, y_k) and its 5th closest neighbor. This idea was generalized to include any $W_k(x, y)$ which have finite support (to make the method local) so long as the $Q_k(x, y)$ interpolate $f(x, y)$ at all (x_j, y_j) where $W_k(x_j, y_j) \neq 0$. Use of approximations $Q_k(x, y)$ in Hilbert spaces, particularly in Sard spaces, was suggested and implemented [17]. One of the chief advantages of this approach is that instead of taking W_k with disks centered at the (x_k, y_k) as support regions, it is easy to use a smaller number of overlapping rectangles in such a fashion that at most four terms in the sum are nonzero, and $\sum W_k(x, y) \equiv 1$. Use of rectangles also simplifies the problem of determining which terms are nonzero and thus results in a faster algorithm. In general, schemes of this sort are given by $F(x, y) = \sum_{\ell} W_\ell(x, y)Q_\ell(x, y)$, with $\sum W_\ell \equiv 1$ and certain interpolation conditions imposed on the Q_ℓ.

The details of the rectangle selection process follow. As an option the user may specify rectangle boundaries, however an automatic selection process is available and is assumed to be the usual option. A parameter NPPR (for number of points per rectangle) is specified, the suggested value being NPPR = 6. In the automatic case we take $n_x = n_y = [2\sqrt{N/NPPR} - \frac{1}{2}]$ and then grid lines in the x direction at $\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_{n_x+1}$ are chosen so that each strip $(x_{i-1}, x_i)\times(-\infty, \infty)$ contains approximately $N/(n_x + 1)$ points. A similar partition $\tilde{y}_0, \tilde{y}_1, \ldots, \tilde{y}_{n_y+1}$ is found in the y direction. Now, weight functions $W_{ij}(x, y)$ with support $[\tilde{x}_{i-1}, \tilde{x}_{i+1}]\times[\tilde{y}_{j-1}, \tilde{y}_{j+1}] = R_{ij}$ are used together with $Q_{ij}(x, y)$ which satisfy $Q_{ij}(x_k, y_k) = f_k$ whenever $(x_k, y_k) \in R_{ij}$ to form the interpolation function

$$F(x, y) = \sum_{i,j} W_{ij}(x, y)Q_{ij}(x, y). \tag{9}$$

Here we choose the W_{ij} so that $\sum_{i,j} W_{ij} \equiv 1$.

Recently, some work due to Junkins, Jancaitus, and coworkers [31], [33] has come to the investigator's attention. This work involves the idea of weighted local approximations in a similar fashion, and was applied to the problem of terrain modeling. In their case the local interpolation functions were replaced by least squares approximations by polynomials and thus interpolation was not achieved.

2.2.1. Franke's Method (Mode One) (6)

In this method the Q_{ij} of Equation (9) are taken as the optimal interpolation function in the Sard corner space $B_{[2,2]}$. These functions are bicubic spline functions and have continuous second derivatives except along two lines $x = a$ and $y = b$. By taking (a, b) outside of the rectangle R_{ij} the function Q_{ij} is then C^2 on R_{ij}. To preserve the approximation under scaling (not necessarily the same in each variable) the optimal approximation is computed after R_{ij} is transformed to $[0, 1]^2$. At least three interpolation points are used, nearest points (in the ℓ_∞ norm) being added if necessary.

To preserve the continuity of the second derivative it is necessary to take W_{ij} with continuous second derivatives. Thus the choice of

$$W_{ij}(x, y) = V_i(x)U_j(y), \text{ where}$$

$$V_1(x) = \begin{cases} 1 & , x < \tilde{x}_1 \\ H_5\left(\dfrac{x - \tilde{x}_1}{\tilde{x}_2 - \tilde{x}_1}\right) & , \tilde{x}_1 \le x < \tilde{x}_2 \\ 0 & , x \ge \tilde{x}_2 \end{cases}$$

$$V_i(x) = \begin{cases} 0 & , x < \tilde{x}_{i-1} \\ 1 - V_{i-1}(x) & , \tilde{x}_{i-1} \le x < \tilde{x}_i \\ H_5\left(\dfrac{x - \tilde{x}_i}{\tilde{x}_{i+1} - \tilde{x}_i}\right) & , \tilde{x}_i \le x < \tilde{x}_{i+1} \\ 0 & , x \ge \tilde{x}_{i+1} \end{cases}$$

for $i = 2, \ldots, n_x - 1$, and

$$V_{n_x}(x) = \begin{cases} 0 & , x < x_{n-1} \\ 1 - V_{n_x-1}(x) & , \tilde{x}_{n_x-1} \le x < \tilde{x}_{n_x} \\ 1 & , x > \tilde{x}_{n_x} , \end{cases}$$

is made, where $H_5(s) = 1 - s^3(6s^2 - 15s + 10)$ is the Hermite quintic satisfying $H_5(0) = 1$, $H_5'(0) = H_5''(0) = H_5(1) = H_5'(1) = H_5''(1) = 0$.

2.2.2. Franke's Method (Mode Three) (1)

Because the optimal approximations in $B_{\lceil 2,2 \rceil}$ have no polynomial precision, another choice for local approximating functions Q_{ij} is available. In this case the Q_{ij} are taken to be the optimal approximation in $B_{\lceil 2,2 \rceil}$ boolean sum the least squares (unweighted) plane fit to all data points in R_{ij}. Since the latter process has linear precision, so does the overall approximation. The process is implemented as

$$B \oplus Lf = B(I - L)f + Lf, \text{ where}$$

B is the optimal approximation and L is the least squares plane fit.

The choice of rectangles and weight functions is identical to that of the previous section.

2.2.3. Franke's Method (Thin Plate Local Functions) (24)

The elegant theory and excellent fitting characteristics of the thin plate approximations given by Duchon [11], (see Schumaker [47] and Section 2.5.4.) lead to their consideration as local approximations in the basic method given by Equation (9). Several other modifications were incorporated as well.

The suggested number of subrectangles remains the same. However the selection of grid lines $\tilde{x}_i$ and $\tilde{y}_j$ is done in a way which preserves symmetry under reflections and also will result in a symmetric interpolant if the data itself is symmetric.

In selecting points to be interpolated by the Q_{ij} a slightly larger rectangle than R_{ij} is considered by including all points in the rectangle $[-.1125, 1.1125]^2$ after R_{ij} has been transformed to $[0, 1]^2$. This rectangle has area approximately 50% larger than unity and interpolation on the larger set of points tends to make the transition between regions somewhat smoother. This choice was made on the basis of computational experience. Again, at least three points must be interpolated and the nearest points (in the ℓ_∞ norm) are added if necessary.

Experience has shown that many C^1 surfaces appear to be smoother than C^1 in that second derivative jumps are apparently small. While the thin plate approximations have discontinuous second derivatives at the data points, the former reason is the primary one for using $H_3(s) = 1 - s^2(3 - 2s)$ in place of $H_5(s)$ in the definition of the W_{ij} for this method.

The local approximations have the form

$$Q_{ij}(x, y) = \sum_{k \in I_{ij}} A_{i,j,k} d_k^2 \log d_k + a_{ij}x + b_{ij}y + c_{ij}$$

where I_{ij} is the set of indices k for which (x_k, y_k, f_k) is a point to be interpolated by Q_{ij}. See Section 2.5.4. for a further discussion of thin plate splines.

2.3.0. Triangle Based Blending Methods

These methods are conceptually the same as given by Equation (4), but a significant difference is that the weight functions are based on a triangulation of the convex hull of the point set $\{(x_k, y_k)\}$. Several such schemes have been proposed, e.g. [9], [18], [19], and [40]. One of those considered here is the one described in [18].

Assume a triangulation of the convex hull, and suppose $(x, y) \in \bar{T}_{ijk}$ where $\bar{T}_{ijk}$ is the triangle with vertices (x_i, y_i), (x_j, y_j), and (x_k, y_k). We then take

(10) $F(x, y) = W_i(x, y)Q_i(x, y) + W_j(x, y)Q_j(x, y) + W_k(x, y)Q_k(x, y)$

where the weight functions are finite element "shape" functions satisfying $W_m(x_\ell, y_\ell) = \delta_{m\ell}$ and $Q_\ell(x_\ell, y_\ell) = f_\ell$ for $m, \ell = i, j, k$. In all previously referenced methods the weight functions may be viewed as nine parameter cubic shape functions with a rational correction to obtain normal derivatives equal to zero, and hence a C^1 approximation overall. There are many ways to obtain such correction terms, all of which appear to lead to the possibility of negative values being taken on by one of the weight functions if the triangle is very obtuse. This is probably not serious, although one has no control over the shape of the triangles in the sense that very obtuse angles cannot be avoided in some instances. The weight functions used here are obtained from a minimum norm problem [43]. Let b_i, b_j, b_k be the barycentric coordinates of (x, y) in $\bar{T}_{ijk}$, and let ℓ_i, ℓ_j, and ℓ_k be the lengths of the sides opposite vertices i, j, and k, respectively. Then the weight function is given by

$$W_k(x, y) = b_k^2(3 - 2b_k) + 6b_ib_jb_k[\alpha_{kj} + \alpha_{ki}]$$

with

$$\alpha_{kj} = \frac{b_kb_j(1 + b_i)}{(1 - b_i)(1 - b_k)}\left[\frac{\ell_k^2 + \ell_i^2 - \ell_j^2}{2\ell_i^2}\right],$$

and the others being obtained by a cyclic permutation of the indices.

While the basic method is defined only on the convex hull of the point set, it is easily extended to be a globally defined function by the following idea. The exterior of the convex hull is divided into semiinfinite rectangles, shown in Figure 4, by constructing perpendiculars to the exterior edges of the convex hull at each exterior vertex.

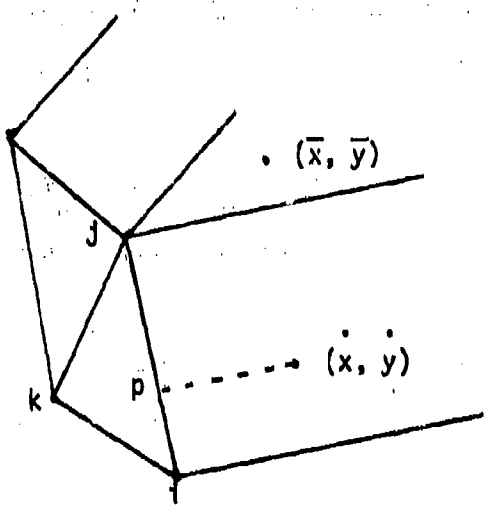


Figure 4

To extend the definition of Equation (10) outside the convex hull we proceed as follows. For a point in an exterior triangle, such as $(\bar{x}, \bar{y})$, we take $F(\bar{x}, \bar{y}) = Q_j(\bar{x}, \bar{y})$. For a point in an exterior rectangle, such as $(\dot{x}, \dot{y})$, let p be the projection of $(\dot{x}, \dot{y})$ onto side ij, and let $(b_i, b_j, 0)$ be the barycentric coordinates of p in $\bar{T}_{ijk}$. Then we take

$$F(\dot{x}, \dot{y}) = h_3(b_i)Q_i(\dot{x}, \dot{y}) + h_3(b_j)Q_j(\dot{x}, \dot{y}),$$

where $h_3(s) = s^2(3 - 2s)$. These extensions yield a globally defined approximation which is C^1.

2.3.1. Nielson-Franke Linear Triangle Method (12)

This method uses the inverse distance weighted least squares plane $Q_k(x, y)$ which is also used in the Modified Linear Shepard's Method. See Section 2.1.3. for details.

Another idea was investigated for determining the slopes to be used in the planar fit, but was abandoned as being a poor idea. Since the idea has been mentioned by a number of persons, it is discussed here. At a vertex k, determine the slopes a_{ij} and b_{ij} of the plane $a_{ij}x + b_{ijy} + c_{ij}$ through the points (x_i, y_i, f_i), (x_j, y_j, f_j), and (x_k, y_k, f_k), where T_{ijk} is a triangle in the triangulation. The nodal function is taken as $Q_k(x, y) = A_k(x - x_k) + B_k(y - y_k) + f_k$, where A_k and B_k are the average values of a_{ij} and b_{ij}, respectively. We can think of this as taking the nodal function at each vertex as the average plane from the piecewise linear approximation on the triangulation. This fails because of the possible existence of long thin triangles. This is especially crucial when the triangle is very obtuse, and the plane through the three points may have very large gradients because the three vertices lie nearly on a straight line while the three points on the surface do not.

2.3.2. Nielson-Franke Quadratic Triangle Method (13)

This method uses the inverse distance weighted quadratic $Q_k(x, y)$ which is also used in the Modified Quadratic Shepard's Method. See Section 2.1.7. for details. A complete description is given in [18].

2.4.0. Finite Element Based Methods

These methods are based on the concept of using C^1 finite element functions on a triangulation of the convex hull of the point set. This requires a scheme for estimating some derivatives (which ones depend on the element used by the method) at the data points. Our test results indicate that accurate estimates of the derivatives are very important and have a pronounced effect on the visual aspects of the surface, particularly, but also the accuracy.

2.4.1. Akima's Method (6)

Akima's method [1] uses the 18 parameter quintic element and thus requires both first and second partial derivatives at each data point. The scheme used is as follows. The user specifies a parameter, n_d. Let $P_k = (x_k, y_k, z_k)$, and form the sum $\vec{V} = \sum \pm \overrightarrow{P_kP_i} \times \overrightarrow{P_kP_j}$, where (i, j), $i \neq j$ ranges over the n_d closest points (x_i, y_i) and (x_j, y_j) to (x_k, y_k), and where the sign is chosen so that the z component of each cross product is positive. The first partial derivatives are taken to be those of the plane normal to $\vec{V}$. The second derivatives are obtained by applying the same process to the derived data. The cross partial is taken as the average of the two so obtained. Akima suggests $n_d = 3$ or 4 as appropriate, but we have found $n_d = 6$ generally works better.

Extrapolation outside the convex hull is achieved by construction of an appropriate polynomial in the exterior rectangle or triangular regions given in Figure 4, and C^1 continuity is maintained. In a triangular region, the conditions at the vertex determine a unique bivariate quadratic. For the rectangular region the conditions at the two vertices determine a unique polynomial of degree two in the direction normal to the boundary segment, matching the quadratic in the adjacent triangular region, and of degree 5 in the tangential direction, matching the value and first two derivatives across the boundary from rectangular to triangular region.

2.4.3. Akima's Method - Modification Two (11)

It was felt that perhaps an inverse distance weighting of the cross products might be desirable, so this scheme formed the vector sum

$$\vec{V}_2 = \sum \pm \frac{\overrightarrow{P_kP_i} \times \overrightarrow{P_kP_j}}{\|P_kP_i\|^2 \|P_kP_j\|^2} .$$

All other aspects of the program were maintained.

2.4.2. Akima's Method - Modification One (10)

It is easily observed that while Akima's scheme for estimating derivatives puts less weight on nearly collinear points, which seems desirable, it also puts more weight on distant points, which does not seem to be desirable. To remove the distance weighting, the scheme was modified to form cross products of unit vectors in the same directions as before, i.e., the sum

$$\vec{V}_1 = \sum \pm \frac{\overrightarrow{P_kP_i} \times \overrightarrow{P_kP_j}}{\|\overrightarrow{P_kP_i}\| \|\overrightarrow{P_kP_j}\|}$$

was formed. All other aspects of the program were maintained.

2.4.3. Akima's Method - Modification Two (11)

It was felt that perhaps an inverse distance weighting of the cross products might be desirable, so this scheme formed the vector sum

$$\vec{V}_2 = \sum \pm \frac{\overline{P_kP_i} \times \overline{P_kP_j}}{\|P_kP_i\|^2 \|P_kP_j\|^2} .$$

All other aspects of the program were maintained.

2.4.4. Akima's Method - Modification Three (16)

This modification incorporated use of the inverse distance weighted quadratic least squares polynomial fit used in the modified Quadratic Shepard method described in Section 2.1.7. The required derivatives were then taken from the quadratic nodal function computed in this manner. All other aspects of the program were maintained.

2.4.5. Nielson's Minimum Norm Network (19)

This scheme [43] uses a cubic element with a rational correction term to obtain C^1 continuity. One of the basis functions is that described in Section 2.3.0. Only first derivatives are required. These are obtained by minimizing the value of $\int[\frac{d^2}{ds^2} F(x(s), y(s))]^2 ds$, where the integral is over the entire network of edges in the triangulation. We note that the interpolation function is a univariate Hermite cubic polynomial along each edge. This is a global method.

The original scheme is not able to extrapolate outside the convex hull, but the following idea was incorporated to achieve extrapolation. In a triangular exterior region as in Figure 4, the function is taken to be the linear function determined by the value and slopes at the vertex. In the rectangular region the function is extended by extrapolating from the projection point, p, with the given slope and value along a straight line. The resulting surface is only C^0 across exterior rectangular to triangular boundaries, but for mild extrapolation this will likely not be noticeable. An appropriate rational correction could probably be made in the triangular area to achieve C^1 continuity.

2.4.6. Lawson's Method (28)

This method is somewhat similar to Akima's in philosophy except for the particular finite element used and the manner of estimating derivatives [34]. The Clough-Tocher cubic element, which requires first derivatives at the vertices, is used. The derivatives are estimated by fitting an inverse distance weighted quadratic at each vertex. The program is presently not set up to extrapolate outside the convex hull of the point set, although a scheme for extrapolation similar to that used in Nielson's program could be incorporated. Time did not permit this, however.

2.5.0. Foley's Methods

Foley's methods [15] involve several ideas. The use of a generalized Newton type interpolant is involved in them prominently and this idea is discussed in Section 2.5.1.

Another idea which is exploited successfully is that of using one interpolant to generate a grid of points on which product type approximations can be constructed. The product approximation will not, in general, interpolate the given data. Hence a correction based on the original approximation is made to the error. This process is termed a "delta sum" by Foley, written $P\Delta Q$, defined by $P\Delta Q = P \oplus Q$, and implemented as $(P\Delta Q)f = P(I - QP)f + QPf$.

The idea has greater generality than considered by Foley, but the application of it seems to be the appropriate one. He considers cases where the product type approximation (taking the part of Q) is either the bivariate product Bernstein polynomial or the bivariate product natural bicubic spline. The first interpolant (taking the part of P) is taken as either the generalized Newton interpolant, or a form of Shepard's method. The delta sum idea is applied in iterated form for two methods.

2.5.1. Generalized Newton Interpolant (25)

The generalized Newton interpolant as considered by Foley takes the form $F(x, y) = T_N(x, y)$, where $T_1(x, y) = f_1$, $T_k(x, y) = T_{k-1}(x, y) + W_k(x, y) \frac{f_k - T_{k-1}(x_k, y_k)}{W_k(x_k, y_k)}$, $k = 2, \ldots, N$, where $W_k(x, y) = \prod_{j=1}^{k-1} L_j(d_j)$, with $d_j = \sqrt{(x - x_k)^2 + (y - y_k)^2}$, and $L_j(0) = 0$, $L_j(t) \neq 0$ if $t \neq 0$. Foley takes $L_j(t) = \frac{t^2}{t^2 + r_j^2}$ where the value of r_j was obtained in an ad hoc fashion as $r_j = \frac{.81}{48} [25d_{j1} + 13d_{j2} + 7d_{j3} + 3d_{j4}]$. Here d_{ji} represents the distance to the $i^{\underline{th}}$ nearest neighbor to (x_j, y_j) in the set of points $\{(x_k, y_k)\}$.

Unfortunately, unlike the univariate Newton polynomial, this function depends on the ordering of the data points. A number of experiments lead Foley to two ordering schemes. Let $(\bar{x}, \bar{y})$ be the centroid of the set $\{(x_k, y_k)\}$, i.e., $(\bar{x}, \bar{y}) = \frac{1}{N} \sum_{k=1}^{N} (x_k, y_k)$. Let $a_k^2 = (\bar{x} - x_k)^2 + (\bar{y} - y_k)^2$, and arrange the points in increasing order of a_k. This is called "inside out" ordering, and the opposite order is called "outside in". The two interpolants based on these orderings are called TIO(f) and TOI(f), respectively, by Foley. Since each appears to work better in the region from which the final points come, i.e., TIO is better in the outer regions, while TOI is better in the central regions, a blending of the two is taken as the final interpolant. The weighting function is given by

$$BL(x, y) = \frac{3}{2} \cdot \frac{(x - \bar{x})^2 + (y - \bar{y})^2}{(x - x)^2 + (y - y)^2 + D^2},$$

where $D^2 = \frac{1}{2} \max_k [(x_k - \bar{x})^2 + (y_k - \bar{y})^2]$. The final interpolant, TF, is then given by

$$F(x, y) = BL(x, y)TIO(f) + (1 - BL(x, y))TOI(f).$$

2.5.2. TF Delta Sum Bernstein Polynomial (26)

The only additional information in the implementation of this scheme is the region on which the data for the Bernstein polynomial is to come (a square in the bivariate case) and the degree of the polynomial. Foley takes the square to be the smallest square containing the set $\{(x_k, y_k)\}$, although notes it might be better to do otherwise in some circumstances. The degree is taken to be 10, although this means a grid of 11 x 11 points is used for the Bernstein approximation. We have followed Foley, but it might be more reasonable to use an m x m grid where $m^2 \approx N$, as is done in the next section. Let the Bernstein polynomial for g(x, y) on (a, b) x (c, d) be denoted by BRN(g). Then the TF delta sum Bernstein polynomial approximation is given by $TF\Delta BRN(f) = TF(f - BRN(TF(f))) + BRN(TF(f))$.

2.5.3. Iterated Delta Sums: TF Delta Sum Bicubic Spline (30)

The TF interpolant is also applied in conjunction with the natural bicubic spline. We again need to give a grid of points on which to compute the bicubic spline. The selection routine supplied by Foley (but not used for the examples in his thesis) is as follows. Let $m = [\sqrt{N}]$ (here $[\cdot]$ denotes the integer part), and $k = [\frac{N}{m} + \frac{1}{2}]$. Sort the x-coordinates so that $x_1 \le x_2 \le x_3 \le \ldots \le x_N$. Then the x-grid lines are given by

$$\tilde{x}_j = \frac{1}{k} \sum_{i=(j-1)k+1}^{k_j} x_i \quad \text{for } j = 1, \ldots, m-1,$$

$$\tilde{x}_m = \frac{1}{N-(m-1)m} \sum_{i=(m-1)m+1}^{N} x_i .$$

The y-grid lines are formed in dual fashion.

We now consider applying the delta sum in iterated fashion to obtain a sequence of operators $G_0, G_1, \ldots$. Let B represent the natural bicubic spline operator on the above grid. Then let $G_0 = TF$, and successively form operators $G_{n+1} = TF \circledcirc BG_n$, $n = 0, 1, \ldots$.

The calculation can then be organized as follows: Compute the current approximation at the grid points; construct the natural bicubic spline interpolant for the grid; correct the spline interpolant for the grid to obtain interpolation at the scattered points by adding in the TF interpolant for the error. Computationally this all amounts to writing $G_{n+1}f$ as $BG_nf + TF(I - BG_n)f$. Under certain conditions the iteration may converge, and can converge to a bicubic spline function which interpolates the original data. In other instances the iteration appears to diverge, unfortunately. We have taken 3 as the nominal number of iterations to be used.

2.5.4. Iterated Delta Sums: A Shepard's Method Delta Sum Bicubic Spline (31)

The basic idea of the previous section was also applied using a modified Shepard's method in place of the TF interpolant. The stable, if somewhat undesirable, behavior of Shepard's method would appear to be well suited for this use.

The basic modification to Shepard's method was one to force a diminished region of influence on the points, taking the weights to be $\frac{r^2}{d_k^2(d_k^2 + r^2)}$, where $r^2 = \frac{\pi D^2}{N}$, $D^2 = \max_j [(x_j - \bar{x})^2 + (y_j - \bar{y})^2]$, and $(\bar{x}, \bar{y})$ is the centroid of the set $\{(x_j, y_j)\}$, as in Section 2.5.1. The modified Shepard's method used here is of the form

$$F(x, y) = \sum_{k=1}^{N} \frac{f_k}{\rho_k(x, y)} \Big/ \sum_{k=1}^{N} \frac{1}{\rho_k(x, y)}, \text{ where}$$

$$\rho_k(x, y) = \frac{r^2}{d_k^2(d_k^2 + r^2)}.$$

The iterated delta sum interpolant is then formed in exactly the same manner as in the previous section, with the modified Shepard operator, SM, replacing TF. Thus we have $G_0 = SM$, $G_{n+1} = SM \circledast BG_n$, $n = 0, 1, \ldots$. We have again taken 3 iterations as the nominal value, and comments regarding convergence/nonconvergence of the iteration of the previous section apply here.

2.6.0. Global Basis Function Type Methods

These methods can be characterized by the following idea. For each (x_k, y_k) simply choose some function $G_k(x, y)$, and then determine coefficients A_k so that $F(x, y) = \sum_k A_k G_k(x, y)$ interpolates the data. Schemes which work are not so simple in that appropriate choices of function G_k are not particularly easy to make. Even if the functions G_k have only local support the methods are global and further they require solution of a system of N linear equations. In all instances we consider, the systems have a symmetric coefficient matrix $(G_i(x_j, y_j))$, but this need not be the case. Usually the G_k are really functions of one variable, $d_k = \sqrt{(x - x_k)^2 + (y - y_k)^2}$. While it seems that functions G_k which diminish as one moves away from the point (x_k, y_k) would be best, this has not been borne out computationally. Numerous colleagues have suggested (among others) B splines, Gaussian distributions, and other basis functions which seem to have an at best shaky mathematical justification. We investigated several methods of this type and have found them to work better than expected. They are, as mentioned, global methods.

2.6.1. Rotated Gaussian (20)

This scheme is mentioned by Bolandi, etal. [7], Arthur [3], and more recently was rediscovered with a slight variation [48]. It consists of using $G_k = \exp(-d_k^2/R^2)$, where R is taken to be a constant for all k. The method is quite sensitive to the choice of R and yields poor results with ease, but will yield quite good results for an appropriate value of R. We used a nominal value of $R = \frac{2.016D}{2\sqrt{N}}$, where D is the diameter of the point set. The factor $\frac{1}{2}\frac{D}{\sqrt{N}}$ represents (approximately) the radius of a disk in which one could anticipate finding one point of the set and in some sense is proportional to the mean distance to the nearest neighbor.

2.6.2. Hardy's Multiquadric (21)

This method has been used extensively by Hardy and his coworkers [22-28], in geographic and related applications. The basis function used is the upper hyperboloid $G_k = ((x - x_k)^2 + (y - y_k)^2 + r^2)^{1/2}$, where r is a parameter which determines the semi-axis of the hyperbola. Hardy [26] indicates that the best value for r is approximately .815d, where d is approximately the mean distance to the nearest neighbor. We have not verified this and have used a nominal value of r = 2.5R, where R is the radius of the disk which could be anticipated to contain one point. The actual parameter used by the program is NPPR and the value of r is computed from $r = \frac{NPPR}{10} R$, $R = \frac{1}{2} D/\sqrt{N}$ where D is the diameter of the point set $\{(x_k, y_k)\}$, and a nominal value of 25 is assumed for NPPR. We observe better results are generally obtained with larger r, but this also leads to poorer conditioning of the coefficient matrix $(G_i(x_j, y_j))$, and we have compromised on the above value which corresponds to approximately 1.6d. Because of the scattered nature of the data this may vary.

2.6.3. Hardy's Reciprocal Multiquadric (27)

For this method the reciprocal hyperboloid $G_k(x, y) = ((x - x_k)^2 + (y - y_k)^2 + r^2)^{-1/2}$ is used. The value of r used is the same as that for the previous method.

2.6.4. Duchon's Radial Cubic Method (22)

Duchon [12] has derived this method as the optimal solution in a certain Hilbert space via construction of the reproducing kernel (see [49] for some details). For practical purposes the user must solve a system of the form

$$\sum_{j=1}^{n} A_j G_j(x_i y_i) + ax_i + by_i + c = f_i,\ i = 1, \ldots, n$$

$$\sum_{j=1}^{n} A_j x_j = 0,\ \sum_{j=1}^{n} A_j y_j = 0,\ \sum_{j=1}^{n} A_j = 0,$$

where $G_k(x, y) = ((x - x_k)^2 + (y - y_k)^2)^{3/2}$. G_k is seen to be of the form d_k^3, where d_k is distance from (x_k, y_k), hence my name for it.

2.6.5. Duchon's Thin Plate Function (23)

This method is similar to the previous one in that it is the optimal solution in some Hilbert space. This one is particularly interesting since over all interpolating functions in the Hilbert space it minimizes the thin plate functional $\iint_{R^2} \left(\frac{\partial^2 F}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 F}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 F}{\partial y^2}\right)^2$. The form of the solution had been previously given by Harder and Desmarais [21]. The method is also discussed by Meinguet [41] and its fitting properties in connection with smoothing has been investigated by Wahba [52].

The system of equations is identical in structure to that of the previous method except that $G_k(x, y) = d_k^2 \log d_k$, where again d_k is the distance from (x_k, y_k).

2.6.6. Rotated B-Splines (29)

This idea has been suggested by various persons. We took the B-spline based on equally spaced points, with knots at $\pm R$, $\pm R/2$, and 0, where R was chosen in a manner similar to other schemes previously described. The nominal value of R was taken as $\frac{4.8384D}{2\sqrt{N}}$, where D is the diameter of the point set, and again the factor $\frac{1}{2}\frac{D}{\sqrt{N}}$ represents the radius of a disk in which one could anticipate finding one point in the set. The basis function was

$$G_k(d_k) = 2(1 - \frac{d_k}{R})_+^3 - (1 - \frac{2d_k}{R})_+^3.$$

3.0.0. Results

The results of the study are discussed in this section in the same sequence as the methods are discussed in Section 2, with corresponding subsection numbers relating to classes of methods. We confine our comments here to absolute merits of each method, insofar as possible, although it is almost certain we are "grading on the curve". Merits of some methods (the better ones or more available ones, in our opinion) versus other methods are discussed in Section 4.

It is hardly possible to discuss the performance of each method on each surface in the detail which would be desirable from a completeness point of view, if not the writer's (and perhaps not the reader's either). In order to point out pertinent behavior which one can look for in various methods, it was decided to discuss in some detail all plots of surfaces for one method. In addition, there are a number of comments about the test surfaces relative to the data sets which apply to all methods. The method dicussed is neither the best, nor the poorest, but simply a method which illustrates some of the behavior one can watch for. The method chosen for discussion is program #30, Foley's iterated generalized Newton delta sum bicubic spline. We discuss each figure separately.

Figure 1.4.1.30. Part a is the test surface. Part b is the interpolant based on 100 points. The peak of the test surface occurs near $(\frac{2}{9}, \frac{2}{9})$ and inspection of the set of points in Figure 0.1.0.0. reveals a relatively large gap in that vicinity. Thus the peak value of the surface is not well defined. Part b shows the left rear of the peak to be the poorest portion of the peak definition. At the left front ofthe surface an undesirable flattening of the surface occurs. Cross sections throughout the interior appear to be quite

smooth. At the rear of the dip there is an extended depressed region. Reference to Figure 0.1.0.0. again reveals that there is a relatively large gap in the points which causes the extent of the dip to be poorly defined. The dip is near $(\frac{4}{9}, \frac{7}{9})$. Figure 4.1.1.30 is a larger plot of the surface in Figure 1.4.1.30b. Part c shows several ill defined portions of the surface corresponding to regions with gaps in the data set, shown in Figure 0.2.0.0. In particular, near the rear edge of the surface behind the peak, and to the right. The dip is completely missed because of a lack of data to define it. The surface basically appears reasonable, except possibly for the behavior at the rear edge near the right. A larger plot of the data is shown in Figure 4.2.1.30. The surface in part d shows basically appropriate behavior. The peak is reasonably well defined, although slightly low. Again, no point is on the top of the peak. The dip is somewhat defined, but spread out because of a lack of nearby points to pull the surface back up. The near corner is somewhat low, however this corresponds to an area of extrapolation. Figure 4.3.1.30 shows a larger plot of the surface.

Figure 1.4.4.30. The test surface is shown in part a, and part b appears almost indistinguishable from it. There is a very slight flattening at the right edge near the center. Part c is also a good approximation, with a somewhat flattened area at the right, in front of the center. While there are many points nearby, there is a relatively large gap along the edge which accounts for poor definition of the surface there. Less noticeable is a slightly raised area to the rear of the center, along the right edge. The most noticeable defect in part d is the poor behavior at the front edge, the surface being slightly high toward the right of the center, and low at the right front corner.

Figure 1.4.5.30. This surface is difficult to fit closely because of its sharply peaked behavior, shown in part a. While the peak in part b is well defined, to the right one observes small "kinks", and generally wavy behavior around the edges, more noticeable at the front and right because of the viewing point. The peak (at $(\frac{1}{2}, \frac{1}{2})$) is poorly defined by the set of points in part c and thus is considerably low. The wavy behavior around the edges is again observed, but somewhat amplified. In part d the peak is higher, but the wavy behavior away from the peak is very pronounced, although the surface is smooth in the sense there are no apparent "kinks" in the surface.

Figure 1.4.6.30. This surface, a part of a sphere, is shown in part a. The surfaces shown in parts b, c, and d show varying amounts of imperfect behavior, mostly appearing as flattened spots on the surface.

Figure 2.0.0.30. Part a shows the cardinal function. The waviness that extends throughout the square is not desirable and is probably an artifact of the underlying polynomial - like interpolant. Part b shows the surface for Akima's data and basically appears reasonable. There is some wavy behavior in the cross section lines near the base of the sharp rise toward the rear. Part c shows a portion of the surface for Ferguson's data. Extrapolation is involved at the front corners and the surface dips a lot toward the right front. The same data scaled by a factor of 3 in the y-direction is shown in part d. Here the surface dips at the left front corner as well and rises at the left rear corner where mild extrapolation occurs. Parts c and d exhibit some of the effects of scaling differently in the two variables.

Figure 3.1.1.30. This figure shows the effects of varying the parameter from its nominal value. For this method the parameter is the number of delta iterates performed. The test surface is shown in part a. Part b is the surface

obtained with the 100 point data set after one iteration. Part c shows the surface after the nominal number of iterations, which is three. Part d shows the surface after five iterations. The surface shows definite improvement with additional iterations, particularly when comparing b and c. Some improvement is seen in d compared to c, particularly in that the peak is filled out, although the flattened portion near the left front persists. The deviation statistics table P.1 shows improvemant in that aspect also.

Figure 3.3.1.30. This figure is the analogue of the previous, but with the 25 point data set rather than the 100 point data set. In this case it is not readily apparent that the surface improves with more iterates, although part c is in some respects more pleasing than part b, since the slight rise toward the rear edge near the right corner has been lessened. However the surface has been depressed at the front edge near the right corner, which continues with more iterations, as seen in part d. The deviation statistics, Table P.3 show improvement in the maximum deviation, but increases in the mean deviation and root mean square deviation after 3 iterations.

Figures 4.n.1.30. These figures are larger copies of the figures given in Figure 1.4.1.30 , and were previously discussed.

Figure 4.1.2.30. This function is probably the most difficult to fit well. There is some irregularity along the sharp rise about three-fourths of the way back along the diagonal. The most obvious defects are near the front corner, on the right, and a waviness along the front edge.

Figure 4.2.2.30. The appearance of large gaps between data points (see Figure 0.2.0.0) leads to a quite wavy surface. With the exception of a few cross sections at about $\frac{1}{3}$ of the way back along the diagonal, the surface is quite smooth, however, with the greatest overshoot occuring near the left

rear and right front corners, as might be expected.

Figure 4.3.2.30. The regularity of the set of data points (see Figure 0.3.0.0.) used here leads to a more regular appearing, if somewhat wavy, surface. The surface appears to be very smooth with no apparent "kinks" observable along the cross-sections. As usual, edge behavior seems to account for the largest deviations.

Figure 4.1.3.30. This surface is almost indistinguishable from the original. The only apparent defect is a slight flattening of the surface at the right edge near the center.

Figure 4.2.3.30. The large gaps in the data show on this surface, but less conspicuously than on some others. The right end of the surface appears depressed, with the front edge also appearing to be lower than the test surface. The surface is quite smooth, however.

Figure 4.3.3.30. This surface is quite smooth and pleasing, but the left rear corner is considerably higher than that of the test surface. The slope of the surface toward the right center seems to be more gentle than that of the test surface .

3.1.0. Inverse Distance Weighted Methods

The performance of schemes within the general class of methods varies a great deal. The basic Shepard's method (program #18) with exponent 2 is unacceptable for a variety of reasons for all but some very special applications (perhaps). For more than a few points the method does not perform as one would be led to believe when observing the method for 5 or 10 points. These are mainly the size examples given in previous literature. As can be seen from the plots, Figures 1.4.1.18 and 2.0.0.18, the surfaces often tend to have sharp peaks and dips at the data points. In fact, the resolution of the plots is not

fine enough to show the exact nature of the surface.

Considerable improvement accrues from localizing the method (program #7). This could likely be accomplished in a variety of ways other than the approach we have taken, for example by using an appropriate (larger) exponent. No experimentation was done in this direction. The plots for this program, Figures 1.4.1.7 and 2.0.0.7 show clearly the increased influence of a data point on the surface nearby. This is especially evident in the cardinal function plot where the influence of a single point is seen. Basically, localizing the scheme causes the well advertised flat spots to become more prominent and the surfaces to become more pleasing. However, for general purpose interpolation the scheme is still basically unacceptable.

Forming boolean sums with other approximations does not seem to work well in this case. It appears that for the idea of boolean sum approximations to work the second approximation in the boolean sum must be a good approximation itself. The least squares plane used in our program (#2) is not suitable since it will consistently allow undershoot near peaks and hence appears to have flat spots (not necessarily with zero slopes) at points where this occurs. This is particularly noticeable in Figure 1.4.1.2, less so in Figure 2.0.0.2. The alternative to the least squares plane, a higher degree approximation such as a quadratic would likely work very well, but is quite expensive as we will note later.

Another way to generalize Shepard's method is through the use of more information about the surface near the data points. Use of the least squares plane passing through each data point in conjunction with a local weighting function (program #3) leads to an improved surface, however the surfaces often tend to look somewhat lumpy, as can be seen from the plots, Figures 1.4.1.3 and 2.0.0.3. It is particularly bad in Figure 1.4.1.3c, probably because of the

(intended) varying sparseness of the data points. The use of planar fits does not seem adequate.

The use of a quadratic function passing through each data point (program #17) leads to virtually no improvement over the basic Shepard's method due to the influence of "far away" points. The plots are shown in Figures 1.4.1.17 and 2.0.0.17, and are similar to these for program #18, although these are generally somewhat nicer in that the number of sharp peaks is reduced.

The use of a quadratic least squares fit at each data point in conjunction with localization of the weights (program #14) leads to a significant improvement over other methods of this type in most (not all) instances, especially for larger numbers of data points. For small numbers of data points the surfaces seem to be adversely affected by what might be termed "edge effects". This is not unique to this scheme but occurs with other methods, particularly local methods. More will be said of this in Section 4. The plots shown in Figures 1.4.n.14, 2.0.0.14, 4.1.n.14, 4.2.n.14, and 4.3.n.14., show some of this, and a particularly good illustration is given by the cardinal function plot, 2.0.0.14a, which can be seen to behave in unseemly fashion near the left rear corner. Within this class of tested methods, the modified quadratic Shepard method is undoubtedly the best performer overall. The effect of changing the parameters in the method are shown in Figures 3.1.1.14 and 3.3.1.14. A greater tolerance to changes in the parameter is seen for the larger data set, while edge effects are more prominent as the radius of influence is decreased, particularly for the smaller data set.

The other approach to modifying Shepard's method is that taken by McLain [39], and implemented here in program #5 for the quadratic approximation. From the point of view of fitting, the method works quite well, although it may be

more prone to "edge effects" as can be seen from the plots, Figures 1.4.1.5 and 2.0.0.5, especially 2.0.0.5 a and c. The main reason for discounting the method is the rather high computational burden. The modified quadratic Shepard's method yields results as good or better, but at much less cost.

The second McLain type of interpolant was the fit with a linear function, but with modified weights (program #8). Because of the necessary undershoot near peaks one can not expect the method to perform in a satisfactory fashion. As we see from the plots, Figures 1.4.1.8 and 2.0.0.8, we are not disappointed in that respect. Overall the surfaces appear to be quite lumpy and generally unacceptable. The cardinal function plot shows a somewhat more peaked function than might be expected, almost like Shepard's method.

3.2.0. Franke's Method

The performance of this class of methods is somewhat uneven, giving quite reasonable results in some cases and not in others. Edge effects seem to come into the method prominently giving poor results for data sets with small numbers of points. When the local approximations are optimal approximations in $B_{[2,2]}$ (program #6) the plots are shown in Figures 1.4.1.6 and 2.0.0.6. Typical edge effects are seen in Figure 2.0.0.6a, the cardinal function.

When the optimal approximations are taken boolean sum with a least squares plane (program #1), the resulting surfaces are virtually unchanged, as can be seen in Figures 1.4.1.1 and 2.0.0.1. Thus it appears that the use of the boolean sum with the plane is mainly to incorporate polynomial precision, and its use with a reasonably good approximation will not improve it very much. This observation was also made by Foley in his thesis [15]. Additional plots are given in Figures 1.4.n.1, 4.1.n.1, 4.2.n.1, and 4.3.n.1, and basically show that the method performs competently for larger data sets and not so well on smaller ones. The variation of the parameter in the method,

Figures 3.1.1.1 and 3.3.1.1 show the effects of localizing too much are drastic (plot b), whereas the reverse has lesser overall impact.

The use of thin plate local functions (program #24) generally results in much more pleasant appearing approximations. This is particularly true for smaller numbers of data points as can be observed in the plots in Figures 1.4.1.24 and 2.0.0.24. While error statistics indicate some improvement over programs #1 and #6, the plots generally appear to be considerably more pleasant. Additional surfaces are shown in Figures 1.4.n.24, 4.1.n.24, 4.2.n.24, and 4.3.n.24. The variation of the parameter shown in Figures 3.1.1.24 and 3.3.1.24 show basically the same trend as before: localizing too much tends to degrade the approximation, while the reverse has less impact. Overall, the use of thin plate functions in the method is a nice improvement.

3.3.0. Triangle Based Blending Methods

The performance of this class of methods is dependent on the type of nodal function used. If they are good local approximations to the surface, the overall approximation will be good. In line with this, the linear nodal function method (program #12) does not perform adequately. The plots, shown in Figures 1.4.1.12 and 2.0.0.12 show the transition between local approximations resulting in apparent creases in some instances. This is due partly to the resolution of the plots. Another defect, but one which is an artifact of the triangulation is the apparent edge especially noticeable in Figure 2.0.0.12c and d. This is due to the occurence of a very long thin triangle along the edge of the convex hull. The result is that the blend of approximations near the middle part of the triangle does not reflect the actual behavior at nearby points (not in the triangle). This is not a defect unique to this

method, but occurs to some extent in all triangle based methods.

The use of quadratic nodal functions (program #13) results in a very reasonable approximation, as can be seen in the plots given in Figures 1.4.1.13 and 2.0.0.13. The apparent discontinuity in Figure 2.0.0.13d near the back is due to the previously mentioned problem of long thin triangles which can occur. The other edge at the left front is not nearly so apparent here. Additional plots are given in Figures 1.4.n.13, 4.1.n.13, 4.2.n.13, and 4.3.n.13. The parameter variation plots 3.1.1.13 and 3.3.1.13 basically show that too much localization degrades the surface. This method generally performs in quite an acceptable manner provided the disadvantages of triangles are acceptable and are outweighed by the advantages of triangles and the overall method.

3.4.0. Finite Element Based Methods

The performance of a method in this class is greatly dependent on the quality of the estimated partial derivatives. This is the major problem with Akima's method (program #4) and causes the surfaces to have a somewhat lumpy and uneven appearance. The plots for Akima's method are given in Figures 1.4.n.4, 2.0.0.4, 4.1.n.4, 4.2.n.4, and 4.3.n.4. The poor derivative estimates are especially noticeable in Figure 1.4.6.4 where the surface has a somewhat crumpled look. The variation of parameters plots, Figures 3.1.1.4 and 3.3.1.4 indicates that the nominal value we have chosen is probably about the right one to use. Figure 3.3.1.4 seems to show less sensitivity to the parameter than 3.1.1.4. Figures 2.0.0.4c and d show the characteristic defect of triangle based methods. Akima's method is very fast, usually faster than other methods by a factor of 4 or 5 or more.

Akima's method, modification one (program #10) performs somewhat better

than the original, but still does not achieve the good fits which the underlying approximations should be capable of making. All plots, Figures 1.4.n.10, 2.0.0.10, 4.1.n.10, 4.2.n.10, and 4.3.n.10 are basically similar to those of the original method, although the statistics on deviations generally show it to be slightly better.

Akima's method, modification two (program #11) would seem to promise to be better than either of the previous. However, the results basically show little or no improvement over either one. The plots, given in Figures 1.4.1.11 and 2.0.0.11 are basically similar to those for programs #4 and #10.

Akima's method, modification three (program #16) is undoubtedly the best performer of the four. The uneveness of the surface is gone from nearly all of the plots, shown in Figures 1.4.n.16, 2.0.0.16, 4.1.n.16, 4.2.n.16, and 4.3.n.16. Even the artifact of an apparent edge due to the triangulation, usually prominent in the analogues to Figure 2.0.0.16c and d, have been reduced a great deal. However, the cardinal function now shows some unbecoming behavior along the left edge near the rear corner. This seems to be caused by the quadratic approximation as it also occurs in programs #13 and #14, where the identical quadratic approximation is used. The method seems to be fairly insensitive to the parameter value, as is shown in Figures 3.1.1.16 and 3.3.1.16, although the larger value in Figure 3.1.1.16d shows some more noticeable defects along the front slope. Incorporation of the quadratic to estimate derivatives results in considerably larger preprocessing time.

Nielson's minimum norm network (program #19) shows the capability of triangle based approximation, given the appropriate values of the derivatives. As can be seen in the plots for the method, Figures 1.4.n.19, 2.0.0.19, 4.1.n.19, 4.2.n.19, and 4.3.n.19, the surface almost always appears quite smooth and

visually pleasing. The one displeasing behavior is seen in Figure 2.0.0.19c and d (and to some extent in b) where the occurence of long slim triangles gives rise to apparent discontinuities across the triangle. Again, this simply reflects the fact that in the middle of such a triangle the function may not be appropriately represented since it is far from the vertices of the triangle, and generally closer to other data points which have minimal influence. The method is reasonably fast and is undoubtedly the best performer in this class of methods.

Lawson's method (program #28) is similar in spirit to Akima's, but basically performs in much better fashion than all but modification three of Akima's method (program #16). The plots, shown in Figures 1.4.n.28, 2.0.0.28, 4.1.n.28, 4.2.n.28, and 4.3.n.16. One caution regarding the plots for this method: The program does not extrapolate outside the convex hull of the set $\{(x_k, y_k)\}$ and the function values at such points have been set to zero in the plots. Care should be taken when viewing the plots based on the 100 point and 25 point data sets, as well as the Ferguson data set to not let these points influence one's perception of the surface. Such points are omitted from the deviation statistics for this method. The usual artifact of triangulation methods is not seen in Figure 2.0.0.28c and d because no grid points fall in the appropriate place. It was extrapolation in the other methods which made it more visible. It is noticeable in Figure 2.0.0.28b along the left edge and the back, just as it is in the other triangulation based methods. Lawson's method is quite efficient in terms of timing, being only slightly slower than Akima's method (#4).

3.5.0. Foley's Methods

Foley's generalized Newton interpolant (program #25) performs in a somewhat

expected manner in that polynomials of high degree generally do not work very well. This is particularly evident when any extrapolation is involved, or in regions where no points are nearby. The plots given in Figures 1.4.1.25 and 2.0.0.25 show this, especially in the latter, Figures 2.0.0.25c and d, where more extrapolation is required than in the other examples. The cardinal function, 2.0.0.25a has some polynomial type (mis)behavior near the front corner, also.

The Newton delta sum Bernstein interpolant (programs #26) is something of an improvement in most instances, although the overall set of surface plots, shown in Figures 1.4.1.26 and 2.0.0.26 show basically the same kind of behavior. The additional program complexity and time required are probably not worth the result obtained here.

The use of bicubic splines and iteration in connection with the generalized Newton polynomial (program #30) often results in vastly improved surfaces as can be seen from the plots in Figures 1.4.n.30, 2.0.0.30, 4.1.n.30, 4.2.n.30, and 4.3.n.30. However, this is not universally true, and in particular the cardinal function shown in Figure 2.0.0.2a is less desirable. Other surfaces based on the 25 point set are also adversely affected. This is shown in Figure 3.3.1.30 as more iterations of the delta sum produce poorer surfaces. On the other hand Figure 3.1.1.30 shows definite improvement as the number of iterations increases. All things said, however, this method seems to be the best of Foley's methods. Computation time is not excessive, although it is slower than most methods for the problems in our tests.

The use of a modified Shepard's method in place of the generalized Newton interpolant in the iterated delta sum with bicubic splines (program #31) generally gave poorer results than program #30. The plots are shown in Figures 1.4.1.31 and 2.0.0.31. As can be seen in 2.0.0.31a, the cardinal function is improved

over that in Figure 2.0.0.30a, as are the surfaces in 2.0.0.31c and d. That the latter are better is no doubt due to the stable extrapolation of Shepard's method.

3.6.0. Global Basis Function Type Methods

The performance of this class of methods varies widely. Most (Duchon's methods are the exception) are dependent on scaling or a parameter specified by the user. We have attempted to reduce these to an automatic value based on some estimate of mean distance between points.

The rotated Gaussian (program #20) did not perform well. The plots, shown in Figures 1.4.1.20 and 2.0.0.20 are quite smooth and give reasonable appearing approximations in the former. The cardinal function in Figure 2.0.0.20a appears to have some undue influence near the front corner. The other plots in Figure 2.0.0.20, especially b, show a tendency of the surface to exhibit local Guassian "bumps". The surface tends to zero far away from the data. Experimentation with the parameter (related to the variance) showed the method to be sensitive to its value, and to depend on the function values rather than only the (x_k, y_k) points. For example, a nicer cardinal function could be obtained by varying the parameter, but this degraded the performance on the surface shown in Figure 1.4.1.20d, which is based on the same (x_k, y_k) sets. For these reasons we don't think this is a suitable idea.

The multiquadric method proposed by Hardy (program #21) performs very well. The plots, shown in Figures 1.4.n.21, 2.0.0.21, 4.1.n.21, 4.2.n.21, and 4.3.n.21 show that the method produces very smooth and pleasing surfaces. The deviations tables, D, and the related tables E show that the method is consistently among the most accurate, as well. The value of the parameter was computed from a formula and is related to the mean distance to nearest neighbor in the set

$\{(x_k, y_k)\}$. The surface is quite stable with respect to changes in the value of the parameter, as can be seen in Figures 3.1.1.21 and 3.3.1.21. The "best" value is probably dependent on the function values as well, but we obtained excellent results without considering that. Larger values of the parameter, r, seemed to give better results, but the system of equations became too ill-conditioned to solve in single precision. Thus a somewhat smaller nominal value was chosen than might have been otherwise. Larger values did degrade the performance on smaller sets of data while they improved it on larger sets of data. The method has also performed well on recent tests with ocean bottom topography [47].

The use of "reciprocal multiquadrics" for the basis functions (program #27) also worked quite well. The plots are shown in Figures 1.4.n.27, 2.0.0.27, 4.1.n.27, 4.2.n.27, and 4.3.n.27. The surfaces are again seen to be very smooth. The basis functions resemble the rotated Guassian (#20) but generally perform much more reliably than it. In particular, the method is much less sensitive to variations in the parameter, although very small values of the parameter will lead to a surface consisting of sharp peaks at each data point (or holes, if the function value is negative). For a range of values near the nominal value chosen for the parameter the method is quite stable. Overall its peformance is nearly as good as the multiquadric method.

The method of Duchon which involves the use of basis functions d_k^3 (program #22) works quite well. The plots, shown in Figures 1.4.1.22 and 2.0.0.22, show very smooth surfaces with a pleasing appearance. The cardinal function, Figure 2.0.0.22a is very nicely shaped. This method was among the better performers overall, however solution of the system of equations often required the use of double precision. For this reason the "thin plate splines"

(program #23) were considered more desirable even though the method was not superior in many cases. The plots for the thin plate splines are shown in Figures 1.4.n.23, 2.0.0.23, 4.1.n.23, 4.n.2.23, and 4.n.3.23. The surfaces are quite smooth, Figures 1.4.1.22 and 1.4.1.23 being very similar. The cardinal function, Figure 2.0.0.23a, is very similar to that of the previous, although it seems to be slightly more peaked with less undershoot at the front. With the exception of Akima's surface, Figure 2.0.0.23b, all calculations were performed in single precision on the IBM 360/67.

The use of rotated B-splines as a basis function for each data point (program #29) yields variable results. The method is very sensitive to the choice of radius at which the function goes to zero. The nominal value used was chosen on the basis of good performance for the surface shown in Figure 1.4.1.29. This resulted in unacceptable behavior in the cardinal function, Figure 2.0.0.29a. In this respect the method seems to be similar to program #20, the rotated Gaussians. The use of a radius which resulted in an acceptable cardinal function seriously degraded its performance on the surface in Figure 1.4.1.29d, which is based on the same (x_k, y_k) points. For that reason the method is judged unacceptable.

4.0. Summary

This summary generally deals only with the extensively tested methods, in the first two groups of Table S. These two groups were selected on the basis of meeting one or both of two criteria, (1) availability (are documented programs readily available), and (2) performance in these tests. The following local methods were selected (given by number; refer to Table S for a pointer to the description): 1, 4, 10, 13, 14, 16, 24, 28. The global methods selected were: 19, 21, 23, 27, 30.

The discussion of overall performance is made in separate sections for local and global methods. As we have noted previously, global methods are not feasible for very large sets of data. If a data set consists of some 100 - 200 points, it is feasible and practical to use a global method. We have demonstrated here that for 100 points or less, some real advantages accrue for global methods. In particular, global methods perform better in terms of their deviations from test surfaces. This is seen in Table E.2, where local methods are "best" in only 4 of 18 cases. Global methods seem to be less likely to exhibit edge effects than local methods. It is possible that this is partially responsible for the results in Table E.2, although the surfaces for global methods are generally much smoother and more pleasant appearing than those for local methods. For very large sets of data the regions where edge effects occur should be a smaller part of the overall region of interest.

The processing time for local methods is generally less than that for global methods, although some global methods are faster than some local methods. The trend of local methods is generally to increase at least linearly with the number of points when total time is considered. Global methods generally increase at least linearly also since all points must be inspected The potential savings for local methods comes from not having to solve a linear system of N equations in the preprocessing phase and use of only nearby points in the evaluation phase. Specific comments are made in the next two sections.

4.1. Local Methods

The best performing local methods are probably the Modified Quadratic Shepard Method and a similar program based on a triangulation of the convex hull, the Nielson-Franke Quadratic Triangle method. Both methods perform

consistently well, with the method based on triangles generally being slightly more accurate and faster. It is, however somewhat prone to misbehave when long slim triangles occur, and the triangulation requires a great deal of auxiliary storage. For general purpose use the Modified Quadratic Shepard's Method is favored for several reasons. It is easy to implement and generalizes in rather easy fashion to higher dimensional spaces. The apparent time penalty in the evaluation phase could be reduced by some additional preprocessing and auxiliary storage to allow quicker determination of points which (potentially) affect the interpolant. This sort of scheme should be incorporated when dealing with very large sets of data to avoid excessive evaluation times. Triangle based methods (whether local or global) have this kind of scheme built in and partly accounts for their efficiency during the evaluation phase.

Akima's method suffers from poor estimates of the derivative values. Modification one results in modest improvement, generally, but still does not perform as well as is possible for the underlying approximation. Both versions are subject to the appearance of extraneous bumps, as seen in Figures 4.1.6.4 and 4.1.6.10, as well as overshoot as seen in Figures 1.4.1.4b and 1.4.1.10b. Lawson's method, based on a similar idea, but with a different element and better estimates of the derivatives, generally performs better than Akima's method. The use of inverse distance weighted least squares quadratics to estimate the derivatives in Akima's method generally (but not always) results in improved accuracy, and even more often gives a much more pleasing surface, as can be seen by comparing the corresponding plots for the three different schemes for estimating derivatives. This latter version of Akima's program performs about on a par with Lawson's method. It requires considerably more preprocessing time, however. It is subject to edge effects in some cases.

Lawson's method generally gives smooth appearing surfaces, although the occurance of long slim triangles can cause problems. Some of these effects are not apparent because no evaluation points fall within the long slim triangles that occur along edges, as in Ferguson's data, for example. While it is not a local method, it should be mentioned that Nielson's Minimum Norm Network method generally performs better than any of the local methods based on triangles. It avoids the usual storage problem involved in solving a large system of equations by solving iteratively, and rapid convergence is obtained. The overall storage requirements are similar to Akima's method (and its variants) but more than for Lawson's method. Timing is somewhat slower than all but modification three to Akima's method in the preprocessing phase. It is also slower in the evaluation phase, but a different implementation of this could result in it being about as fast as Lawson's method. The underlying approximation is somewhat more complicated than Akima's, but use of an evaluation phase following a strategy similar to Akima's should not be slower by a factor of more than about two. So far this modification has not been made.

The remaining two local methods are due to the investigator. The underlying idea of partitioning the plane into rectangles seems to be sound, resulting in reasonable (not fast, but nearly independent of N) evaluation times. The use of thin plate splines as the local approximations is a definite improvement in both the appearance and accuracy of the method. Overall, however, performance of the method is somewhat disappointing. It seems there is no inherent reason why its performance should not be nearly as good as the local approximations, which are very good, according to our results on global methods. It seems that the amount of overlap in the local interpolating functions is not sufficient to prevent transition from one rectangle to another resulting in

transition to a fundamentally different surface, as can be seen, for example, in Figure 1.4.5.24c. Use of much larger values of NPPR could result in better approximations, although our tests (in Tables P.1 and P.3) show conflicting evidence. No further experimentation has been performed. The amount of auxiliary storage required is mild, compared to the methods based on triangles, particularly. Evaluation times are nearly independent of the number of data points, which could be useful for very large N. Triangle based methods also possess this property and are faster than rectangle based methods because faster evaluation of the local interpolants is possible.

All things considered, the method of choice here seems to be the Modified Quadratic Shepard's Method. Its advantages of simplicity and mild auxiliary storage requirements overcome its relatively expensive evaluation phase. A preprocessing phase to determine (potential) points which affect the interpolant in various regions could be implemented at a modest cost in time (probably less than one second) and storage (about 5/3 N locations), which for large data sets would probably result in evaluation times of 10 seconds or less (independent of N), but this has not been implemented as yet.

4.2. Global Methods

The most impressive method in these tests is the multiquadric method of Hardy. It is consistently best or near best in terms of accuracy, and always results in visually pleasant surfaces. Nonetheless, a certain skepticism persists because the method has no apparent mathematical basis to explain its efficacy. In some respects the basis functions are somewhat similar to the thin plate splines of Duchon in that they take on large values at points far away from the data point. Further, they appear (for $r = 0$) to fit the class of approximations discussed by Meinguet [41], but the proofs do not hold.

In the degenerate case (r = 0), initially investigated by Hardy, the multiquadrics are cones with zero value at the data point (just as $d^2 \log d$ is zero at the data point). Given only the basis functions $d^2 \log d$ one might also be perplexed at how well they work. Perhaps there is an equally elegant (but unknown) theory to explain the abilities of the multiquadric method. On the basis of our tests we can recommend the use of either the multiquadric method or the thin plane splines as the best of the global basis function methods, and perhaps the best of all global methods considered. The reciprocal multiquadric has some potential bad effects and for too small a value of r will give poor results, as noted in Section 3.6.0. There seems to be no reason to use it rather than the multiquadric method.

Nielson's minimum norm network has been discussed a bit in connection with local methods in the previous section. Computationally it is a viable method for larger sets of data than the methods requiring solution of a full system of N or more equations since it uses iteration on a sparse system of equations. It does use considerable storage which will probably limit the method before excessive computation time. The use of the method must be done with the knowledge that poor behavior can occur in long slim triangles, a caution that applies to all methods based on triangles. Nielson's method is reportedly soon to be available in a version which does not require a convex region, and this could easily be used to eliminate undesirable triangles along the edges. Extrapolation will not be so easily implemented in this version, however, so if that is important, it is a consideration.

Foley's TF delta sum bicubic spline is a relatively poor performer here. Results using the method have been discussed in great detail as an example in Section 3.0.0. While the method yields some very nice interpolants, it is

rather inconsistent and often has undesirable ripples.

5.0. Epilog

This investigation has consumed a great deal of time and effort. Thanks are due to numerous colleagues, among them Greg Nielson, Bob Barnhill, Frank Little, Tom Foley, Rosemary Chang, and others with whom I discussed many ideas and who made valuable (sometimes followed!) suggestions. Thanks are also due to those who supplied working programs, among them Greg Nielson, Hiroshi Akima, Charles Lawson, and Tom Foley. Last, but hardly least, thanks go to Linda Dent for her patience and good humor during the period when the manuscript was being typed and revised, and (especially) during the process of pasting up plots.

Despite the number of ideas explored and programs written and tested, there are a number of ideas which were not investigated. Among them are several from the CAGD group at the University of Utah. Many of these are based on the use of triangulations, which the investigator feels are much more suitable for the design problem (where long slim triangles can be avoided) than for the scattered data interpolation problem. It was not possible to test Vittitow's [55] variations of Maude's method [37], although it appears they may perform reasonably well. Another idea which was not tested has its genesis in Briggs [8], and is available commercially [59]. The user's manual contains some impressive material, but no tests of the software have been conducted. There are no doubt more ideas worthy of investigation appearing in the literature.

In terms of the data considered here, it was for the most part rather nice data, even though some effort was made to include some data with varying densities. Real data exists which is very sparce in certain regions, or

lies in clumps. Some methods will not work in a reasonable fashion for this type of data, although we have not tried to determine which methods will and which will not. Methods based on quadratic approximations will likely misbehave for such data. In addition, local methods based on distance weighting may have holes in the domain of definition when density varies greatly or when data appears in clumps. Some additional work is necessary to see if there are suitable local methods for such data.

The investigator is willing to make further tests (at least for the supplier and perhaps for wider dissemination) of working programs, under the following (negotiable) guidelines: (1) The program is to be supplied on cards (preferably EBCDIC punch). (2) The program is to be in the form of one or more subroutines, and a grid of interpolant values is to be returned by calling one of them with the appropriate data and workspace. (3) The program is to be in ANSI standard Fortran. (4) Program documentation will be supplied to enable use of the program. (5) A sample driver program will be supplied. The investigator will supply at least the plots of the type 1.4.1.n and 2.0.0.n and the corresponding error and timing statistics. Depending on those results, additional tests may be performed and reported to the supplier.

BIBLIOGRAPHY

[1] AKIMA, Hiroshi - "A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points," ACM TOMS 4 (1978) 148-159

[2] AKIMA, Hiroshi - "Algorithm 526: Bivariate interpolation and smooth surface fitting for irregularly distributed data points," ACM TOMS 4 (1978) 160-164

[3] ARTHUR, D. W. G. - "Interpolation of a function of many variables II," Photogram Eng 39 (1973) 261-266

[4] BARNHILL, R. E. - "Representation and Approximation of Surfaces," pp 69-120 in Mathematical Software III, J. R. Rice, ed., Academic Press

[5] BARNHILL, R. E.; DUBE, R. P.; LITTLE, F. F. - "Shepard's surface interpolation formula: Properties and extensions." Manuscript

[6] BARNHILL, R. E.; GREGORY, J. - "Polynomial interpolation to boundary data on triangles," Math Comp, 29 (1975) 726-735

[7] BOLONDI, G.; ROCCA, F.; ZANALETTI, S. - "Automatic contouring of faulted subsurfaces," Geophysics 41 (1976) 1377-1393

[8] BRIGGS, Jan C. - "Machine contouring using minimum curvature," Geophysics 39 (1974) 39-48

[9] BROWN, Jim; DUBE, Peter; LITTLE, Frank - "Smooth interpolation with vertex functions," manuscript.

[10] CZEGLEDY, P. F. - "Surface fitting by orthogonal local polynomials," Biometrical Journal 19 (1977) 257-264

[11] DUCHON, Jean - "Fonctions - Spline du Type Plaque Mince en Dimencion 2," Report #231, Univ. of Grenoble, 1975

[12] DUCHON, Jean - "Fonctions - Spline a Energie Invariate par Rotation," Report #27, Univ. of Grenoble, 1976

[13] DUCHON, Jean - "Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces," R. A. I. R. O. Analyse Numerique 10 (1976) 5-12

[14] FERGUSON, James C. - "Multivariable curve interpolation," J. ACM 11 (1964) 221-228

[15] FOLEY, Jr., Thomas Alfred - "Smooth multivariate interpolation to scattered data," Ph.d. dissertation, Arizona State University, 1979

[16] FRANKE, Richard - "Locally determined smooth interpolation at irregularly spaced points in several variables," JIMA 19 (1977) 471-482

[17] FRANKE, Richard - "Smooth surface approximation by a local method of interpolation at scattered points," Naval Postgraduate School NPS53-78-002, 1978

[18] FRANKE, Richard; NIELSON, Gregory - "Smooth interpolation of large sets of scattered data," Naval Postgraduate School TR #NPS-53-79-005, 1979

[19] GOLD, C. M.; CHARTERS, J. D.; RAMSDEN, J. - "Automated contour mapping using triangular element data structures and an interpolant over each irregular triangular domain," Computer Graphics 11 (1977) 170-175

[20] GORDON, William J.; WIXOM, James A. - "On Shepard's method of 'metric interpolation' to bivariate data," Math Comp 32 (1978) 253-264

[21] HARDER, R. L.; DESMARAIS, R. N. - "Interpolation using surface splines," J. Aircraft 9 (1972) 189-191

[22] HARDY, Rolland L. - "Multiquadric equations of topography and other irregular surfaces," J. of Geophysical Research 76 (1971) 1905-1915

[23] HARDY, Rolland L. - "Analytical topographic surfaces by spatial intersection," Photogrammetric Engineering 38 (1972) 452-458

[24] HARDY, Rolland L. - "Research results in the application of multiquadric equations to surveying and mapping problems," Surveying and Mapping 35 (1975) 321-332

[25] HARDY, Rolland L. - "Geodetic applications of multiquadric equations," Iowa State Univ. TR# 76245 (NTIS PB 255296) 1976

[26] HARDY, Rolland L. - "Least squares prediction," Photogrammetric Eng. and Remote Sensing 43 (1977) 475-492

[27] HARDY, Rolland L. - "The application of multiquadric equations and point mass anomaly models to crustal movement studies," NOAA TR NOS 76 NGS 11, 1978

[28] HARDY, Rolland L.; GOPFERT, W. M. - "Least squares prediction of gravity anomalies, geoidal undulations, and deflections of the vertical with multiquadric harmonic functions," Geophysical Research Letters 10 (1975) 423-426

[29] HEEP, B. R. - "Algorithms for the production of contour maps over an irregular triangular mesh," National Physical Laboratory TR NAC 10, Teddington, G. B.

[30] JANCAITUS, J. R.; JUNKINS, J. L. - "Modeling irregular surfaces," Photogrammetric Engineering and Remote Sensing 39 (1973) 413-420

[31] JANCAITUS, J. R. ; JUNKINS, J. L. - "Modeling in n dimensions using a weighting function approach," J. of Geophysical Research 79 (1974) 3361-3366

[32] JANCAITUS, J. R. - "Modeling and contouring irregular surfaces subject to constraints," U. S. Army Engineer Topographic Labs, Ft. Belvoir, VA 22060, 1975

[33] JUNKINS, J. L.; MILLER, G. W.; JANCAITUS, J. R. - "A weighting function approach to modeling of irregular surfaces," J. of Geophysical Research 78 (1973) 1794-1803

[34] LAWSON, C. L. - "Software for C' surface interpolation," pp 159-192 in Software III, J. R. Rice, ed., Academic Press, 1977

[35] LEBERL, Franz - "Photogrammetric Interpolation," Photogrammetric Engineering and Remote Sensing 41 (1975) 603-612

[36] LOGERLOF, Rolf O. E. - "Interpolation with rounded ramp functions," CACM 8 (1974) 476-479

[37] MAUDE, A. D. - "Interpolation - mainly for graph plotters," The Computer Journal 16 (1973) 64-65

[38] MC LAIN, Dermot H. - "Computer construction of surfaces through arbitrary points," Information Processing 74, North Holland Publishing Co.

[39] MC LAIN, Dermot H. - "Drawing contours from arbitrary data points," The Computer Journal 17 (1974) 318-324

[40] MC LAIN, Dermot H. - "Two dimensional interpolation from random data," The Computer Journal 19 (1976) 178-181; also errata, The Computer Journal 19 (1976) 384

[41] MEINGUET, Jean - "Multivariate interpolation at arbitrary points made simple," ZAMP 30 (1979) 292-304

[42] NEILSON, G. M. - "Multivariate smoothing and interpolating splines," SIAM J. Numer Anal 11 (1974) 435-446

[43] NEILSON, G. M. - "Minimum norm interpolation in triangles", to appear in SIAM J. Numer Anal

[44] NEILSON, Gregory M. - "A method for interpolating scattered data based upon a minimum norm network," (to appear in Math Comp)

[45] NEILSON, Gregory M.; THOMAS, Donald H.; WIXOM, James A., "Boundary data interpolation on triangular domains," GM T. R. GMR-2834, 1978

[46] PEUCKER, T. K.; FOWLER, R. J.; LITTLE, J. J.; MARK, D. M. - "Digital representation of three-dimensional surfaces by triangulated irregular networks," TR #10 (revised), Simon Fraser Univ., Burnaby, B. C., Canada

[47] PICKRELL, Alan J. - "Representation of hydographic surveys and ocean bottom topography by analytical models," M. S. Thesis Naval Postgraduate School, 1979

[48] SCHAGEN, I. P. - "Interpolation in two dimensions - a new technique," JIMA 23 (1979) 53-59

[49] SCHUMAKER, L. L. - "Fitting surfaces to scattered data," in Approximation Theory II," pp 203-268, G. G. Lorentz, C. K. Chui, L. L. Schumaker, eds. Academic Press, 1976

[50] SHEPARD, Donald - "A two dimensional interpolation function for irregularly spaced data," Proc. 23rd Nat. Conf. ACM (1968) 517-523

[51] TIPPER, John C. - "Three-dimensional analysis of geological forms," J. of Geology 85 (1977) 591-611

[52] WAHBA, Grace - "How to smooth surfaces with splines and cross-validation," Proc. 24th Design of Experiments Conference, Madison, WI, 1978

[53] VITTITOW, W. L. - "Interpolation to arbitrarily spaced data," Ph.D. dissertation, Dept. of Math, Univ. of Utah 1978

[54] WHITTEN, E. H. T.; KOELLING, M. E. V. - "Geological use of multidimensional spline functions," Math Geology 5 (1973)

[55] WHITTEN, E. H. T. - "Orthogonal polynomial trend surfaces for irregularly spaced data," Mathematical Geology 2 (1970) 141-152, and letter "More on 'orthogonal polynomial trend surfaces for irregularly spaced data'," Mathematical Geology 4 (1972) 83

[56] WHITTEN, E. H. T.; KOELLING, M. E. V. - "Spline surface interpolation, spatial filtering, and trend surfaces for geologically mapped variables," Math Geology 5 (1973) 111-126

[57] WHITTEN, E. H. T. - "Orthogonal-polynomial contoured trend-surface maps for irregularly-spaced data," TR #2, 1974, Dept. of Geological Sciences, Northwestern Univ., Evanston, Illinois 60211

[58] "Contour: A surface fitting and mapping program," Tech. Report #CSUG-049, Univ. of Calgary, Dept. of Computer Services

[59] User manual for "Surface Gridding Library," Dynamic Graphics, 2150 Shattuck Avenue, Berkeley, CA 94704, 1978

X	Y	X	Y
0.022703	-0.031021	0.053984	0.158074
0.021701	0.257692	0.011751	0.341401
0.001903	0.494380	-0.050968	0.578285
0.039541	0.699342	-0.048706	0.747014
0.031582	0.910765	-0.041878	0.996284
0.132419	0.350114	0.109627	0.091855
0.125444	0.269997	0.093454	0.338152
0.076758	0.417112	0.145187	0.561556
0.062649	0.556223	0.145273	0.752407
0.055667	0.914655	0.069550	0.963442
0.264500	0.029494	0.029164	0.000230
0.208899	0.266878	0.227733	0.369004
0.171473	0.480174	0.226478	0.594
0.190921	0.667980	0.186765	0.818
0.230463	0.904651	0.142	0.9
0.266217	0.139655	0.338765	0.058
0.288259	0.338955	0.317909	0.312
0.346632	0.490299	0.377659	0.519
0.387816	0.644952	0.381292	0.682
0.379536	0.838408	0.480351	0.971
0.414977	0.028462	0.427768	0.156
0.420001	0.246247	0.406566	0.337
0.485566	0.389142	0.410203	0.508
0.479250	0.612425	0.481523	0.751
0.397776	0.844971	0.422732	0.997
0.584669	0.071999	0.537307	0.127
0.606385	0.247	0.501389	0.347
0.574131	0.429	0.410695	0.600
0.555510	0.673	0.558042	0.723
0.609697	0.924241	0.502619	1.038
0.661693	0.085796	0.442784	0.070
0.635647	0.260824	0.370356	0.325
0.700118	0.439070	0.633359	0.509
0.655855	0.659713	0.539501	0.775
0.671809	0.936610	0.622747	1.006
0.773694	0.128937	0.763533	0.102
0.741042	0.193855	0.855899	0.323
0.737005	0.471422	0.808661	0.565
0.821453	0.650555	0.779064	0.802
0.807604	0.847675	0.817095	1.051
0.842457	0.038050	0.868405	0.090
0.836652	0.208109	0.941946	0.331
0.847812	0.432556	0.859558	0.591
0.917570	0.650738	0.859633	0.814
0.927987	0.904231	0.851288	1.069
1.044382	-0.014090	0.907062	0.113
0.985798	0.240594	0.940631	0.379
1.012929	0.439605	0.965704	0.614
1.001985	0.694152	1.035941	0.745
1.041468	0.899204	0.947151	0.950

10 X 10 RANDOM, 100 POINTS

Table 1

X	Y	X	Y
.000	.000	1.000	.000
.000	.150	1.000	.150
.000	.350	1.000	.350
.000	.550	1.000	.550
.000	.750	1.000	.750
.000	.850	1.000	.850
.000	.950	1.000	.950
.150	.000	.850	.000
.150	.150	.850	.150
.150	.350	.850	.350
.150	.550	.850	.550
.150	.750	.850	.750
.150	.850	.850	.850
.150	.950	.850	.950
.350	.000	.650	.000
.350	.150	.650	.150
.350	.350	.650	.350
.350	.550	.650	.550
.350	.750	.650	.750
.350	.850	.650	.850
.350	.950	.650	.950
.450	.000	.550	.000
.450	.150	.550	.150
.450	.350	.550	.350
.450	.550	.550	.550
.450	.750	.550	.750
.450	.850	.550	.850
.450	.950	.550	.950

SPARSE DATA, 33 POINTS

Table 2

X	Y	X	Y
0.13750	0.97500	0.45000	1.03750
0.51250	0.58750	1.08750	0.55000
0.71250	0.76250	0.53750	0.81250
0.22500	0.83750	-0.03750	0.75000
-0.05000	0.41250	0.18750	0.57500
0.47500	0.63750	0.71250	0.55000
0.85000	0.43750	1.00000	0.26250
0.70000	0.31250	0.50000	0.46250
0.27500	0.42500	0.18750	0.26250
0.45000	0.26750	0.58750	0.12500
0.81250	0.18750	1.05000	-0.06125
0.45000	-0.03750	0.10000	0.11250
0.05000	-0.05000		

GREG'S DATA, 25 POINTS

Table 3

X	Y	Z	X	Y	Z
11.16	1.24	22.15	24.20	16.25	2.81
19.85	10.76	7.97	10.35	4.11	22.32
19.72	1.39	16.83	0.0	20.00	34.60
20.87	20.00	5.74	19.95	4.60	14.73
10.28	15.40	41.59	4.51	20.00	15.61
0.0	4.48	61.77	19.70	19.32	0.31
6.08	4.55	35.14	25.00	11.37	4.41
14.90	3.12	21.70	0.0	0.0	58.20
5.66	20.00	4.73	9.22	14.69	43.35
11.77	10.47	13.64	15.10	17.15	12.87
25.00	3.87	8.74	25.00	0.0	12.00
14.55	8.71	14.81	15.20	0.0	21.60
5.23	10.74	20.50	4.14	15.33	53.10
0.51	8.37	45.43	25.00	20.00	1.00
21.67	14.35	5.52	3.31	0.13	44.00
12.55	3.06	27.11	24.67	2.91	10.25
15.51	7.74	15.30	6.71	8.24	30.97
3.45	12.78	41.24	14.26	17.87	10.74
17.43	3.46	18.60	20.80	12.14	5.47
7.58	1.92	25.87	1.55	5.50	51.21
3.22	16.78	34.93	2.56	3.12	50.55
22.33	6.21	10.25	11.52	8.53	15.74
7.54	10.69	19.31	17.36	13.78	12.11
22.59	19.43	3.25	3.47	17.15	48.43
17.25	14.57	6.43	12.13	10.79	13.71

ARIMAIS DATA, 50 POINTS

Table 5

X	Y	Z	X	Y	Z
2.000	5.000	2.500	2.490	-2.547	3.600
2.561	0.097	3.400	3.471	-2.354	3.500
3.961	-4.800	3.500	7.450	4.001	2.500
7.350	2.004	3.500	7.251	0.305	3.500
7.151	-1.991	2.000	7.051	-2.989	2.500
10.701	3.005	2.000	10.751	1.512	1.925
10.602	0.026	1.950	10.453	-1.473	1.776
10.304	-2.965	1.700	14.055	2.503	1.500
14.194	2.261	1.300	14.331	1.013	1.500
14.465	-0.224	2.100	14.007	-1.465	1.750
15.000	4.000	2.500	15.729	2.689	2.500
16.457	1.378	1.700	17.185	0.066	1.100
17.914	-1.245	1.700			

FERGUSON'S DATA, 25 POINTS

Table 6

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0919	.00842	.0148
4:	Akima	.0647	.00787	.0125
10:	Akima Mod. I	.0856	.00784	.0133
13:	Nielson - Franke Q	.0782	.00741	.0122
14:	Mod. Quad. Shepard	.0573	.00785	.0128
16:	Akima Mod. III	.0520	.00729	.0117
24:	Franke - TPS	.0940	.00887	.0164
28:	Lawson	.0951	.00783	.0124
19:	Nielson MinNorm	.0492	.00537	.00940
21:	Hardy Quadric	.0225	.00181	.00357
23:	Duchon TPS	.0518	.00525	.00947
27:	Hardy Recip. Quad.	.0247	.00283	.00518
30:	Foley III	.0636	.00473	.00941
2:	Mod. Shepard @ Plane	.156	.0137	.0254
3:	Mod. Linear Shepard	.104	.00982	.0172
5:	McLain M_{10}	.0601	.00747	.0124
6:	Franke -1	.108	.0103	.0188
7:	Mod. Shepard	.224	.0272	.0440
8:	Mod. McLain M_8	.194	.0167	.0316
11:	Akima Mod. II	.105	.00875	.0152
12:	Nielson - Franke L.	.125	.0101	.0189
17:	Quad Shepard	.264	.0396	.0594
18:	Shepard	.273	.0417	.0620
20:	Rotated Gaussians	.0624	.00599	.0112
22:	Duchon	.0247	.00311	.00578
25:	Foley I	.201	.0153	.0305
26:	Foley II	.144	.0120	.0229
29:	Rotated B-Splines	.0488	.00790	.0112
31:	Foley IV	.128	.0113	.0204

Deviations from Exponential test surface, 100 points

Table D.1.1

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0518	.00286	.00586
4:	Akima	.0520	.00303	.00609
10:	Akima Mod. I	.0473	.00257	.00542
13:	Nielson - Franke Q	.0721	.00265	.00683
14:	Mod. Quad. Shepard	.0468	.00264	.00551
16:	Akima Mod. III	.0958	.00293	.00809
24:	Franke - TPS	.0295	.00243	.00483
28:	Lawson	.0280	.00221	.00448
19:	Nielson MinNorm	.0424	.00181	.00434
21:	Hardy Quadric	.0244	.00177	.00330
23:	Duchon - TPS	.0344	.00210	.00436
27:	Hardy Recip. Quad.	.0379	.00192	.00388
30:	Foley III	.0281	.00223	.00419

Deviations from Cliff test surface, 100 points

Table D.1.2

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0198	.00164	.00294
4:	Akima	.0274	.00224	.00423
10:	Akima Mod. I	.0254	.00198	.00367
13:	Nielson - Franke Q	.0168	.00110	.00206
14:	Mod. Quad. Shepard	.0125	.00112	.00194
16:	Akima Mod. III	.0142	.00105	.00202
24:	Franke - TPS	.0165	.00157	.00273
28:	Lawson	.0565	.00149	.00359
19:	Nielson MinNorm	.0195	.00091	.00200
21:	Hardy Quadric	.00461	.00025	.00052
23:	Duchon - TPS	.00597	.00049	.00092
27:	Hardy Recip. Quad.	.00928	.00068	.00136
30:	Foley III	.0117	.00117	.00196

Deviations from Saddle test surface, 100 points

Table D.1.3

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0114	.00122	.00189
4:	Akima	.0101	.00124	.00177
10:	Akima Mod. I	.00675	.00102	.00143
13:	Nielson - Franke Q	.00517	.00058	.00083
14:	Mod. Quad. Shepard	.00388	.00065	.00089
16:	Akima Mod. III	.00330	.00049	.00070
24:	Franke - TPS	.00560	.00103	.00141
28:	Lawson	.00899	.00061	.00109
19:	Nielson MinNorm	.00303	.00047	.00069
21:	Hardy Quadric	.00102	.00005	.00011
23:	Duchon - TPS	.00294	.00017	.00030
27:	Hardy Recip. Quad.	.00227	.00034	.00050
30:	Foley III	.00604	.00083	.00117

Deviations from Gentle test surface, 100 points

Table D.1.4

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0358	.00228	.00447
4:	Akima	.0434	.00242	.00510
10:	Akima Mod. I	.0317	.00215	.00436
13:	Nielson - Franke Q	.0206	.00176	.00337
14:	Mod. Quad. Shepard	.0218	.00182	.00361
16:	Akima Mod. III	.0212	.00171	.00337
24:	Franke - TPS	.0284	.00212	.00418
28:	Lawson	.0216	.00154	.00323
19:	Nielson MinNorm	.0195	.00101	.00229
21:	Hardy Quadric	.00280	.00012	.00031
23:	Duchon - TPS	.0175	.00088	.00217
27:	Hardy Recip. Quad.	.00736	.00030	.00078
30:	Foley III	.0143	.00172	.00282

Deviations from Steep test surface, 100 points

Table D.1.5

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0119	.00126	.00206
4:	Akima	.0196	.00196	.00313
10:	Akima Mod. I	.0172	.00173	.00286
13:	Nielson - Franke Q	.00343	.00022	.00043
14:	Mod. Quad. Shepard	.00361	.00026	.00050
16:	Akima Mod. III	.00796	.00058	.00094
24:	Franke - TPS	.0111	.00138	.00206
28:	Lawson	.00954	.00038	.00099
19:	Nielson MinNorm	.0117	.00077	.00165
21:	Hardy Quadric	.0106	.00041	.00111
23:	Duchon - TPS	.0170	.00053	.00150
27:	Hardy Recip. Quad.	.0241	.00117	.00263
30:	Foley III	.00965	.00127	.00203

Deviations from Sphere test surface, 100 points

Table D.1.6

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.347	.0477	.0732
4:	Akima	.158	.0384	.0535
10:	Akima Mod. I	.197	.0400	.0570
13:	Nielson - Franke Q	.150	.0326	.0455
14:	Mod. Quad. Shepard	.184	.0340	.0478
16:	Akima Mod. III	.164	.0372	.0521
24:	Franke - TPS	.218	.0346	.0517
28:	Lawson	.287	.0462	.0657
19:	Nielson MinNorm	.150	.0305	.0437
21:	Hardy Quadric	.137	.0181	.0269
23:	Duchon - TPS	.153	.0293	.0421
27:	Hardy Recip. Quad.	.140	.0153	.0244
30:	Foley III	.296	.0350	.0546
2:	Mod. Shepard @ Plane	.208	.0402	.0604
3:	Mod. Linear Shepard	.321	.0566	.0870
5:	McLain M_{10}	.217	.0438	.0625
6:	Franke - 1	.357	.0484	.0741
7:	Mod. Shepard	.377	.0571	.0872
8:	Mod. McLain M_8	.193	.0379	.0566
11:	Akima Mod. II	.232	.0401	.0582
12:	Nielson - Franke L.	.274	.0446	.0651
17:	Quad. Shepard	.223	.0701	.0915
18:	Shepard	.225	.0709	.0922
20:	Rotated Gaussians	.137	.0174	.0287
22:	Duchon	.140	.0235	.0338
25:	Foley I	.162	.0277	.0387
26:	Foley II	.161	.0281	.0383
29:	Rotated B-Splines	.137	.0210	.0337
31:	Foley IV	.273	.0422	.0626

Deviations from Exponential Test Surface, 33 points

Table D.2.1

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0776	.0124	.0190
4:	Akima	.0543	.00850	.0133
10:	Akima Mod. I	.0518	.00747	.0122
13:	Nielson - Franke Q	.0878	.0137	.0219
14:	Mod. Quad. Shepard	.0876	.0121	.0206
16:	Akima Mod. III	.0680	.0106	.0176
24:	Franke - TPS	.0561	.00913	.0147
28:	Lawson	.0956	.0126	.0205
19:	Nielson MinNorm	.0582	.00800	.0140
21:	Hardy Quadric	.0577	.0129	.0170
23:	Duchon - TPS	.0526	.00777	.0134
27:	Hardy Recip. Quad.	.0500	.00853	.0130
30:	Foley III	.0914	.0165	.0262

Deviations from Cliff test surface, 33 points

Table D.2.2

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.111	.0121	.0224
4:	Akima	.0578	.0110	.0165
10:	Akima Mod. I	.0578	.0104	.0156
13:	Nielson - Franke Q	.0679	.00939	.0146
14:	Mod. Quad. Shepard	.0724	.00907	.0139
16:	Akima Mod. III	.0597	.0104	.0162
24:	Franke - TPS	.0662	.0109	.0175
28:	Lawson	.0685	.0133	.0199
19:	Nielson MinNorm	.0571	.0102	.0159
21:	Hardy Quadric	.0262	.00442	.00689
23:	Duchon - TPS	.0574	.00912	.0140
27:	Hardy Recip. Quad.	.0505	.00571	.00970
30:	Foley III	.0885	.00888	.0148

Deviations from Saddle test surface, 33 points

Table D.2.3

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0446	.00608	.0101
4:	Akima	.0167	.00487	.00623
10:	Akima Mod. I	.0160	.00442	.00573
13:	Nielson - Franke Q	.0312	.00422	.00637
14:	Mod. Quad. Shepard	.0272	.00451	.00679
16:	Akima Mod. III	.0204	.00394	.00565
24:	Franke - TPS	.0339	.00681	.0107
28:	Lawson	.0269	.00552	.00815
19:	Nielson MinNorm	.0214	.00371	.00563
21:	Hardy Quadric	.00724	.00121	.00204
23:	Duchon - TPS	.0259	.00415	.00714
27:	Hardy Recip. Quad.	.0188	.00266	.00485
30:	Foley III	.0349	.00438	.00674

Deviations from Gentle test surface, 33 points

Table D.2.4

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.143	.0162	.0298
4:	Akima	.115	.0120	.0240
10:	Akima Mod. I	.109	.0113	.0227
13:	Nielson - Franke Q	.0835	.0104	.0181
14:	Mod. Quad. Shepard	.110	.0113	.0220
16:	Akima Mod. III	.115	.0119	.0240
24:	Franke TPS	.150	.0148	.0305
28:	Lawson	.139	.0129	.0289
19:	Nielson MinNorm	.115	.0106	.0228
21:	Hardy Quadric	.0716	.00850	.0148
23:	Duchon - TPS	.149	.0130	.0296
27:	Hardy Recip. Quad	.0963	.00878	.0180
30:	Foley III	.110	.0143	.0249

Deviations from Steep test surface, 33 points

Table D.2.5

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0464	.00614	.0106
4:	Akima	.0383	.00796	.0110
10:	Akima Mod. I	.0393	.00732	.0104
13:	Nielson - Franke Q.	.0983	.00585	.0177
14:	Mod. Quad. Shepard	.101	.00400	.0136
16:	Akima Mod. III	.0819	.00556	.0139
24:	Franke - TPS	.0307	.00629	.00886
28:	Lawson	.0137	.00210	.00313
19:	Nielson MinNorm	.0186	.00273	.00460
21:	Hardy Quadric	.0203	.00278	.00473
23:	Duchon - TPS	.0232	.00315	.00545
27:	Hardy Recip. Quad.	.0351	.00414	.00737
30:	Foley III	.0269	.00493	.00726

Deviations from Sphere test surface, 33 points

Table D.2.6

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.240	.0359	.0486
4:	Akima	.134	.0282	.0386
10:	Akima Mod. I	.129	.0280	.0390
13:	Nielson - Franke Q	.153	.0350	.0478
14:	Mod. Quad. Shepard	.158	.0353	.0486
16:	Akima Mod. III	.155	.0355	.0484
24:	Franke - TPS	.129	.0267	.0374
28:	Lawson	.202	.0327	.0458
19:	Nielson MinNorm	.124	.0235	.0328
21:	Hardy Quadric	.119	.0235	.0322
23:	Duchon TPS	.121	.0253	.0348
27:	Hardy Recip. Quad.	.119	.0214	.0294
30:	Foley III	.165	.0196	.0310
2:	Mod. Shepard @ Plane	.167	.0328	.0466
3:	Mod. Linear Shepard	.254	.0418	.0593
5:	McLain M_{10}	.255	.0369	.0529
6:	Franke - 1	.241	.0356	.0484
7:	Mod. Shepard	.212	.0481	.0661
8:	Mod. McLain M_8	.262	.0377	.0579
11:	Akima Mod. II	.126	.0284	.0396
12:	Nielson - Franke L.	.249	.0366	.0513
17:	Quad. Shepard	.233	.0550	.0670
18:	Shepard	.238	.0559	.0709
20:	Rotated Gaussians	.118	.0237	.0321
22:	Duchon	.117	.0246	.0330
25:	Foley I	.200	.0375	.0517
26:	Foley II	.166	.0333	.0449
29:	Rotated B-Splines	.131	.0279	.0368
31:	Foley IV	.121	.0195	.0276

Deviations from Exponential Test Surface, 25 points

Table D.3.1

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.161	.0225	.0408
4:	Akima	.0999	.0148	.0257
10:	Akima Mod. I	.0987	.0143	.0252
13:	Nielson - Franke Q	.148	.0166	.0304
14:	Mod. Quad. Shepard	.163	.0166	.0314
16:	Akima Mod. III	.146	.0164	.0305
24:	Franke - TPS	.106	.0148	.0257
28:	Lawson	.132	.0164	.0283
19:	Nielson MinNorm	.0942	.0138	.0242
21:	Hardy Quadric	.0995	.0143	.0231
23:	Duchon - TPS	.101	.0135	.0235
27:	Hardy Recip. Quad	.105	.0139	.0236
30:	Foley III	.0832	.0165	.0250

Deviations from Cliff test surface, 25 points

Table D.3.2

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0688	.0111	.0171
4:	Akima	.0864	.0121	.0202
10:	Akima Mod. I	.0866	.0119	.0203
13:	Nielson - Franke Q	.0794	.0115	.0189
14:	Mod. Quad. Shepard	.0759	.0114	.0183
16:	Akima Mod. III	.0787	.0116	.0189
24:	Franke - TPS	.0714	.00983	.0171
28:	Lawson	.0875	.0126	.0205
19:	Nielson MinNorm	.0704	.0100	.0172
21:	Hardy Quadric	.0397	.00570	.00952
23:	Duchon - TPS	.0588	.00810	.0137
27:	Hardy Recip. Quad.	.0443	.00528	.00955
30:	Foley III	.0823	.00853	.0165

Deviations from Saddle test surface, 25 points

Table D.3.3

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0247	.00491	.00651
4:	Akima	.0256	.00541	.00686
10:	Akima Mod. I	.0248	.00541	.00681
13:	Nielson - Franke Q	.0340	.00562	.00746
14:	Mod. Quad. Shepard	.0227	.00529	.00669
16:	Akima Mod. III	.0232	.00575	.00760
24:	Franke - TPS	.0245	.00440	.00556
28:	Lawson	.0234	.00399	.00541
19:	Nielson MinNorm	.0161	.00307	.00433
21:	Hardy Quadric	.00709	.00107	.00158
23:	Duchon - TPS	.0128	.00265	.00351
27:	Hardy Recip. Quad.	.00528	.00055	.00089
30:	Foley III	.0224	.00436	.00588

Deviations from Gentle test surface, 25 points

Table D.3.4

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.113	.0178	.0257
4:	Akima	.0534	.0108	.0149
10:	Akima Mod. I	.0520	.0103	.0140
13:	Nielson - Franke Q	.0550	.00890	.0127
14:	Mod. Quad. Shepard	.0468	.00911	.0126
16:	Akima Mod. III	.0510	.00908	.0128
24:	Franke - TPS	.0317	.00756	.0100
28:	Lawson	.0455	.0129	.0286
19:	Nielson MinNorm	.0314	.00487	.00694
21:	Hardy Quadric	.0189	.00453	.00595
23:	Duchon - TPS	.0233	.00462	.00653
27:	Hardy Recip. Quad.	.0144	.00288	.00386
30:	Foley III	.0743	.0107	.0161

Deviations from Steep test surface, 25 points

Table D.3.5

Method		Maximum Deviation	Mean Deviation	RMS Deviation
1:	Franke - 3	.0323	.00498	.00748
4:	Akima	.0646	.00903	.0132
10:	Akima Mod. I	.0634	.00811	.0121
13:	Nielson - Franke Q	.0174	.00199	.00324
14:	Mod. Quad. Shepard	.0190	.00200	.00336
16:	Akima Mod. III	.0231	.00190	.00303
24:	Franke - TPS	.0482	.00690	.0106
28:	Lawson	.0212	.00216	.00378
19:	Nielson MinNorm	.0412	.00470	.00765
21:	Hardy Quadric	.0371	.00403	.00650
23:	Duchon - TPS	.0581	.00557	.00925
27:	Hardy Recip. Quad.	.0628	.00774	.0123
30:	Foley III	.0305	.00568	.00822

Deviations from Sphere test surface, 25 points

Table D.3.6

Test Surface	1	2	3	4	5	6
Data Set						
100 points	16	28	14	16	13-16-28	13
33 points	13	10	14	16	13	28
25 points	24	10	1-24	28	24	16

Local Method With Smallest Deviation

Table E.1

Test Surface	1	2	3	4	5	6
Data Set						
100 points	21	21	21	21	21	13
33 points	27	10	21	21	21	28
25 points	31	30	21	21	27	16

Method With Smallest Deviation

Table E.2

Method	Parameter Value	Max.Dev.	Mean Dev.	RMS Dev.
1: Franke - 3	NPPR = 4	.144	.0113	.0210
	6	.0919	.00842	.0148
	8	.0831	.00864	.0145
4: Akima	NCP = 4	.130	.00857	.0153
	6	.0647	.00787	.0125
	8	.0934	.00925	.0156
10: Akima Mod. I	NCP = 4	.152	.00898	.0169
	6	.0856	.00784	.0133
	8	.0696	.00874	.0146
13: Nielson - Franke Q	NPPR = 12	.0997	.00729	.0126
	18	.0782	.00741	.0122
	24	.0899	.00831	.0139
14: Mod.Quad.Shepard	NPPR = 6 - 12	.0663	.00704	.0117
	9 - 18	.0573	.00785	.0128
	12 - 24	.0735	.00894	.0148
16: Akima Mod. III	NCP = 12	.101	.00709	.0124
	18	.0520	.00729	.0117
	24	.0599	.00821	.0133
21: Hardy Quadric	NPPR = 15	.0287	.00303	.00578
	25	.0225	.00181	.00357
	35	.0185	.00138	.00257
24: Franke - TPS	NPPR = 4	.146	.0104	.0203
	6	.0940	.00887	.0164
	8	.0919	.00804	.0150
27: Hardy Recip. Quad.	NPPR = 15	.0912	.00601	.0129
	25	.0247	.00283	.00518
	35	.0220	.00217	.00399
30: Foley III	NIT = 1	.104	.00745	.0155
	3	.0636	.00473	.00941
	5	.0449	.00376	.00707

Deviations from Exponential test surface, 100 points, varying parameters

Table P.1

Method	Parameter Value	Max.Dev.	Mean Dev.	RMS Dev.
1: Franke - 3	NPPR = 4	.600	.0446	.0775
	6	.240	.0359	.0486
	9	.234	.0428	.0595
4: Akima	NCP = 4	.133	.0256	.0369
	6	.134	.0282	.0386
	8	.153	.0302	.0430
10: Akima Mod. I	NCP = 4	.133	.0255	.0371
	6	.129	.0280	.0390
	8	.146	.0301	.0432
13: Nielson - Franke Q	NPPR = 12	.214	.0394	.0570
	18	.153	.0350	.0478
	24	.132	.0322	.0433
14: Mod. Quad. Shepard	NPPR = 6 - 12	.230	.0372	.0549
	9 - 18	.158	.0353	.0486
	12 - 24	.135	.0338	.0456
16: Akima Mod. III	NCP = 12	.176	.0394	.0560
	18	.155	.0355	.0484
	24	.127	.0319	.0433
21: Hardy Quadric	NPPR = 15	.120	.0225	.0307
	25	.119	.0235	.0322
	35	.129	.0280	.0397
24: Franke - TPS	NPPR = 4	.186	.0318	.0455
	6	.129	.0267	.0374
	9	.143	.0281	.0404
27: Hardy Recip. Quad.	NPPR = 15	.122	.0239	.0333
	25	.119	.0214	.0294
	35	.119	.0234	.0323
30: Foley III	NIT = 1	.191	.0238	.0355
	3	.165	.0196	.0310
	5	.154	.0209	.0324

Deviations from Exponential test surface, 25 points, varying parameters

Table P.3

Program Number	Description	Section	Global/Local	Continuity (Continuous Derivatives)	Precision (Polynomial)	Storage	Domain	Sensitivity	Complexity	Accuracy	Visual	Timing (Eval/100 point total)
1	Franke's Method - 3	2.2.2	L	2	1	≈11N	R^2	C	D	B	B	C/B
4	Akima's Method	2.4.1	L	1	1	<33N	R^2	C	D	B	C	A/A
10	Akima's Method - Mod I	2.4.2	L	1	1	<33N	R^2	C	D	B	C	A/A
13	Nielson-Franke Quad.	2.3.2	L	1	2[a]	<32N	R^2	B	D	A^-	A^-	A/B^-
14	Mod. Quad. Shepard	2.1.7	L	1	2[a]	5N	B[d]	B	B	A^-	A^-	C/C
16	Akima's Method - Mod III	2.4.4	L	1	1	<33N	R^2	B	D	B^+	B^+	A/B^-
24	Franke's Method - TPS	2.2.3	L	1	1	≈11N	R^2	C	D	B^+	B^+	B/B
28	Lawson's Method	2.4.6	L	1	2[a]	<18N	$CH\{(x_k,y_k)\}$[c]	N[g]	D	B	B	A/A
19	Nielson's Min Norm Net.	2.4.5	G	1	1	<32N	$CH\{(x_k,y_k)\}$[f]	N	D	A^-	A^-	B/B
21	Hardy's Multiquadrics	2.6.2	G	∞	-1	$\frac{1}{2}N(N+4)$	R^2	B	A	A	A	B^-/C^-
23	Duchon's Thin Plate Splines	2.6.5	G	1[b]	1	$\frac{1}{2}(N+3)(N+7)$	R^2	N	A	A	A	C/D
27	Hardy's Recip. Multiquadric	2.6.3	G	∞	-1	$\frac{1}{2}N(N+4)$	R^2	B	A	A	A	B^-/C^-
30	Foley's TFA Cubic Spline	2.5.3	G	2	-1	≈8N	R^2	C	C	B^+	B	B/D
2	Mod. Shepard @ Plane	2.1.4	L	1	1	0	B	C	B	C	C	D/D
3	Mod. Linear Shepard	2.1.3	L	1	1	2N	B	C	B	C	C	C/C
5	McLain's Method M_{10}	2.1.8	G	∞(?)	1	0	R^2	C	B	B^+	B^+	F/F
6	Franke's Method -1	2.2.1	L	1	-1[e]	≈10N	R^2	C	D	B	B	C/B

7	Mod. Shepard's Method	2.1.2	L	1	0	0	B	C	A	D	D	B^-/B^-
8	Mod. McLain's M_8	2.1.5	L	1	1	0	B	C	B	C	C^-	C/C
11	Akima's Method - II	2.4.3	L	1	1	<33N	R^2	C	D	B	C	A/A
12	Nielson-Franke Linear	2.3.1	L	1	1	<32N	R^2	C	D	C	C	A/A
17	Quad. Shepard's Method	2.1.6	G	∞	2	5N	R^2	N	B	F	F	D/F
18	Shepard's Method	2.1.1	G	∞	0	0	R^2	N	A	F	F	C/C
20	Rotated Gaussians	2.6.1	G	∞	-1	$\frac{1}{2}N(N+4)$	R^2	D	A	B^+	B^+	B^-/C^-
22	Duchon's Radial Cubic	2.6.4	G	2^b	1	$\frac{1}{2}(N+3)(N+7)$	R^2	N	A	A	A	C/D
25	Foley's Generalized Newton	2.5.1	G	∞	-1	$\approx 8N$	R^2	N	C	B^-	B	B^-/C
26	Foley's TFA Bernstein	2.5.2	G	∞	-1	$\approx 8N$	R^2	B	C	B	B	D/C^-
29	Rotated B-Splines	2.6.6	G	2	-1	$\frac{1}{2}N(N+4)$	R^2 (=0 outside B)	D	A	B^+	B	D/D^-
31	Foley's Shep. Δ Cub. Spl.	2.5.4	G	2	0		R^2	C	C	B	B	B/C

TABLE S

Footnotes

[a] except for certain possible dispositions of points

[b] C^∞ except at data points

[c] convex hull of $\{(x_k, y_k)\}$

[d] $B = \{(x, y): d_k(x, y) < R_w, \text{ some } k\}$

[e] no polynomial precision

[f] program modified by this investigator to extrapolate to all of R^2

[g] no parameter

Method		Preprocessing	Evaluation	Total
1:	Franke - 3	1.1	8.8	9.9
4:	Akima	2.2	0.8	3.0
10:	Akima Mod. I	2.8	0.8	3.6
13:	Nielson - Franke Q	10.	1.9	12.
14:	Mod.Quad. Shepard	8.6	15.	24.
16:	Akima Mod. III	11.	0.8	12.
24:	Franke - TPS	2.7	6.5	9.2
28:	Lawson	1.8	1.7	3.5
19:	Nielson MinNorm	5.7	3.8	9.5
21:	Hardy Quadric	7.1	13.	20.
23:	Duchon-TPS	7.5	17.8	24.
27:	Hardy Recip. Quad.	7.1	13.	20.
30:	Foley III	15.	11.	26.
2:	Mod. Shepard @ Plane	2.1	25.	27.
3:	Mod. Linear Shepard	1.2	15.	16.
5:	McLain M_{10}	---	110.	110.
6:	Franke - 1	1.0	8.0	9.0
7:	Mod. Shepard	---	12.	12.
8:	Mod. McLain M_8	---	14.	14.
11:	Akima Mod. II	2.8	.8	3.6
12:	Nielson - Franke L.	2.4	1.5	3.9
17:	Quad. Shepard	33.	22.	55.
18:	Shepard	---	17.	17.
20:	Rotated Gaussians	7.1	13.	20.
22:	Duchon	7.4	15.	22.
25:	Foley I	---	13.	13.
26:	Foley II	4.0	16.	20.
29:	Rotated B-Splines	7.7	23.	31.
31:	Foley IV	11.	6.3	17.

Timing : 100 points

Table T.1

Method		Preprocessing	Evaluation	Total
1:	Franke - 3	0.3	7.8	8.1
4:	Akima	0.5	0.7	1.2
10:	Akima Mod. I	0.7	0.7	1.4
13:	Nielson - Franke Q	2.4	1.8	4.2
14:	Mod. Quad. Shepard	2.1	5.6	7.7
16:	Akima Mod. III	2.7	0.7	3.4
24:	Franke - TPS	0.6	4.6	5.2
28:	Lawson	0.5	1.5	2.0
19:	Nielson MinNorm	1.9	3.1	5.0
21:	Hardy Quadric	0.5	4.0	4.5
23:	Duchon TPS	0.5	5.3	5.8
27:	Hardy Recip. Quad.	0.5	4.0	4.5
30:	Foley III	1.6	4.0	5.6
2:	Mod. Shepard @ Plane	0.2	9.9	10.
3:	Mod. Linear Shepard	0.2	5.1	5.3
5:	McLain M_{10}	---	50.	50.
6:	Franke - 1	0.3	6.9	7.2
7:	Mod. Shepard	---	4.6	4.6
8:	Mod. McLain M_8	---	5.7	5.7
11:	Akima Mod. II	0.7	0.7	1.4
12:	Nielson - Franke L.	0.3	1.5	1.8
17:	Quad. Shepard	4.1	7.1	11.
18:	Shepard	---	6.4	6.4
20:	Rotated Gaussians	0.5	4.0	4.5
22:	Duchon	0.5	5.0	5.5
25:	Foley I	---	4.0	4.0
26:	Foley II	0.9	8.2	9.1
29:	Rotated B-Splines	0.5	7.8	8.3
31:	Foley IV	1.1	2.7	3.8

Timing: 33 points

Table T.2

Method		Preprocessing	Evaluation	Total
1:	Franke - 3	0.2	7.7	7.9
4:	Akima	0.3	0.6	0.9
10:	Akima Mod. I	0.5	0.6	1.1
13:	Nielson - Franke Q	1.7	1.7	3.4
14:	Mod. Quad. Shepard	1.5	4.2	5.7
16:	Akima Mod. III	1.8	0.6	2.4
24:	Franke - TPS	0.4	4.5	4.9
28:	Lawson	0.4	1.4	1.8
19:	Nielson MinNorm	1.4	2.9	4.3
21:	Hardy Quadric	0.2	3.1	3.3
23:	Duchon - TPS	0.2	4.0	4.2
27:	Hardy Recip. Quad.	0.2	3.1	3.3
30:	Foley III	1.1	3.1	4.2
2:	Mod. Shepard @ Plane	0.1	7.6	7.7
3:	Mod. Linear Shepard	0.1	4.0	4.1
5:	McLain M_{10}	---	40.	40.
6:	Franke - 1	0.2	6.7	6.9
7:	Mod. Shepard	---	3.7	3.7
8:	Mod. McLain M_8	---	4.4	4.4
11:	Akima Mod. II	0.5	0.6	1.1
12:	Nielson - Franke L.	0.2	1.4	1.6
17:	Quad. Shepard	2.5	5.5	8.0
18:	Shepard	---	4.1	4.1
20:	Rotated Gaussians	0.2	3.1	3.3
22:	Duchon	0.2	3.8	4.0
25:	Foley I	---	3.0	3.0
26:	Foley II	0.7	7.3	8.0
29:	Rotated B-Splines	0.3	5.9	6.2
31:	Foley IV	0.8	1.8	2.6

Timing : 25 Points

Table T.3

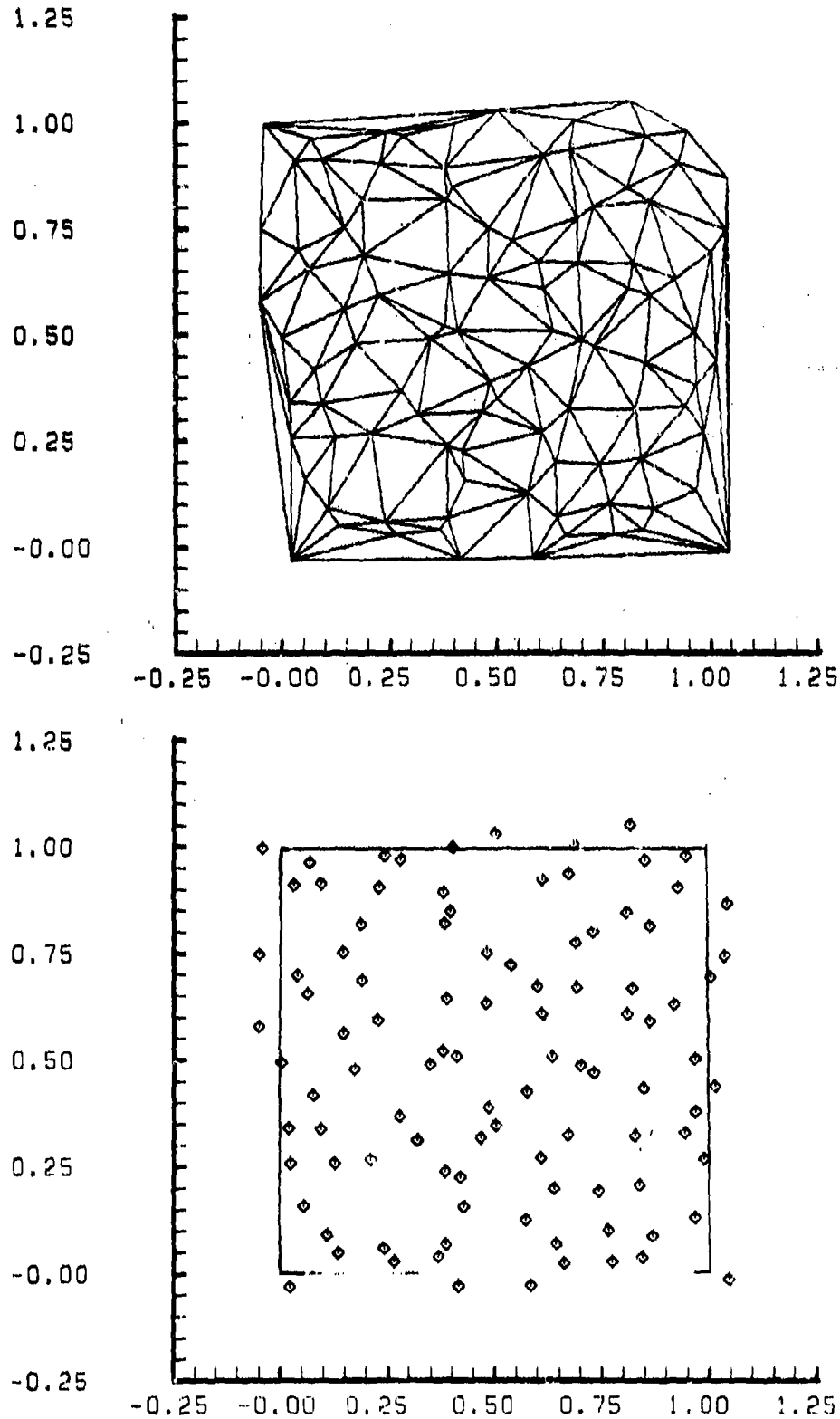

1.25
1.00
0.75
0.50
0.25
-0.00
-0.25
-0.25 -0.00 0.25 0.50 0.75 1.00 1.25
1.25
1.00
0.75
0.50
0.25
-0.00
-0.25
-0.25 -0.00 0.25 0.50 0.75 1.00 1.25
10 X 10 RANDOM

Figure 0.1.0.0.

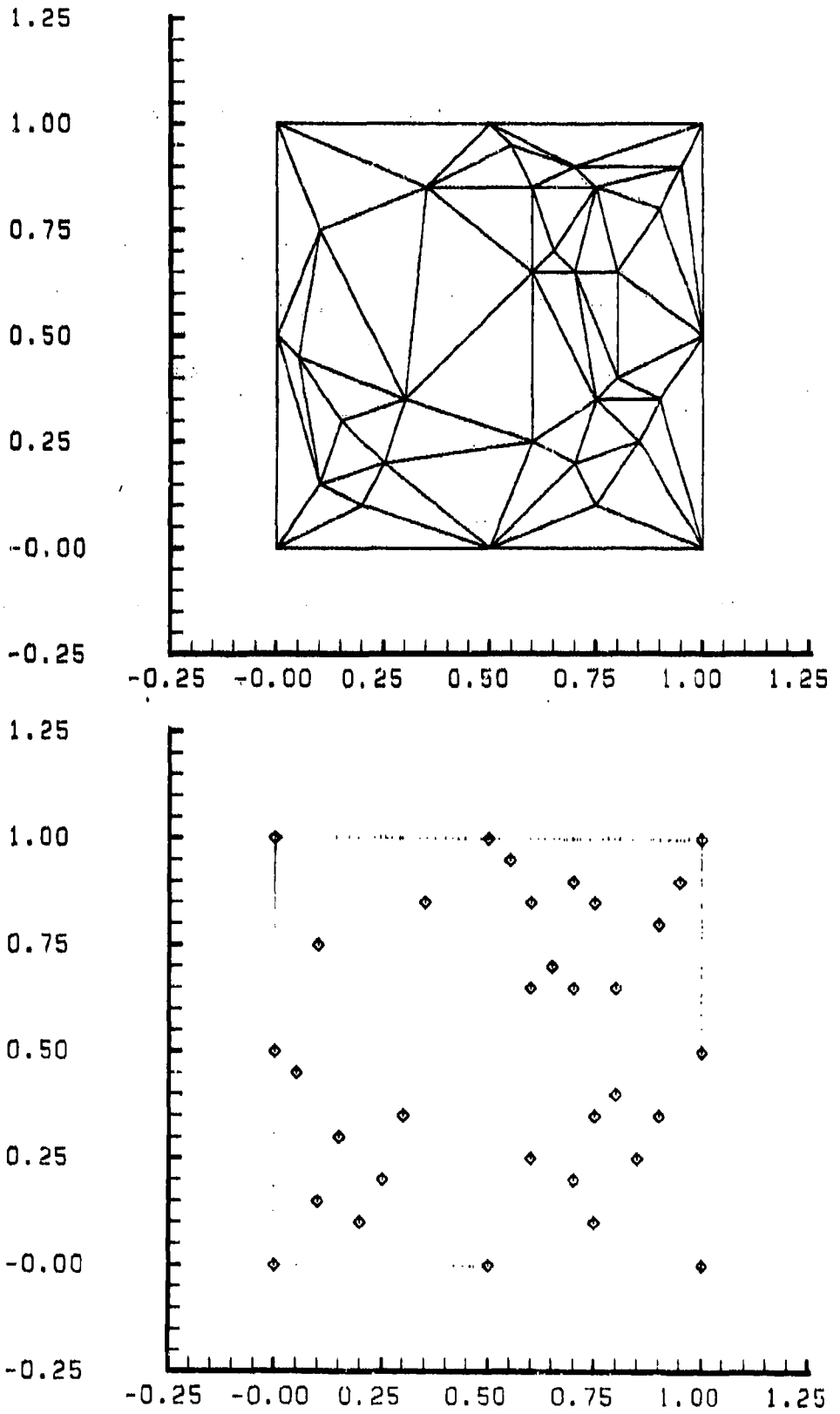

1.25
1.00
0.75
0.50
0.25
-0.00
-0.25
-0.25 -0.00 0.25 0.50 0.75 1.00 1.25
1.25
1.00
0.75
0.50
0.25
-0.00
-0.25
-0.25 -0.00 0.25 0.50 0.75 1.00 1.25
33 SPARSE

Figure 0.2.0.0.

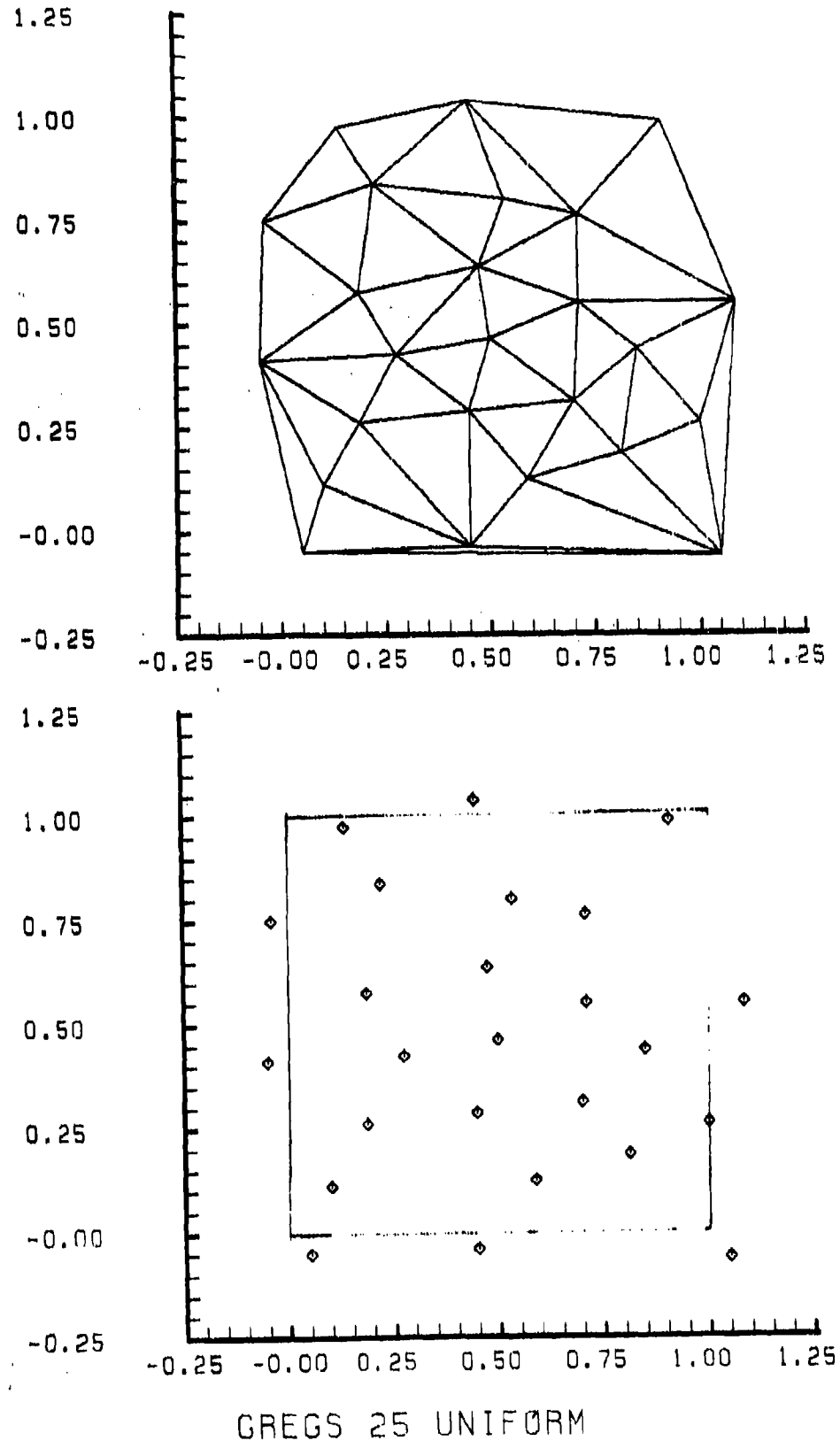

1.25
1.00
0.75
0.50
0.25
-0.00
-0.25
-0.25 -0.00 0.25 0.50 0.75 1.00 1.25
1.25
1.00
0.75
0.50
0.25
-0.00
-0.25
-0.25 -0.00 0.25 0.50 0.75 1.00 1.25
GREGS 25 UNIFORM

Figure 0.3.0.0.

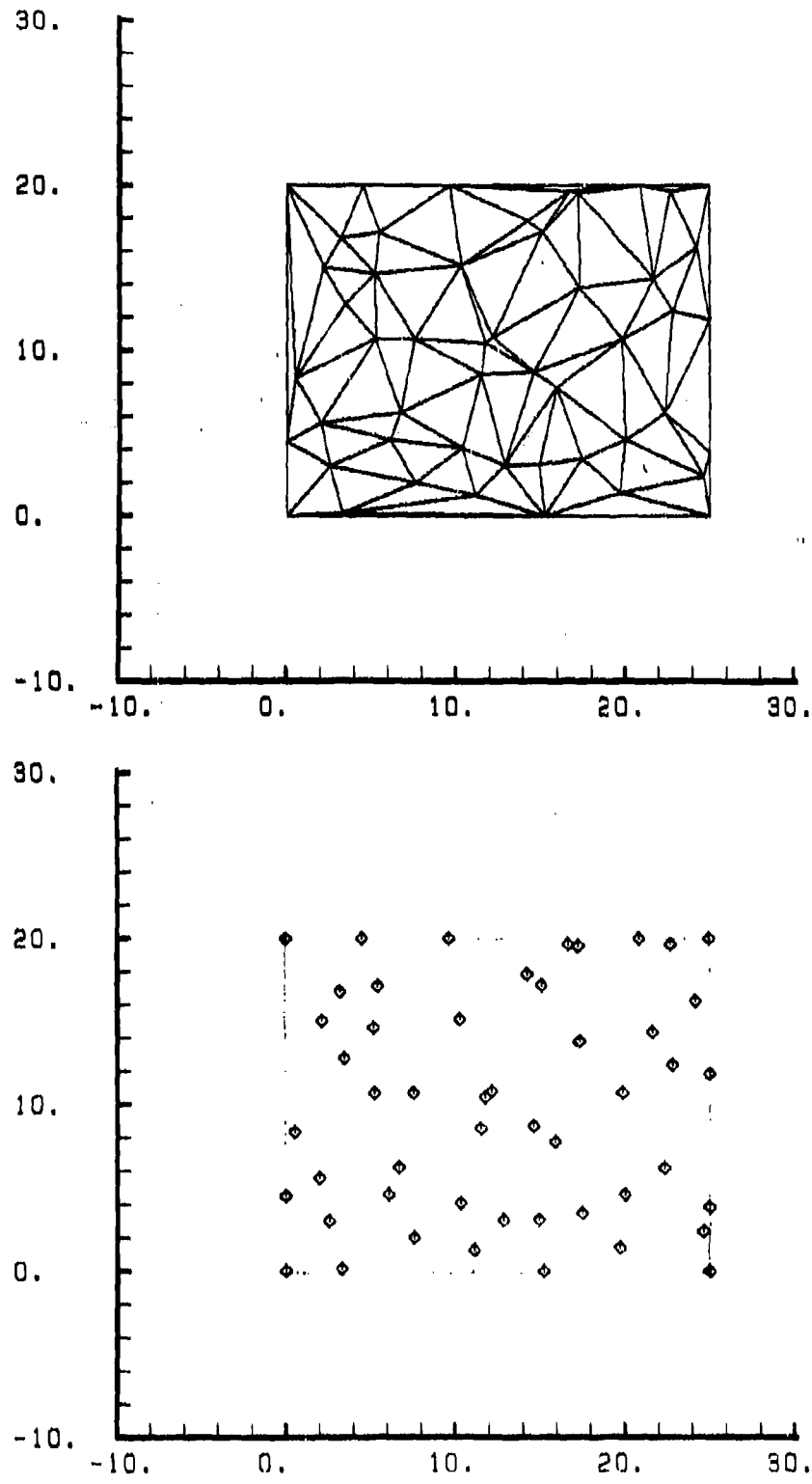
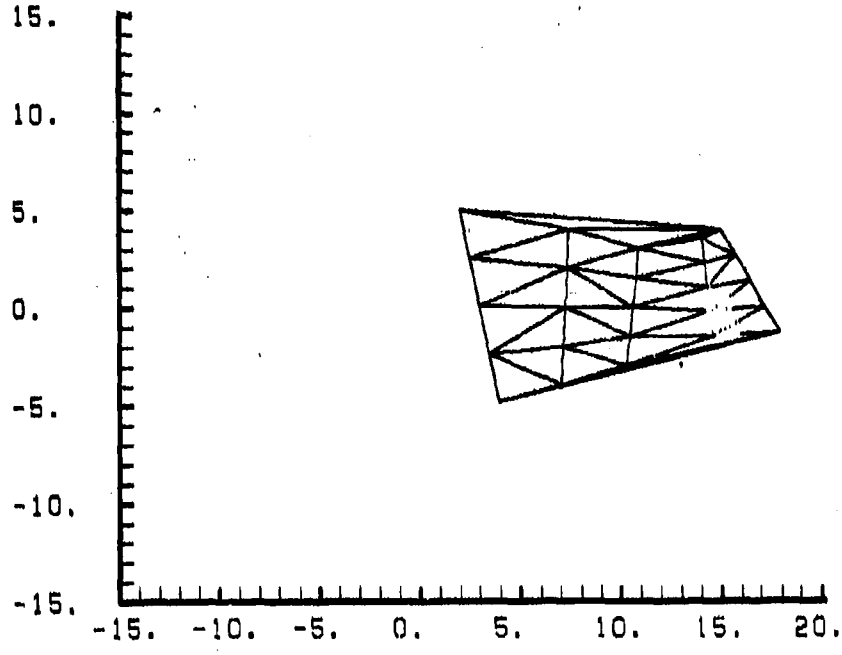
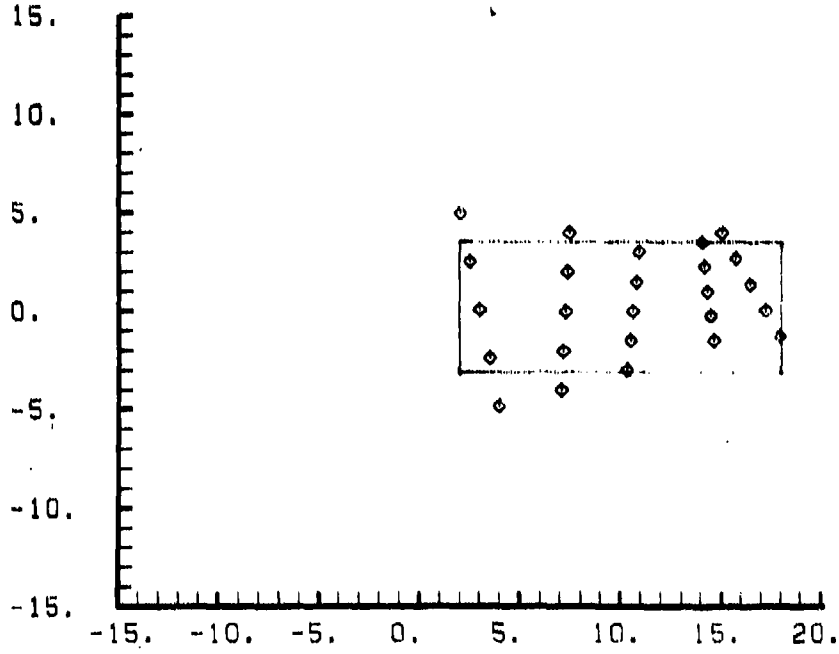

30.
20.
10.
0.
-10.
-10.
0.
10.
20.
30.
30.
20.
10.
0.
-10.
-10.
0.
10.
20.
30.
AKIMA'S 50 POINTS

Figure 0.5.0.0.





FERGUSON'S POINTS

Figure 0.6.0.0.

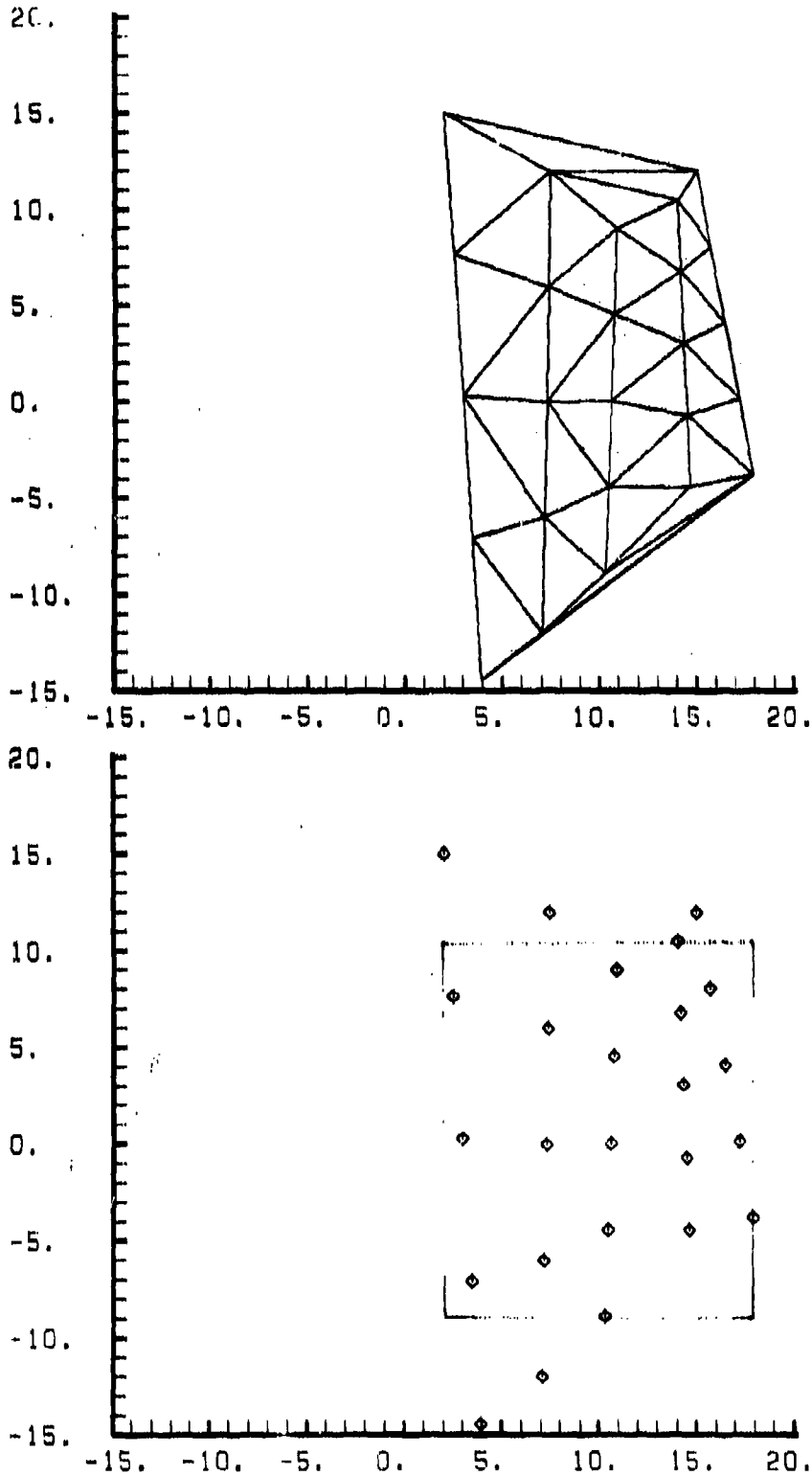
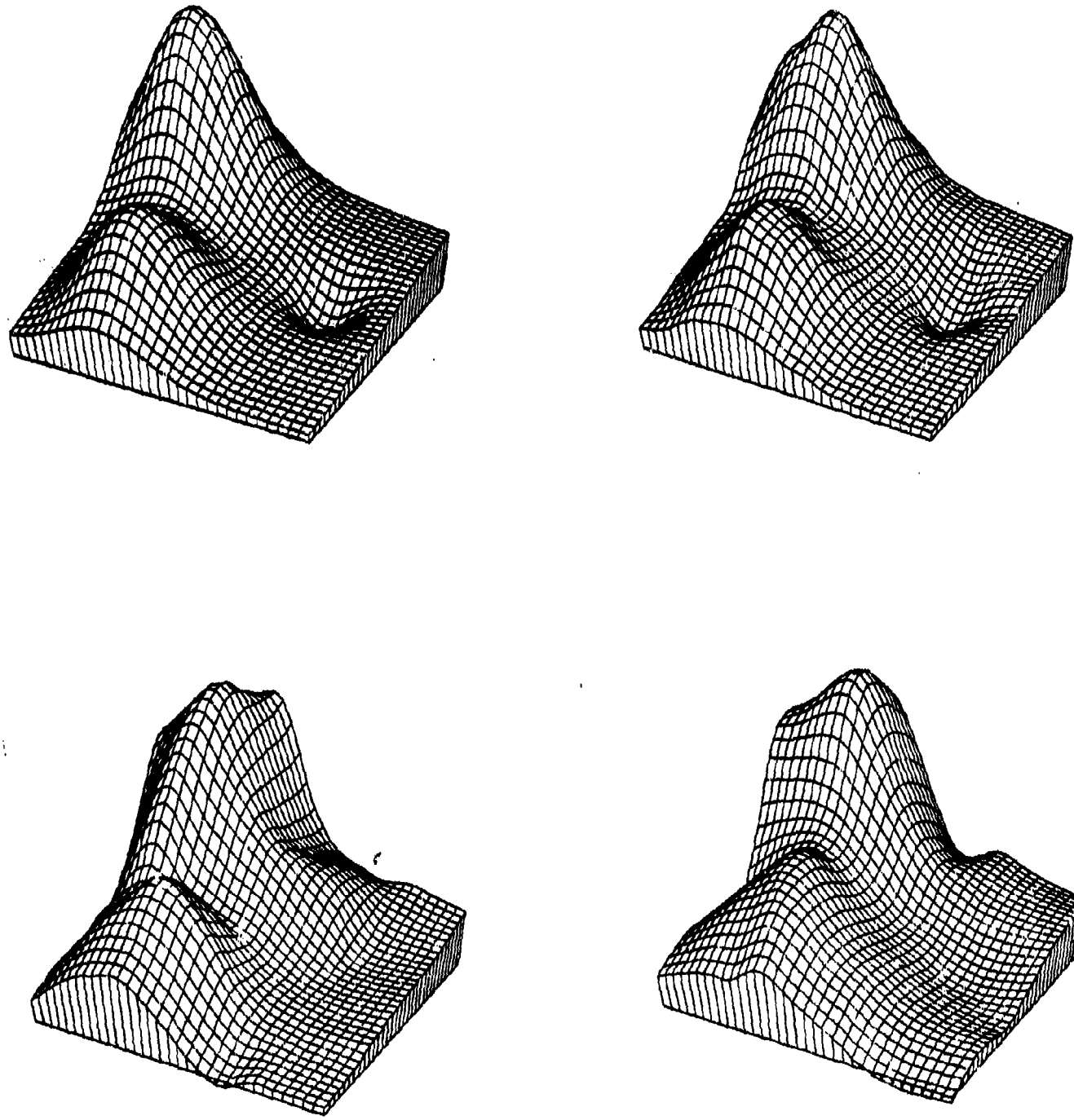
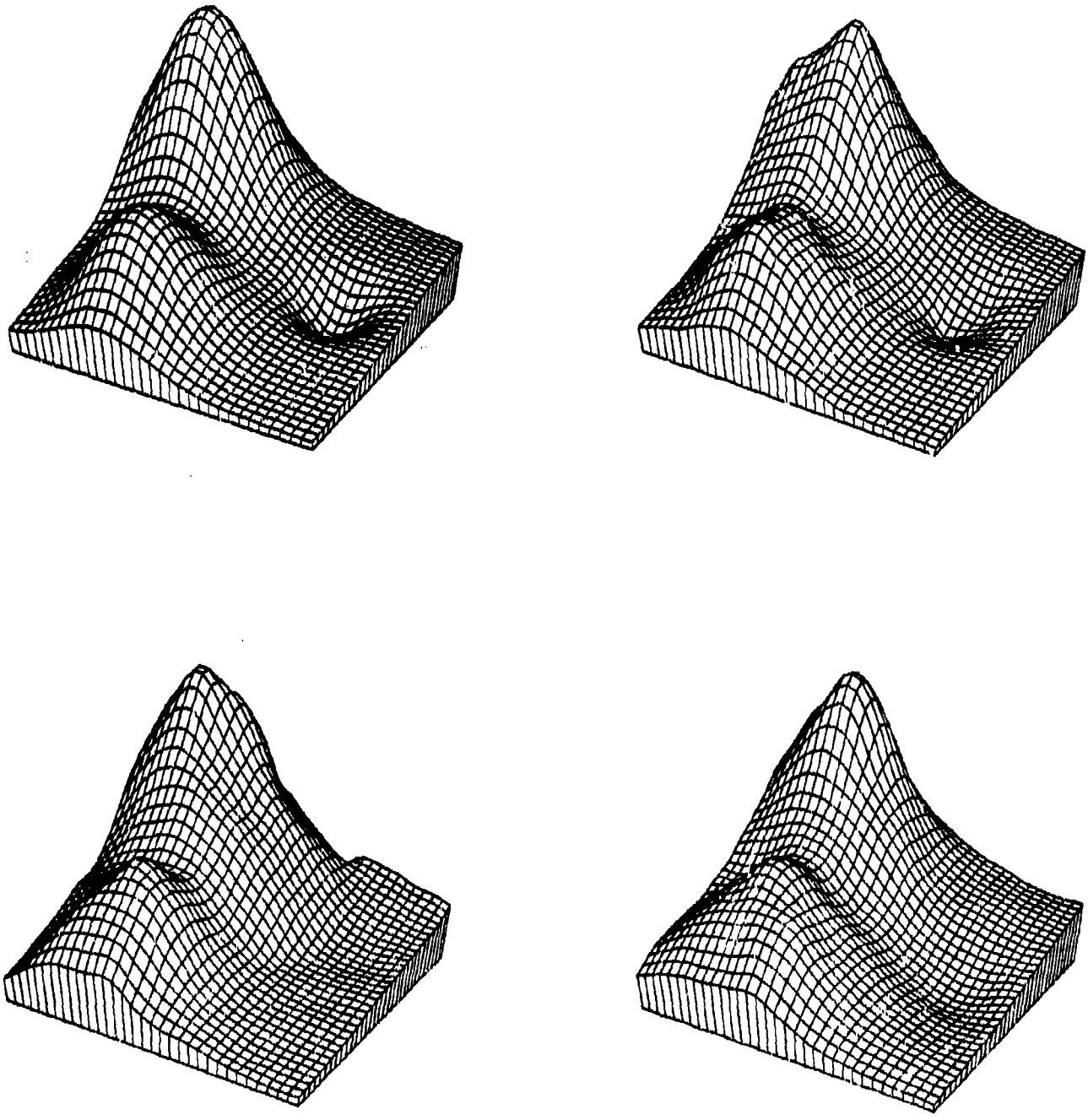

20.
15.
10.
5.
0.
-5.
-10.
-15.
-15. -10. -5. 0. 5. 10. 15. 20.
20.
15.
10.
5.
0.
-5.
-10.
-15.
-15. -10. -5. 0. 5. 10. 15. 20.
FERGUSON, Y X 3

Figure 0.7.0.0.



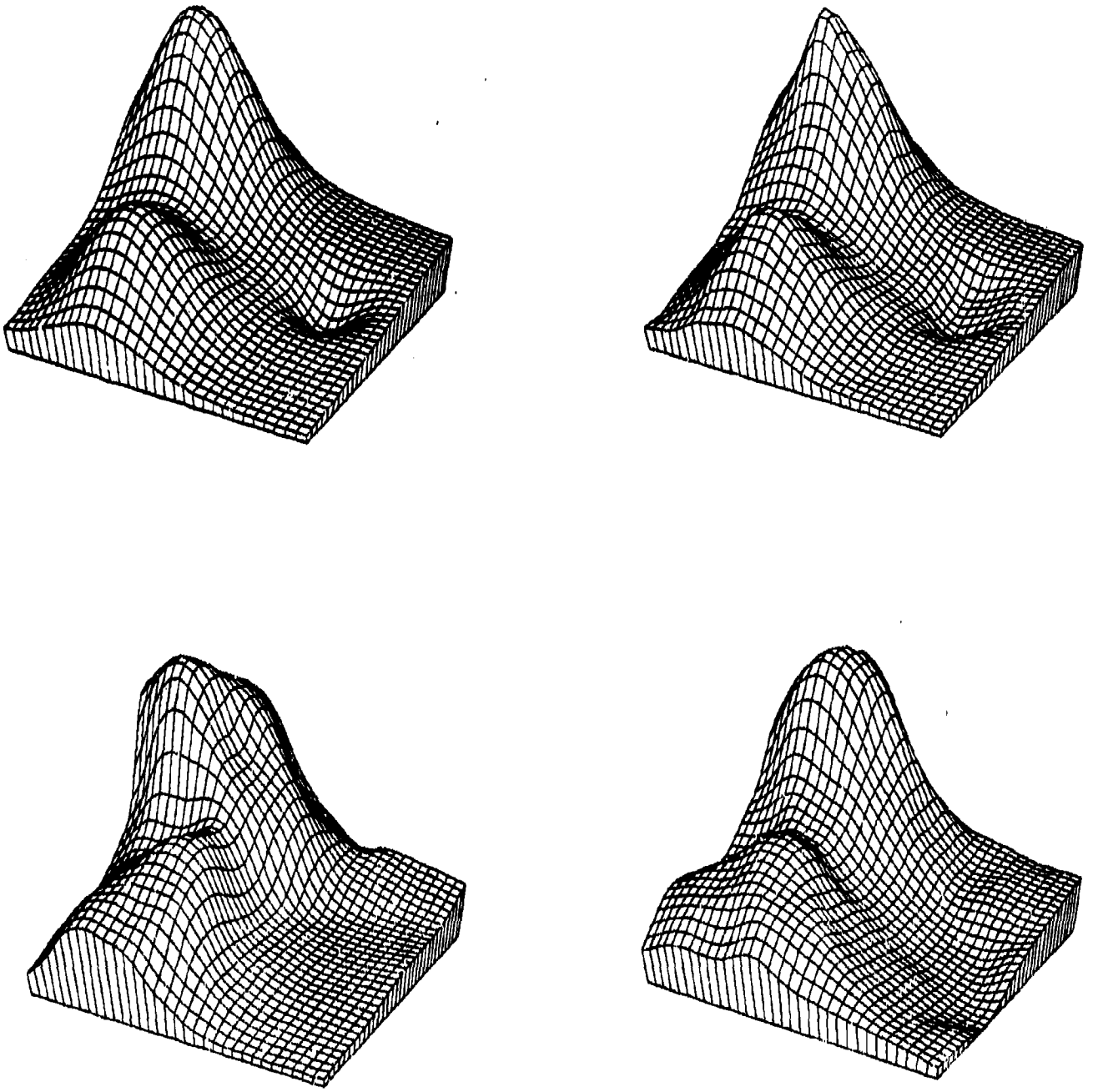
Franke's Method, Mode = 3

Figure 1.4.1.1



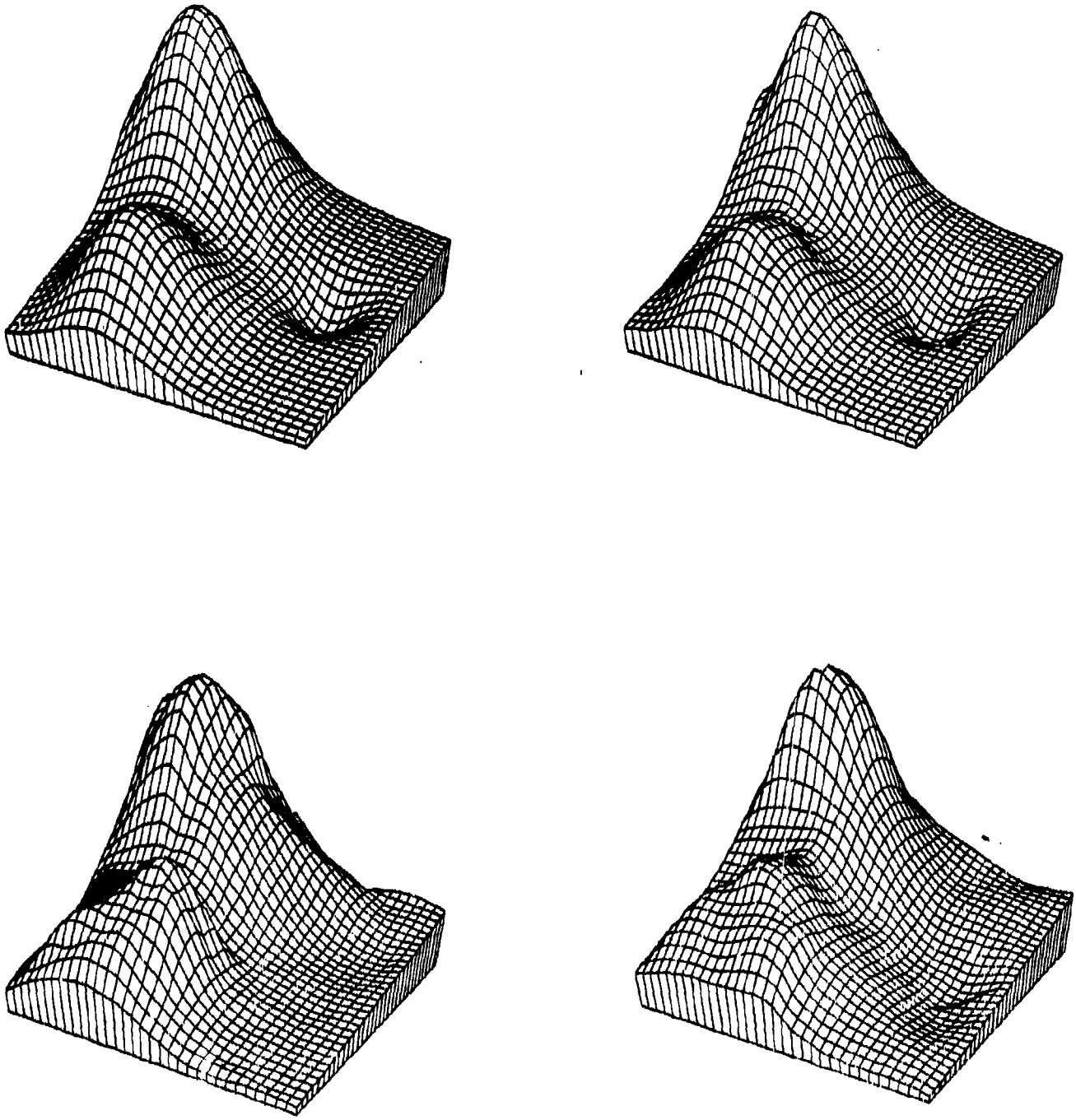
Modified Shepard's Method Boolean Sum Plane

Figure 1.4.1.2



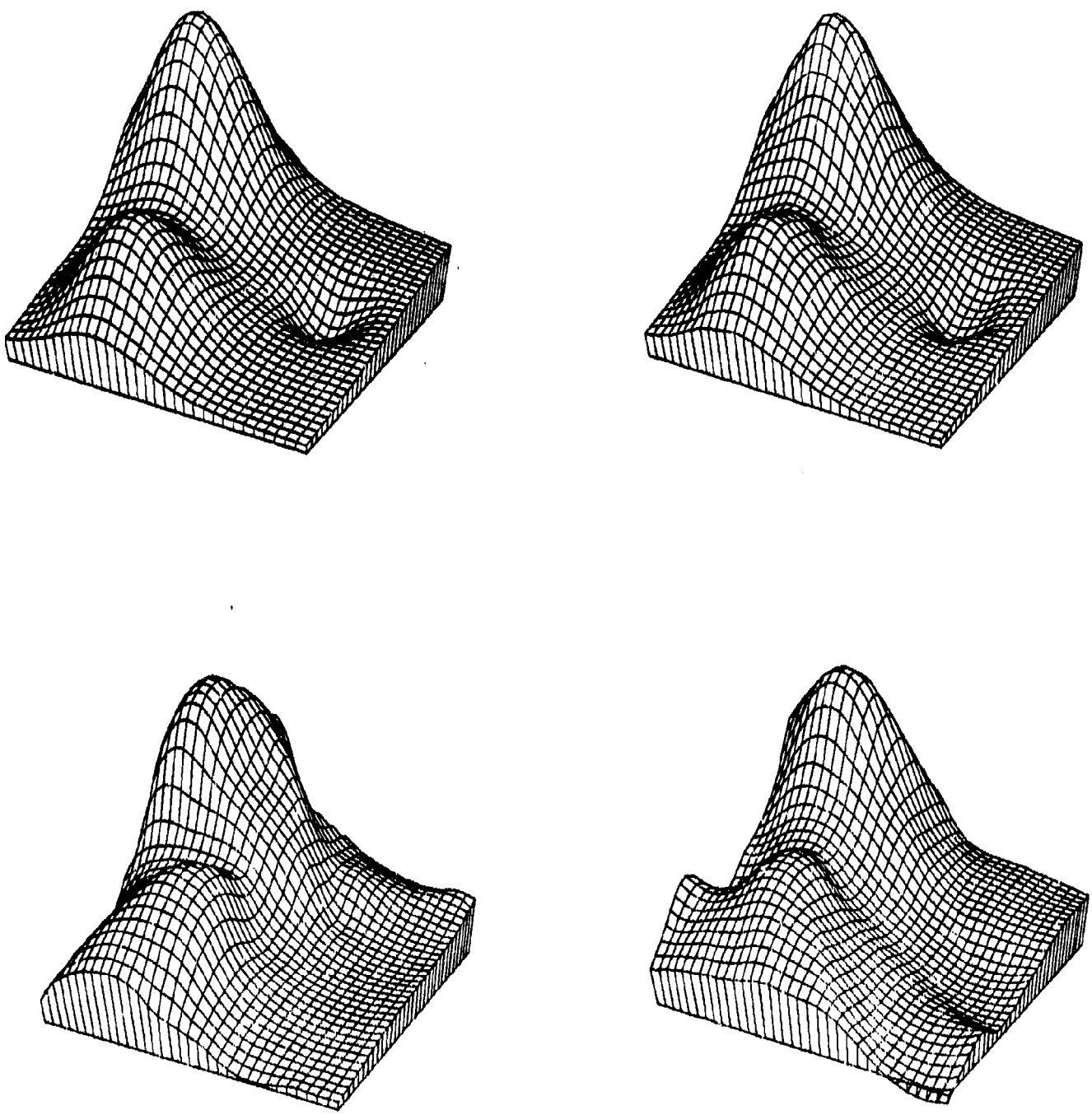
Modified Linear Shepard's Method

Figure 1.4.1.3



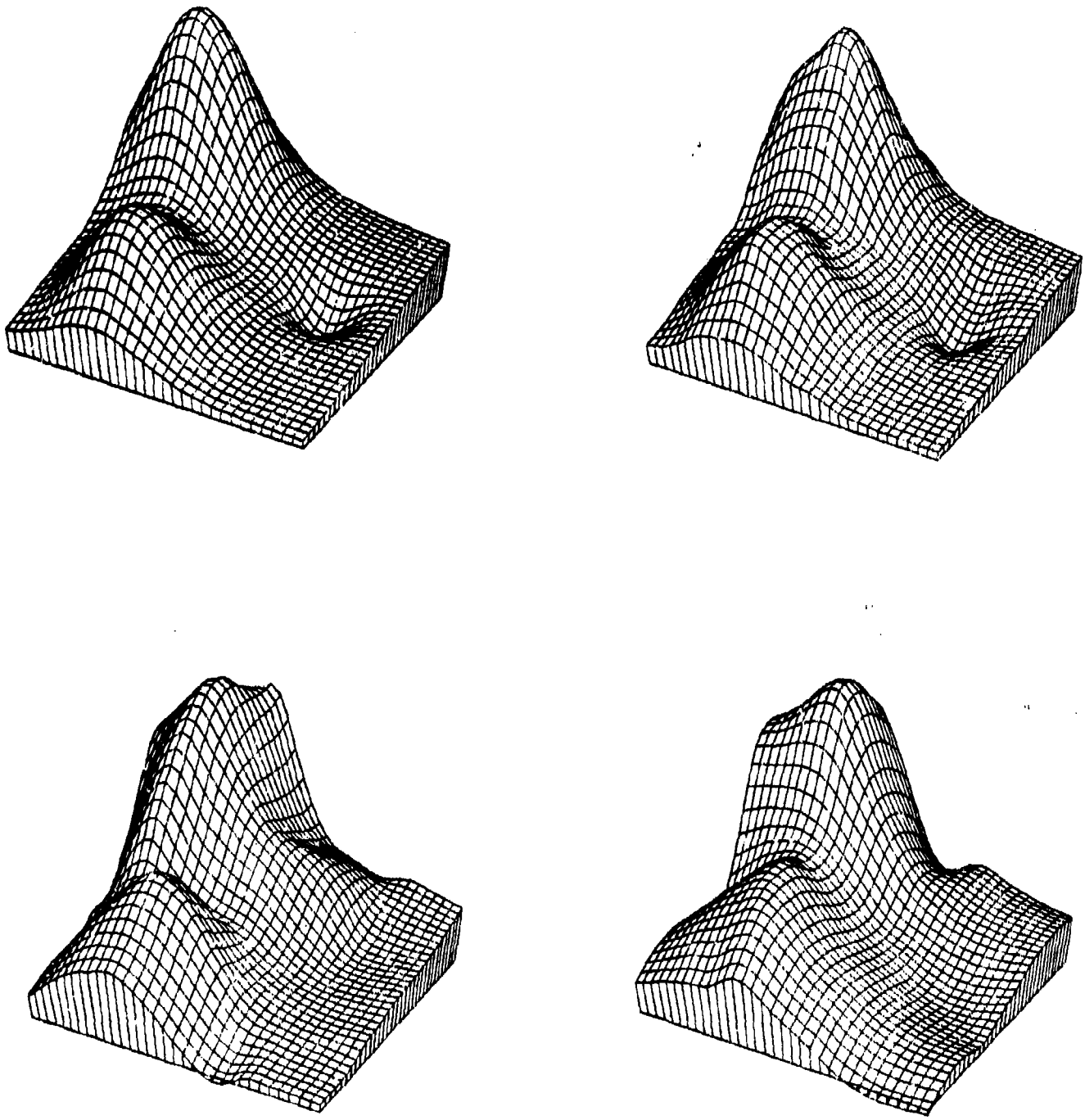
Akima's Method

Figure 1.4.1.4



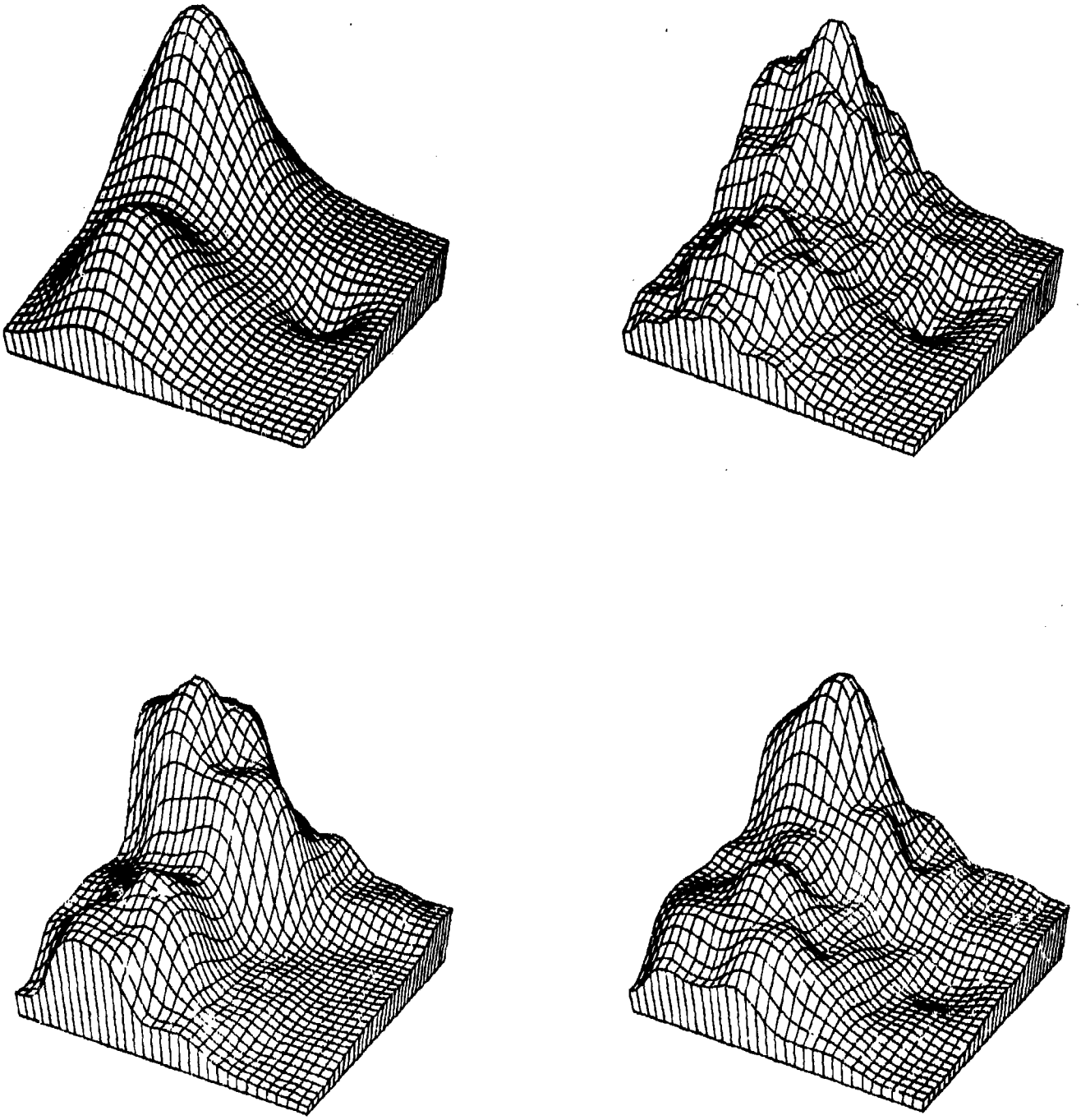
McLain's Inverse Distance Weighted Quadratic

Figure 1.4.1.5



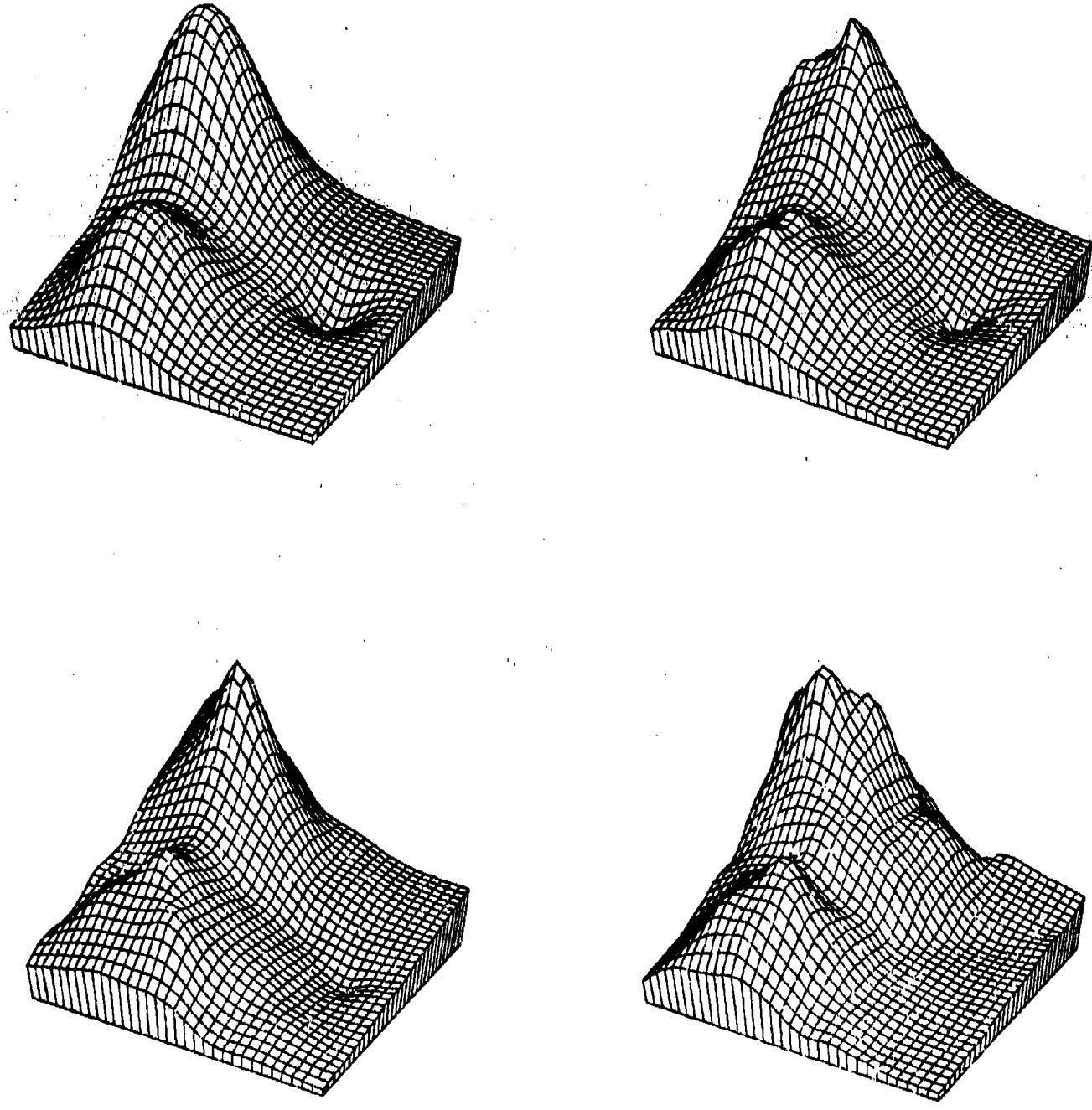
Franke's Method, Mode = 1

Figure 1.4.1.6



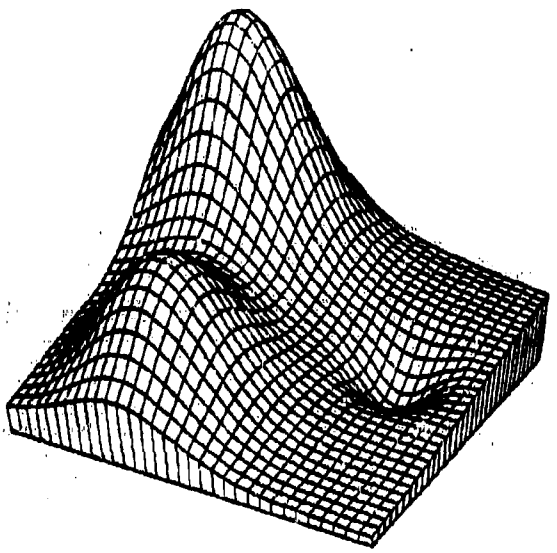
Modified Shepard's Method

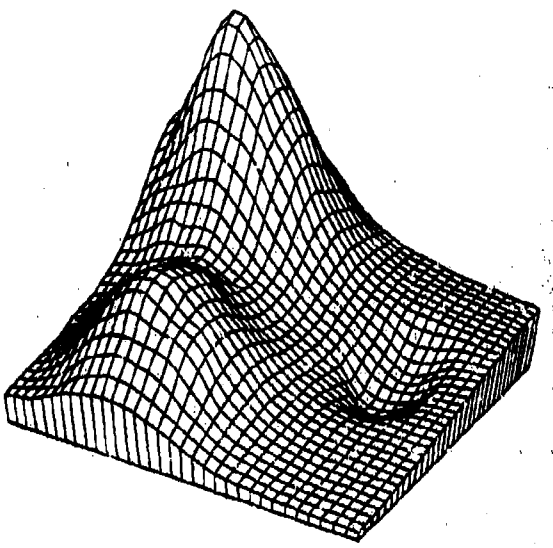
Figure 1.4.1.7

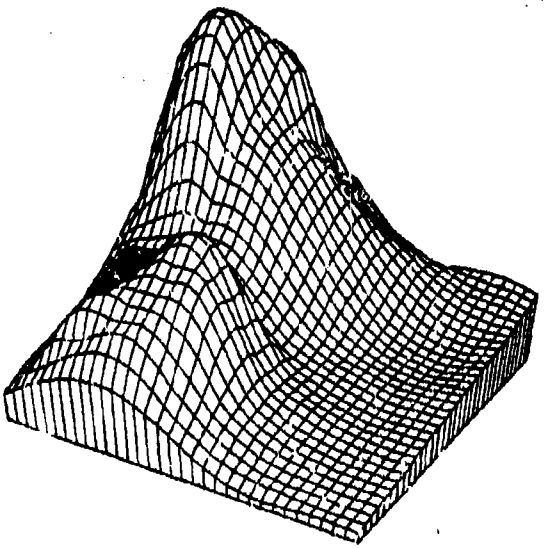


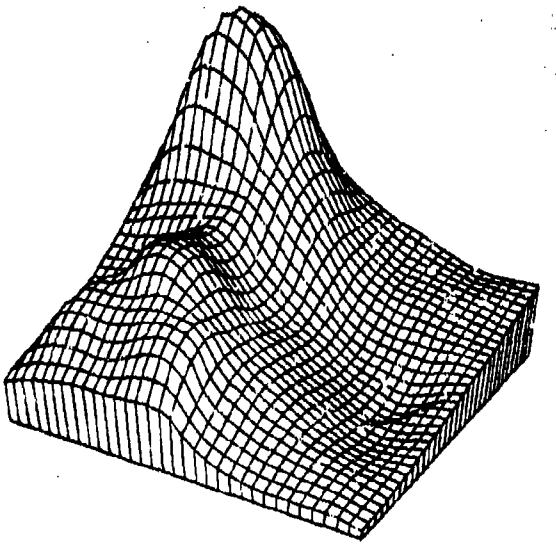
Modified McLain's Inverse Distance Weighted Plane

Figure 1.4.1.8



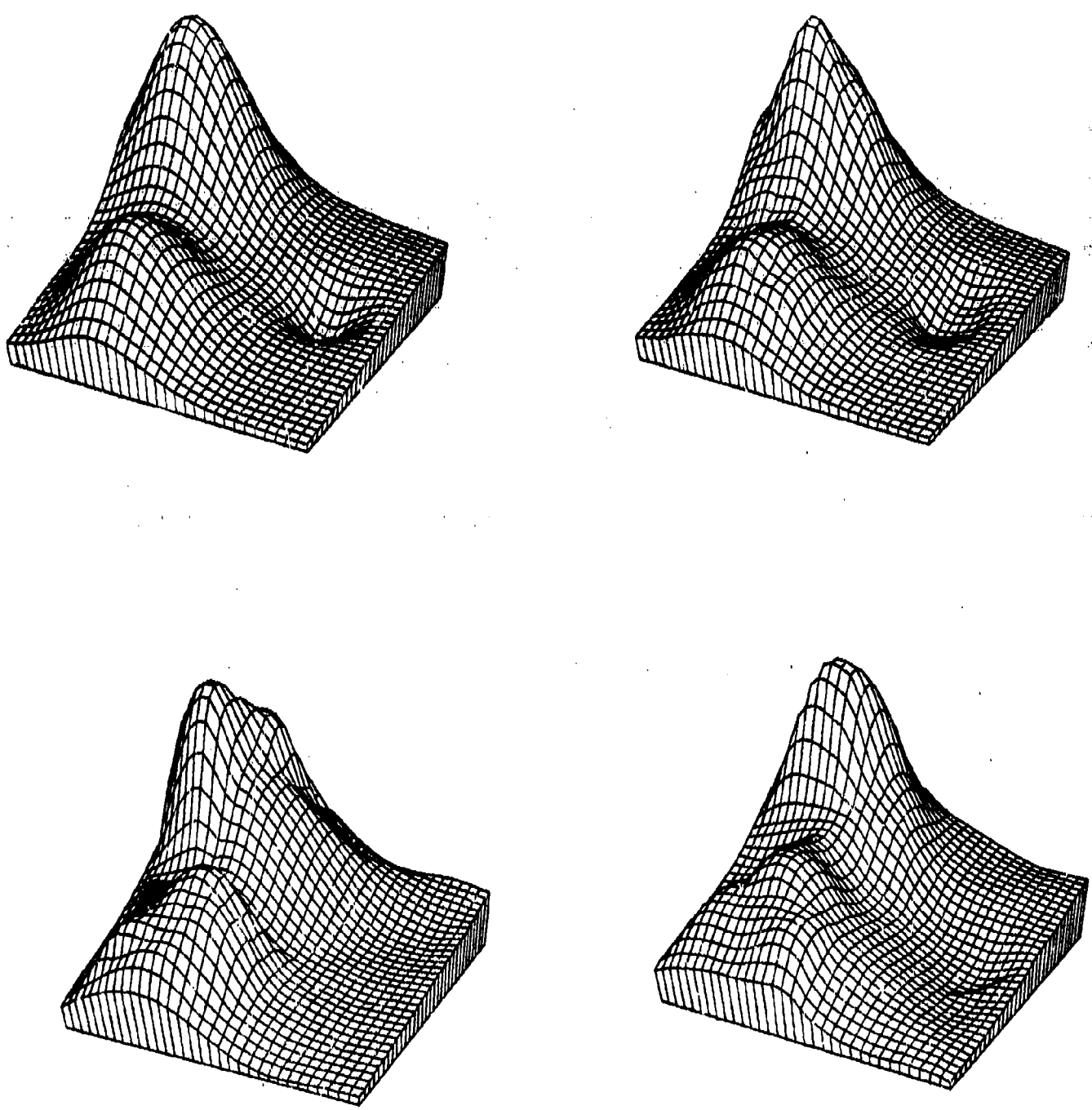






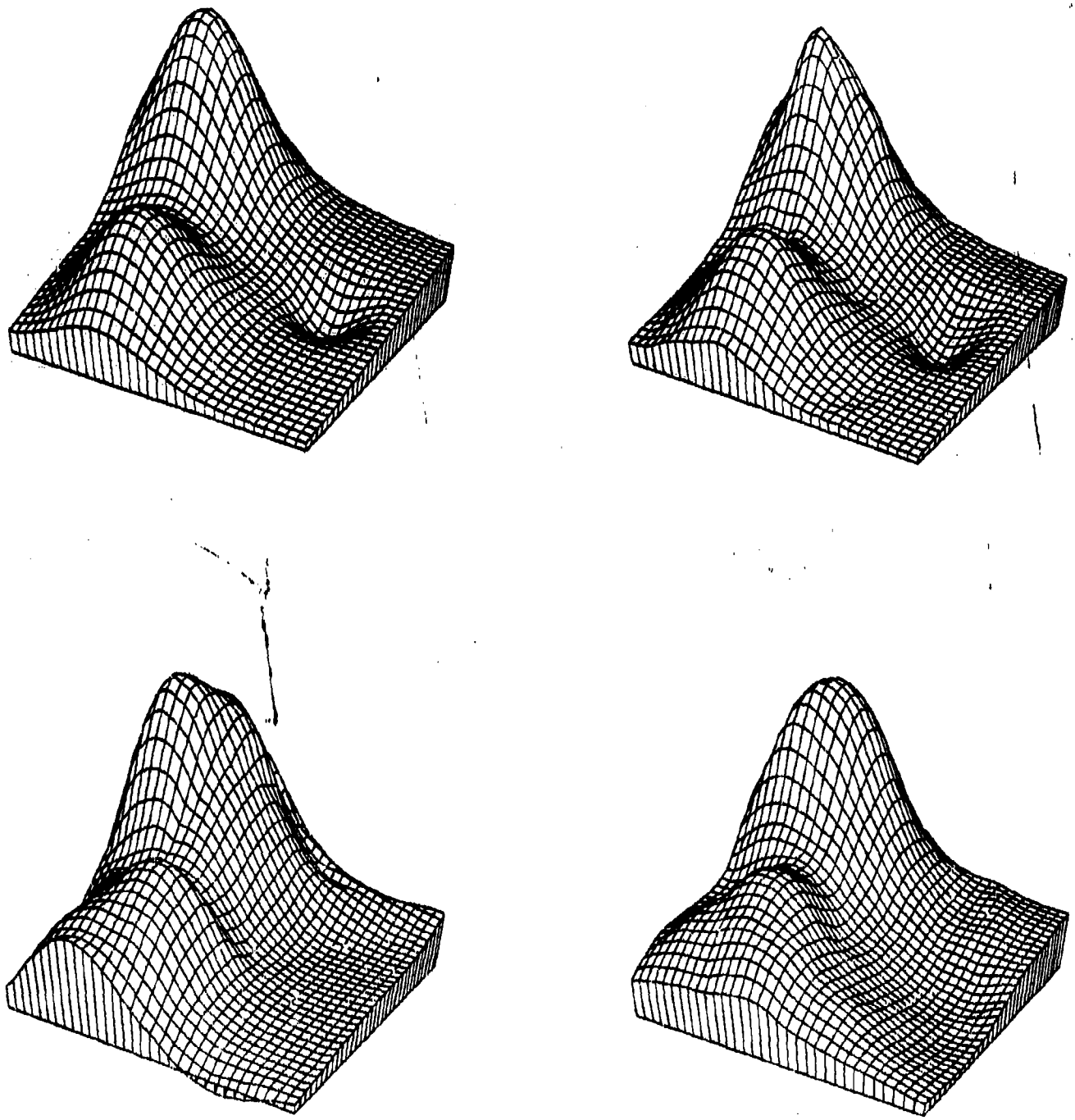
Akima's Method, Modification One

Figure 1.4.1.10



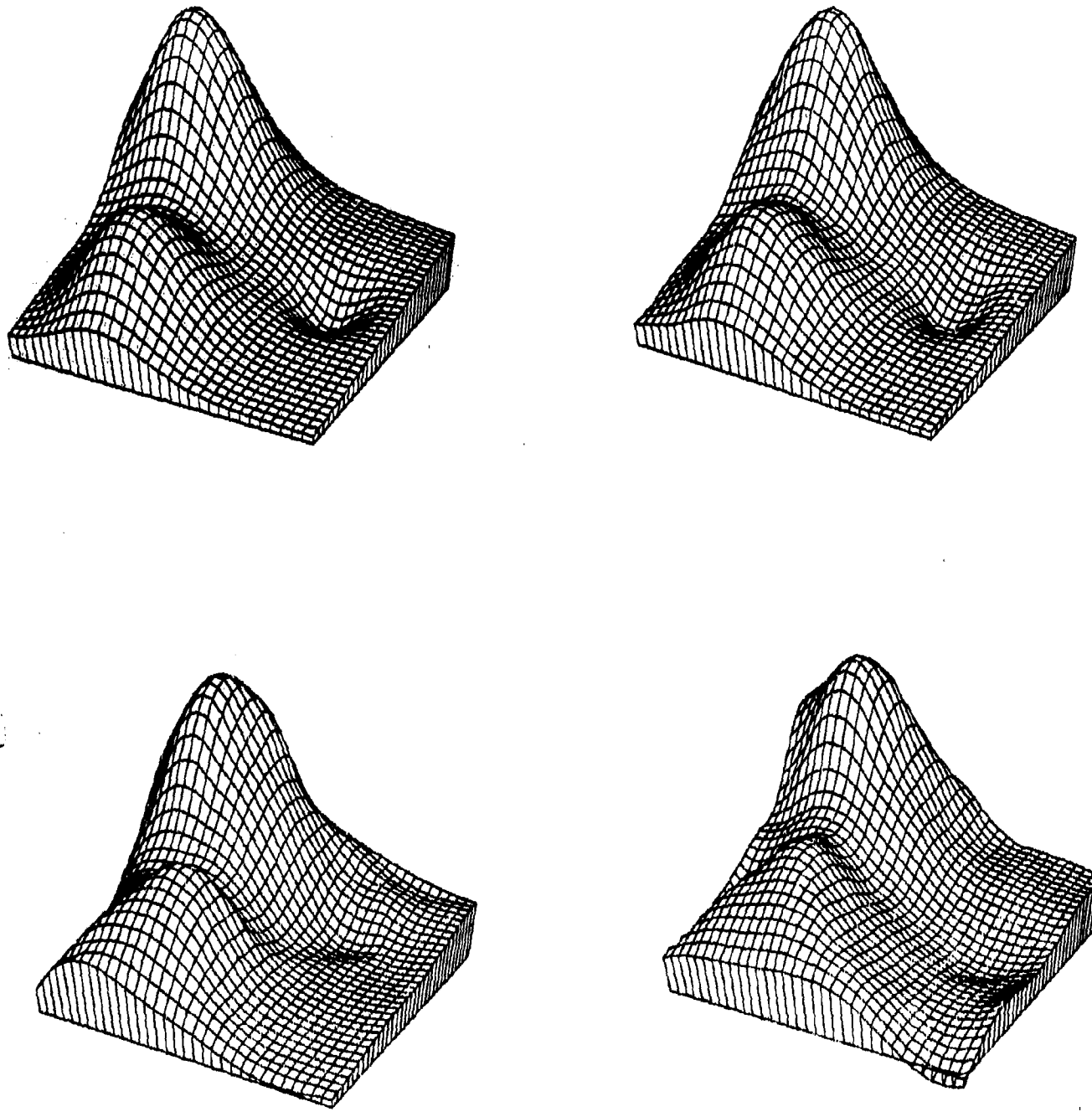
Akima's Method, Modification Two

Figure 1.4.1.11



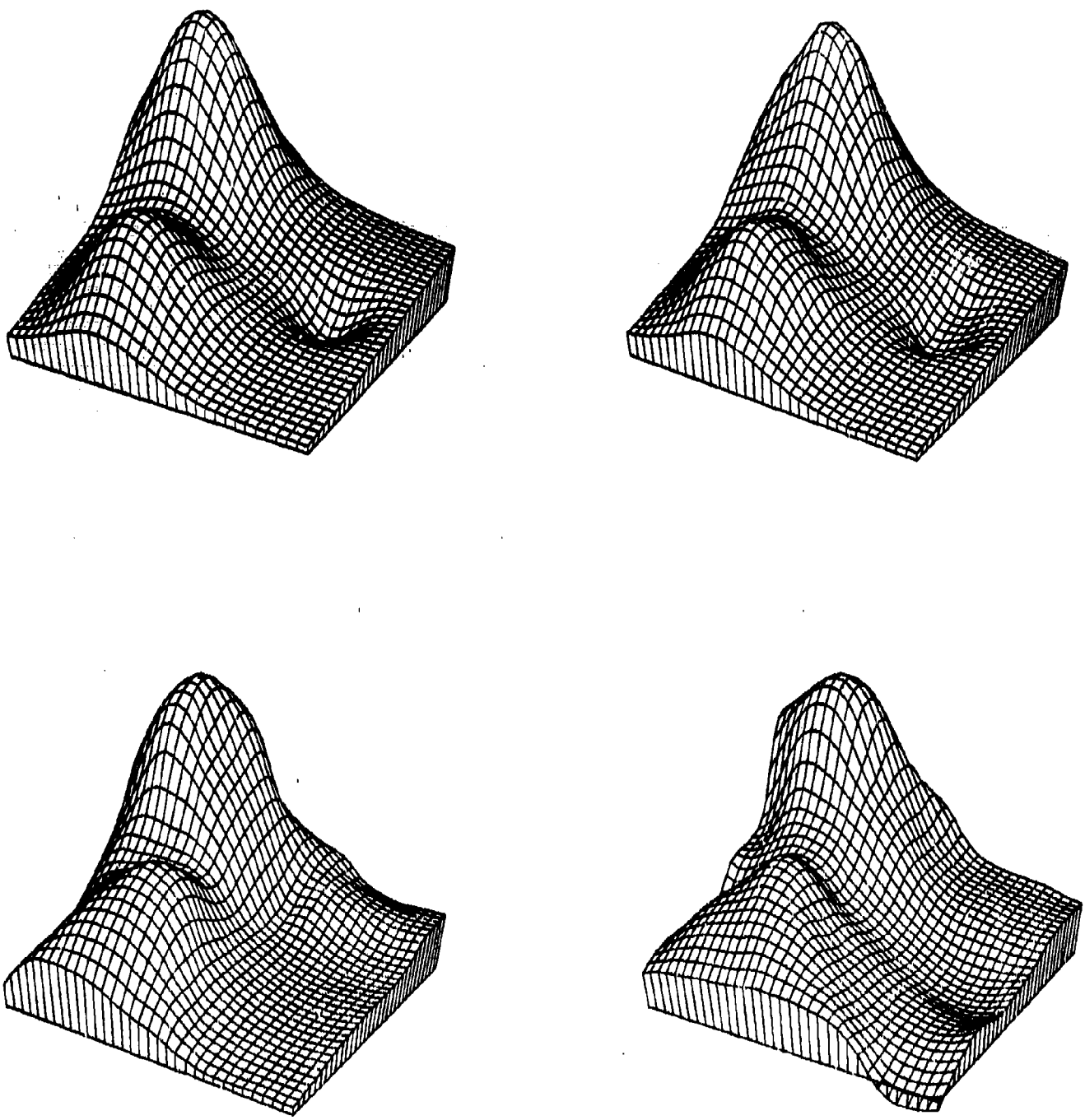
Nielson-Franke Linear Triangle Method

Figure 1.4.1.12



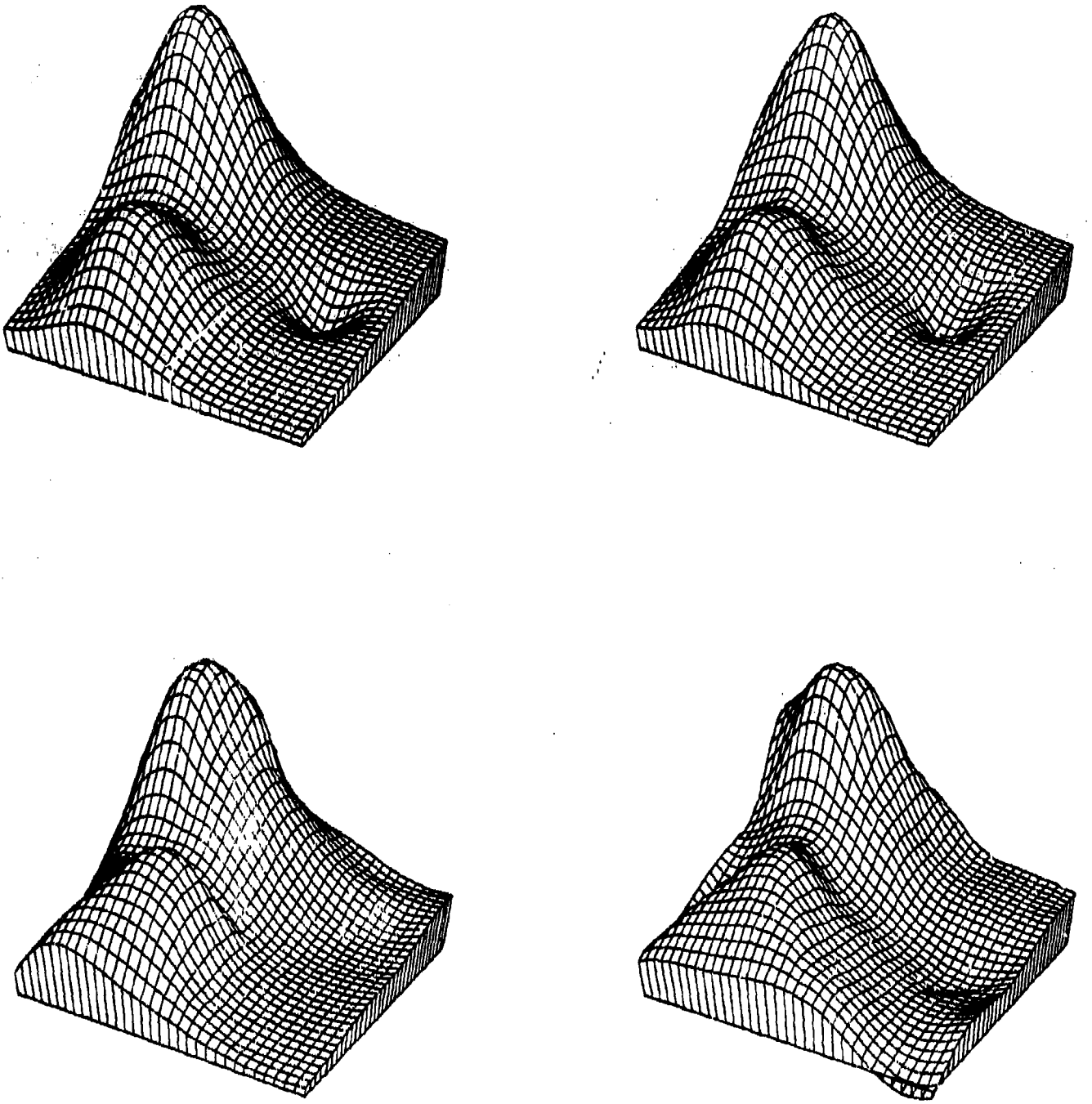
Nielson-Franke Quadratic Triangular Method

Figure 1.4.1.13



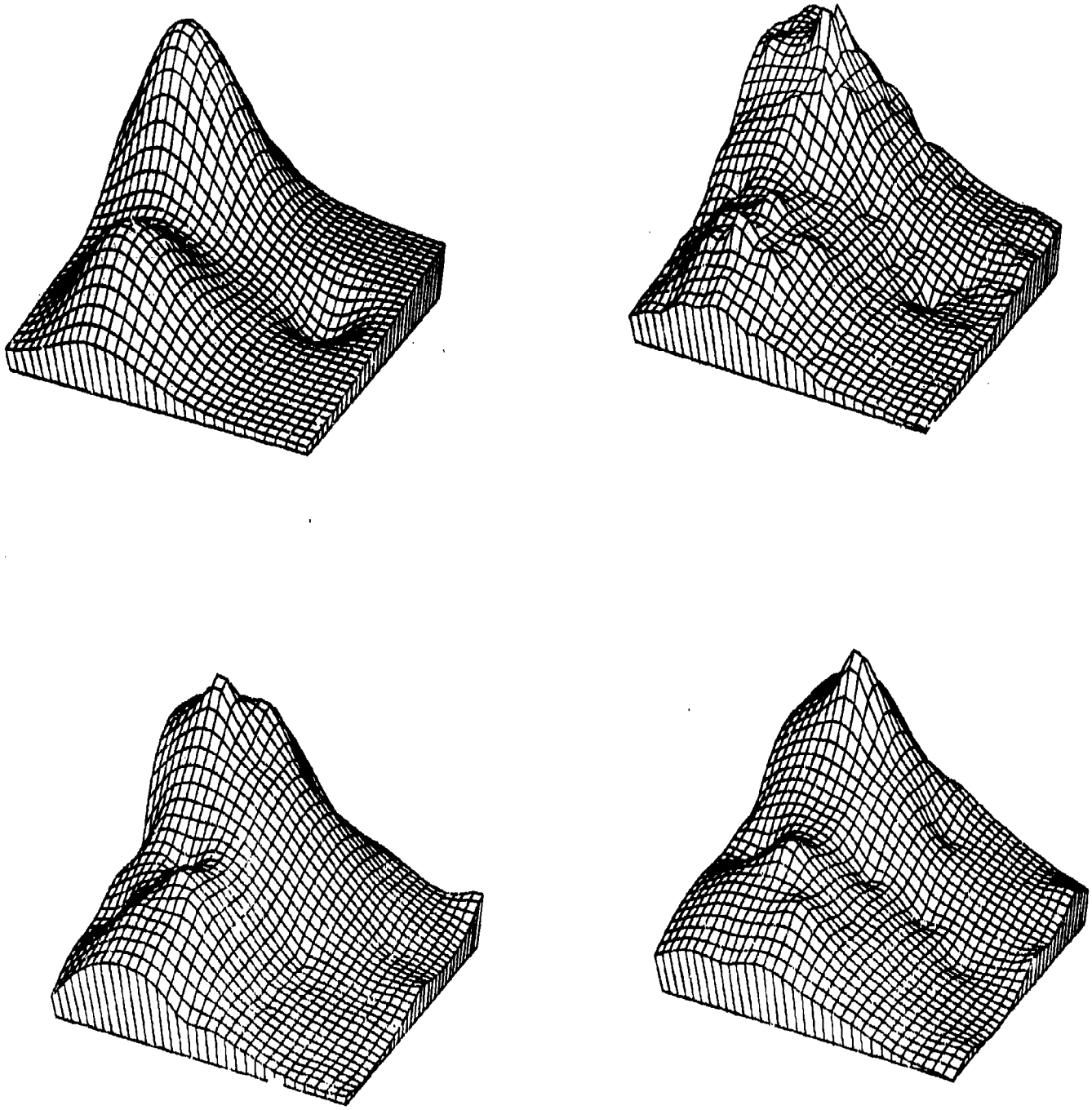
Modified Quadratic Shepard Method

Figure 1.4.1.14



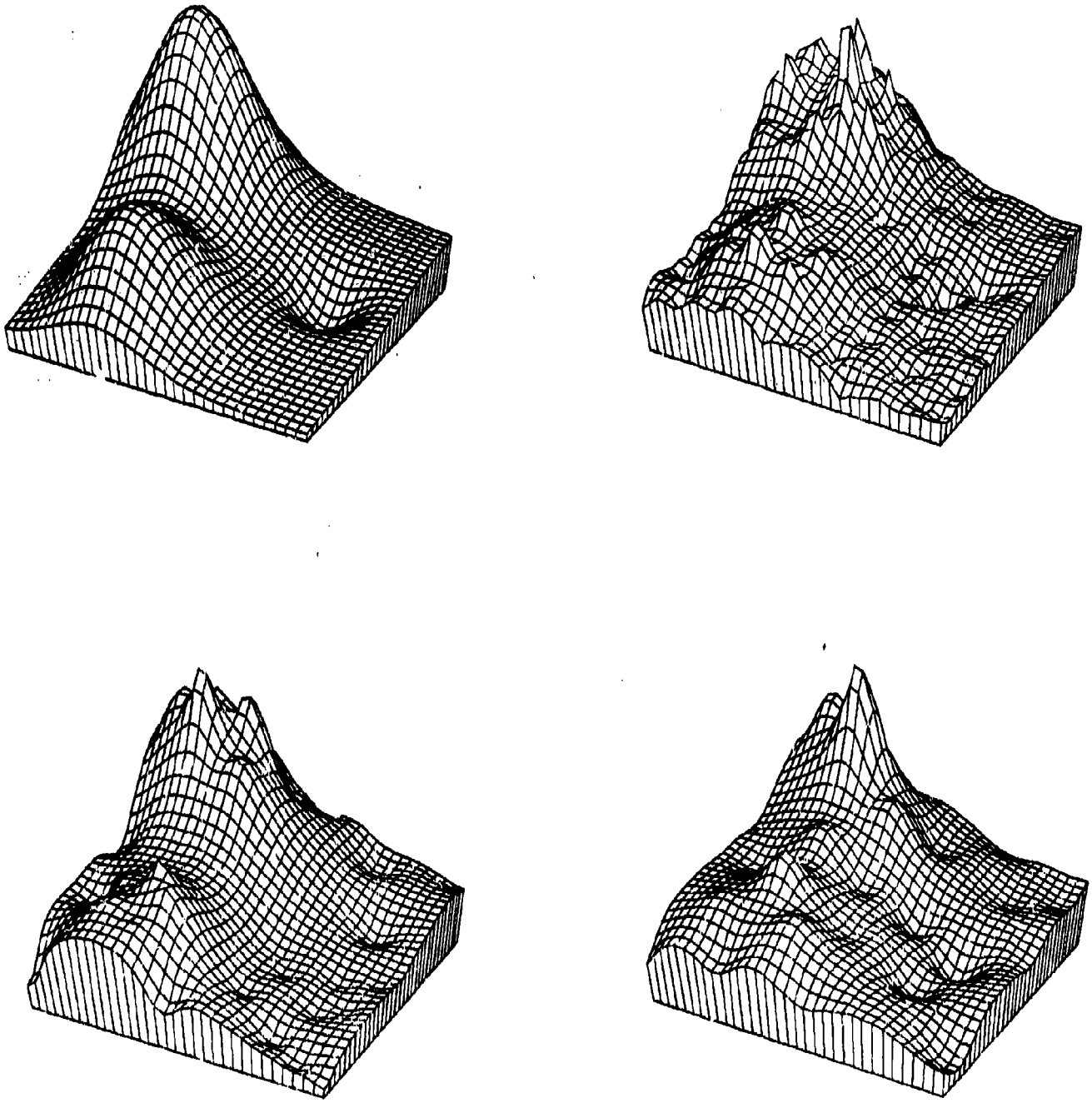
Akima's Method, Modification Three

Figure 1.4.1.16



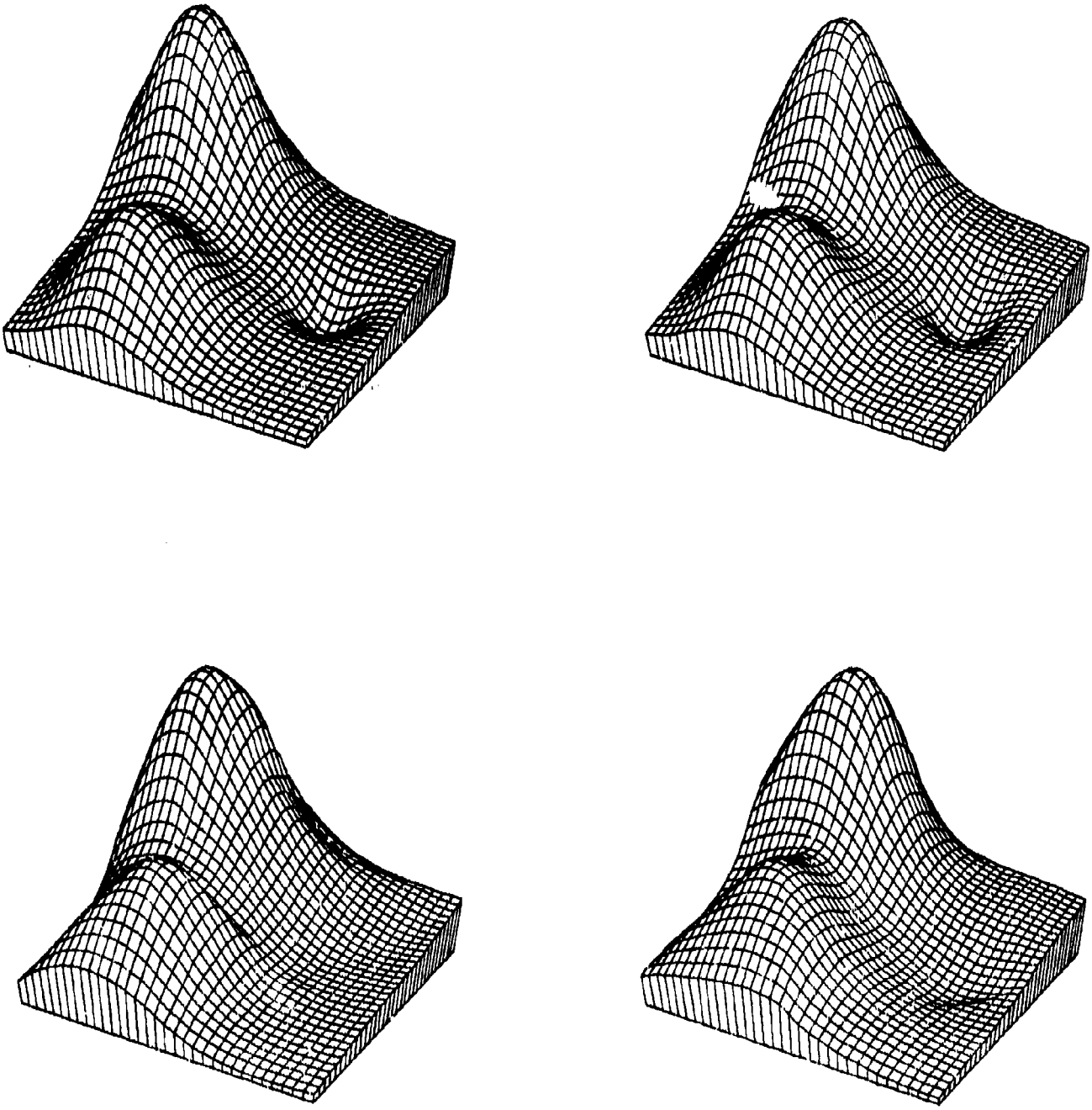
Quadratic Shepard's Method

Figure 1.4.1.17



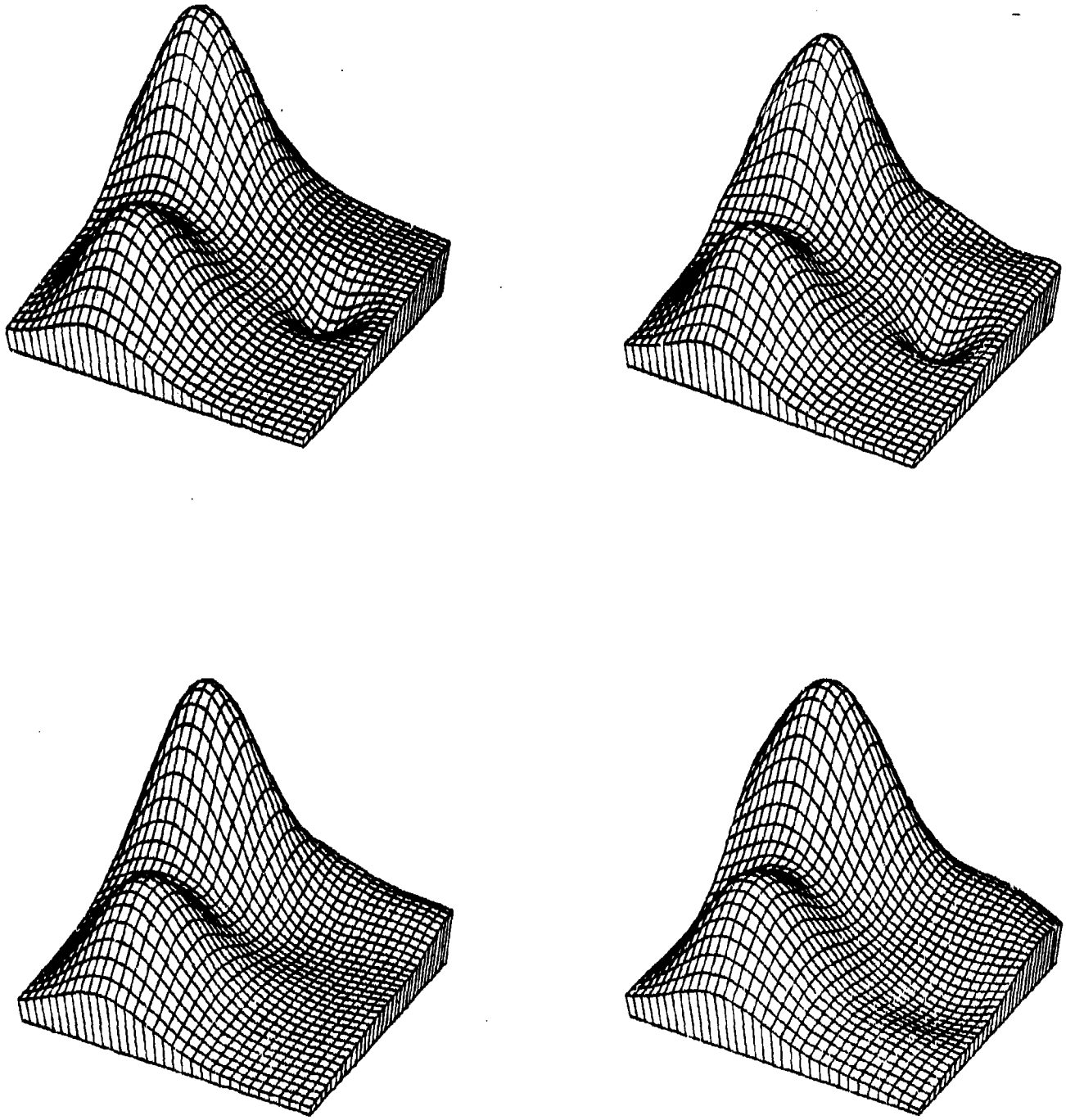
Shepard's Method

Figure 1.4.1.18



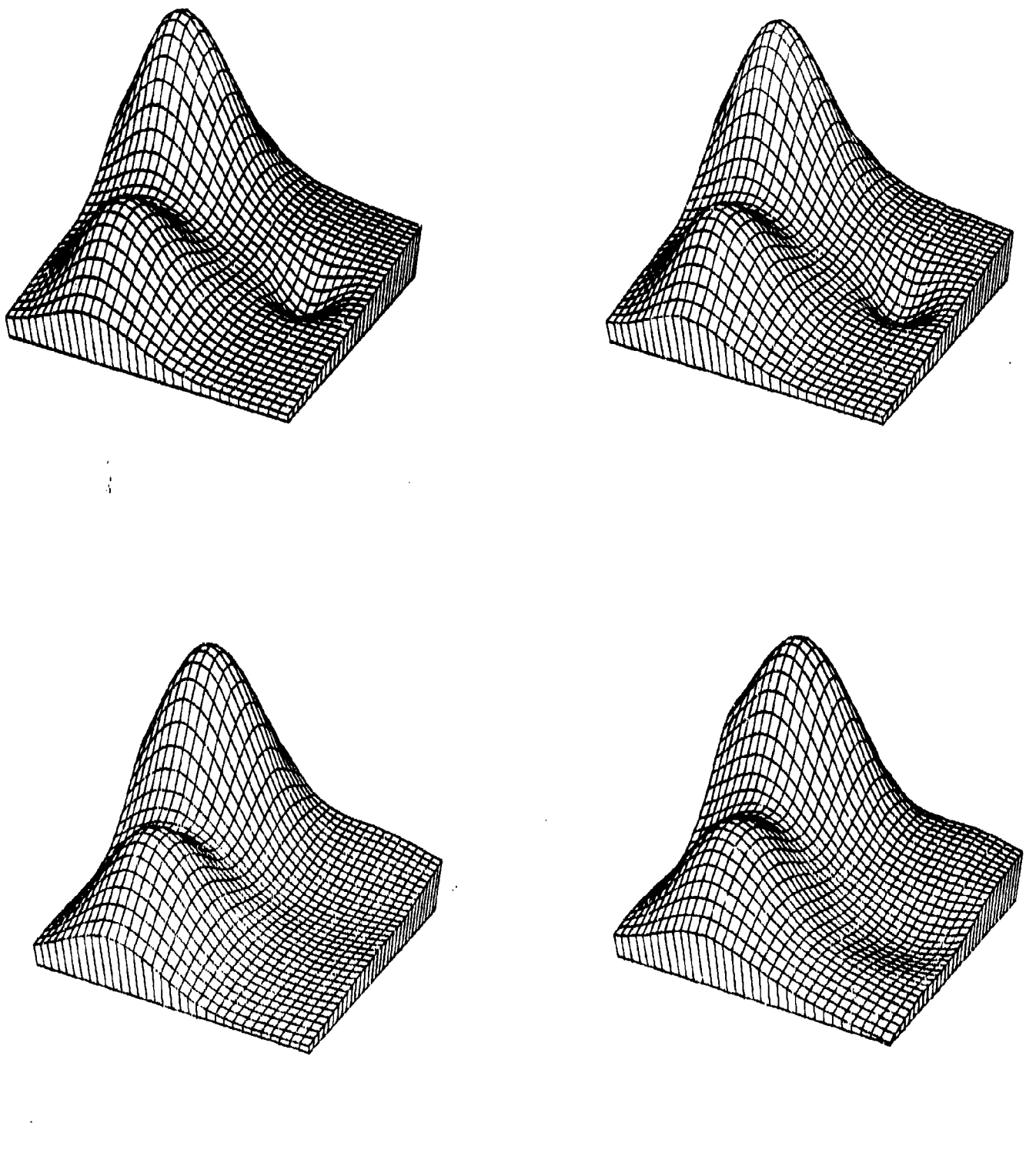
Nielson Minimum Norm Network

Figure 1.4.1.19



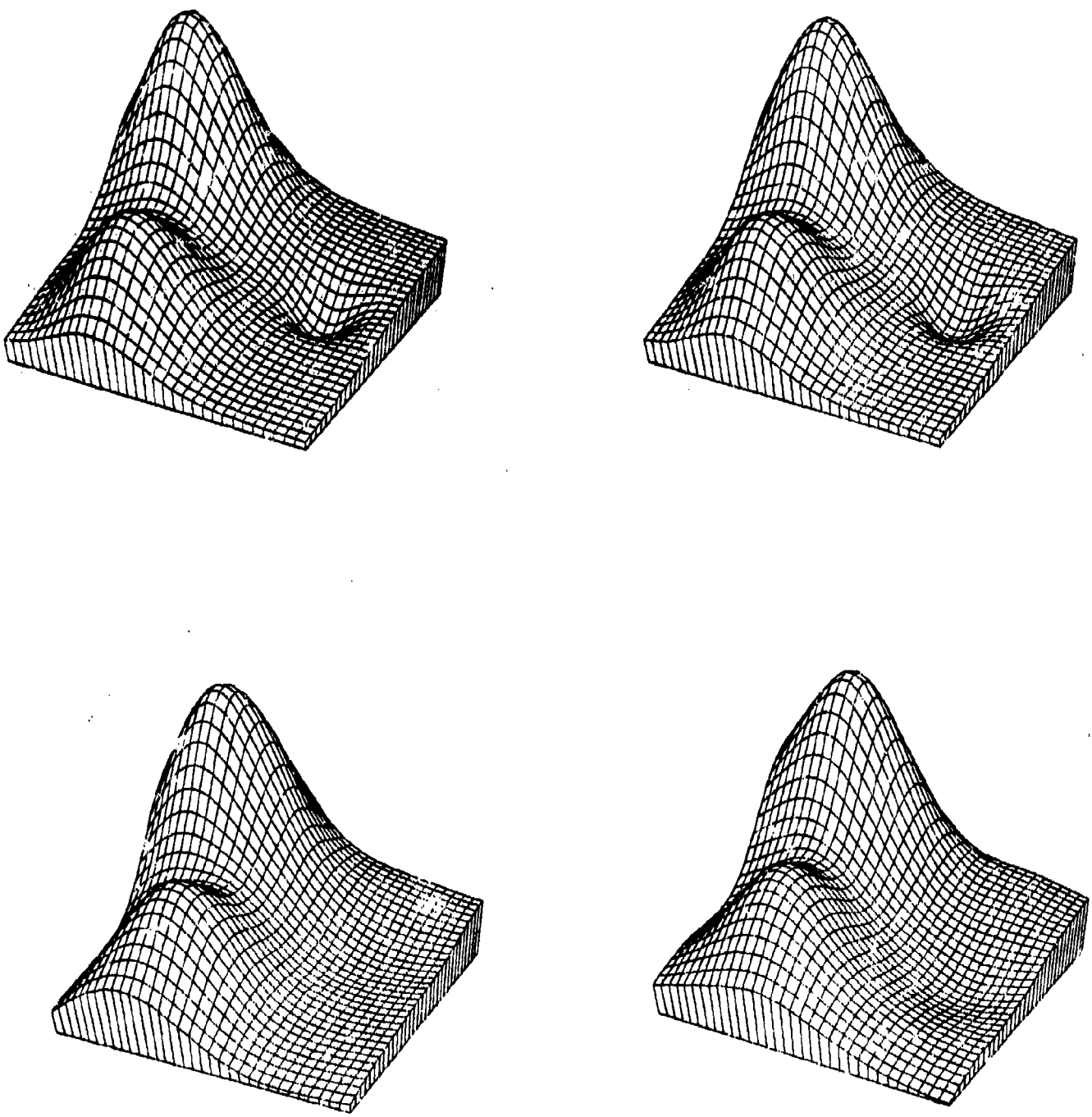
Rotated Gaussian Basis Functions

Figure 1.4.1.20



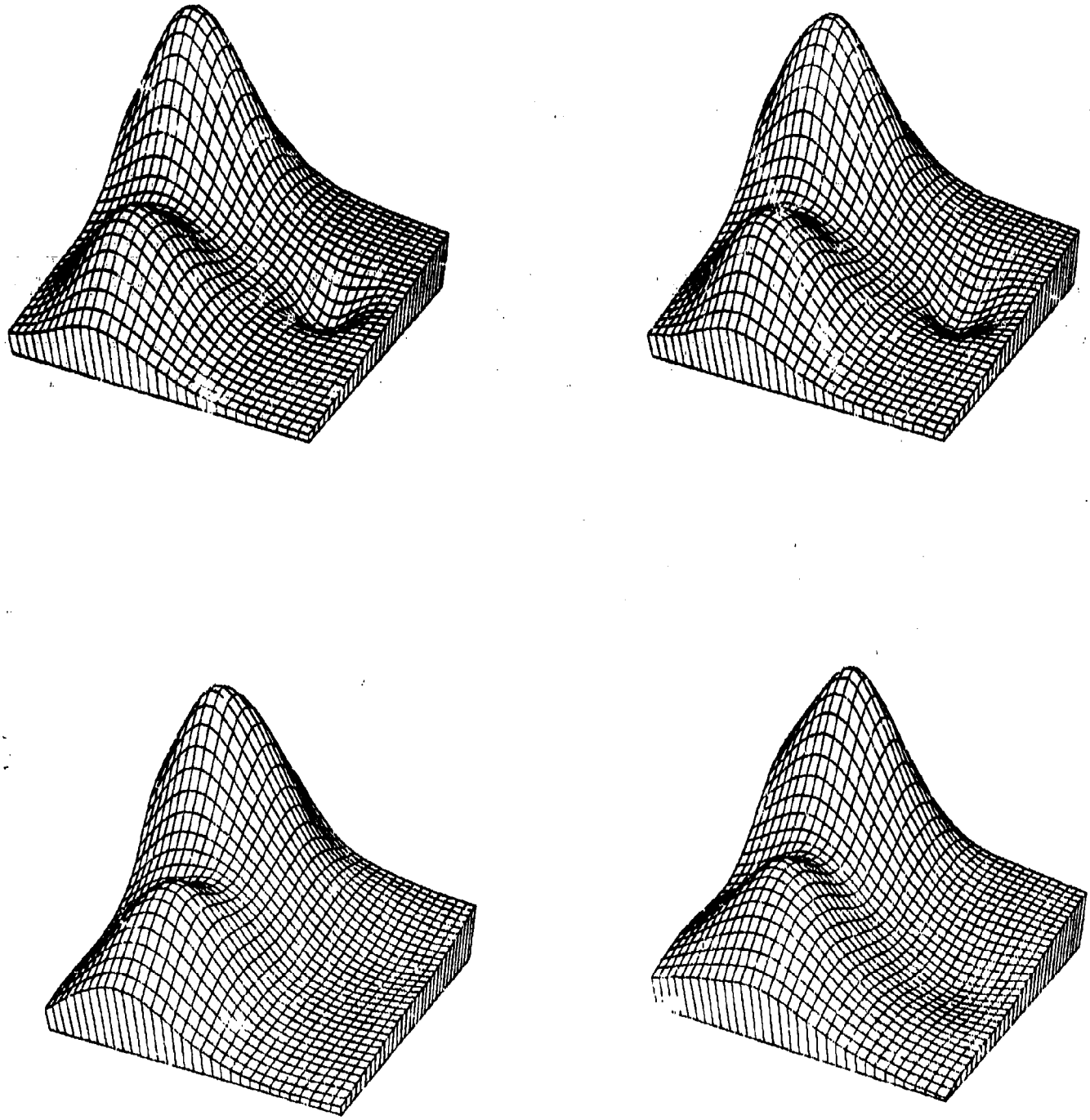
Hardy Multiquadric Method

Figure 1.4.1.21



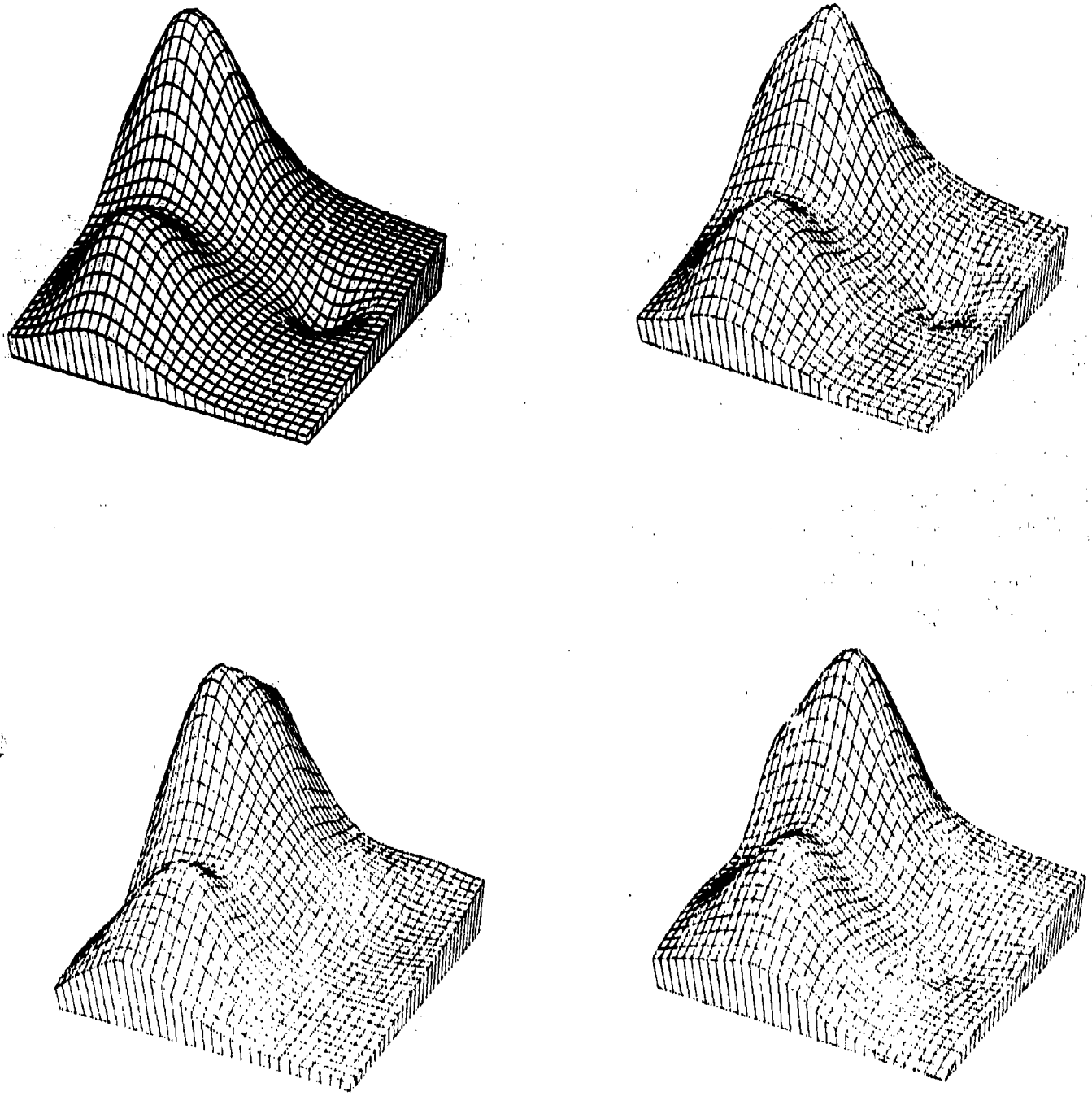
Duchon's Radial Cubics

Figure 1.4.1.22



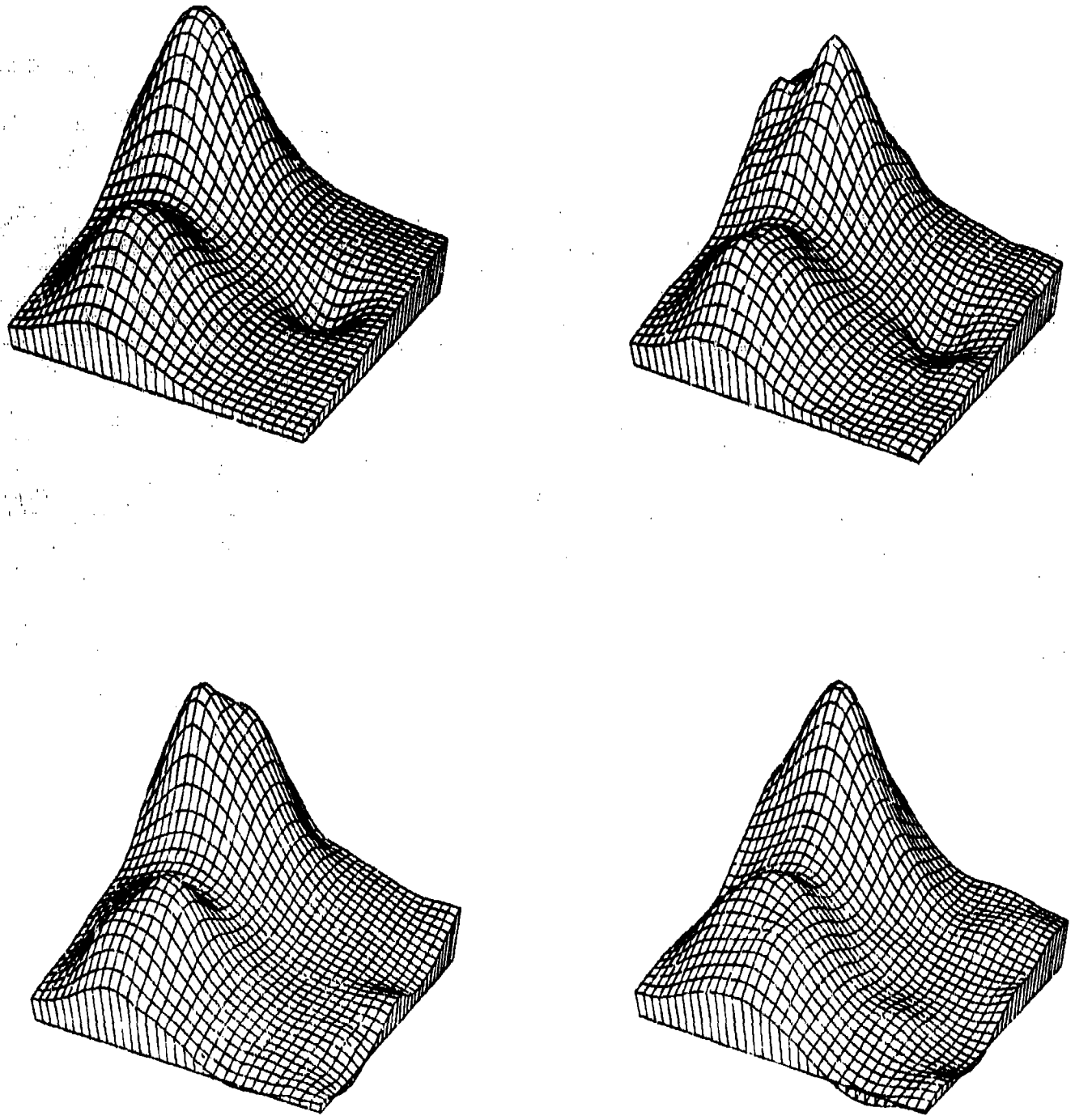
Durhon's Thin Plate Method

Figure 1.4.1.23



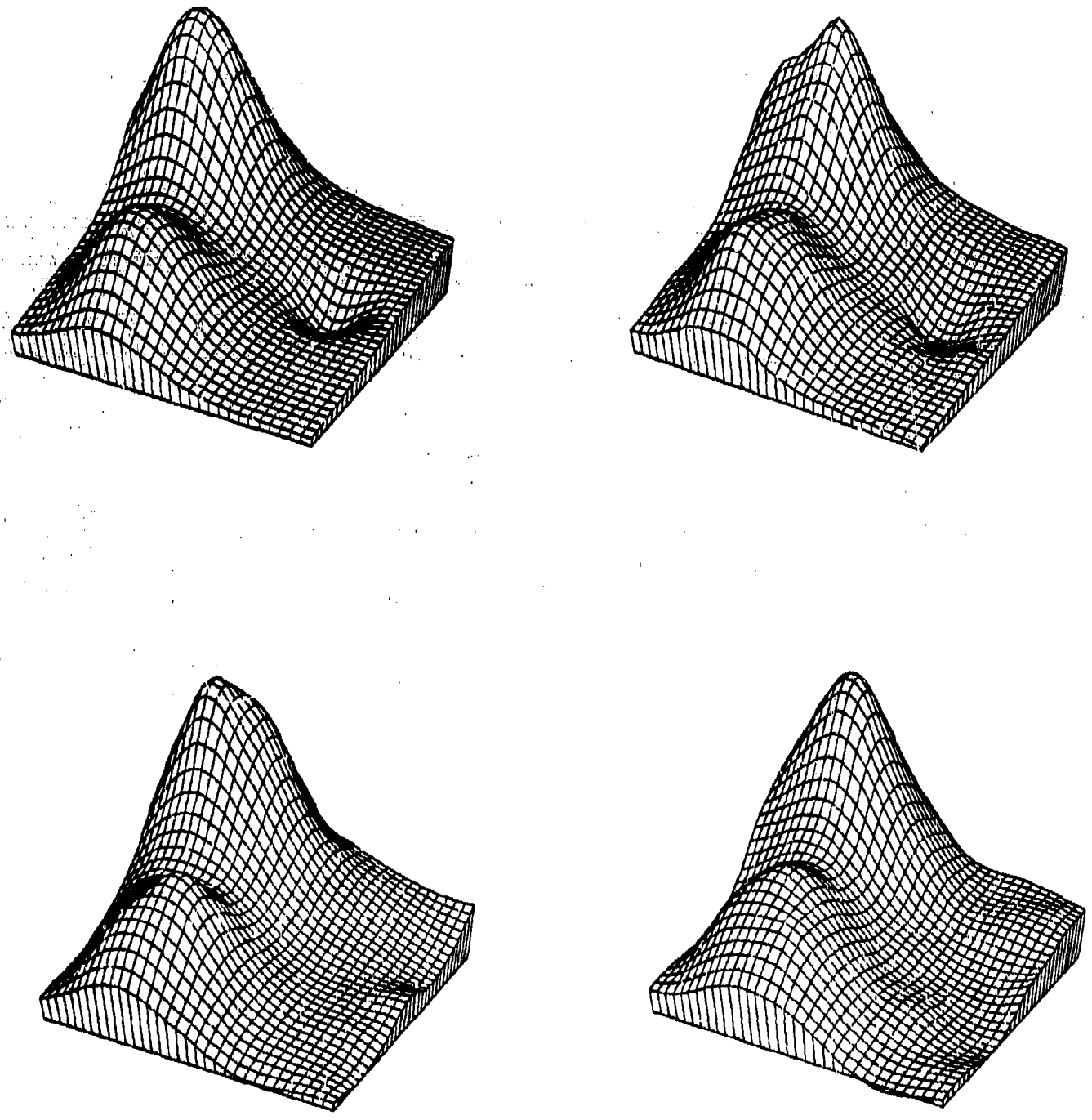
Franke's Method, Thin Plate Local Functions

Figure 1.4.1.24



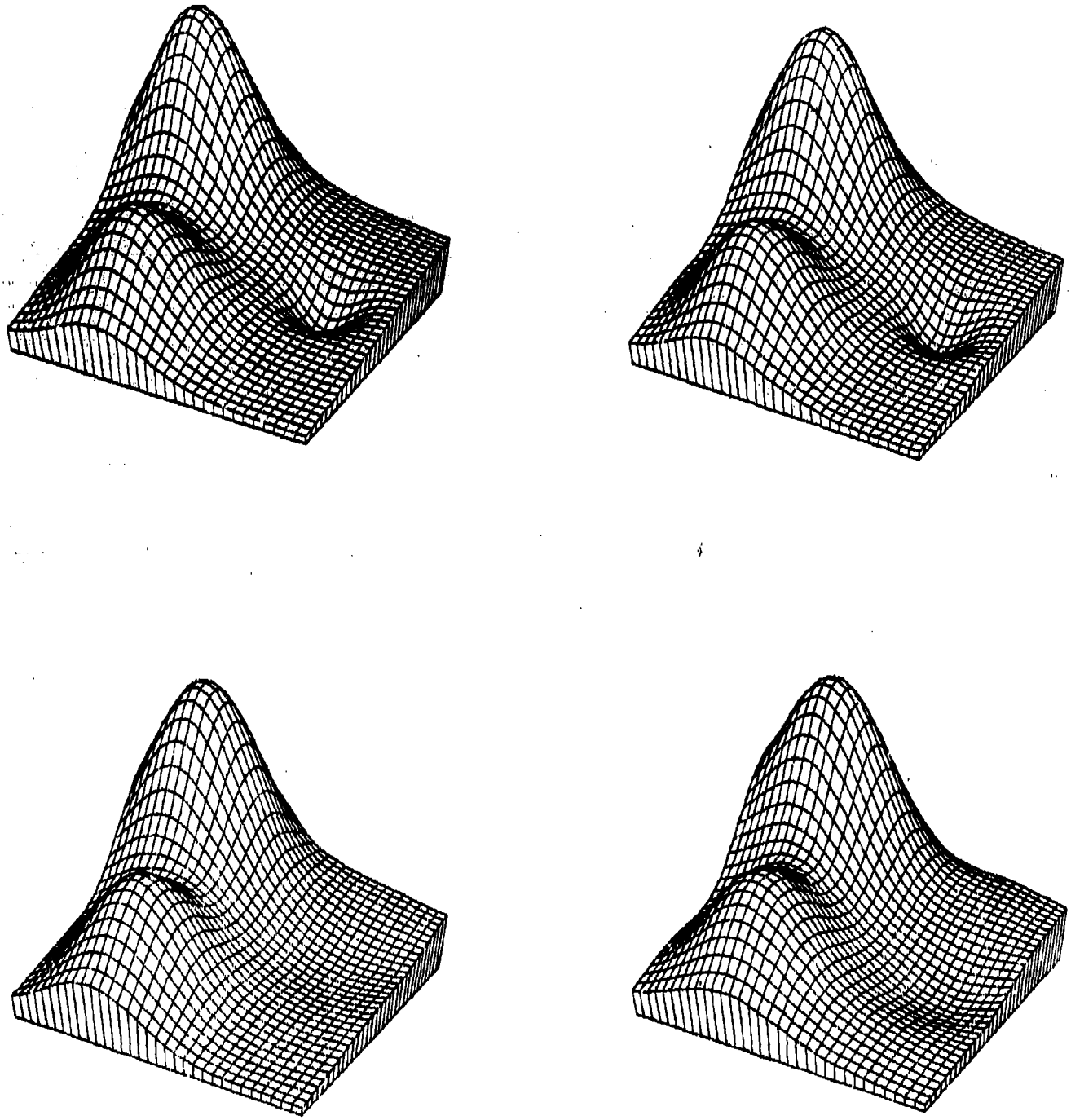
Foley's Generalized Newton Polynomial

Figure 1.4.1.25



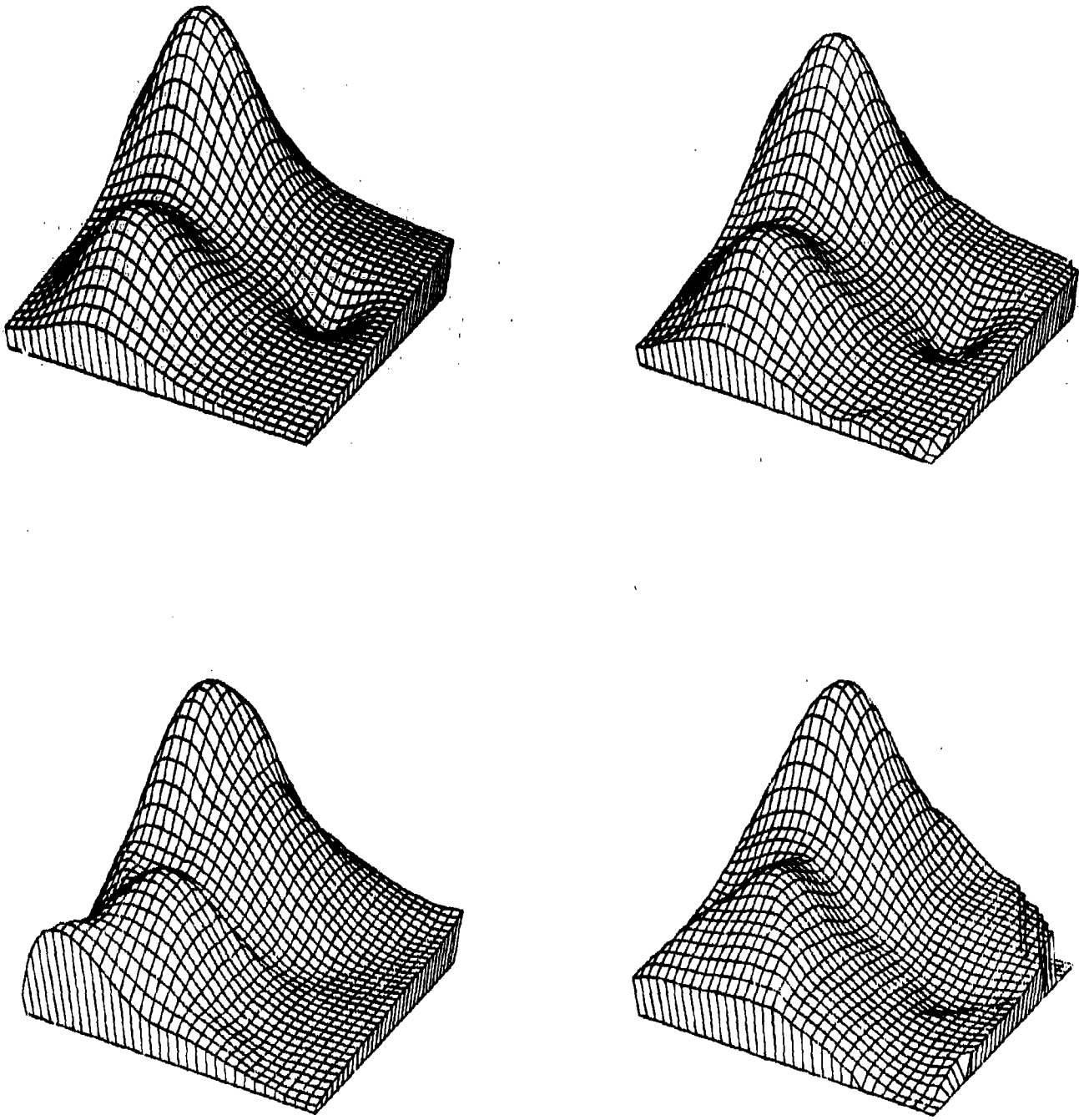
Foley's Generalized Newton Boolean Sum Bernstein - G. N.

Figure 1.4.1.26



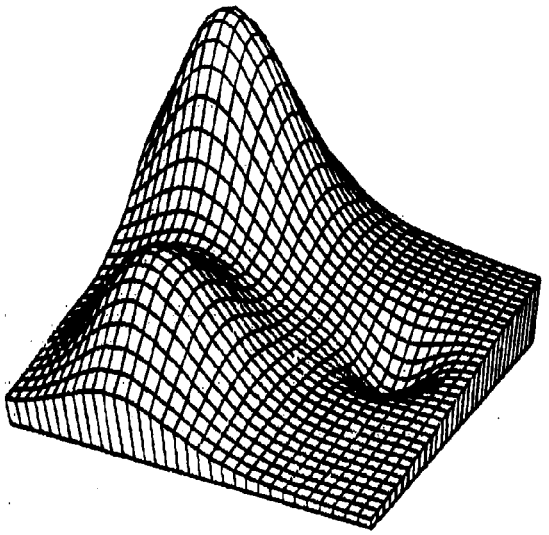
Hardy Reciprocal Multiquadric Method

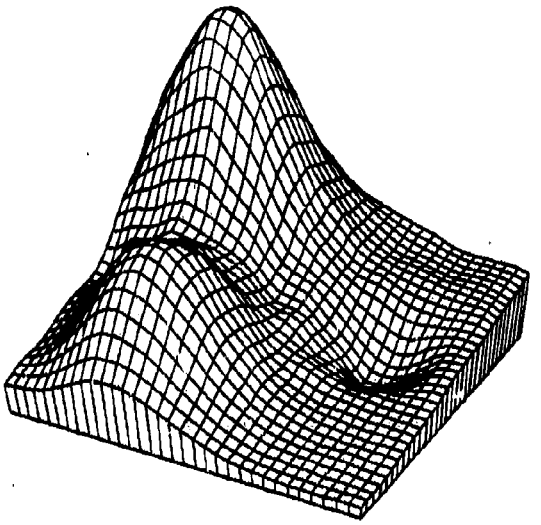
Figure 1.4.1.27

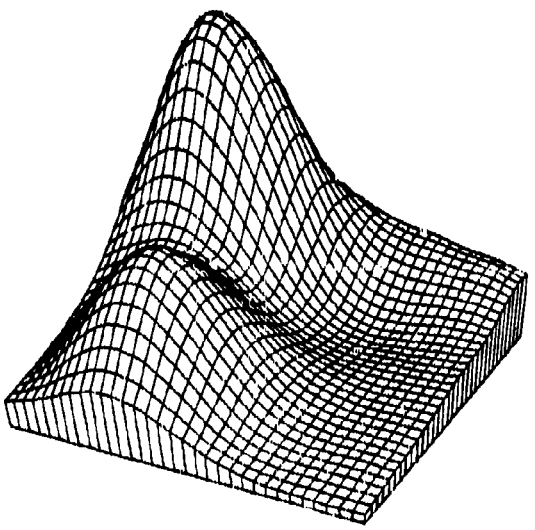


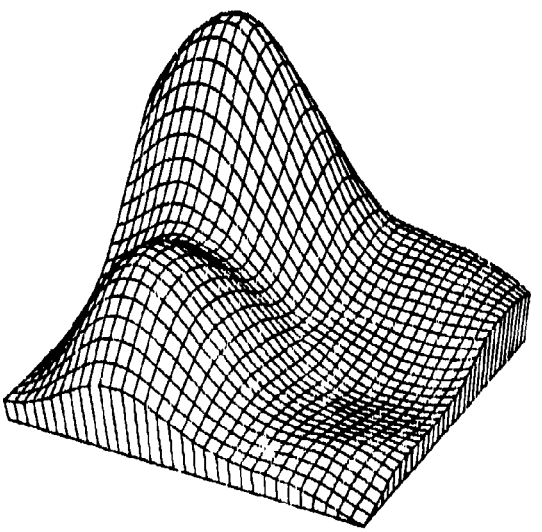
Lawson's Method

Figure 1.4.1.28



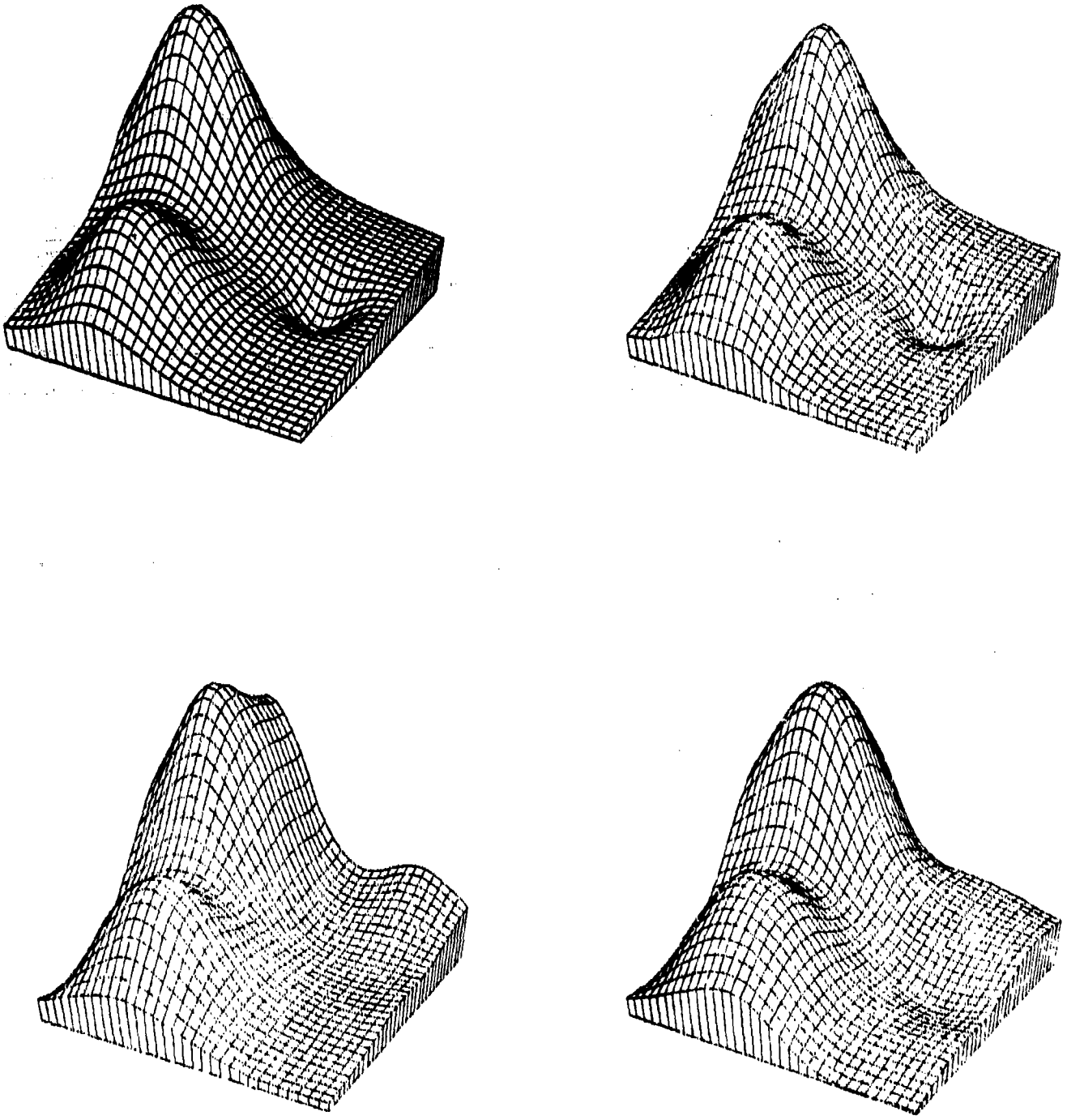






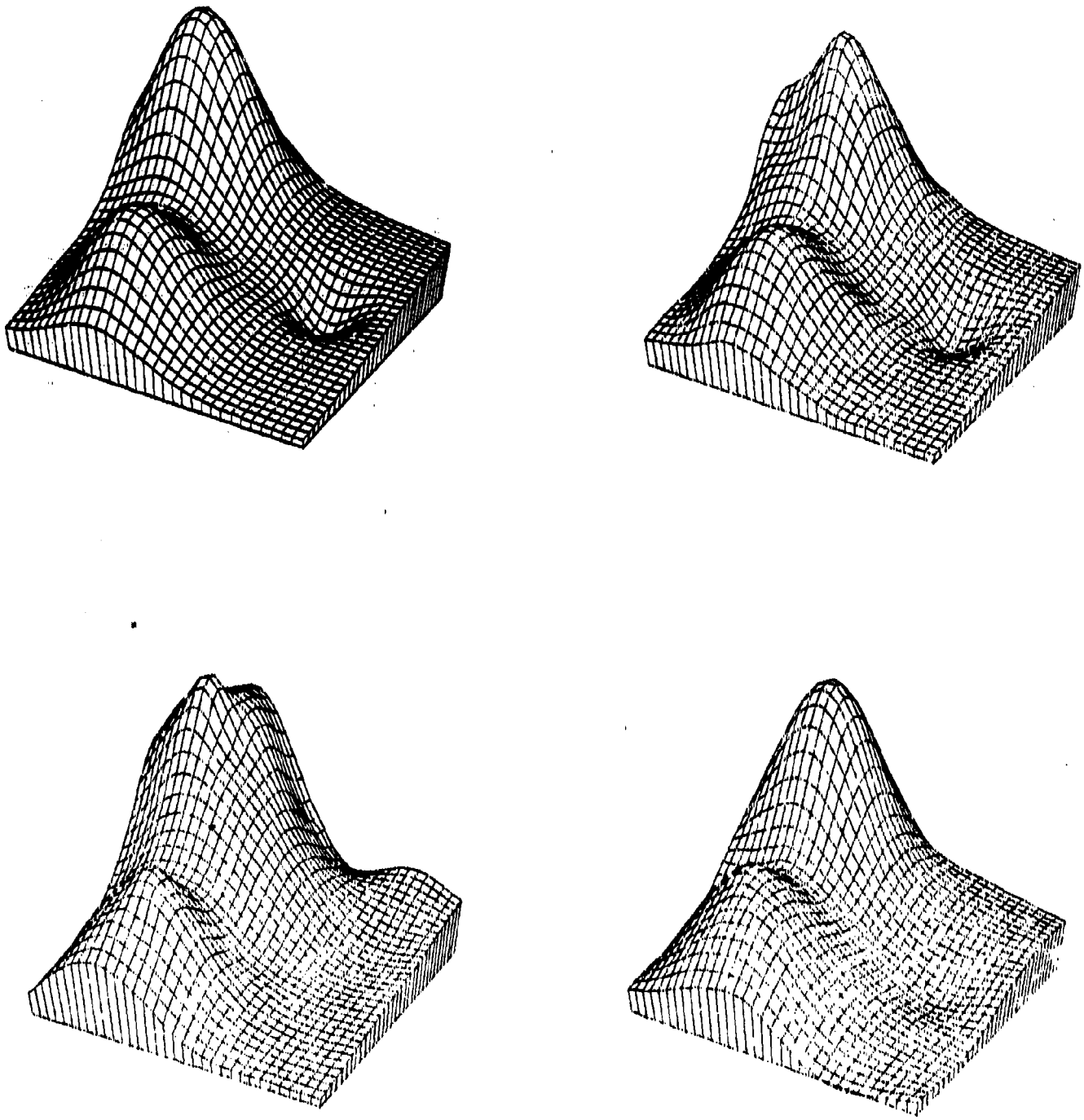
Rotated Cubic B-Spline Basis

Figure 1.4.1.29



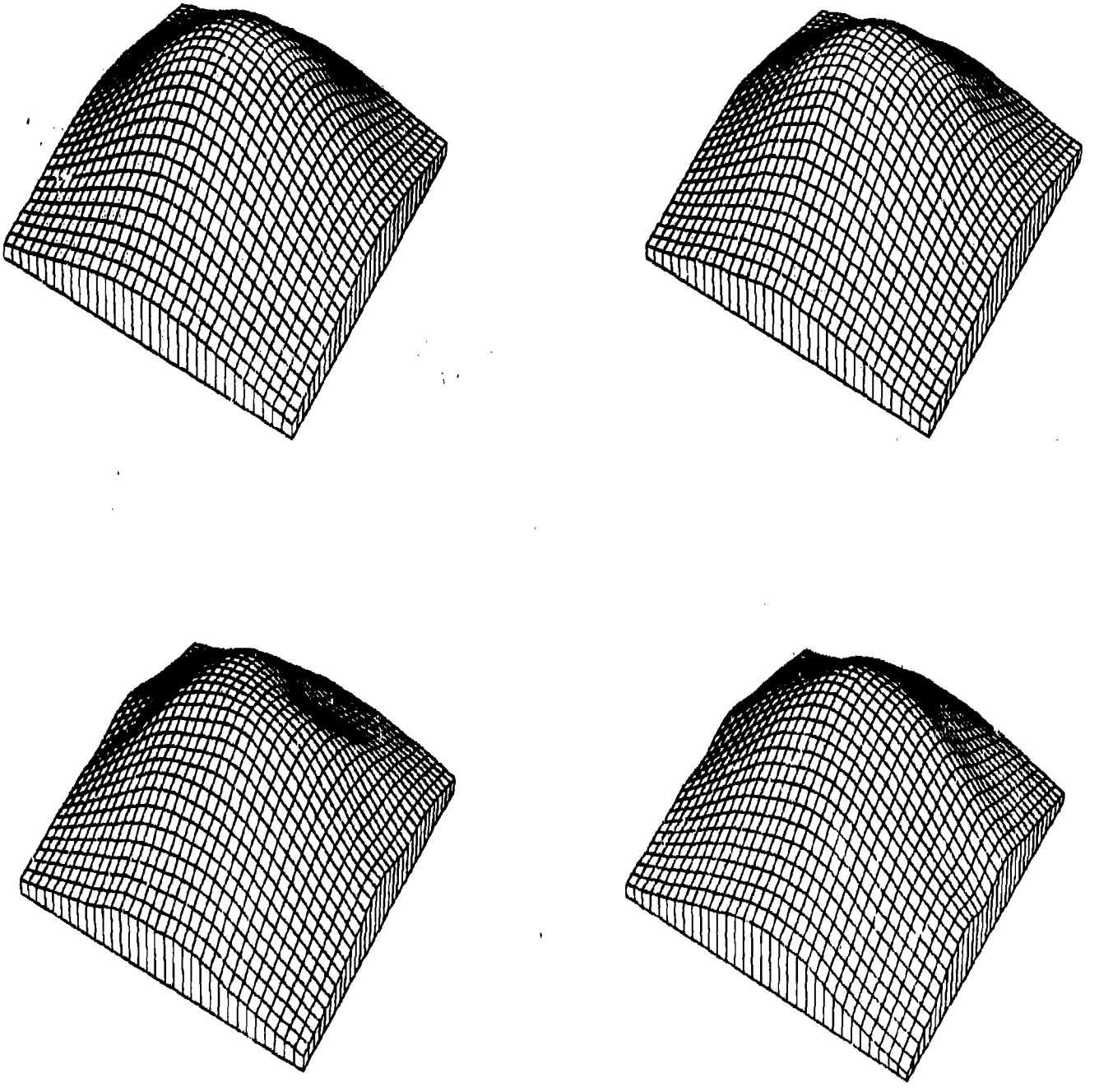
Foley's Iterated Generalized Newton
Delta Sum Bicubic Spline

Figure 1.4.1.30



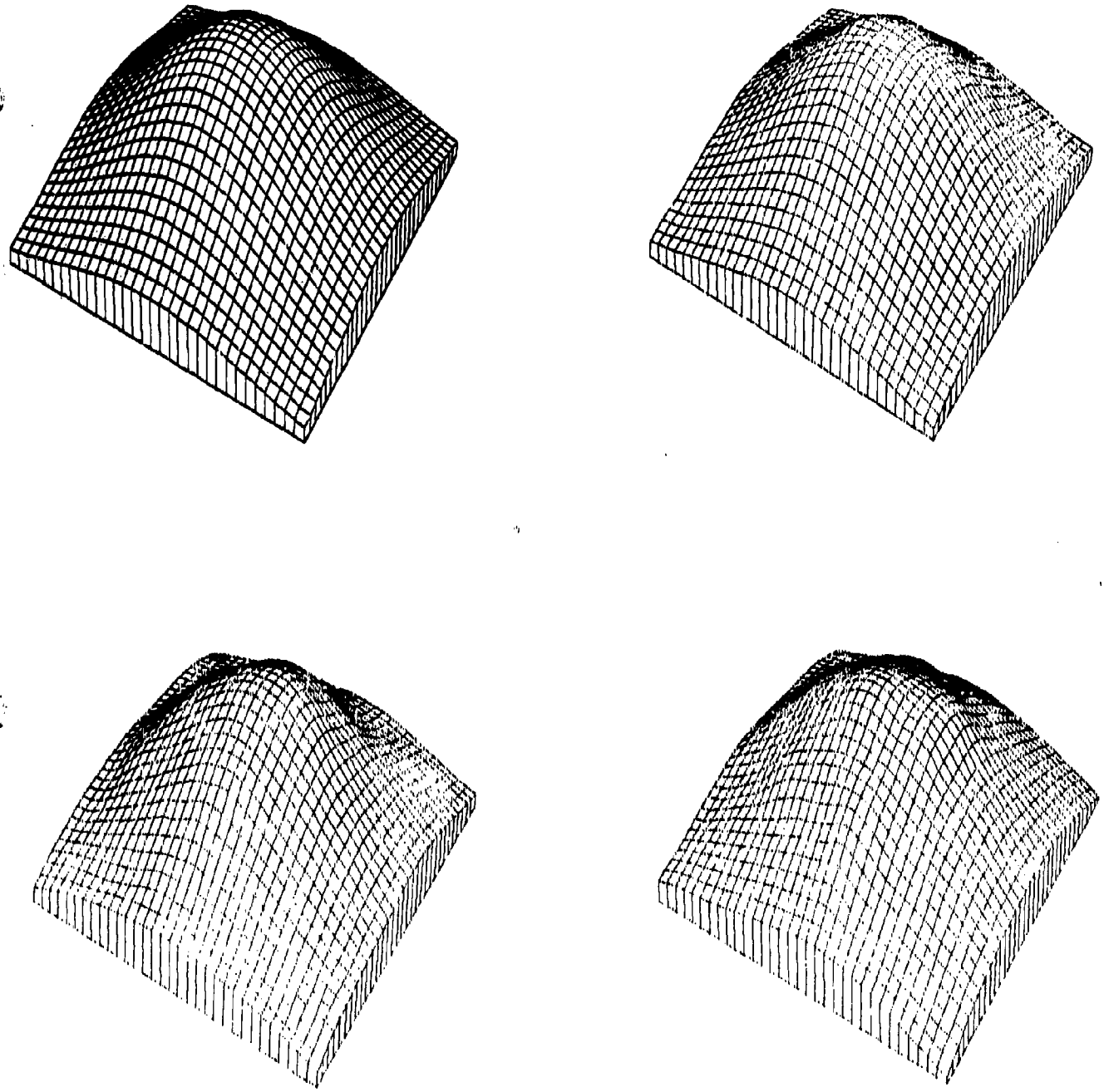
Foley's Iterated Modified Shepard
Delta Sum Bicubic Spline

Figure 1.4.1.31



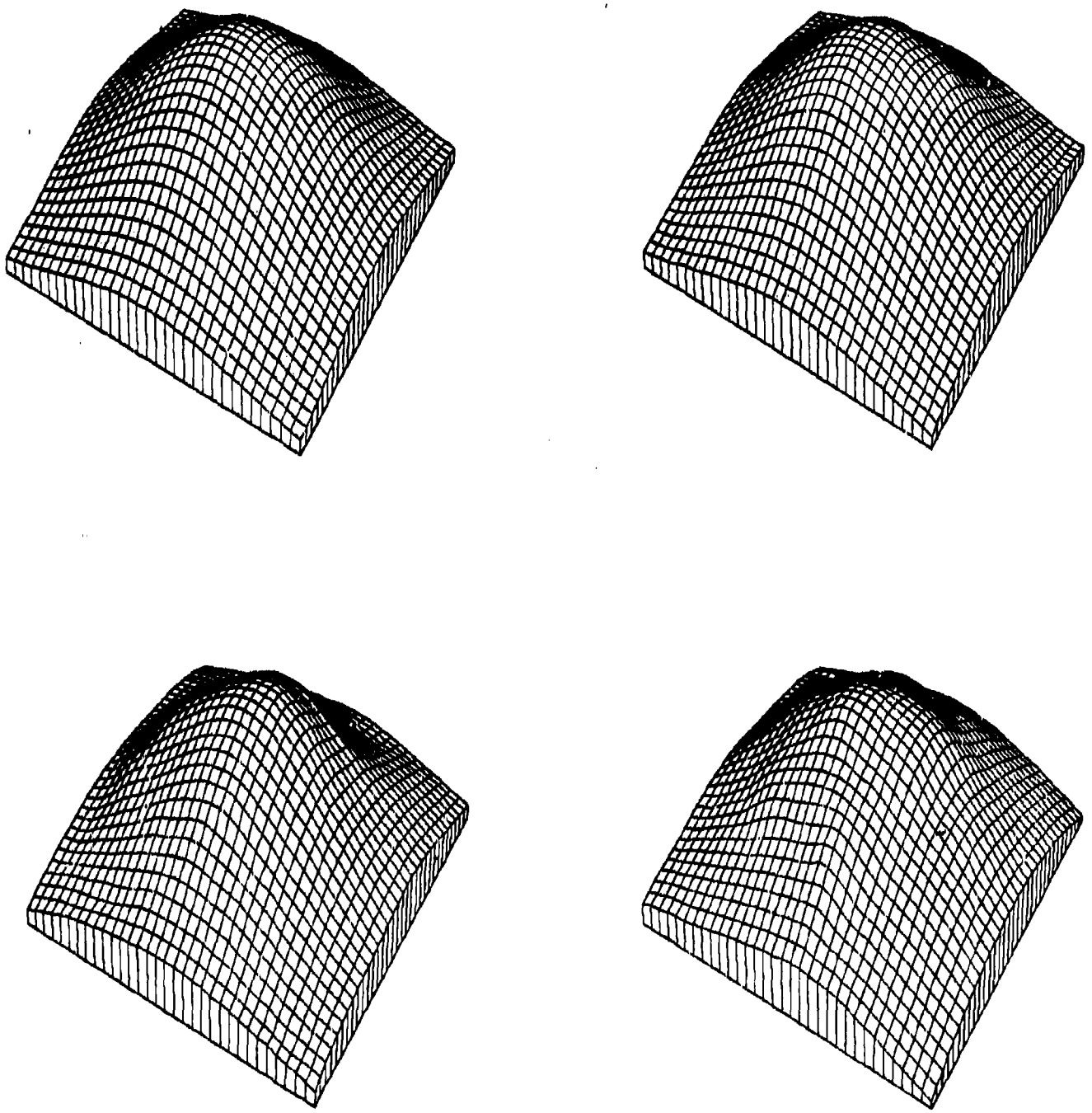
Franke's Method, Mode = 3

Figure 1.4.4.1



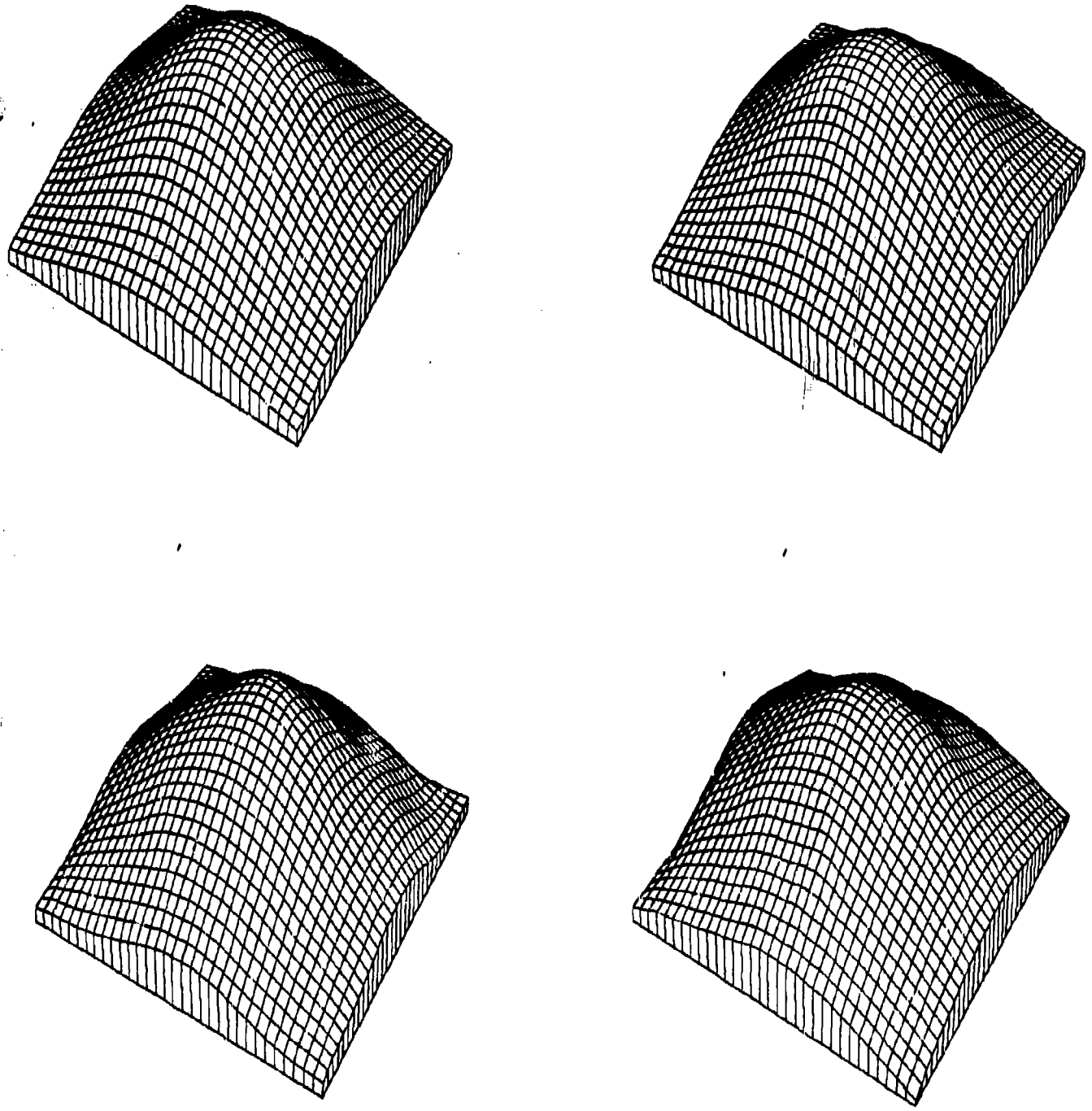
Akima's Method

Figure 1.4.4.4



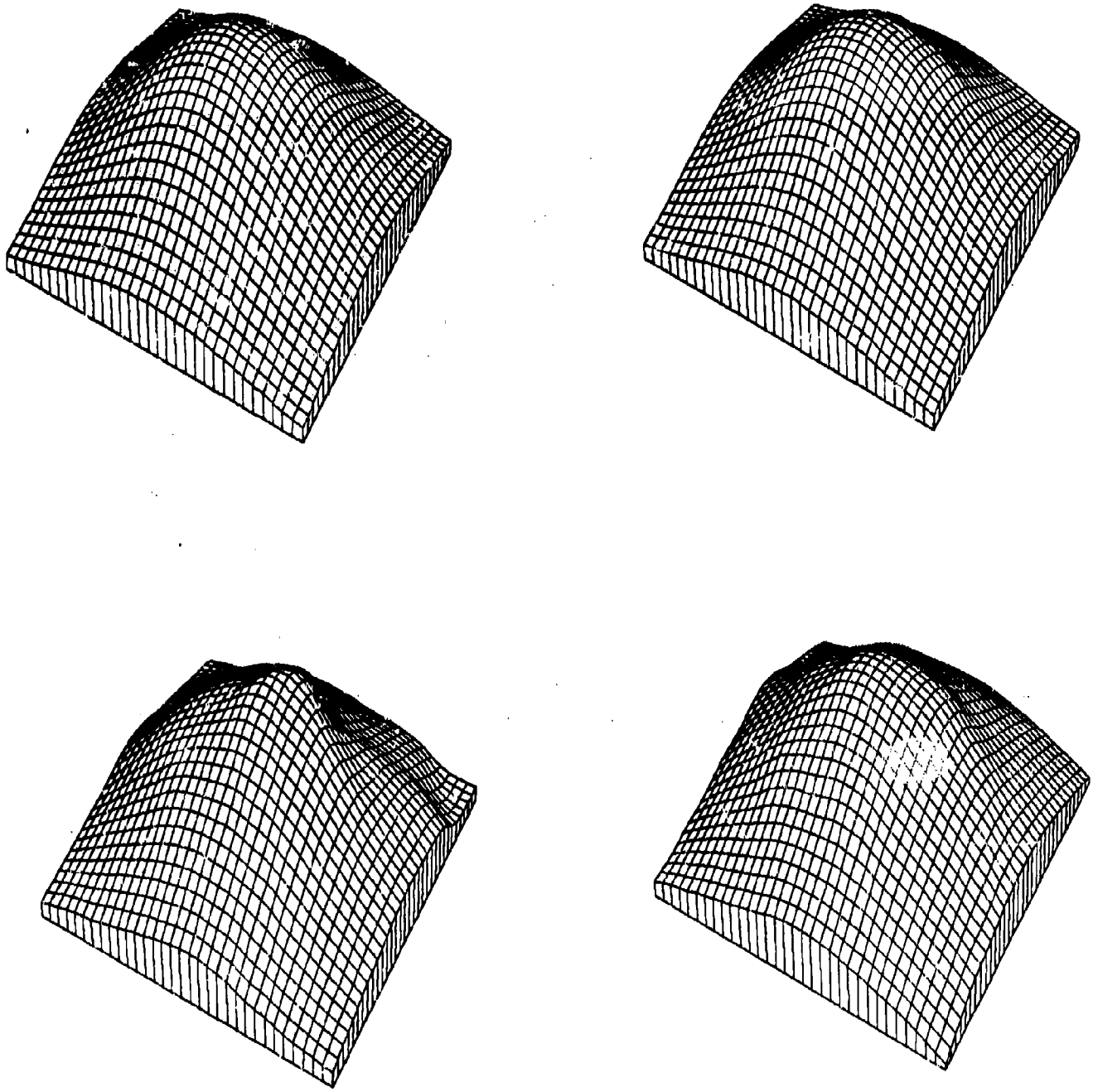
Akima's Method, Modification One

Figure 1.4.4.10



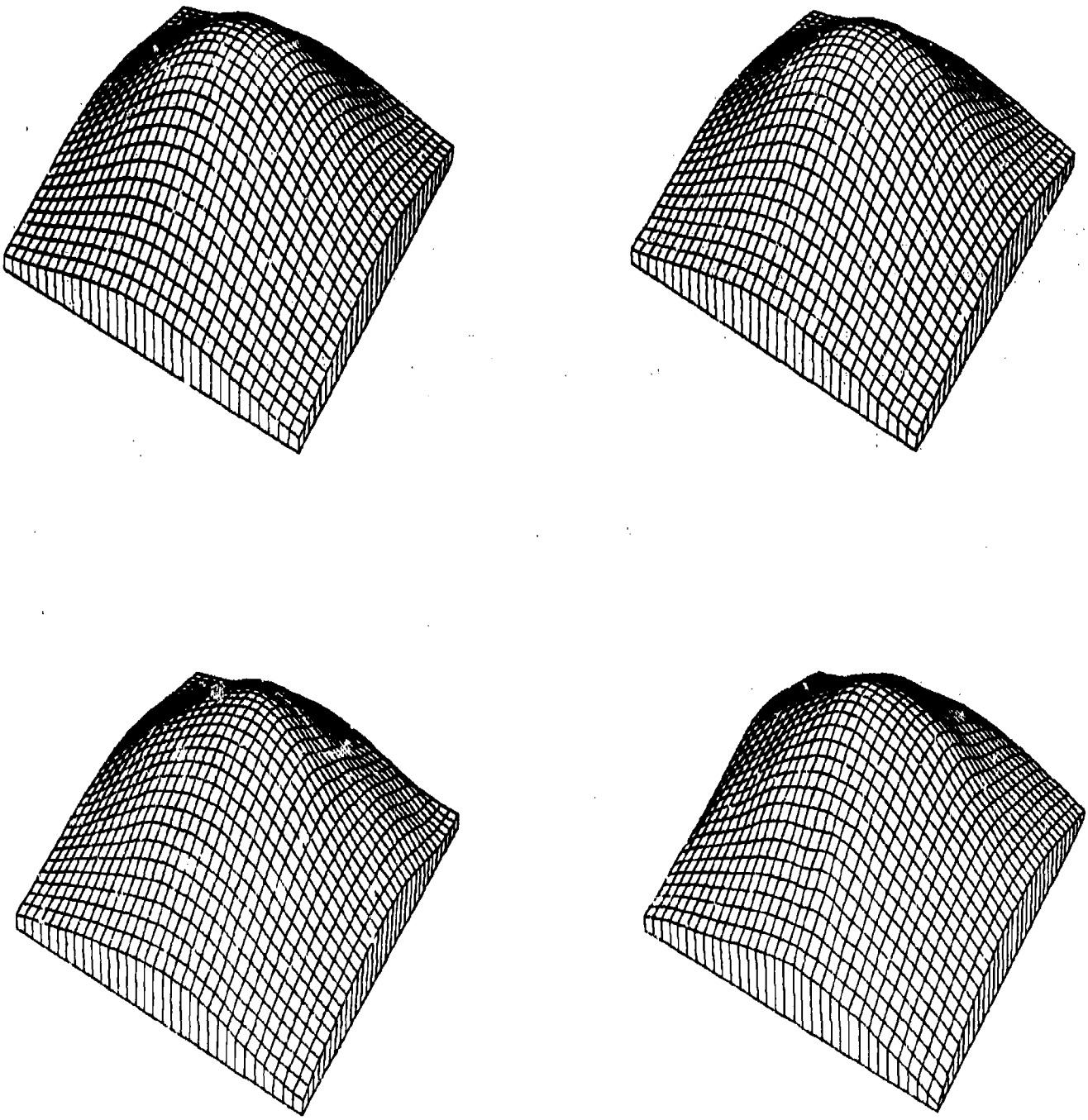
Nielson-Franke Quadratic Triangular Method

Figure 1.4.4.13



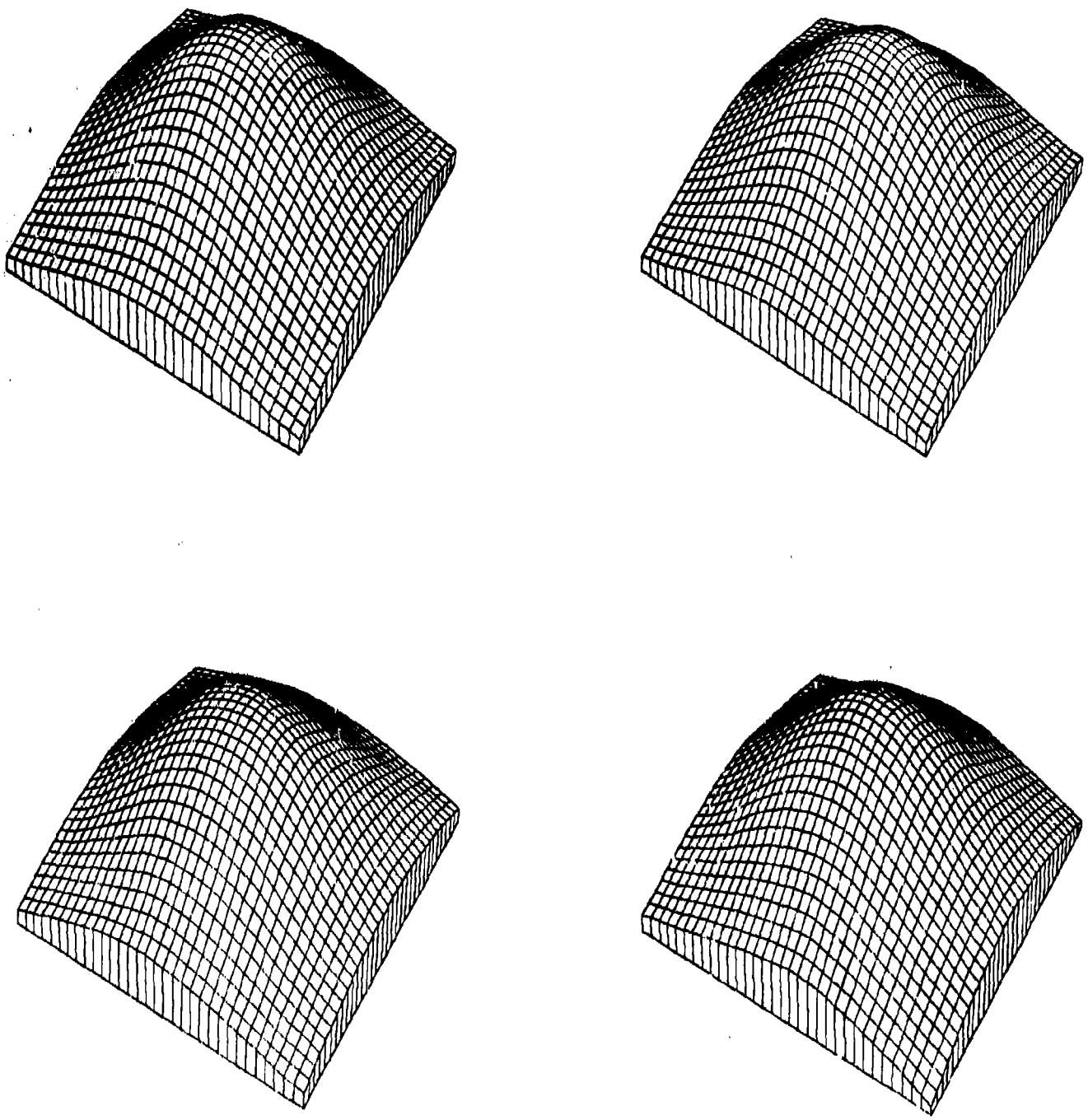
Modified Quadratic Shepard Method

Figure 1.4.4.14



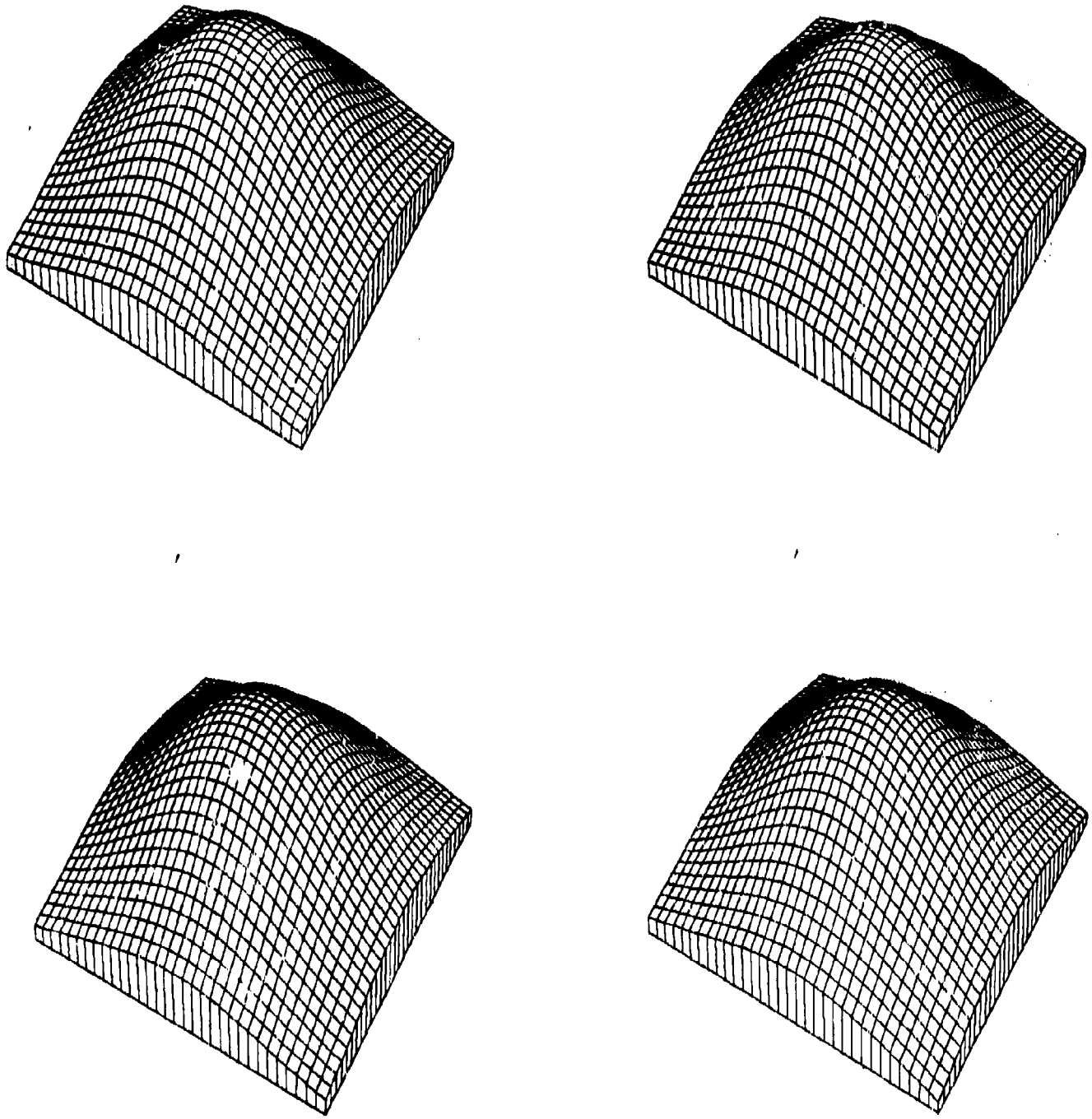
Akima's Method, Modification Three

Figure 1.4.4.16



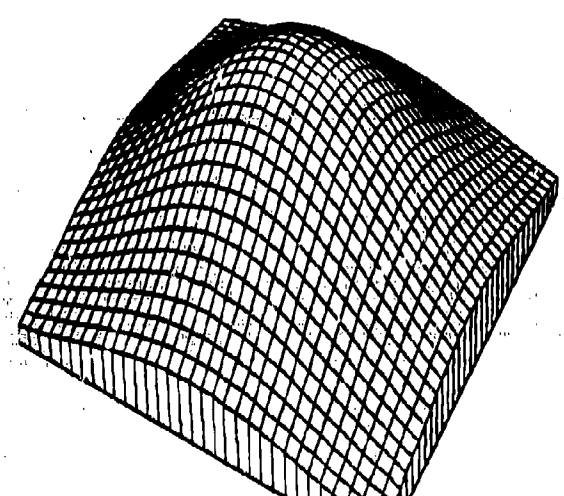
Nielson Minimum Norm Network

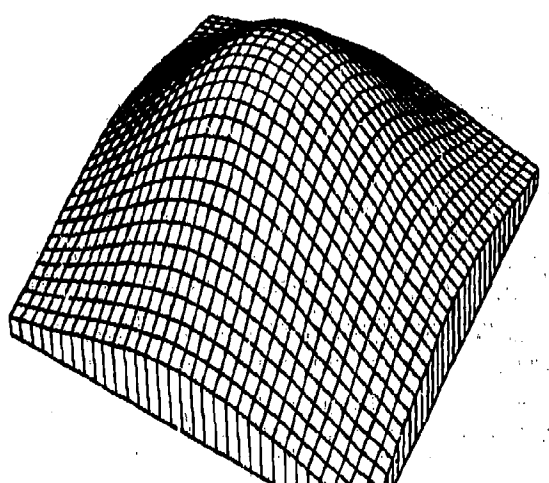
Figure 1.4.4.19

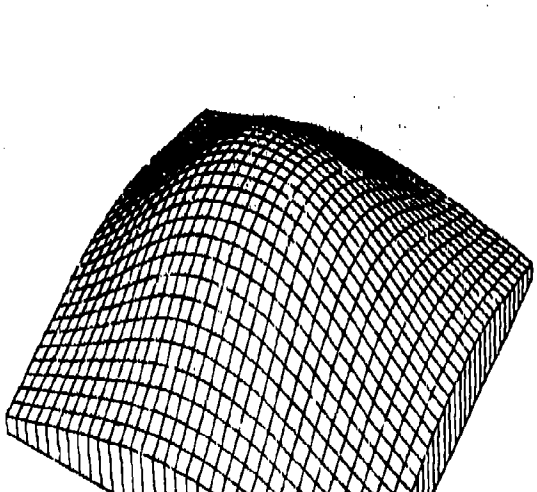


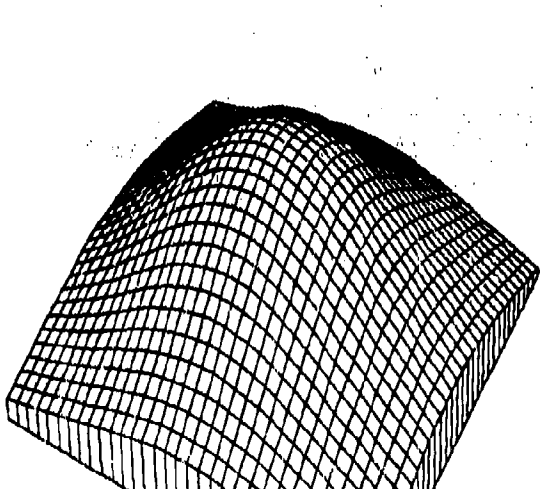
Hardy Multiquadric Method

Figure 1.4.4.21



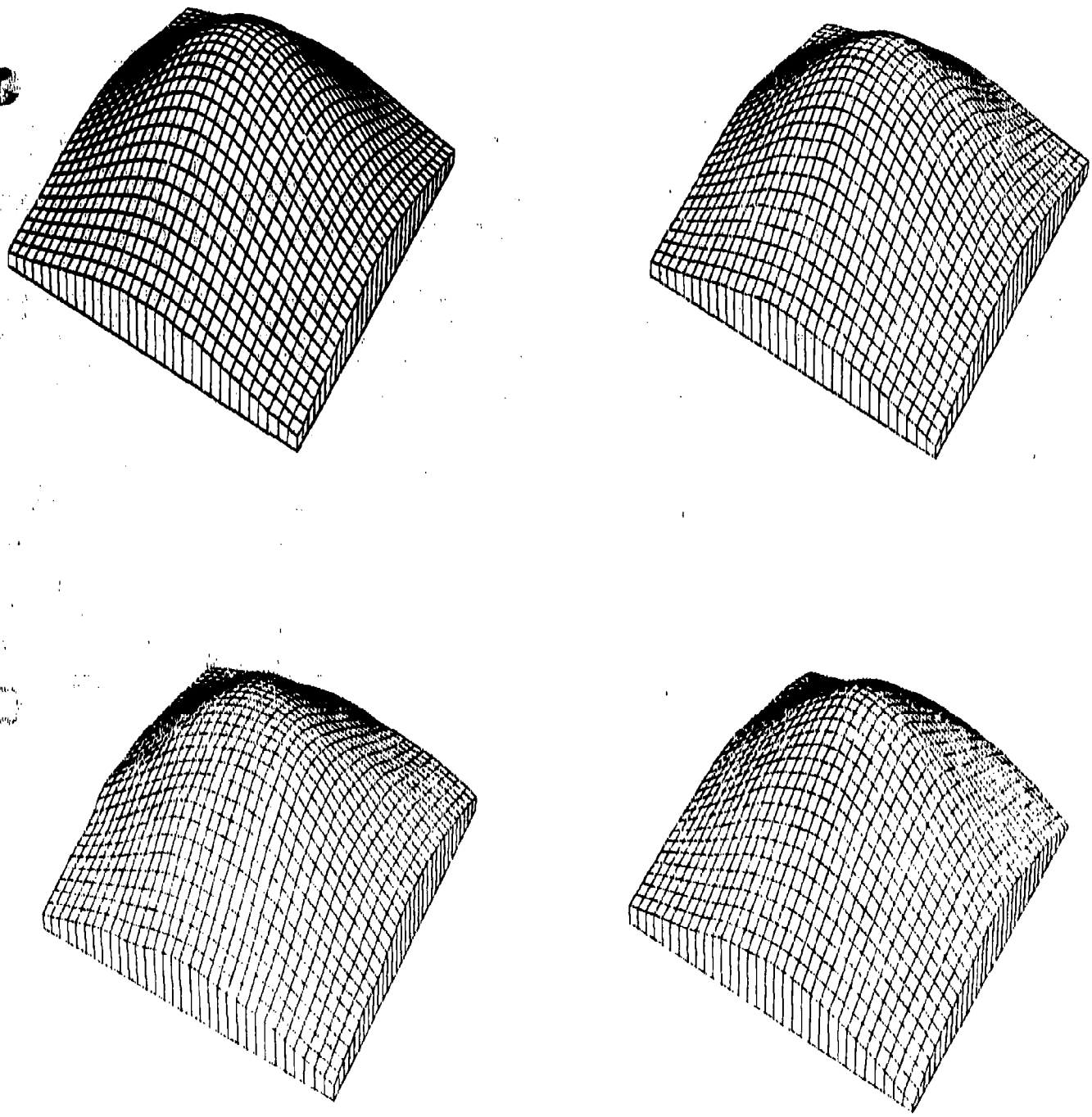






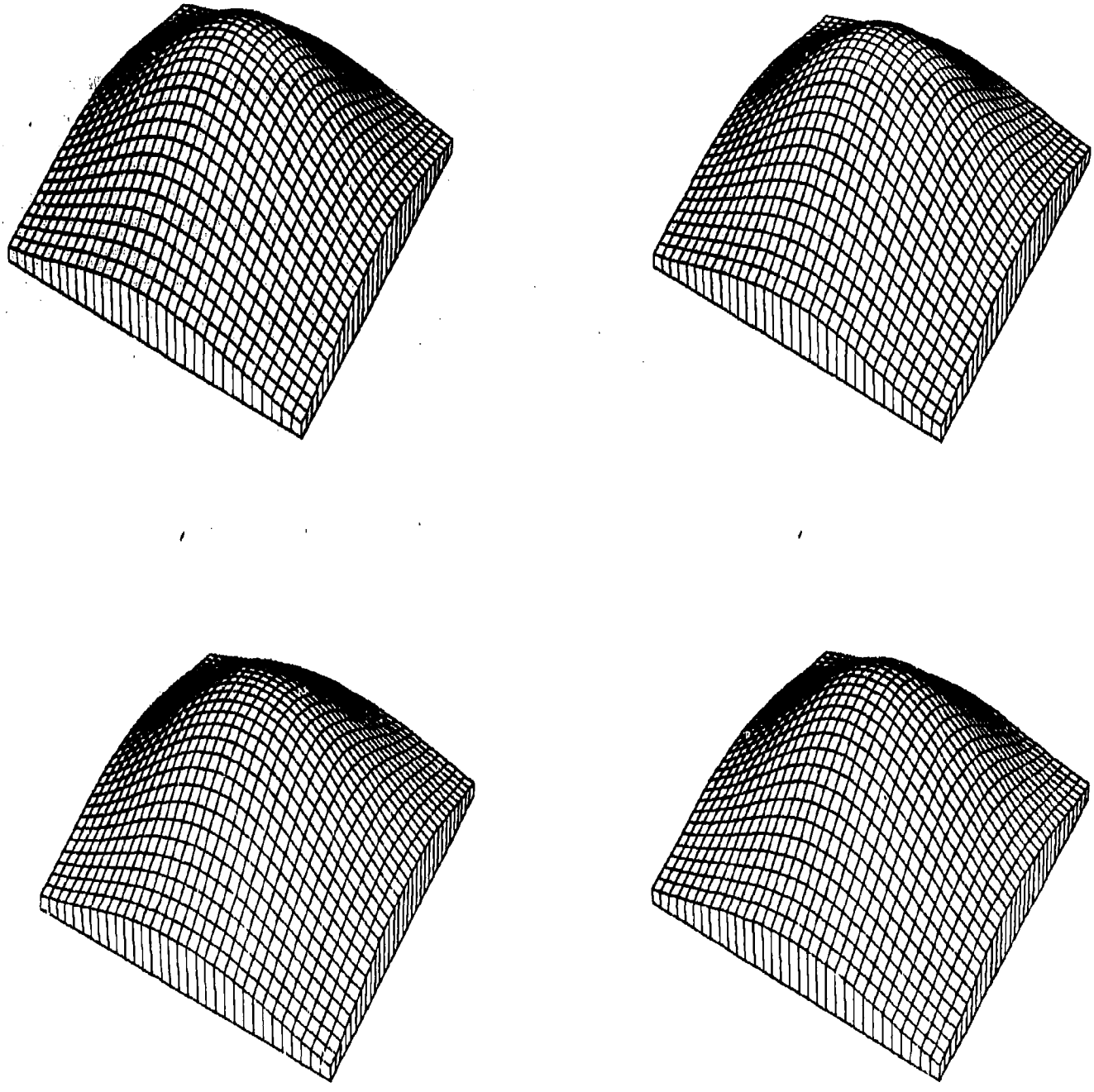
Duchon's Thin Plate Method

Figure 1.4.4.23



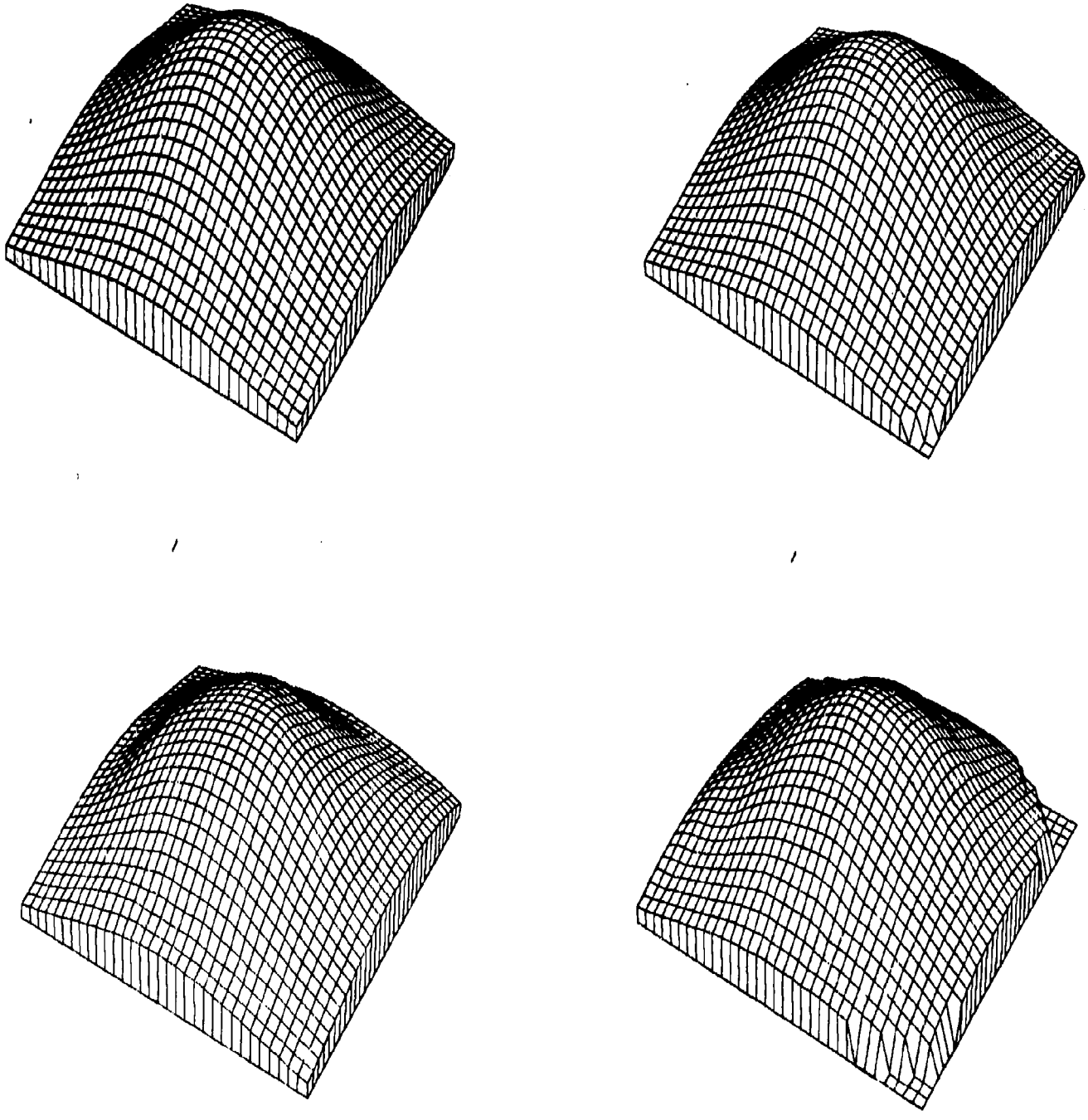
Franke's Method, Thin Plate Local Functions

Figure 1.4.4.24



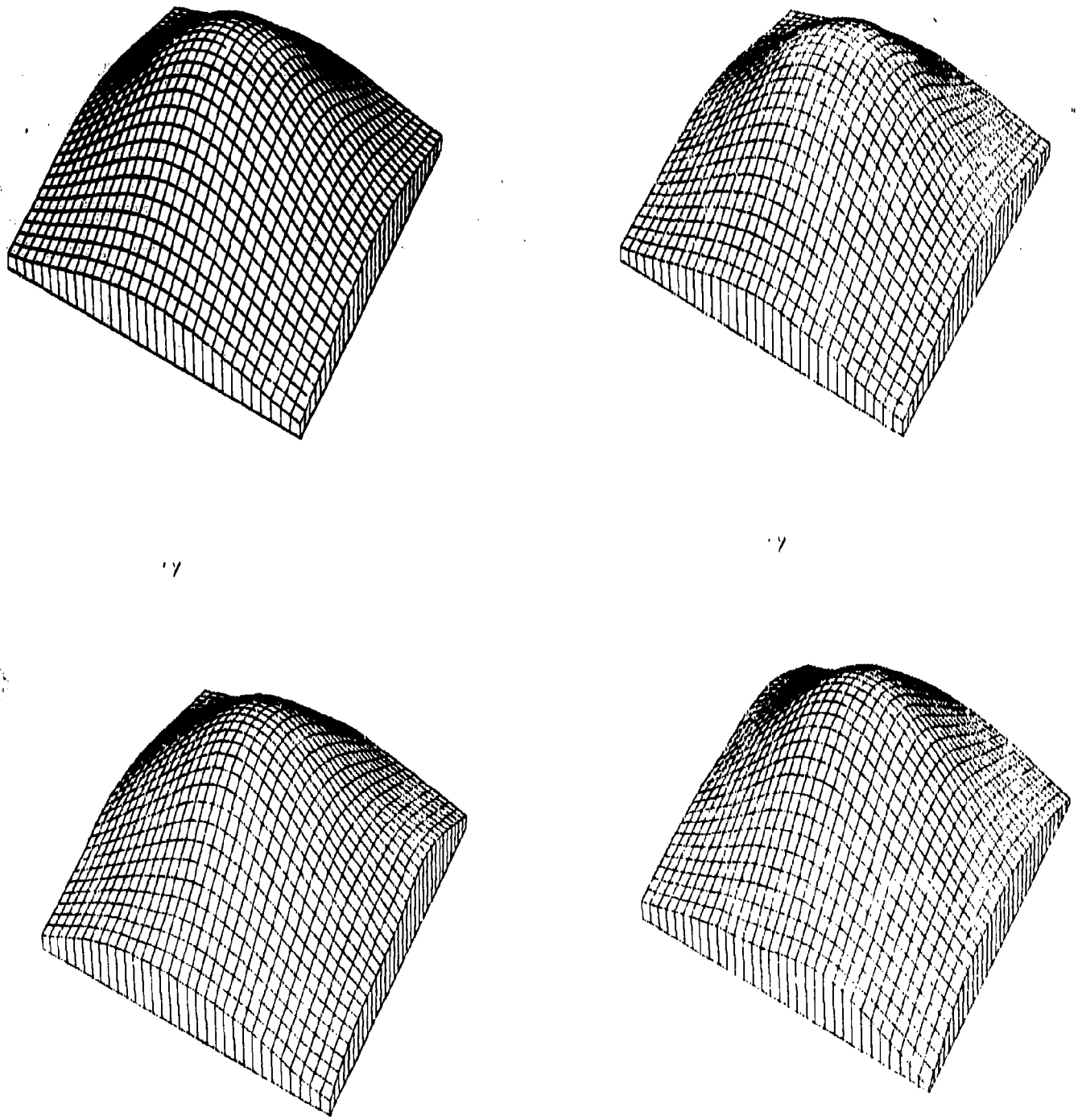
Hardy Reciprocal Multiquadric Method

Figure 1.4.4.27



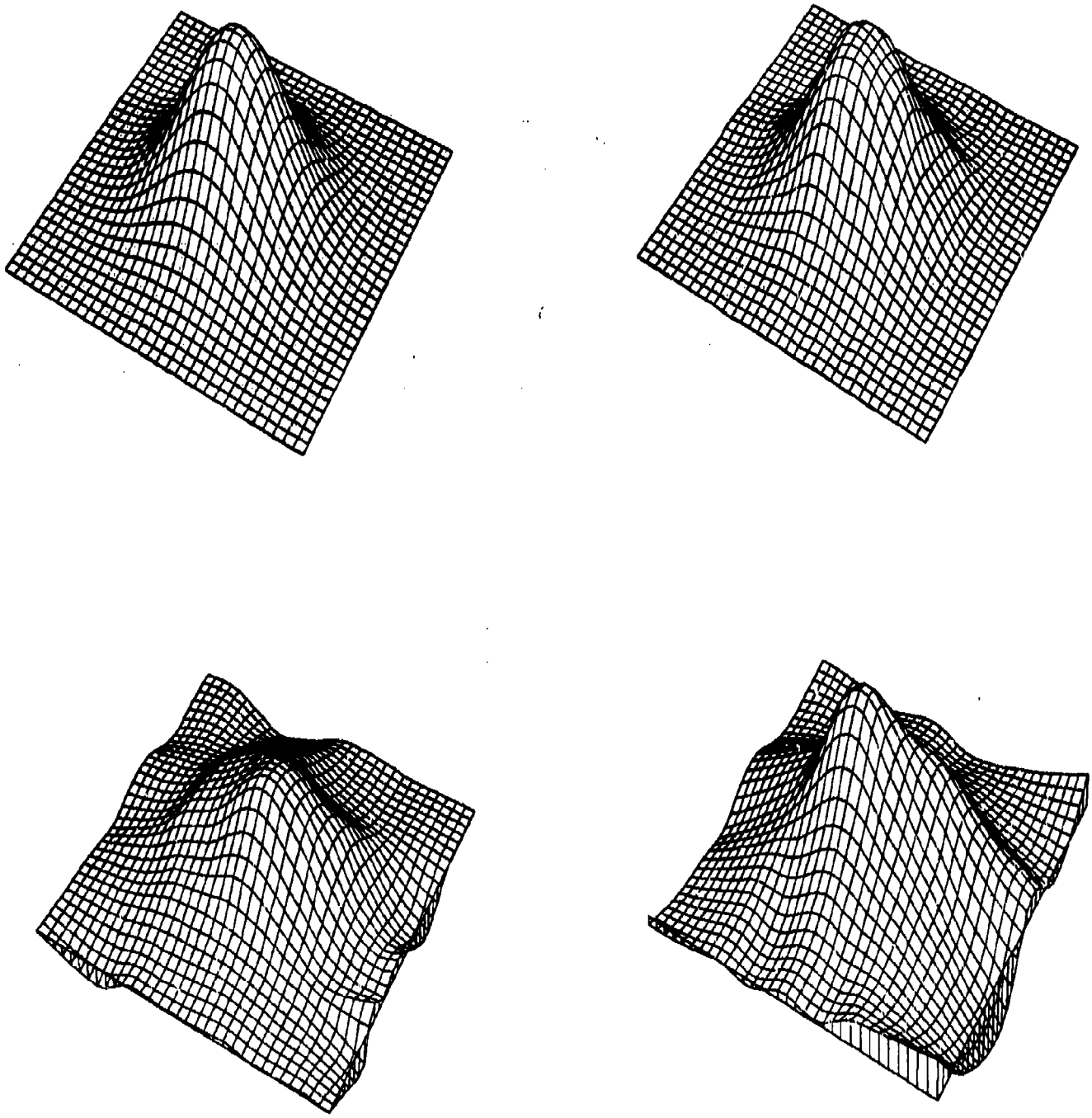
Lawson's Method

Figure 1.4.4.28



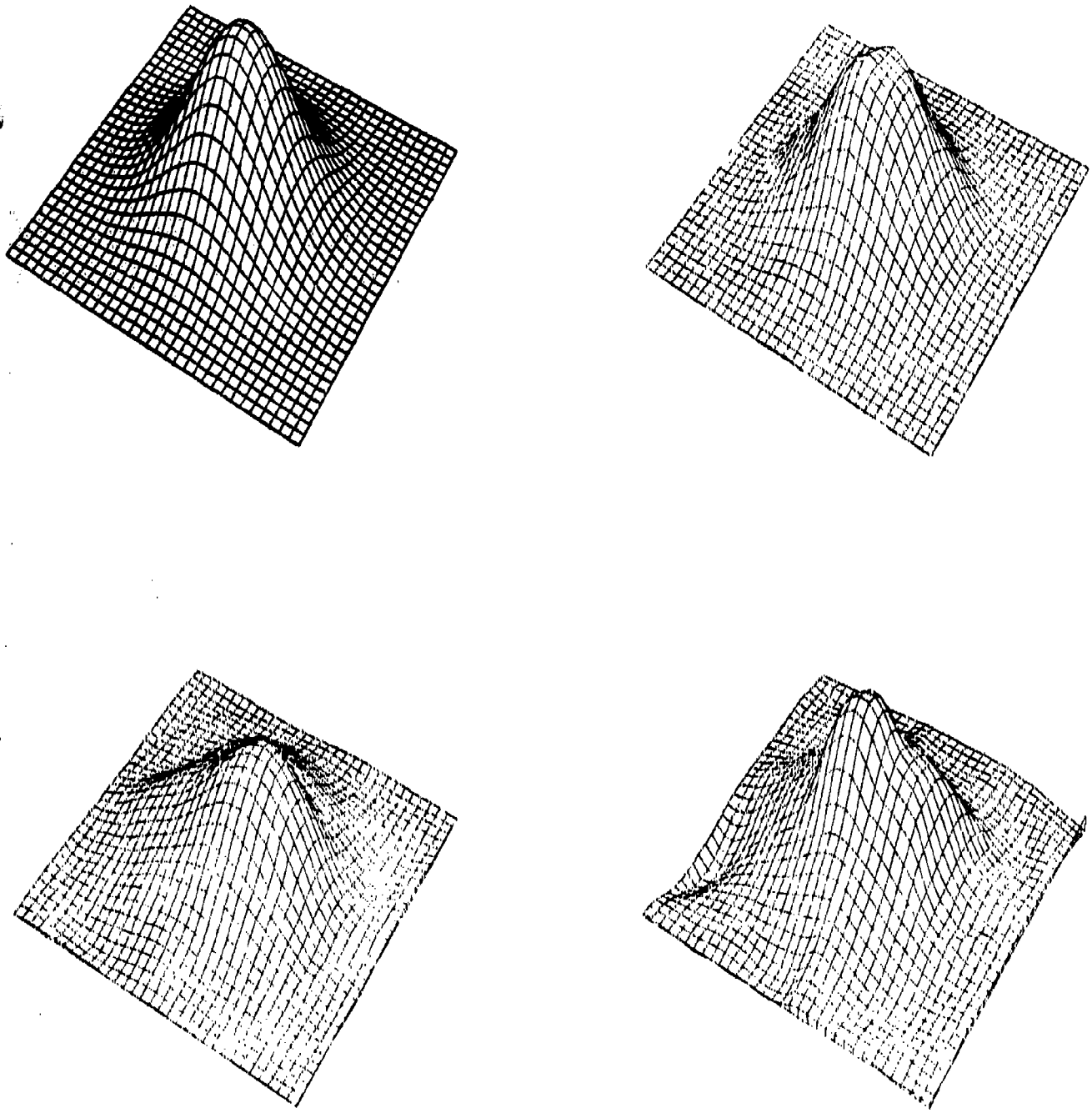
Foley's Iterated Generalized Newton
Delta Sum Bicubic Spline

Figure 1.4.4.30



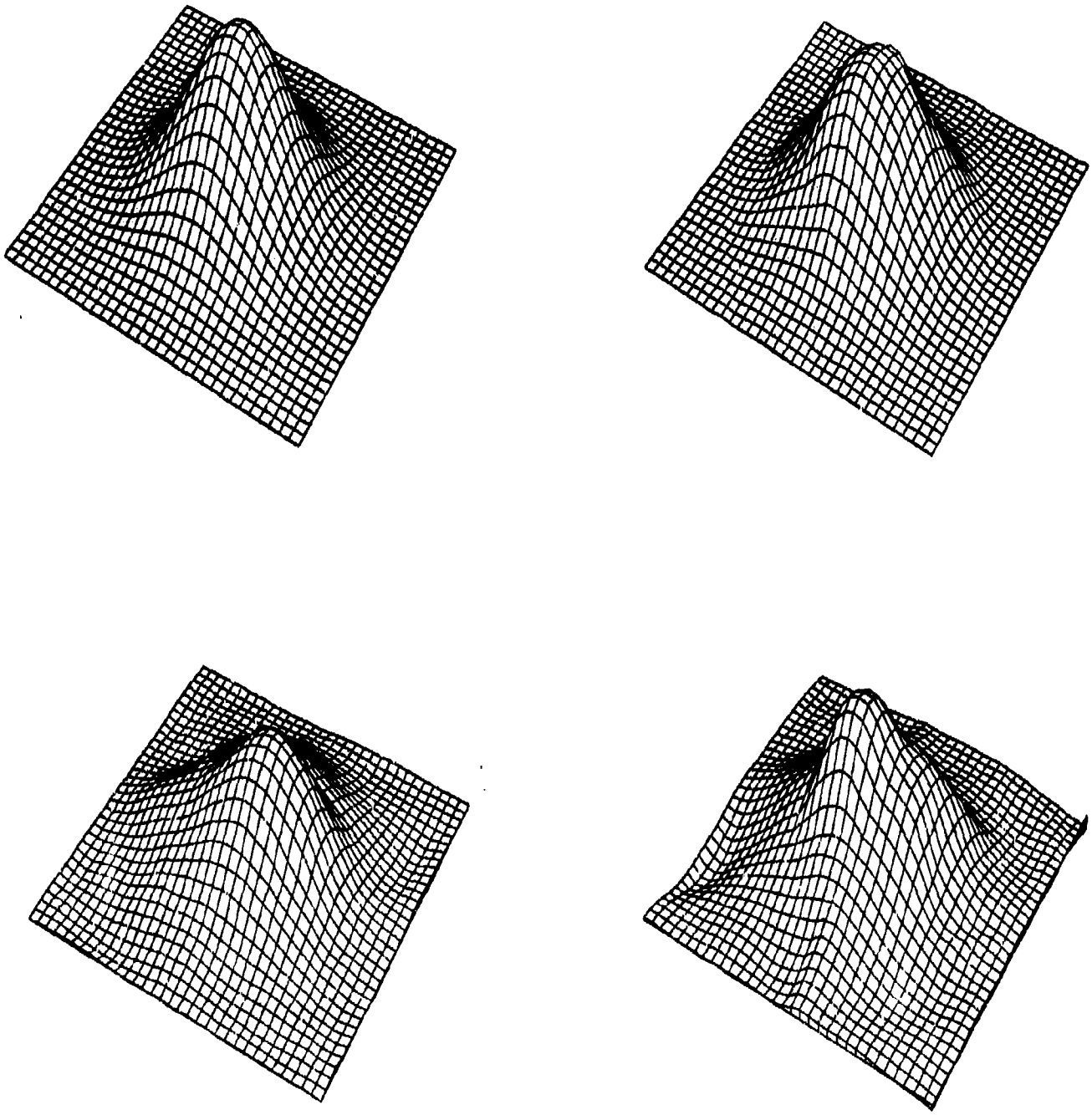
Franke's Method, Mode = 3

Figure 1.4.5.1



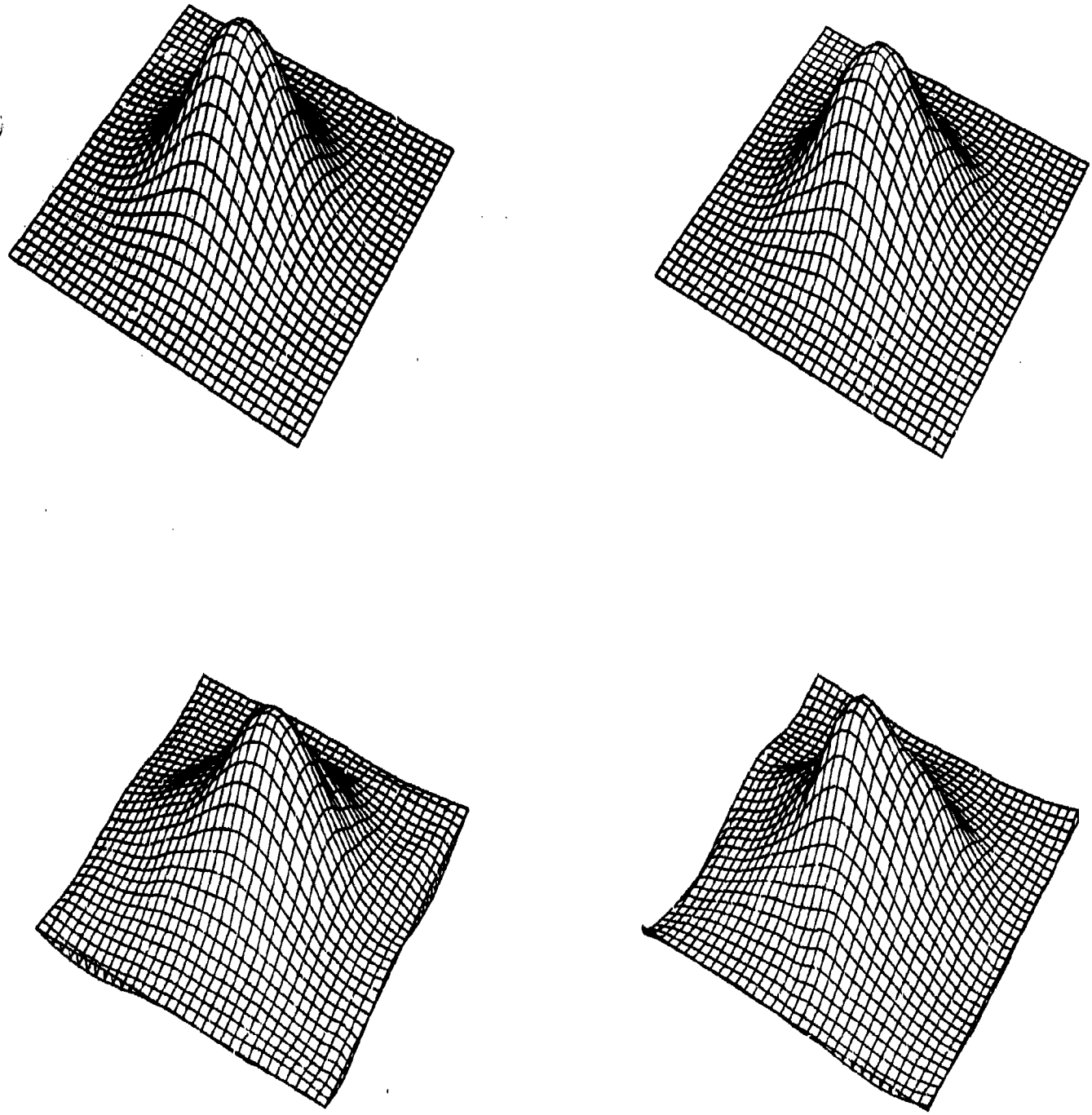
Akima's Method

Figure 1.4.5.4



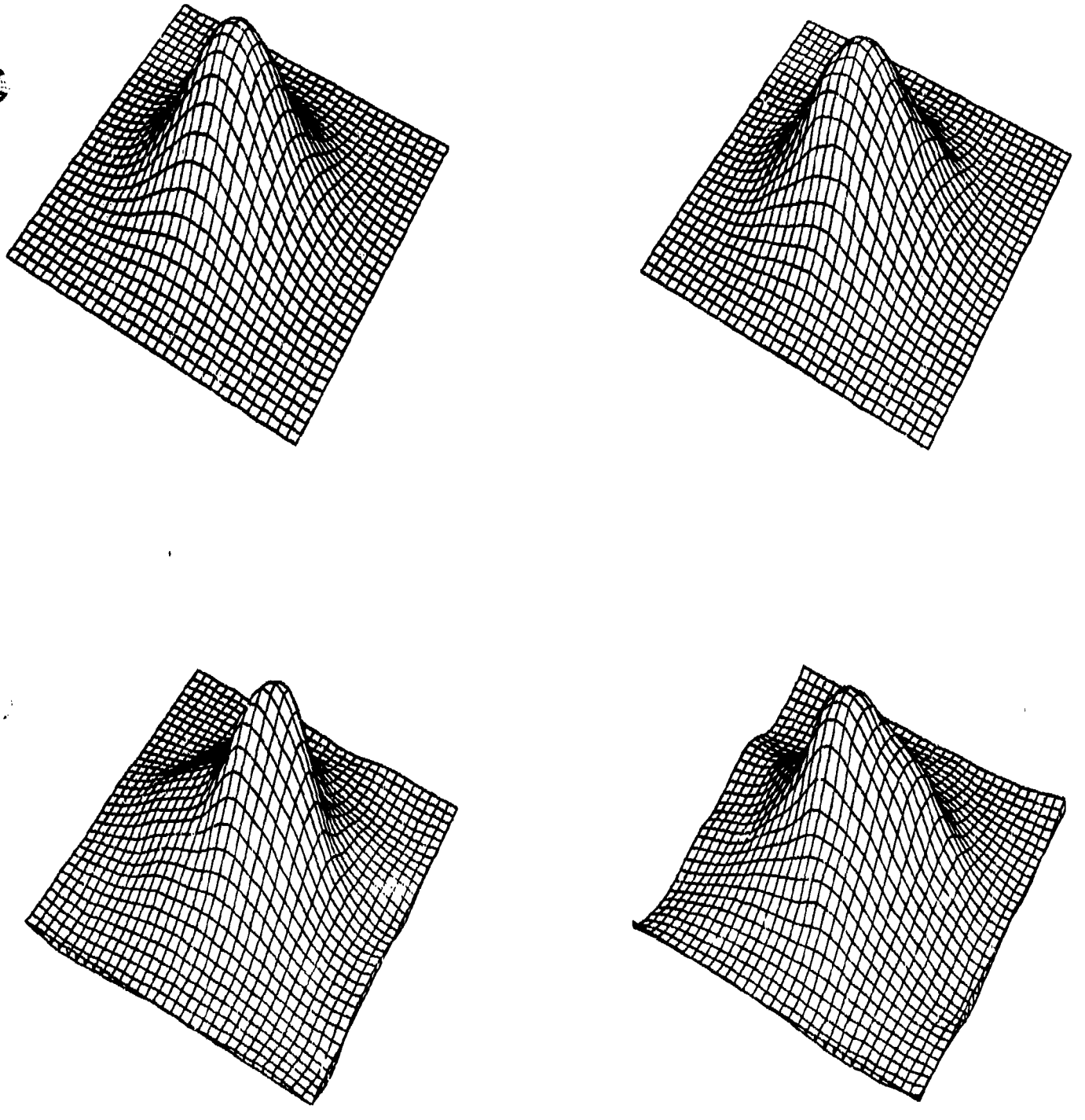
Akima's Method, Modification One

Figure 1.4.5.10



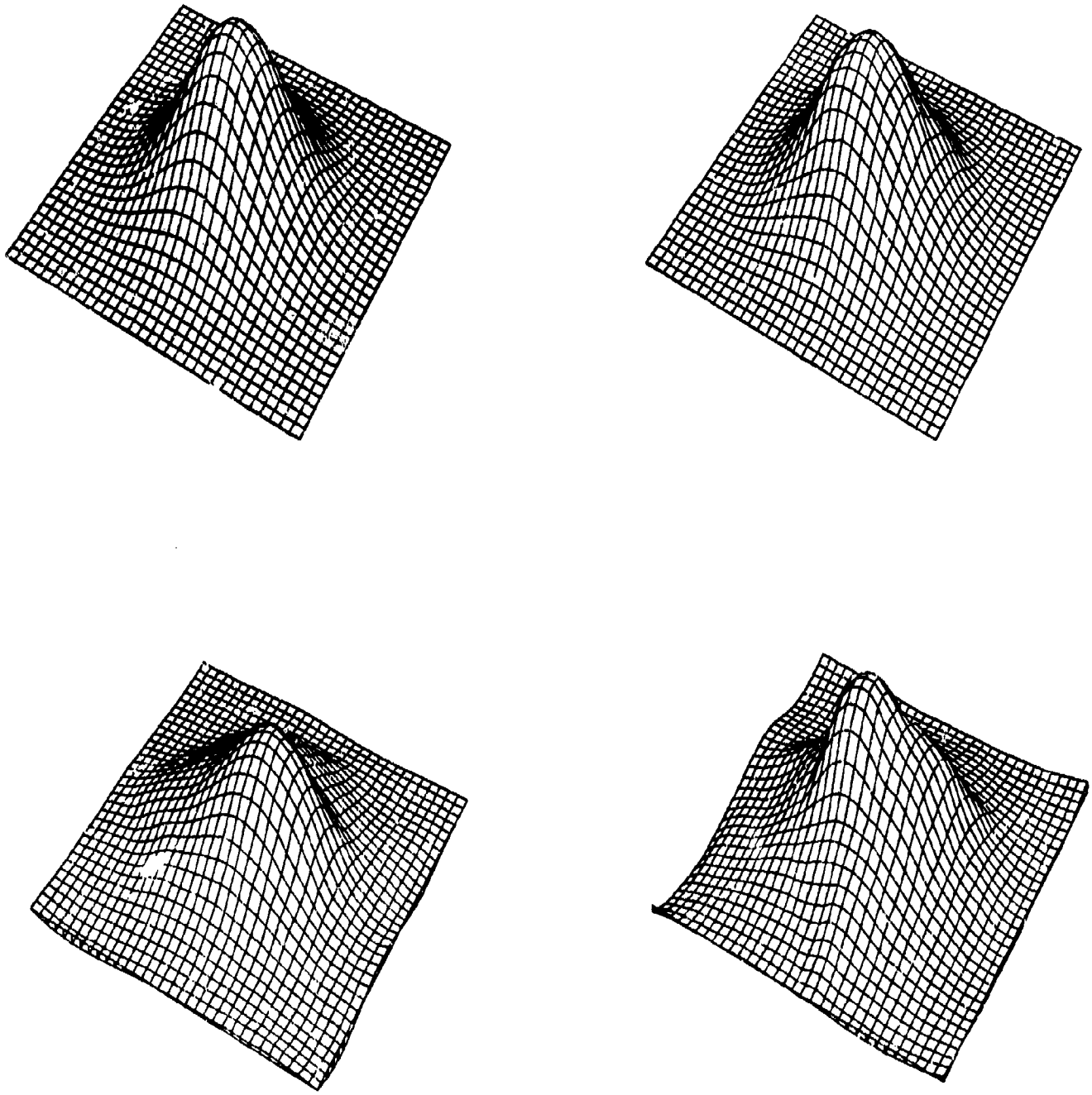
Nielson-Franke Quadratic Triangular Method.

Figure 1.4.5.13



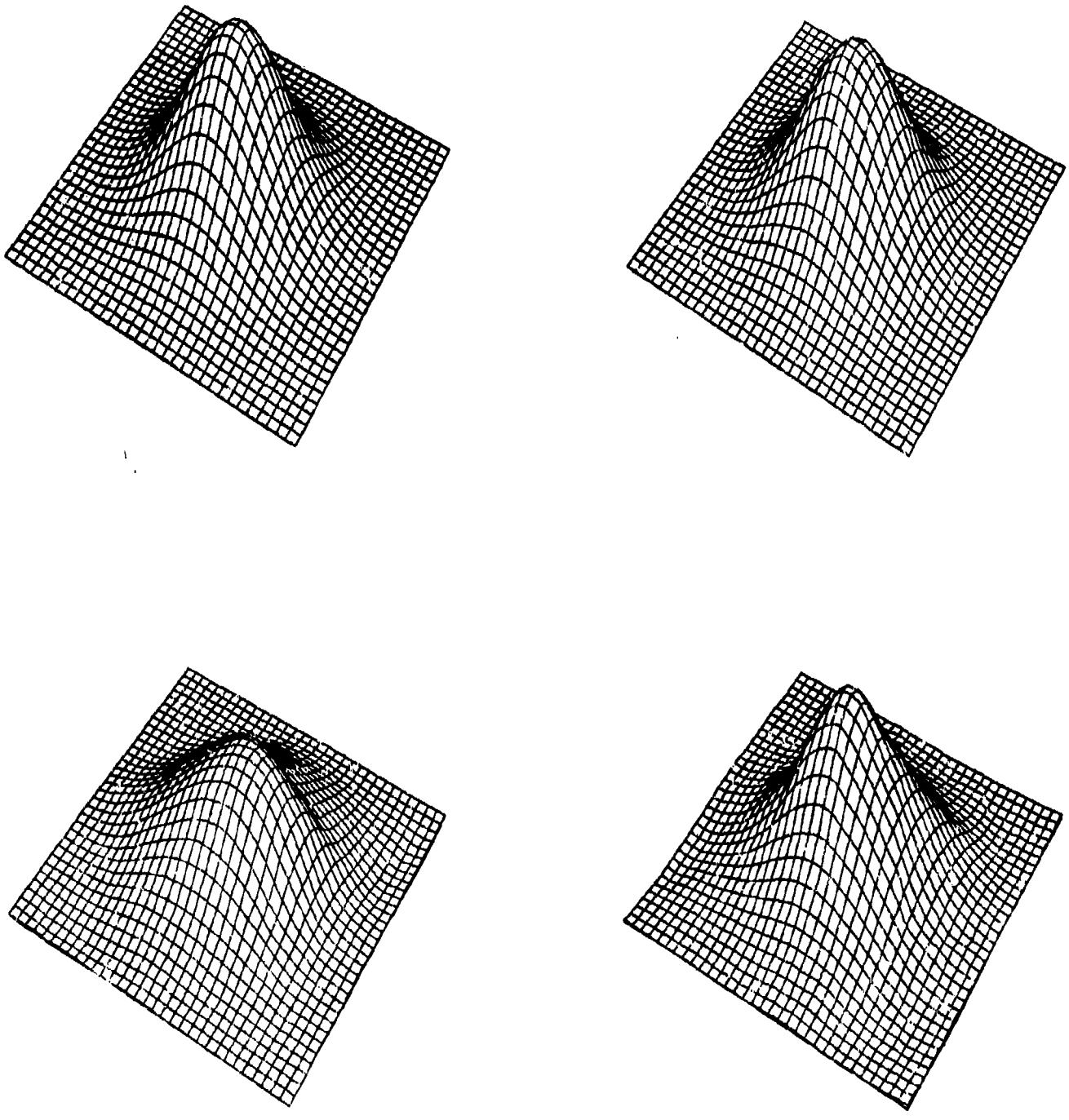
Modified Quadratic Shepard Method

Figure 1.4.5.14



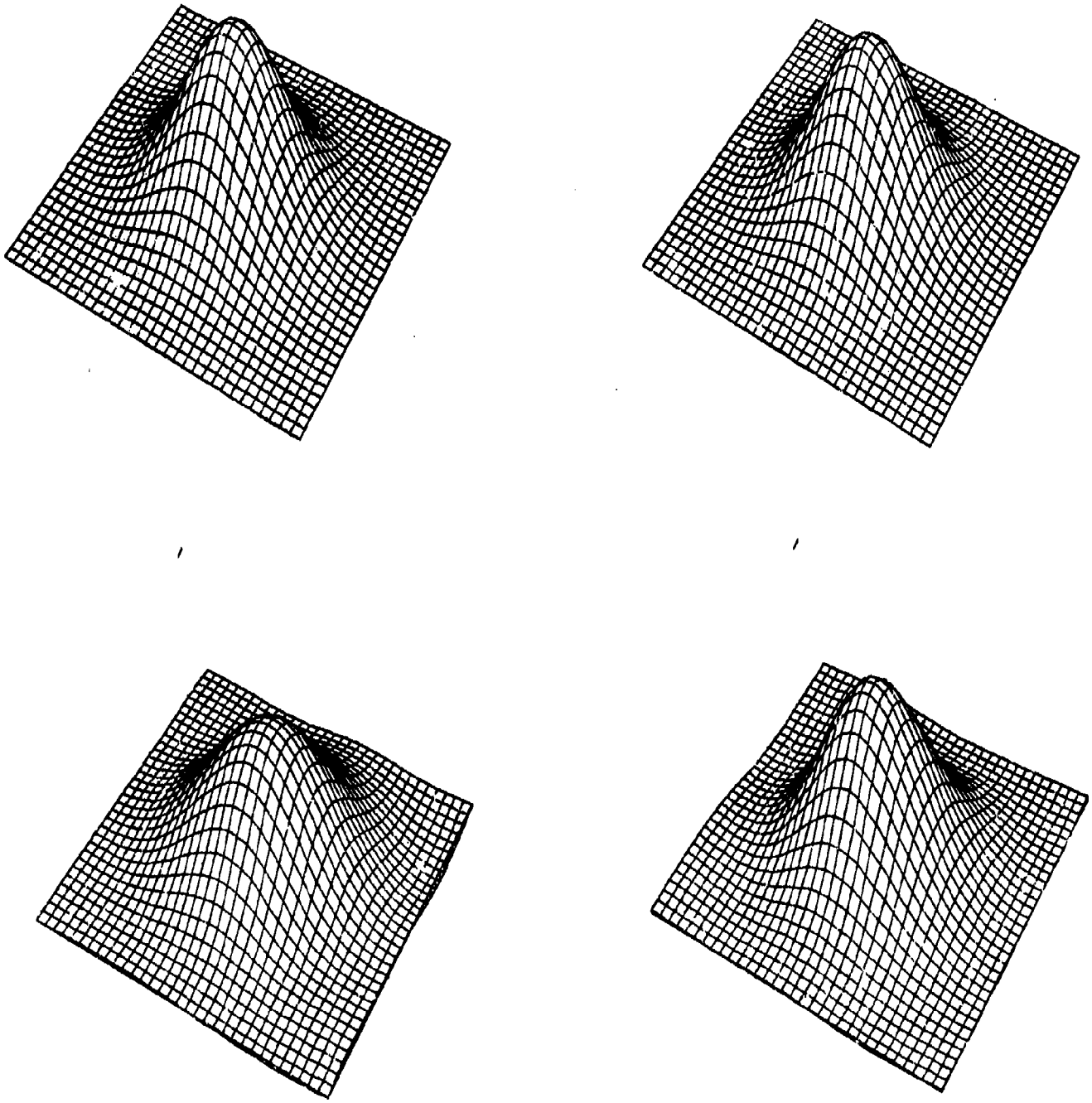
Akima's Method, Modification Three

Figure 1.4.5.16



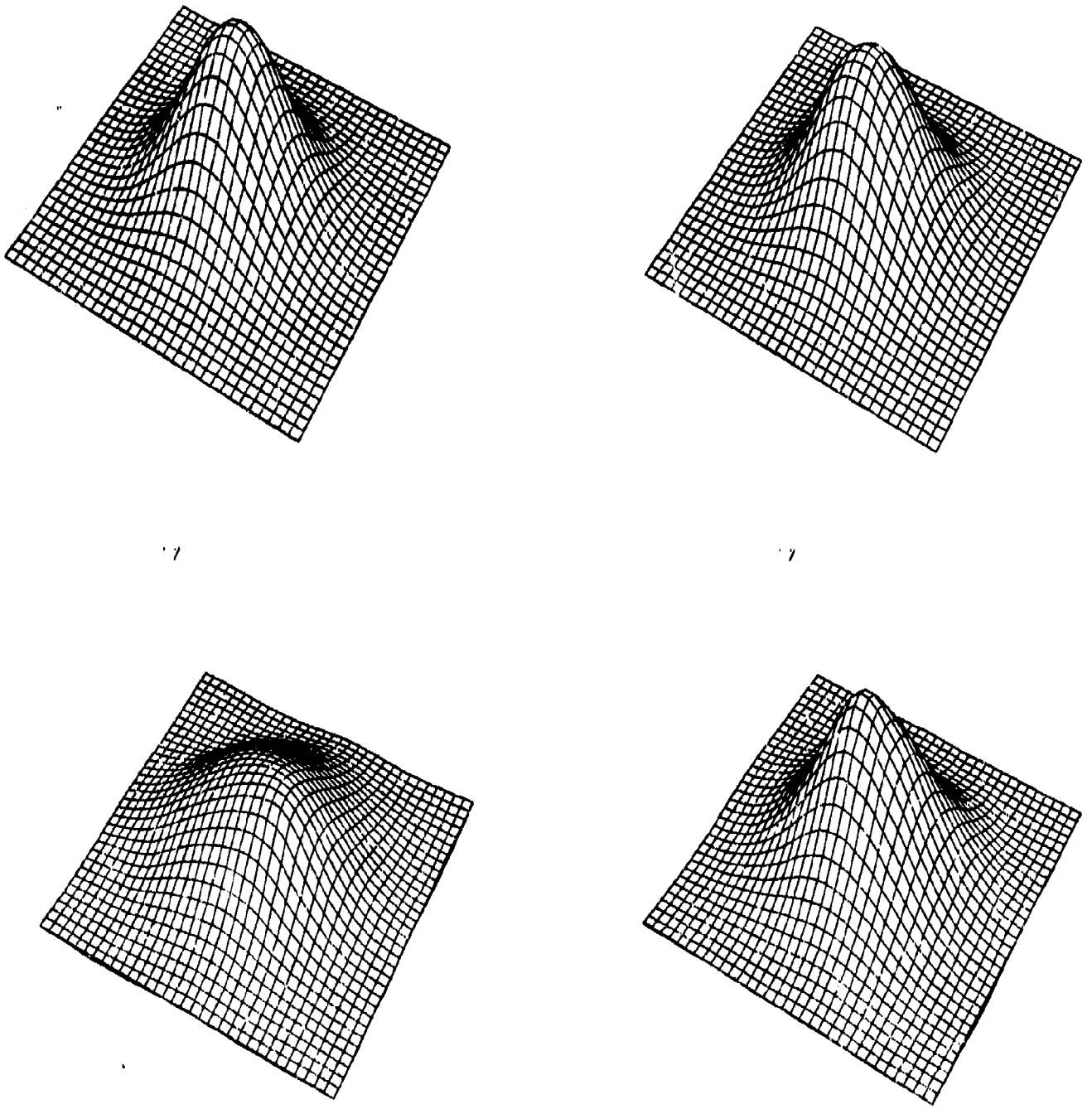
Nielson Minimum Norm Network

Figure 1.4.5.19



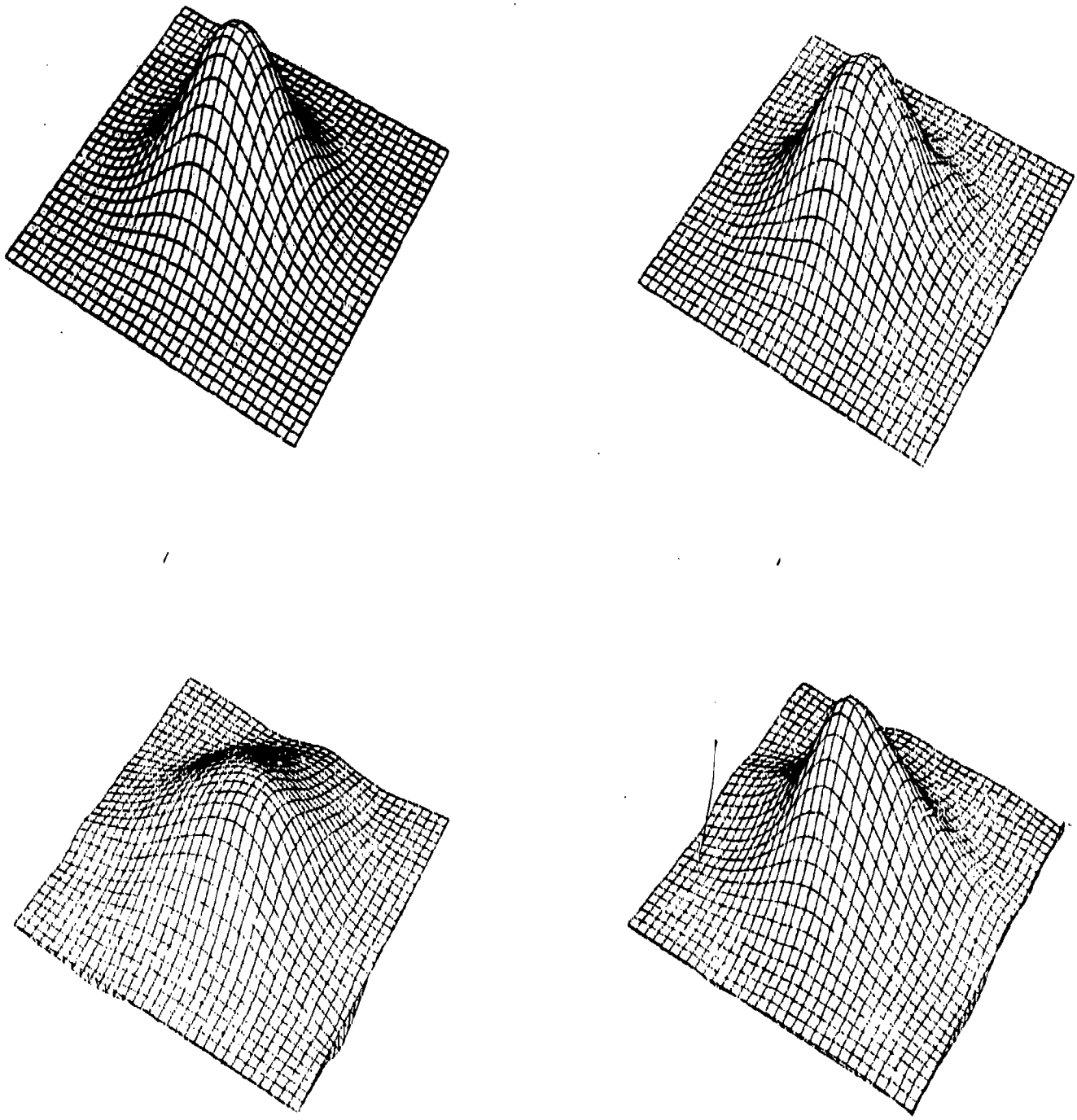
Hardy Multiquadric Method

Figure 1.4.5.21



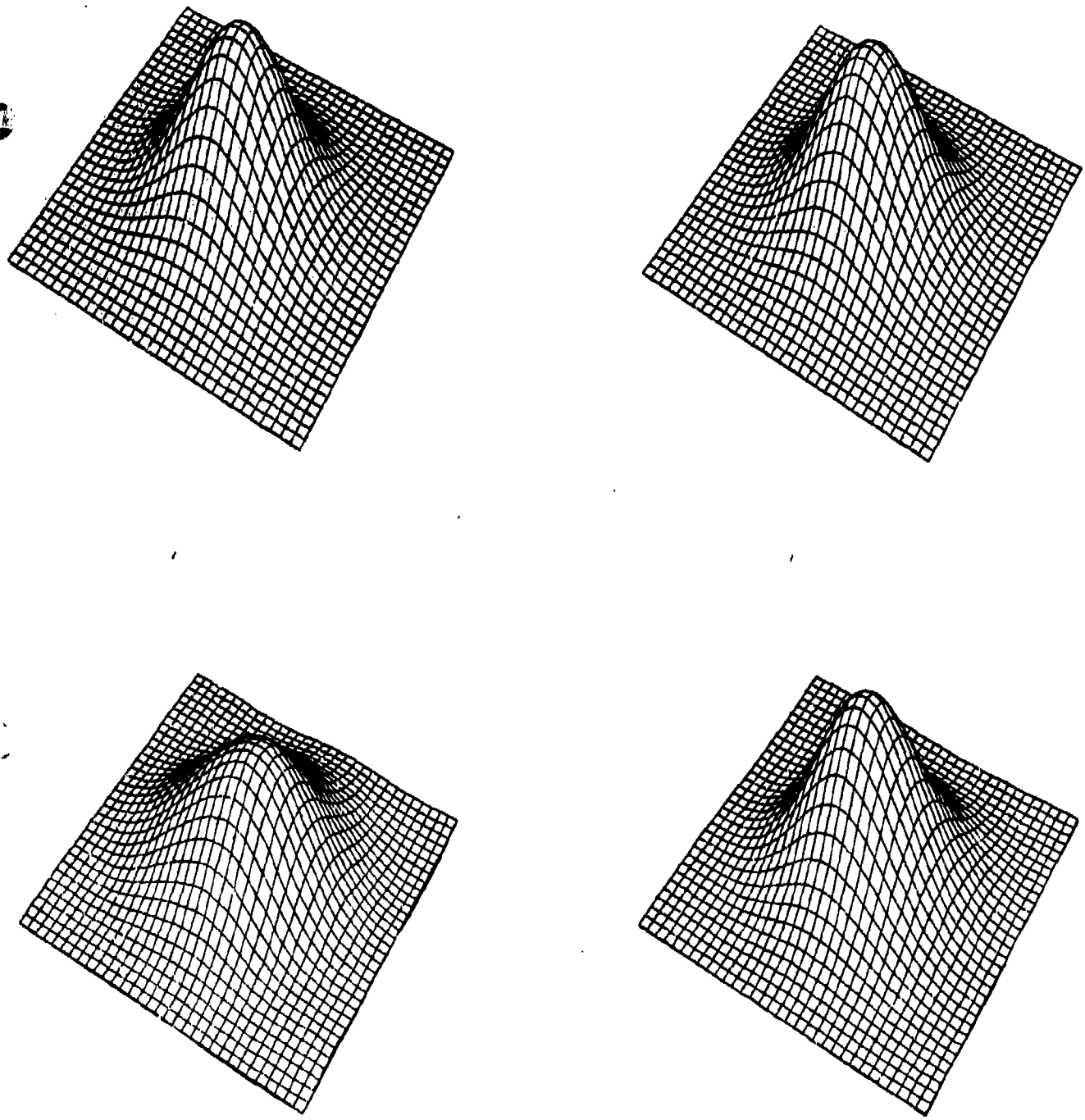
Duchon's Thin Plate Method

Figure 1.4.5.23



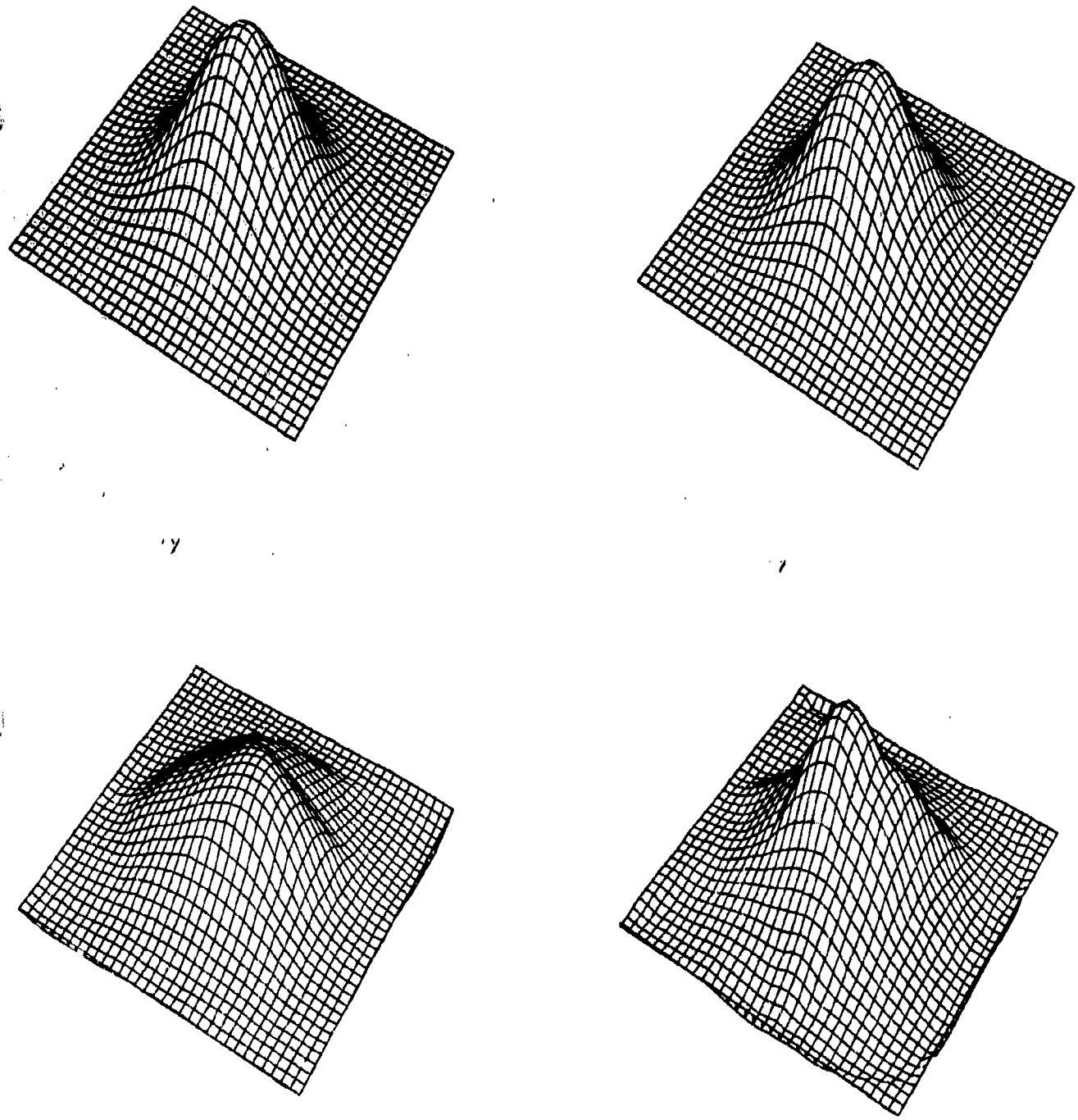
Franke's Method, Thin Plate Local Functions

Figure 1.4.5.24



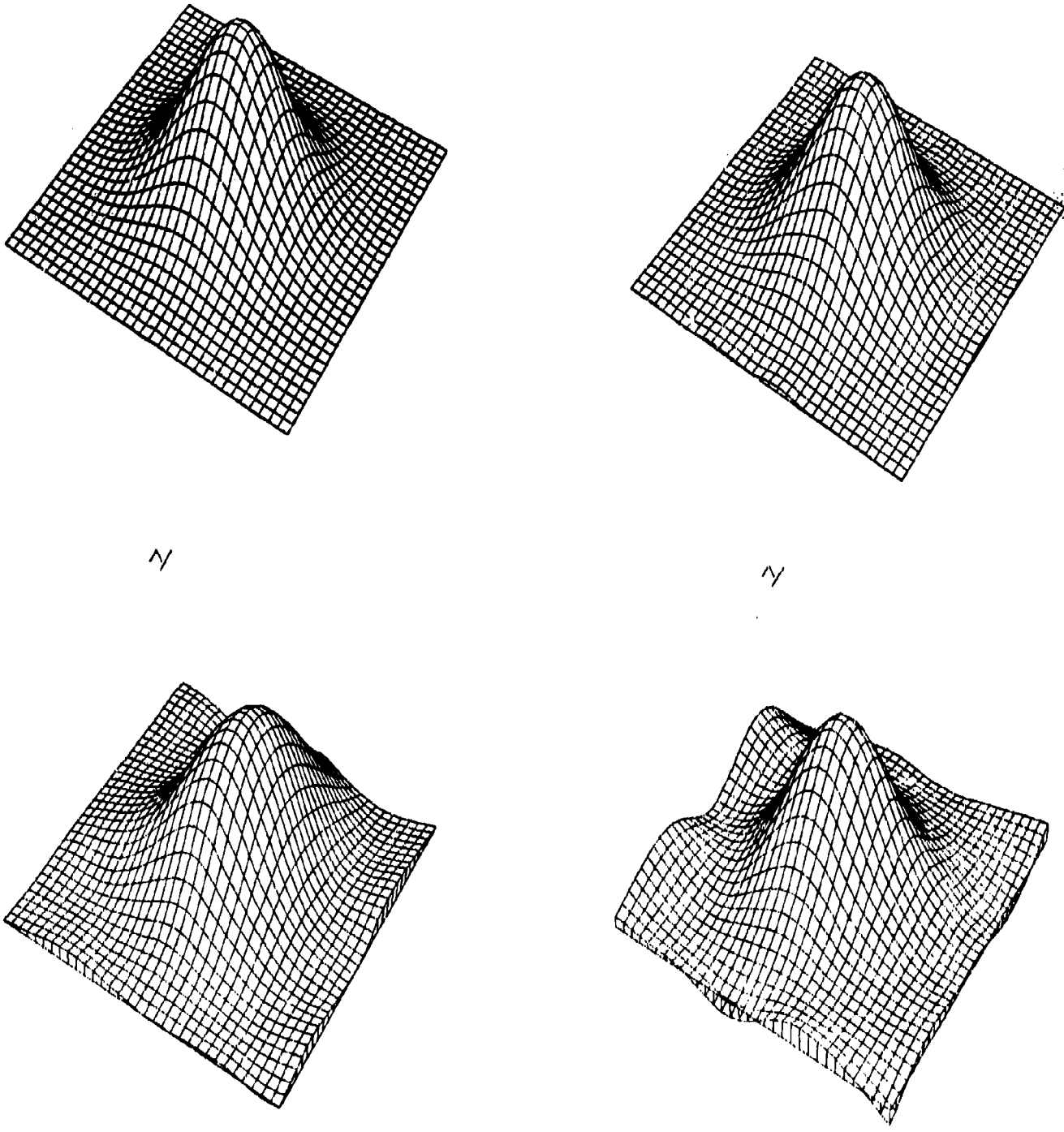
Hardy Reciprocal Multiquadric Method

Figure 1.4.5.27



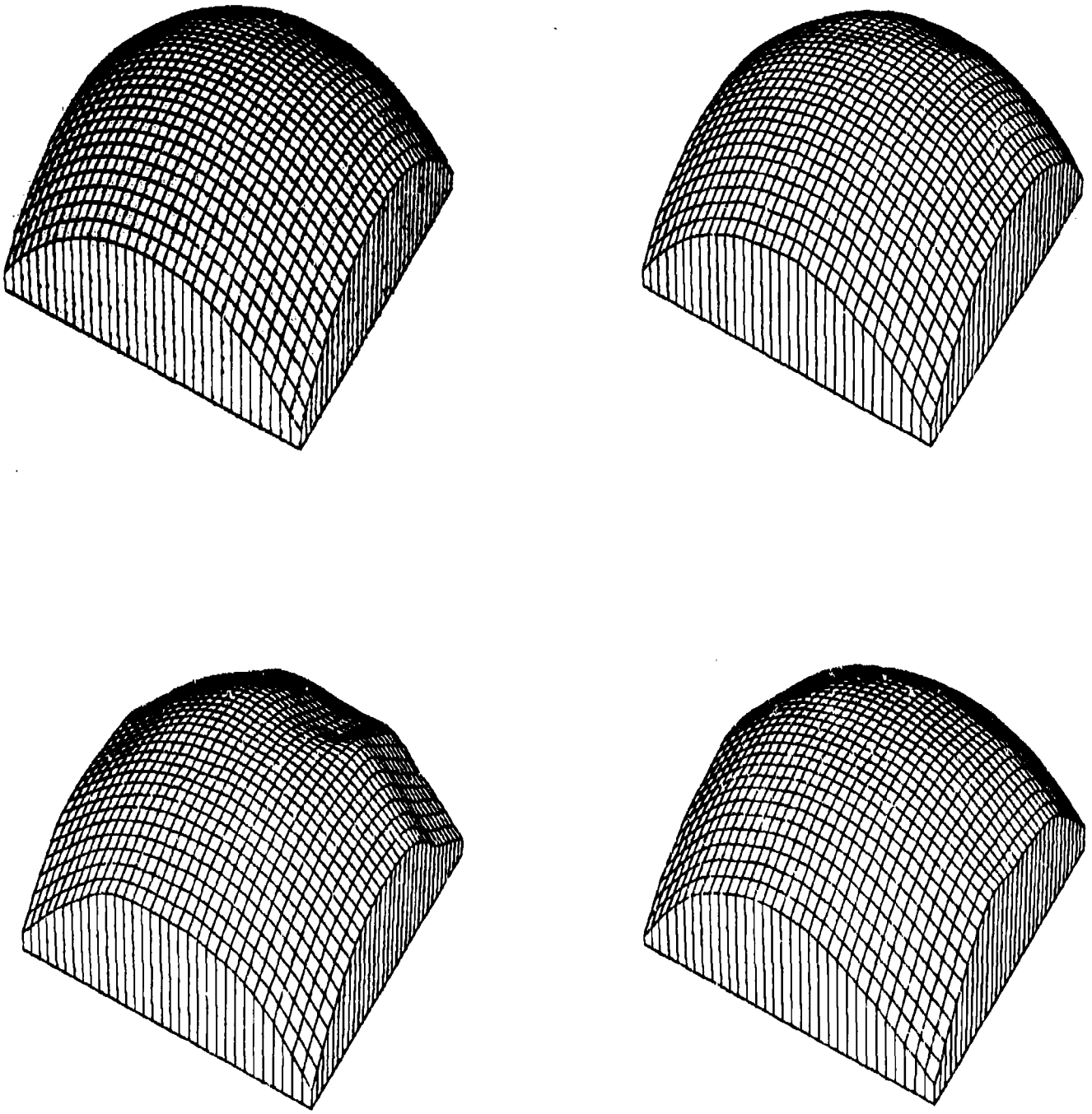
Lawson's Method

Figure 1.4.5.28



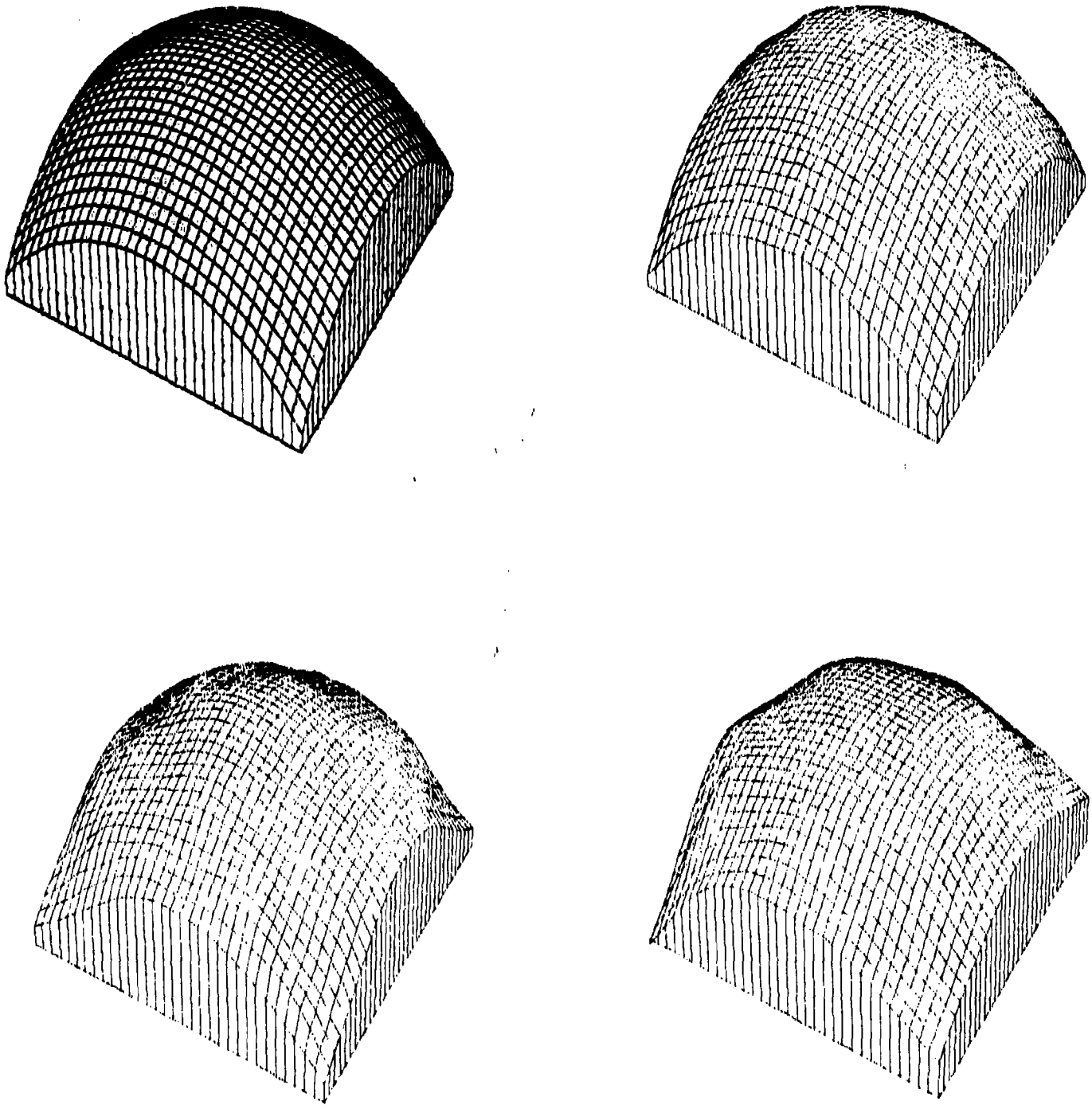
Foley's Iterated Generalized Newton
Delta Sum Bicubic Spline

Figure 1.4.5.30



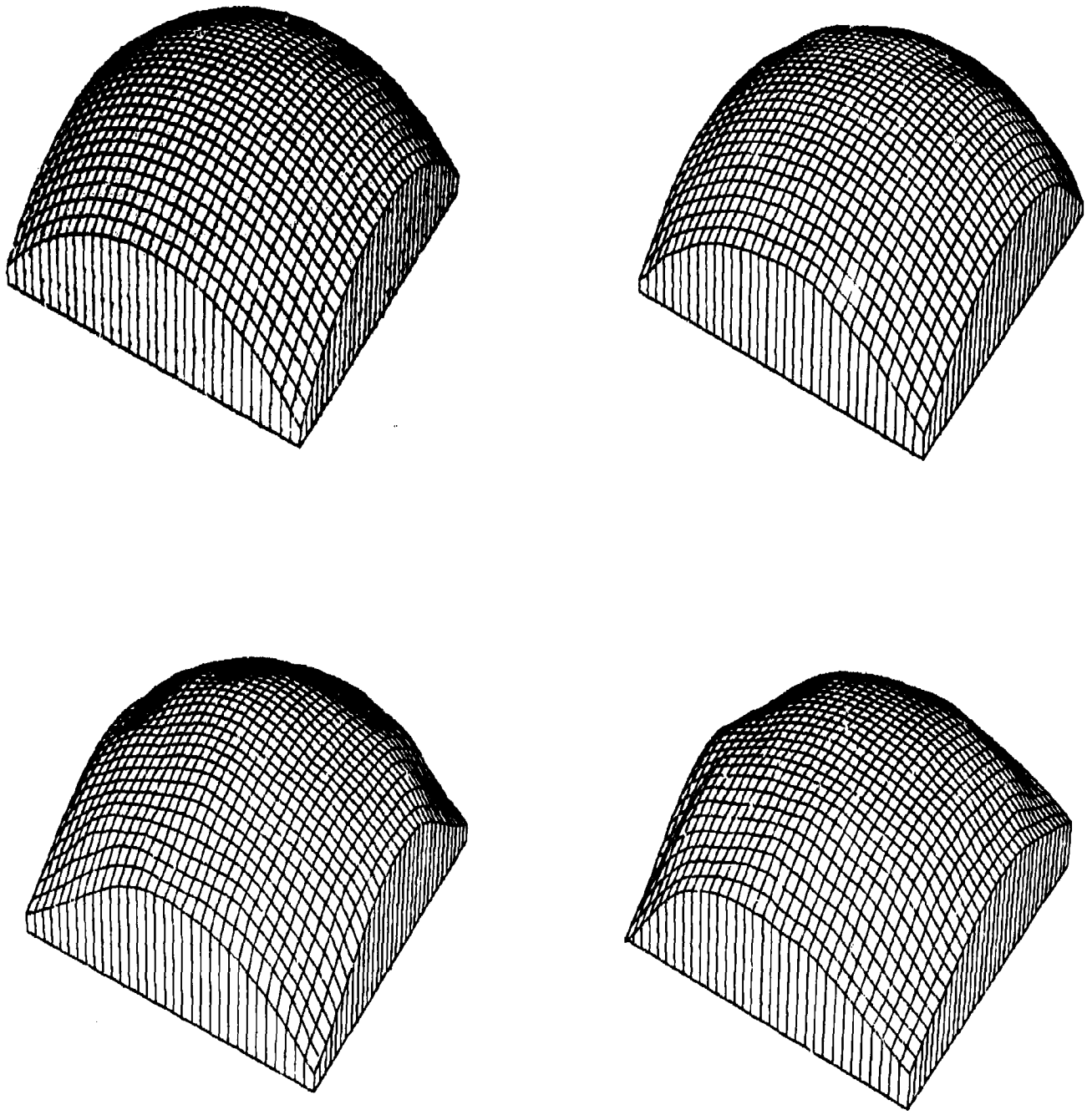
Franke's Method, Mode = 3

Figure 1.4.6.1



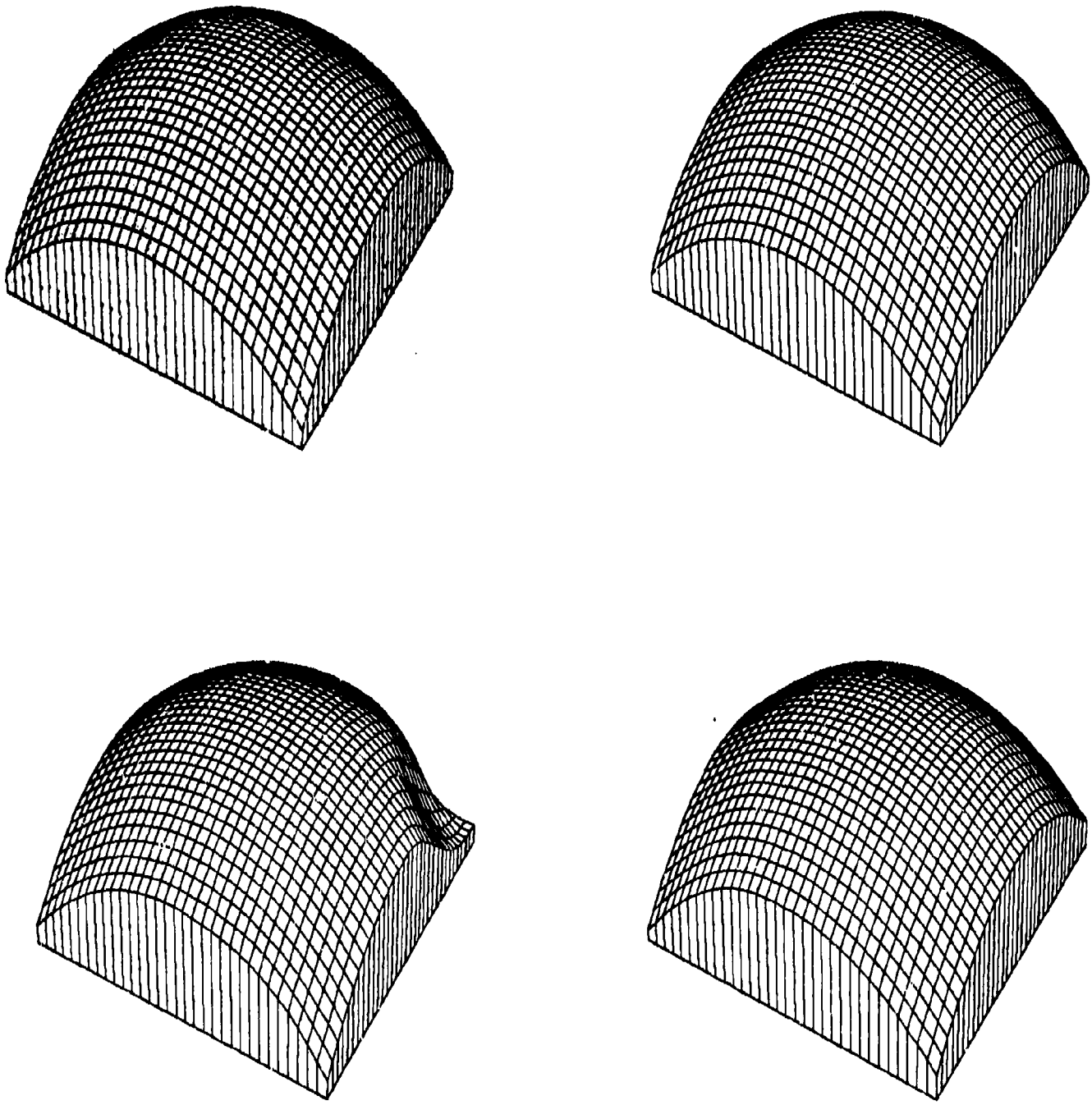
Akima's Method

Figure 1.4.6.4



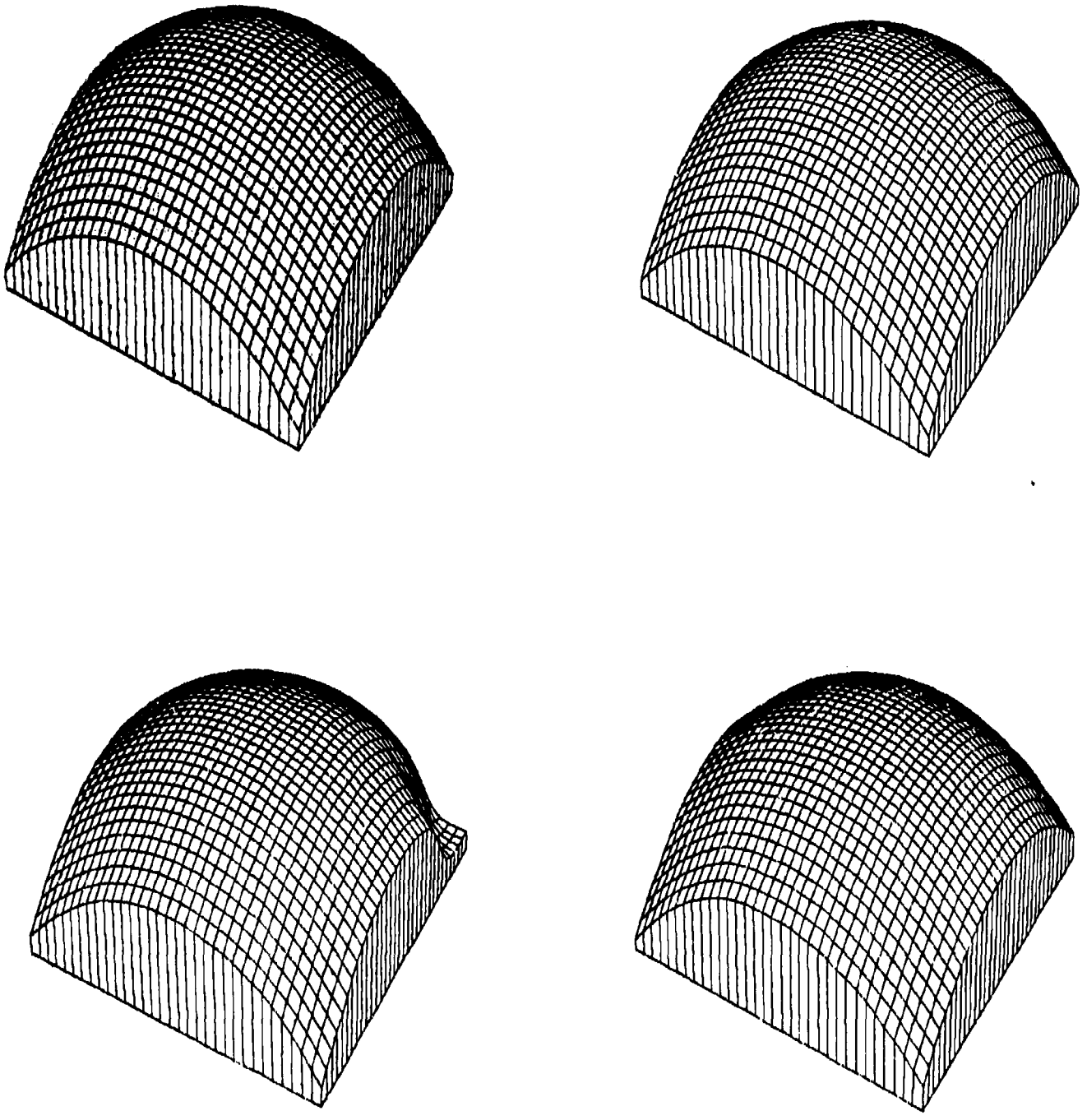
Akima's Method, Modification One

Figure 1.4.6.10



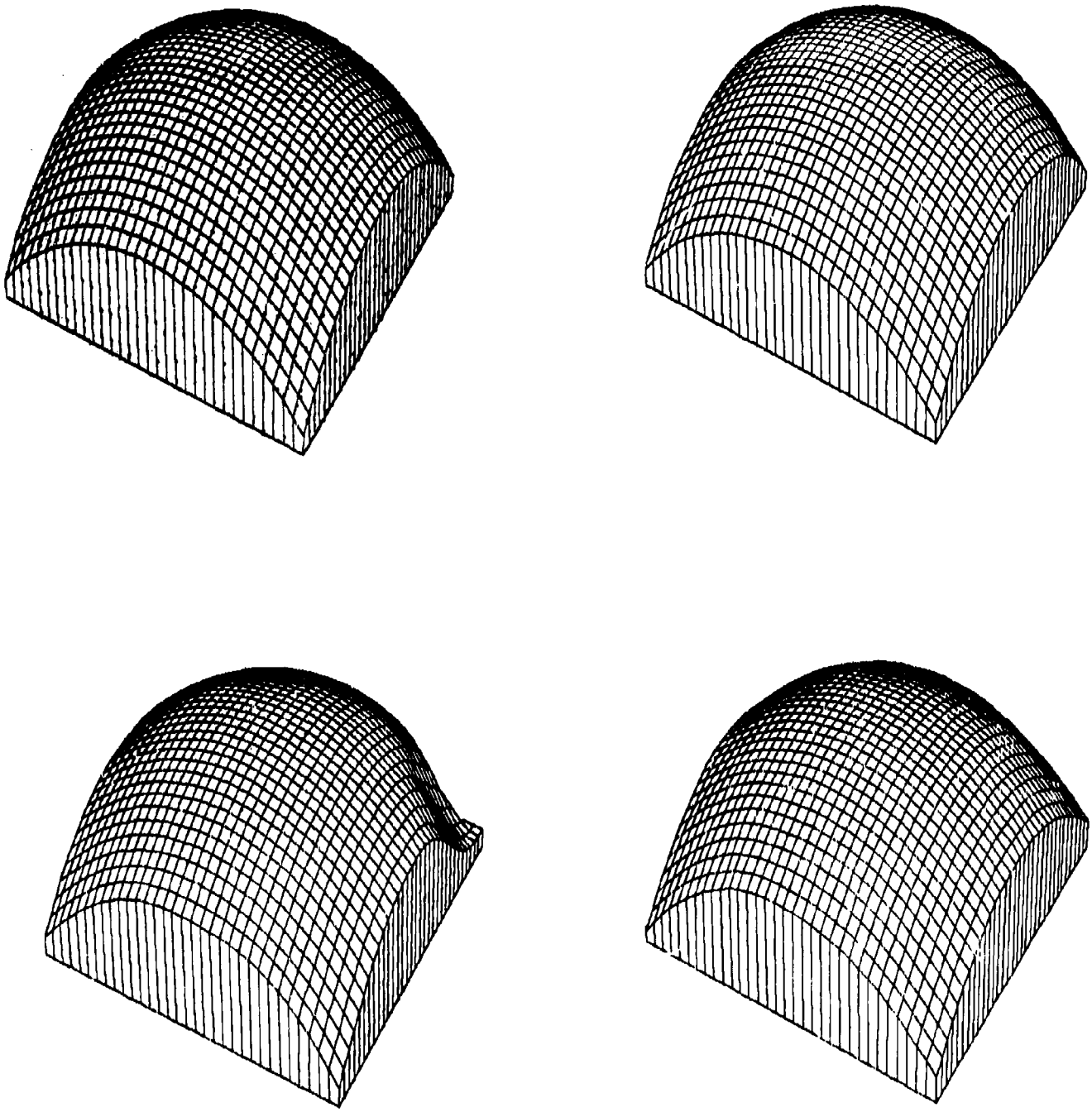
Nielson-Franke Quadratic Triangular Method

Figure 1.4.6.13



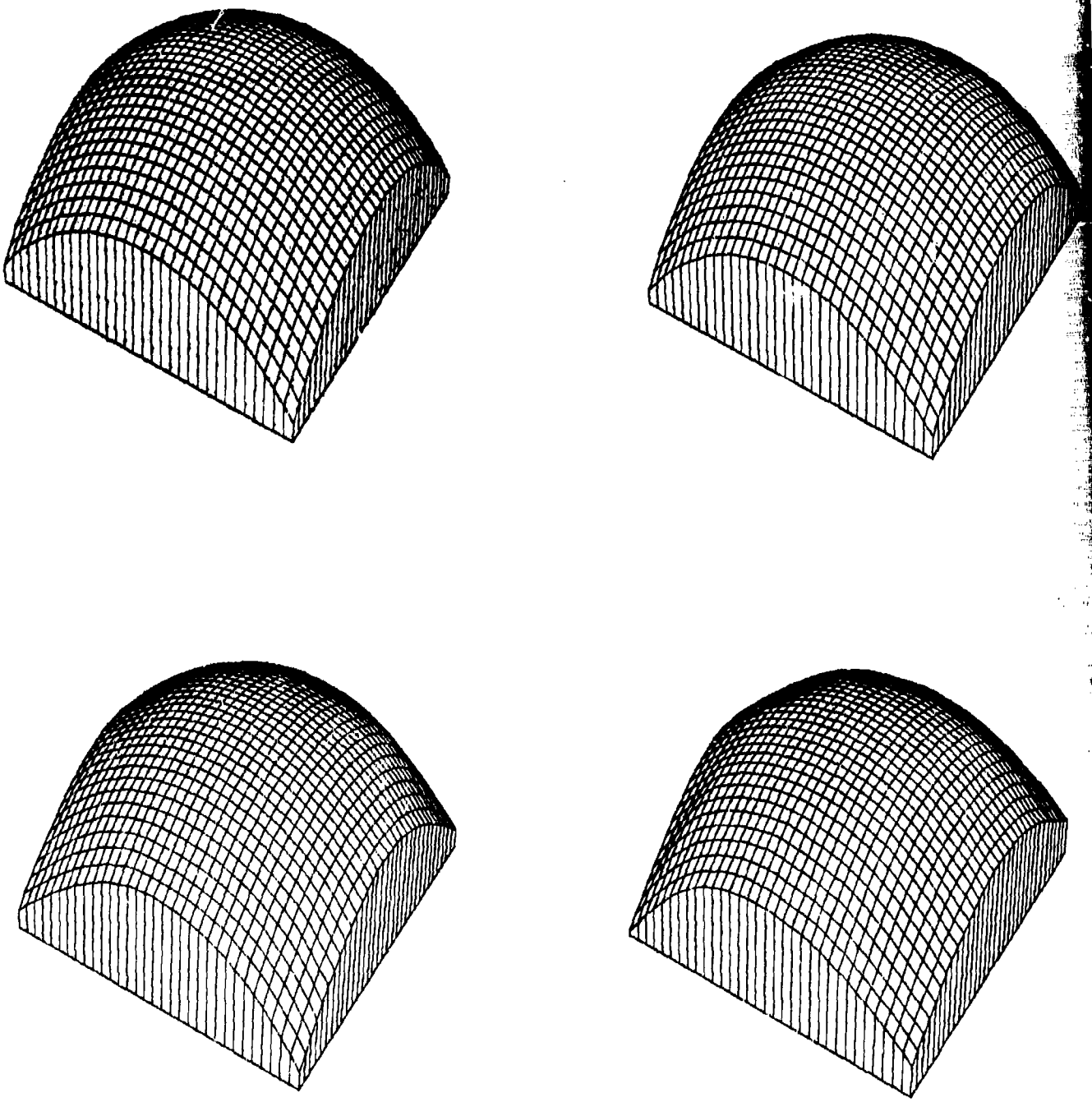
Modified Quadratic Shepard Method

Figure 1.4.6.14



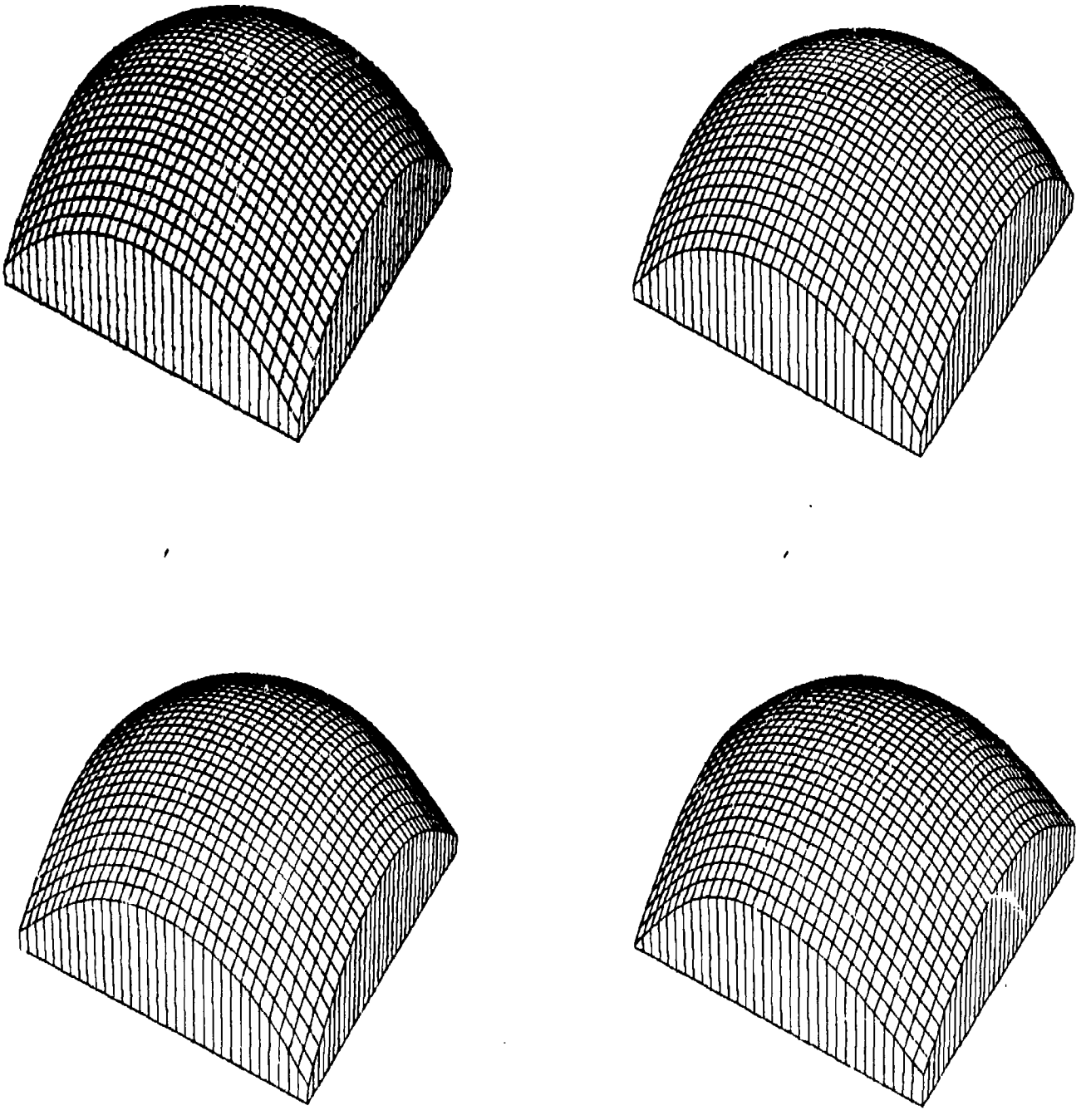
Akima's Method, Modification Three

Figure 1.4.6.16



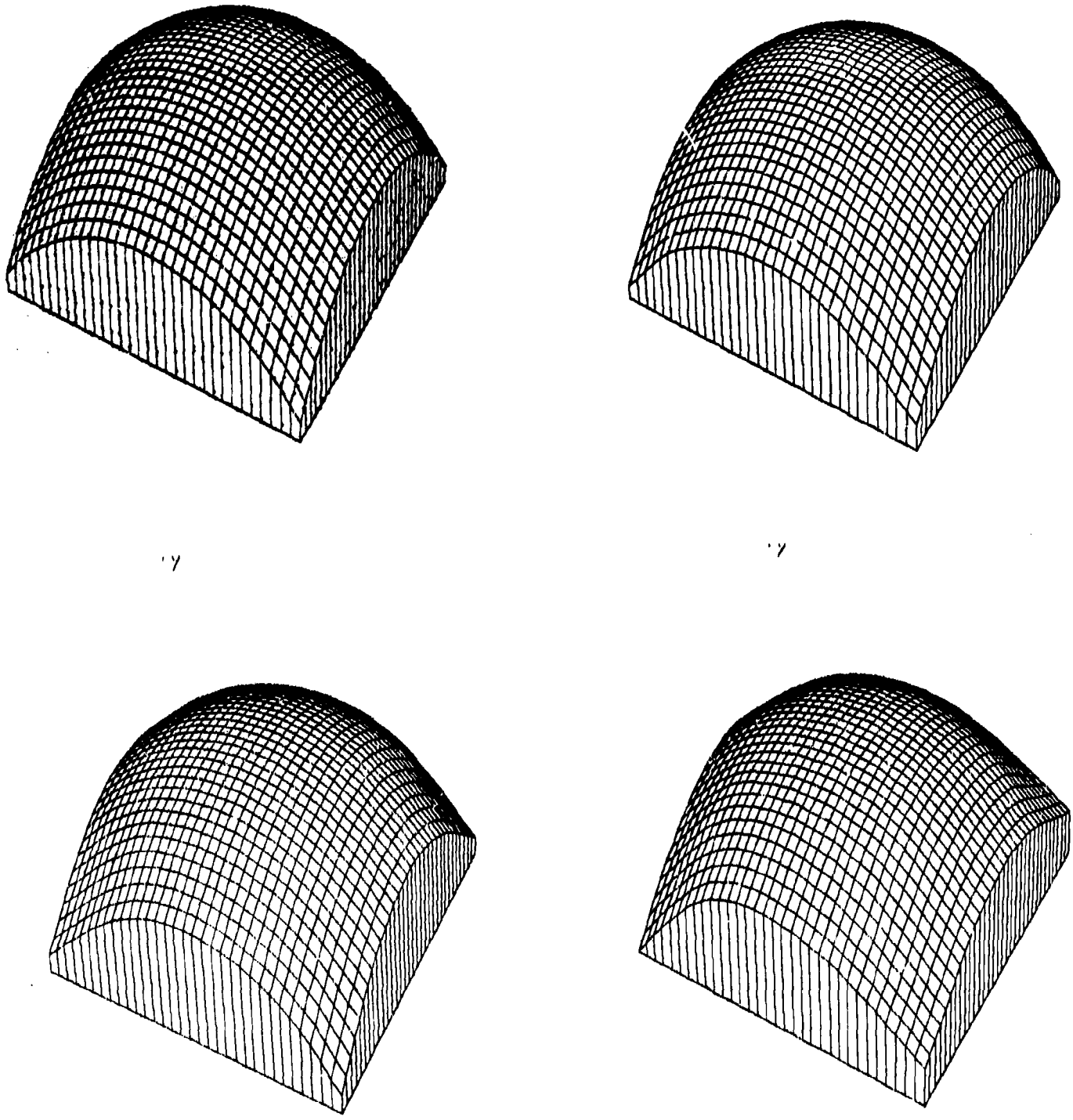
Nielson Minimum Norm Network

Figure 1.4.6.19



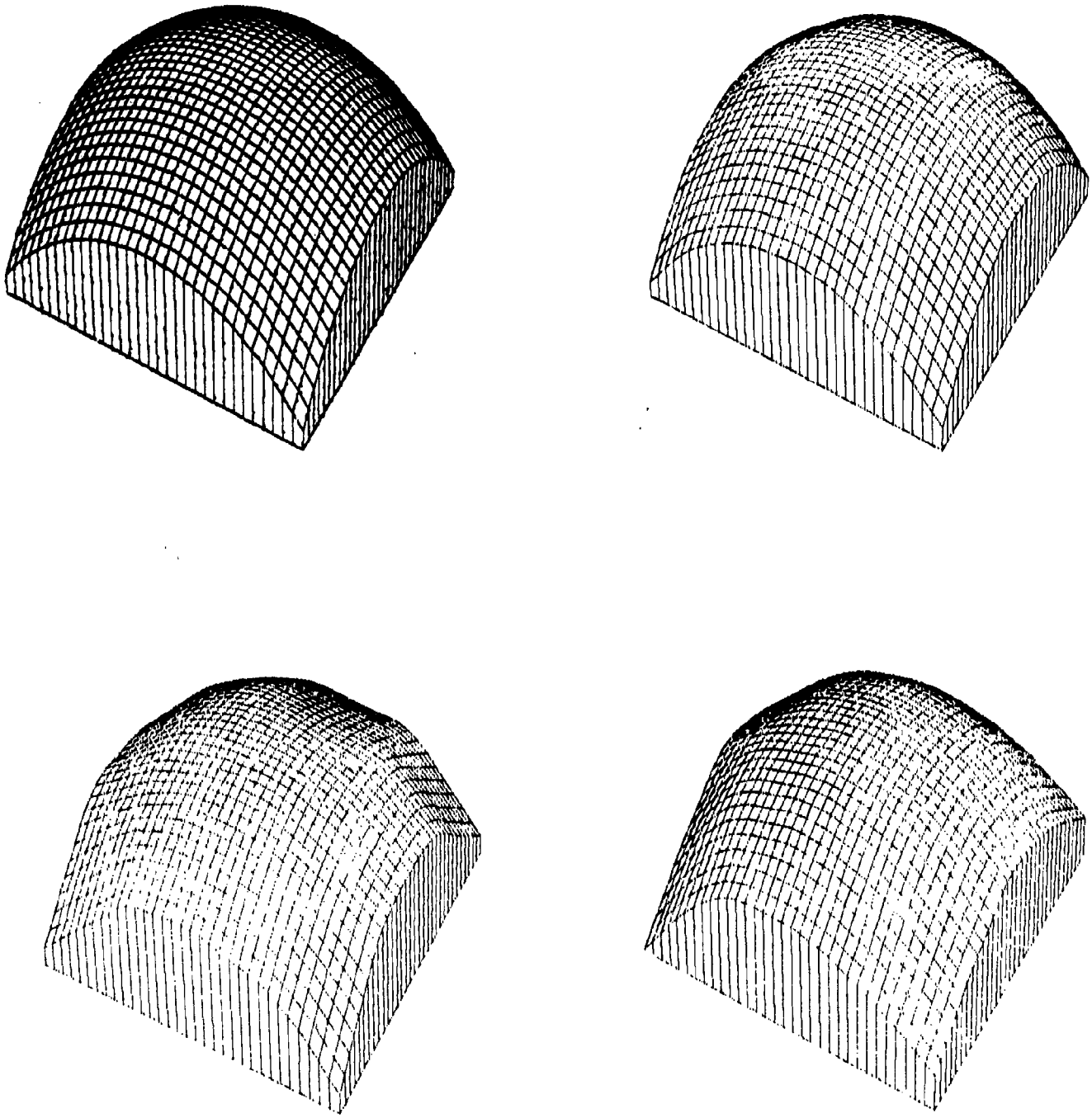
Hardy Multiquadric Method

Figure 1.4.6.21



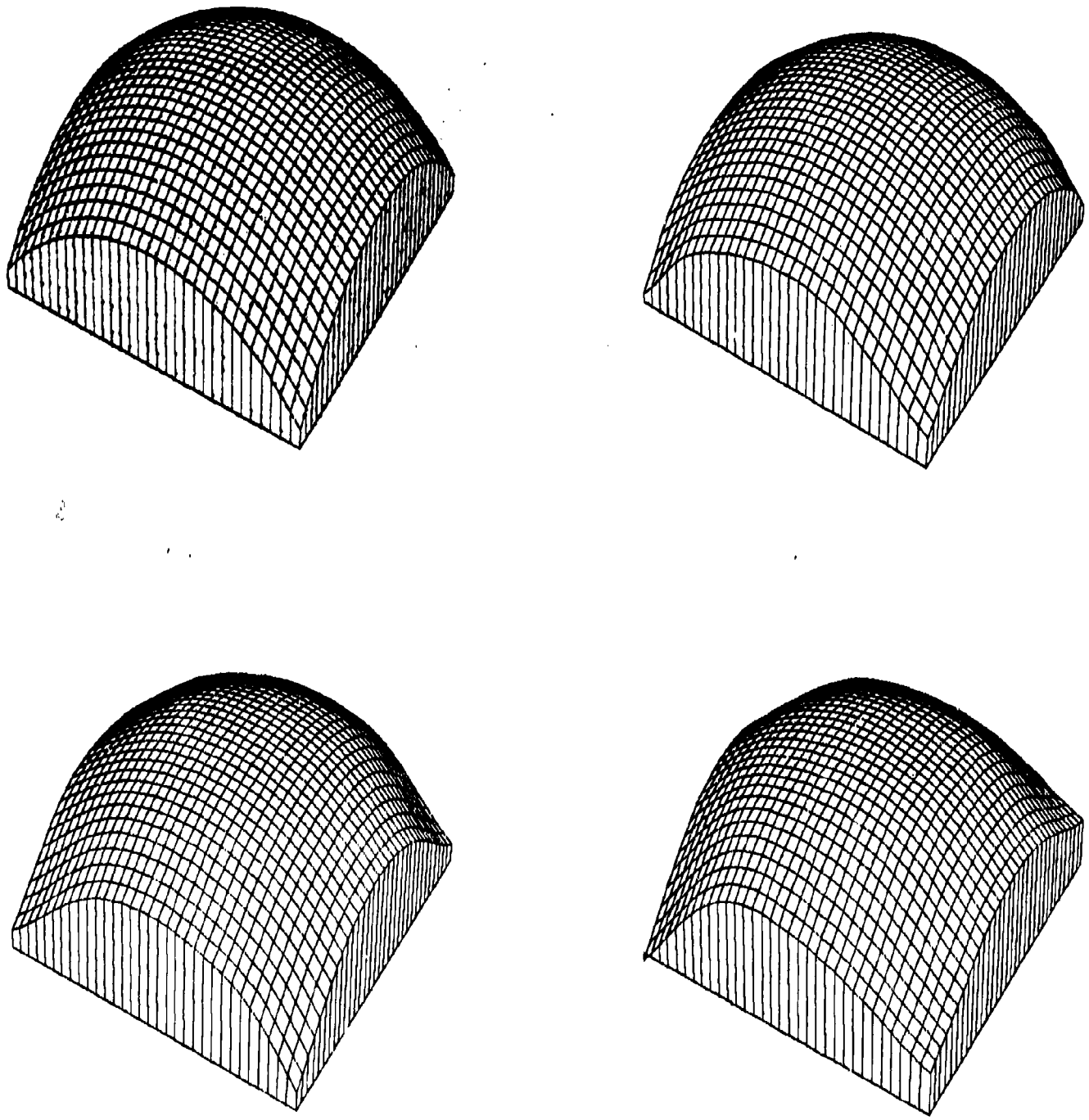
Duchon's Thin Plate Method

Figure 1.4.6.23



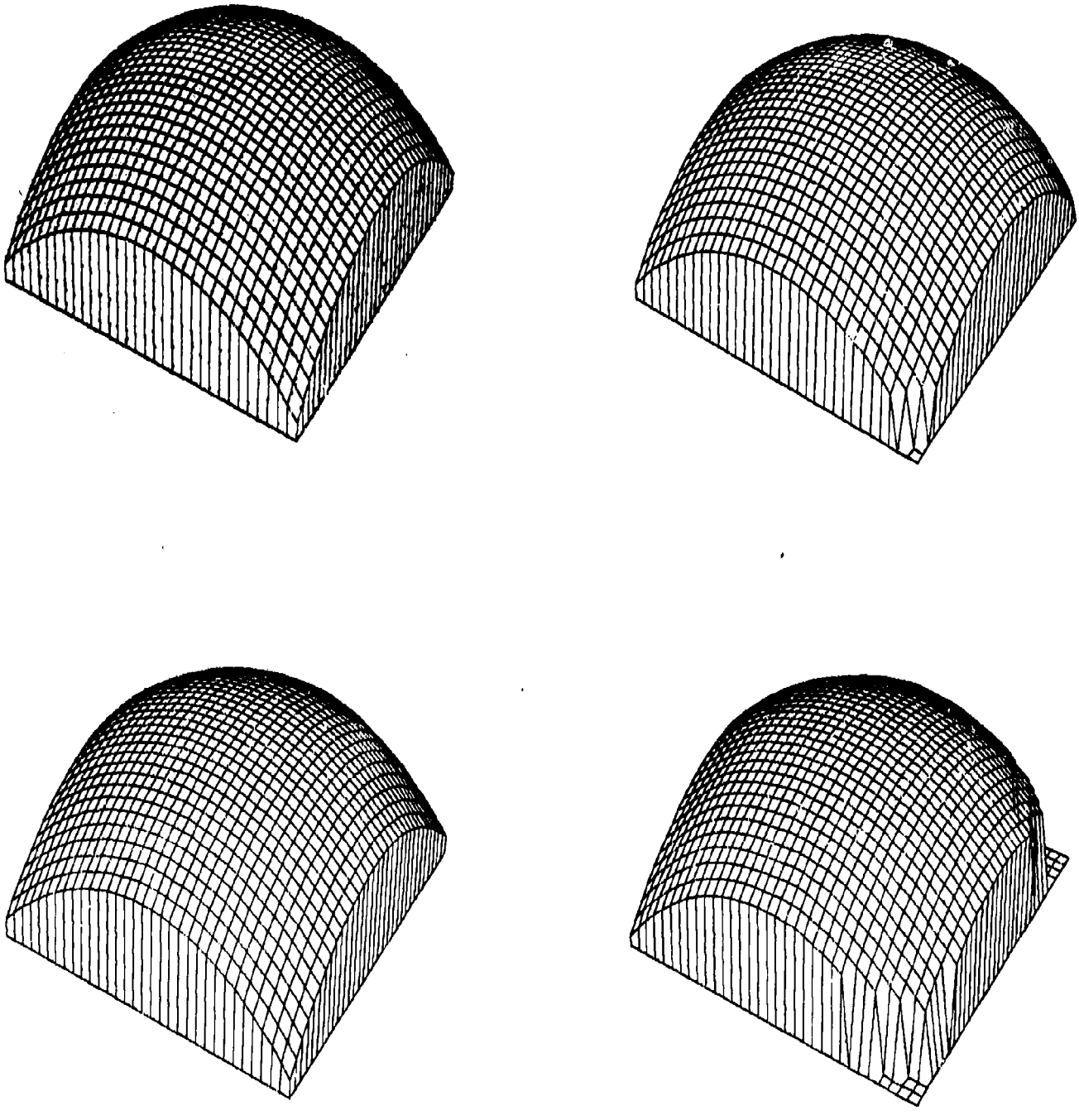
Franke's Method, Thin Plate Local Functions

Figure 1.4.6.24



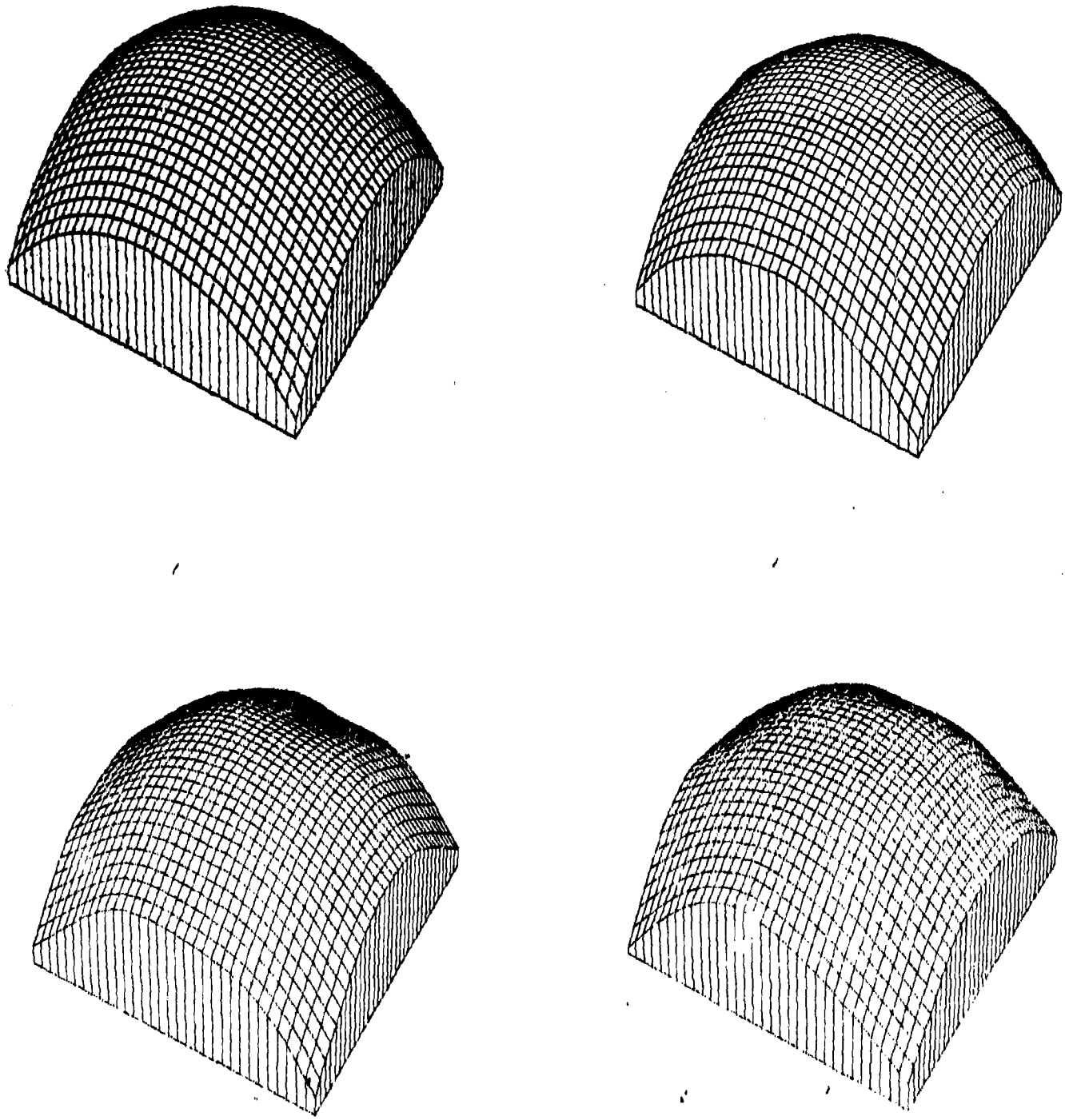
Hardy Reciprocal Multiquadric Method

Figure 1.4.6.27



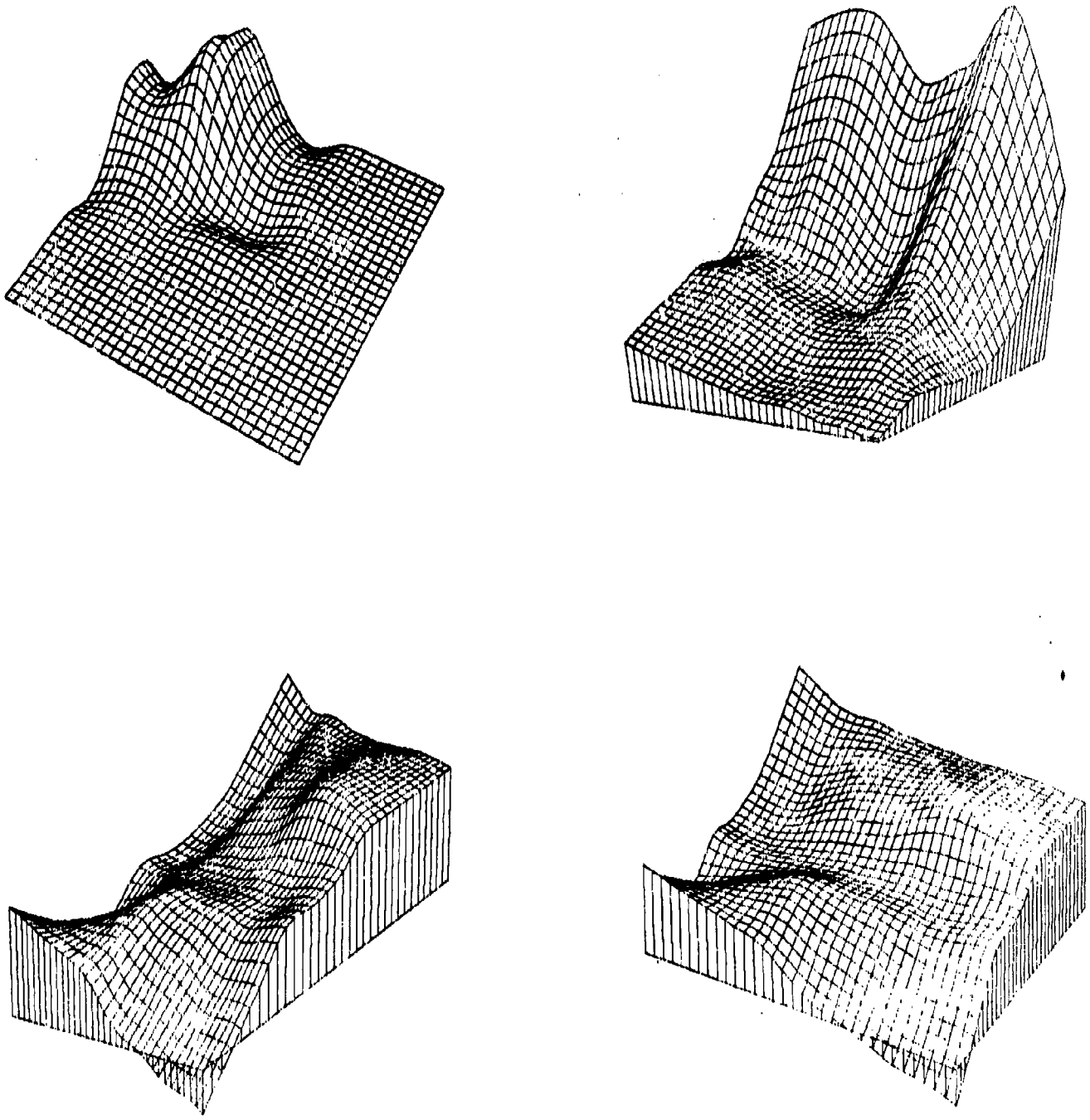
Lawson's Method

Figure 1.4.6.28



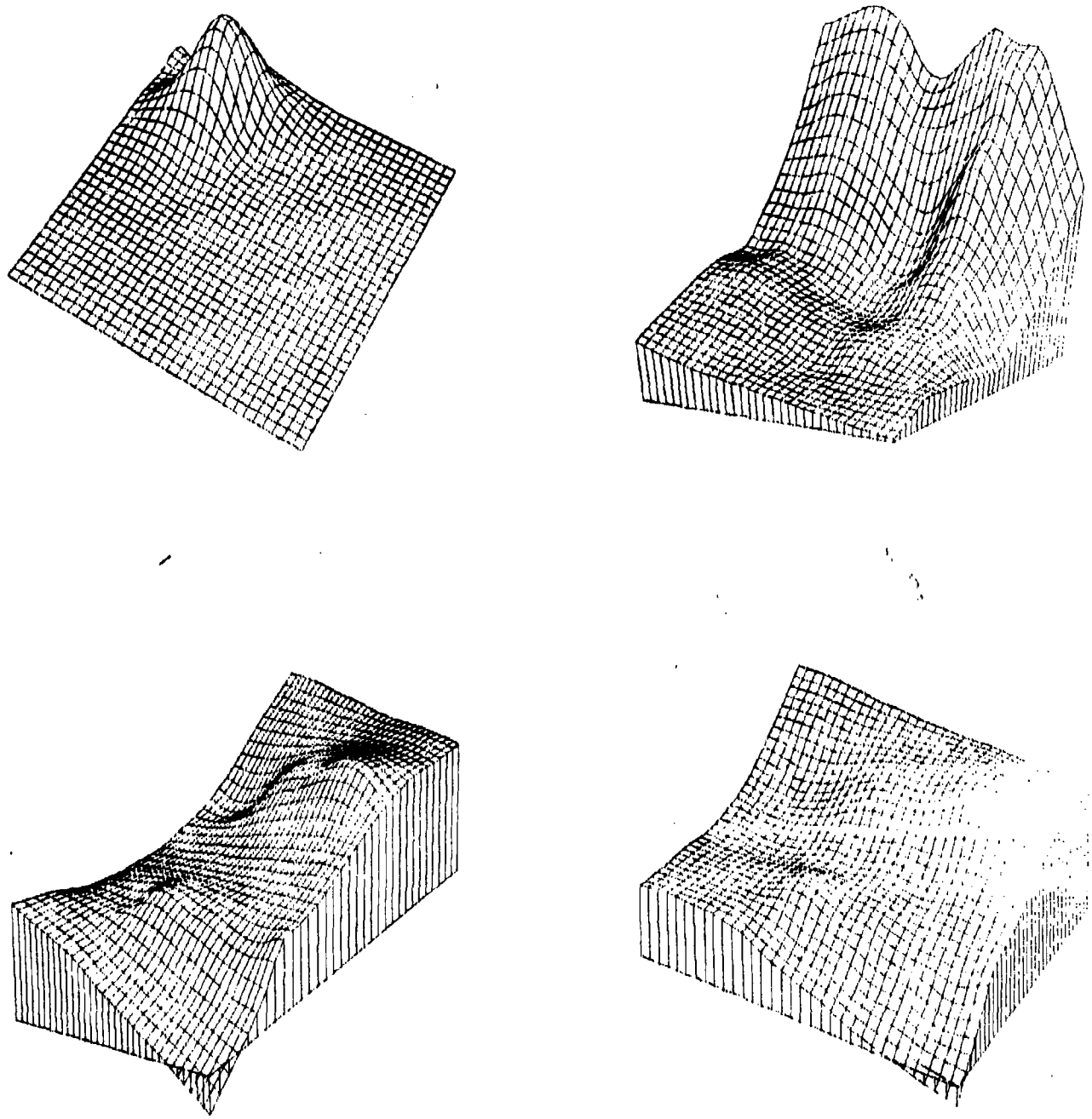
Foley's Iterated Generalized Newton
Delta Sum Bicubic Spline

Figure 1.4.6.30



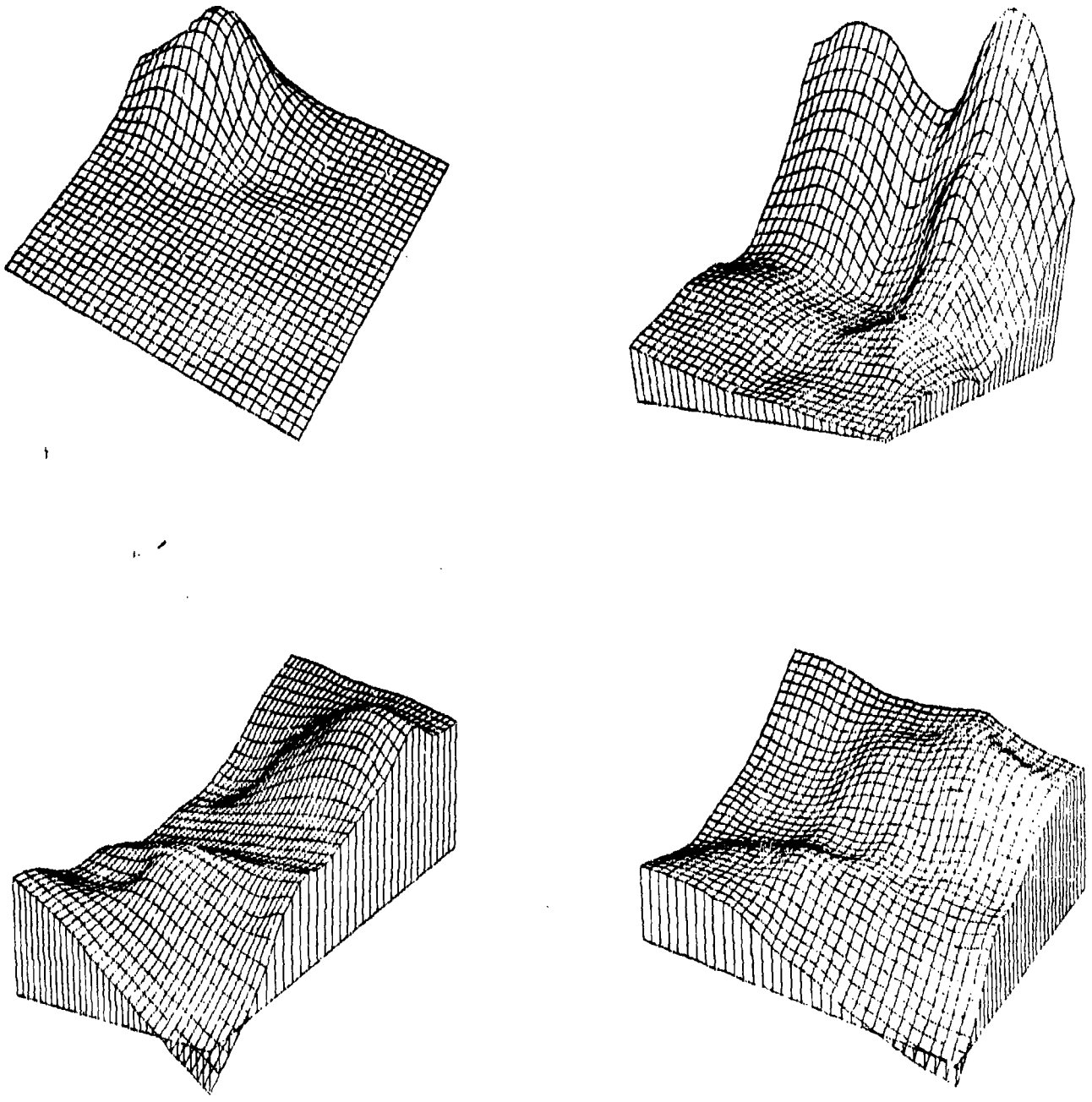
Franke's Method, Mode = 3

Figure 2.0.0.1



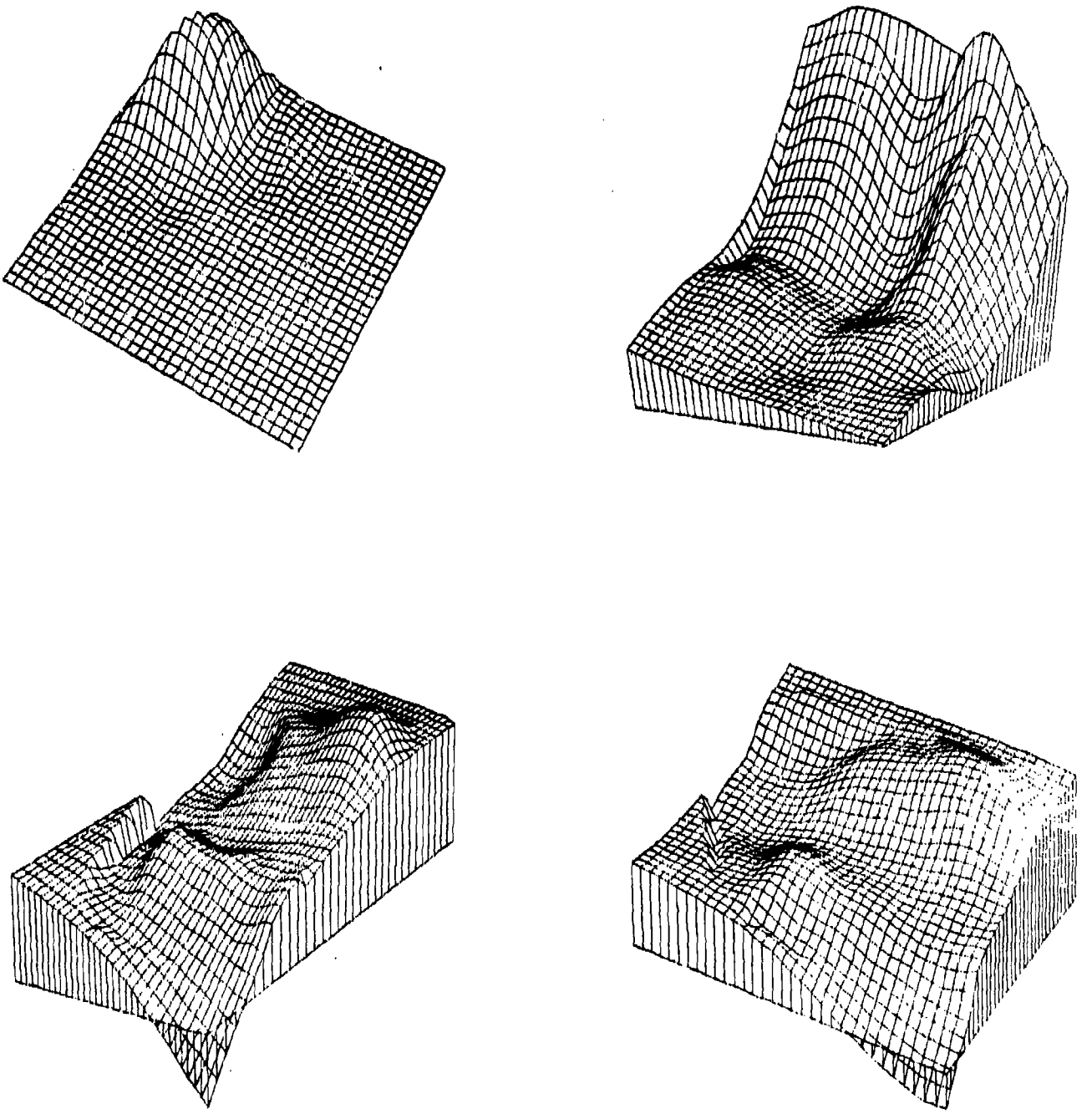
Modified Shepard's Method Boolean Sum Plane

Figure 2.0.0.2



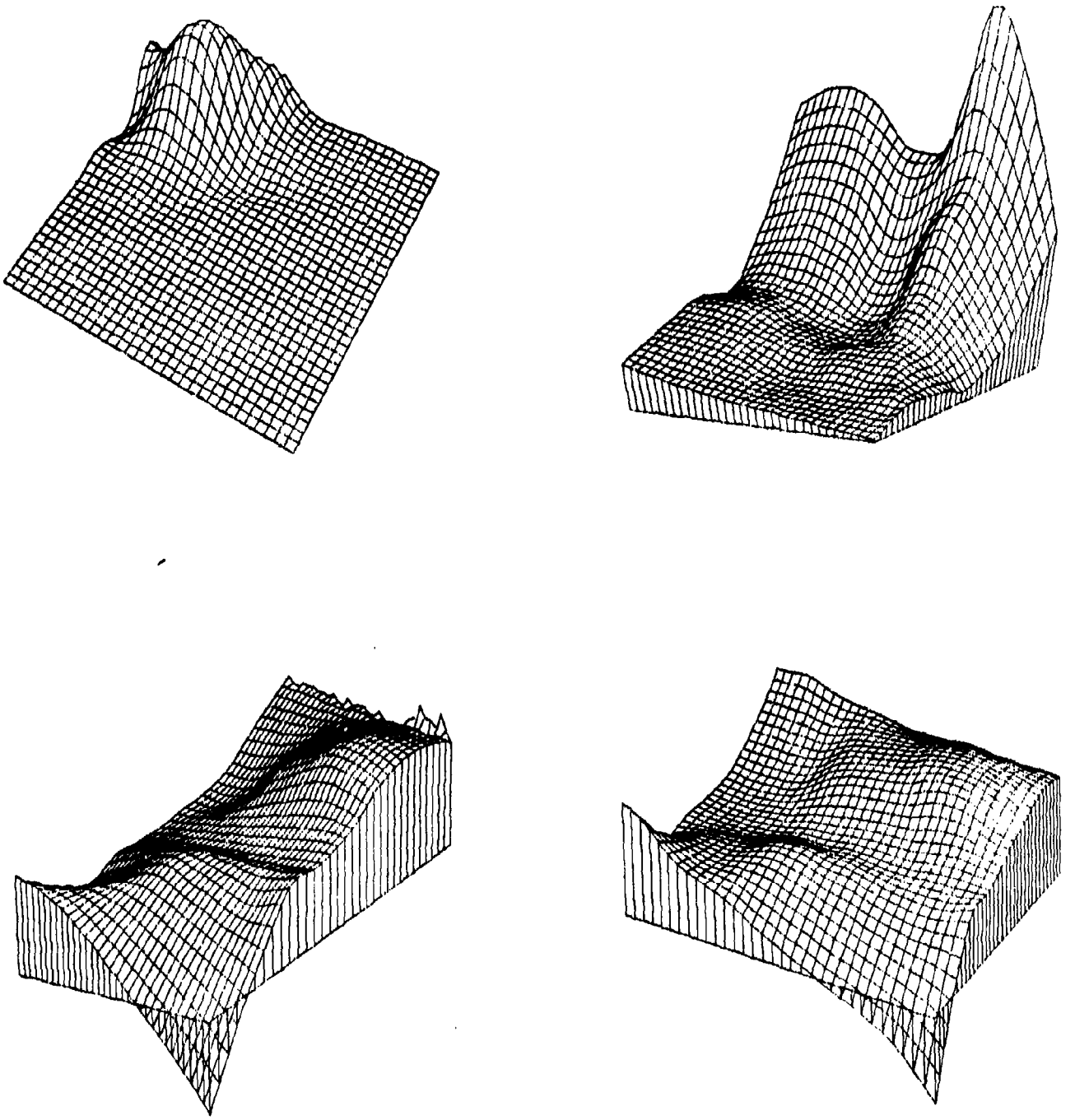
Modified Linear Shepard's Method

Figure 2.0.0.3



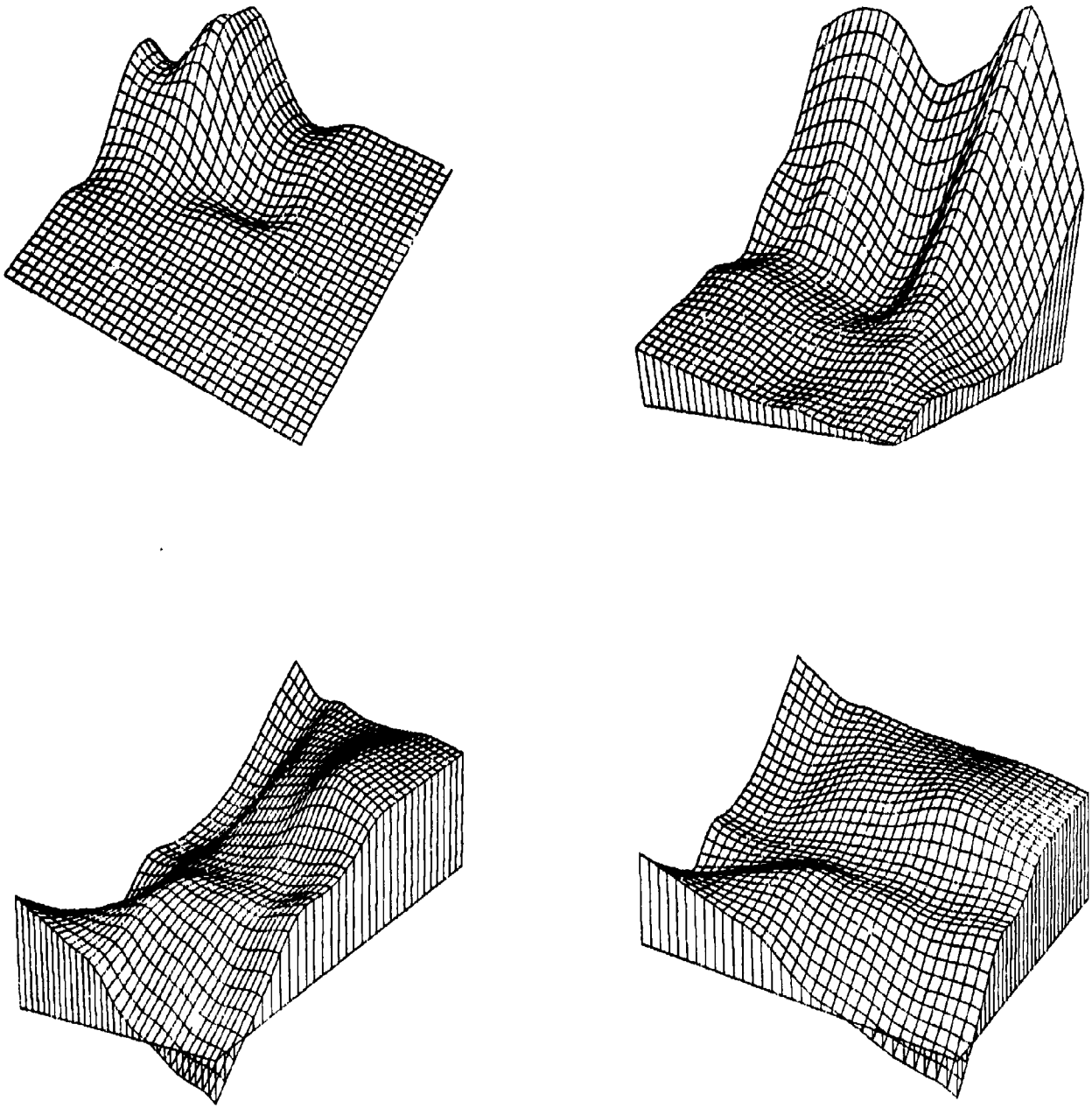
Akima's Method

Figure 2.0.0.4



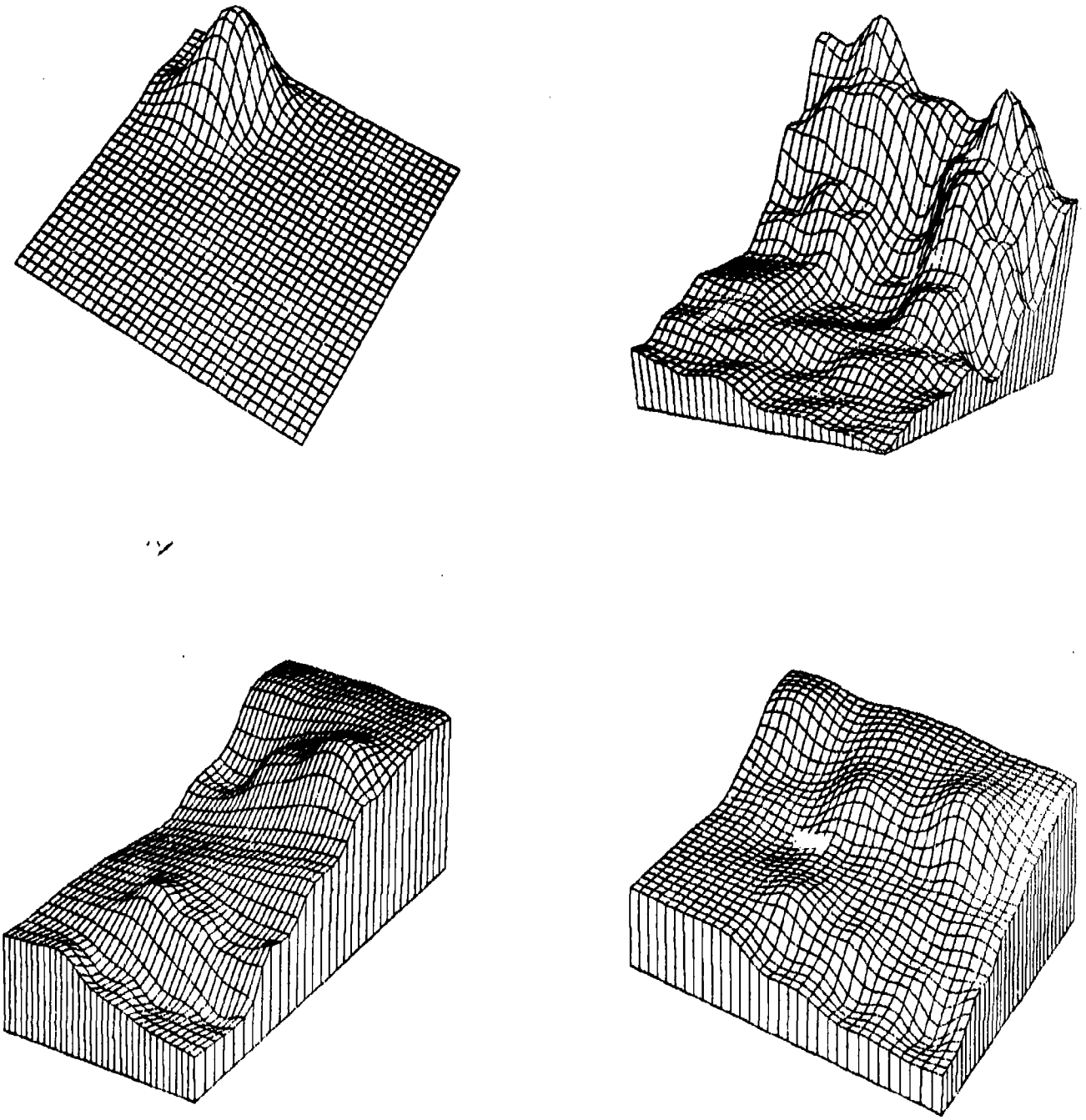
McLain's Inverse Distance Weighted Quadratic

Figure 2.0.0.5



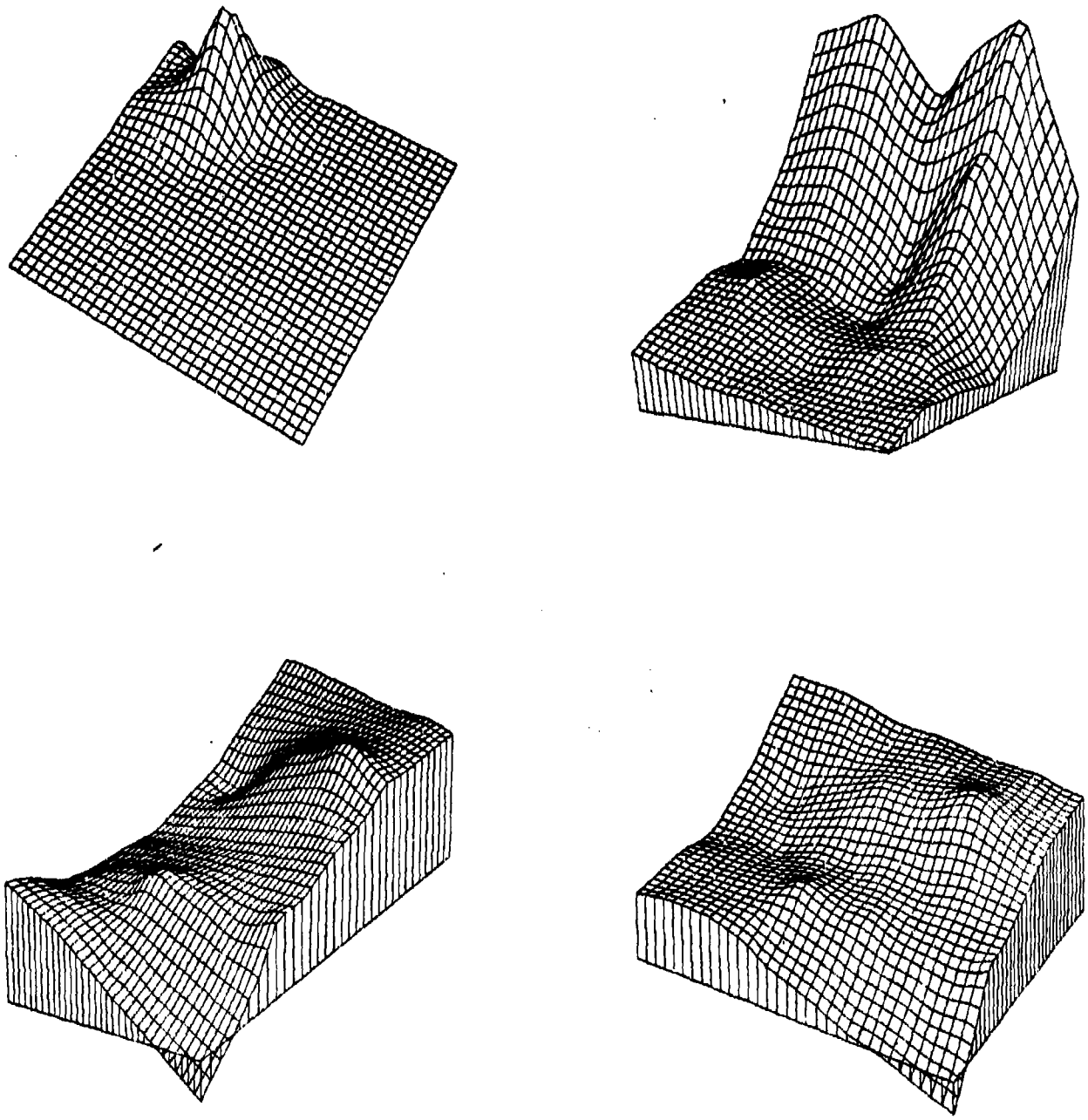
Franke's Method, Mode = 1

Figure 2.0.0.6



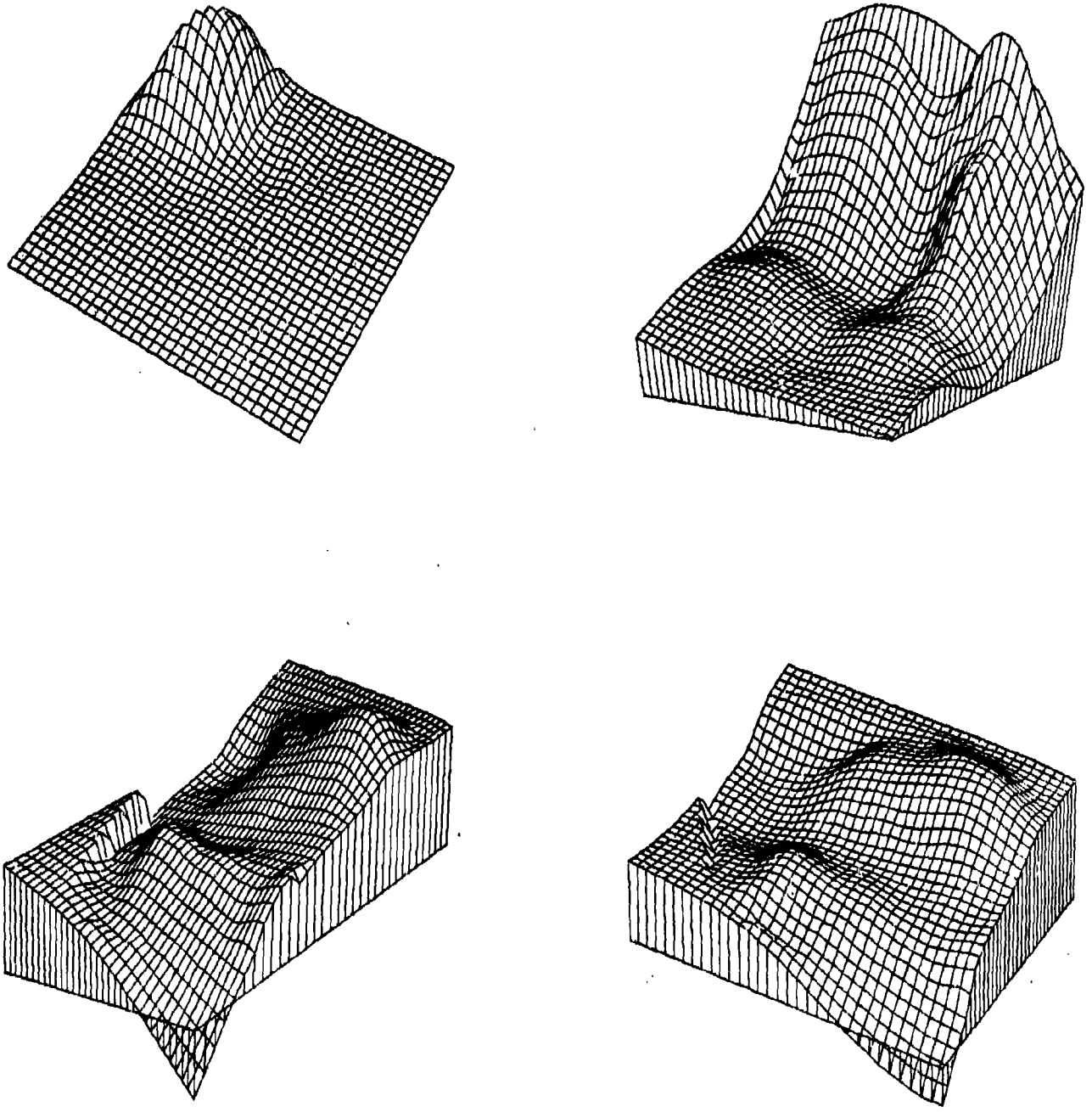
Modified Shepard's Method

Figure 2.0.0.7



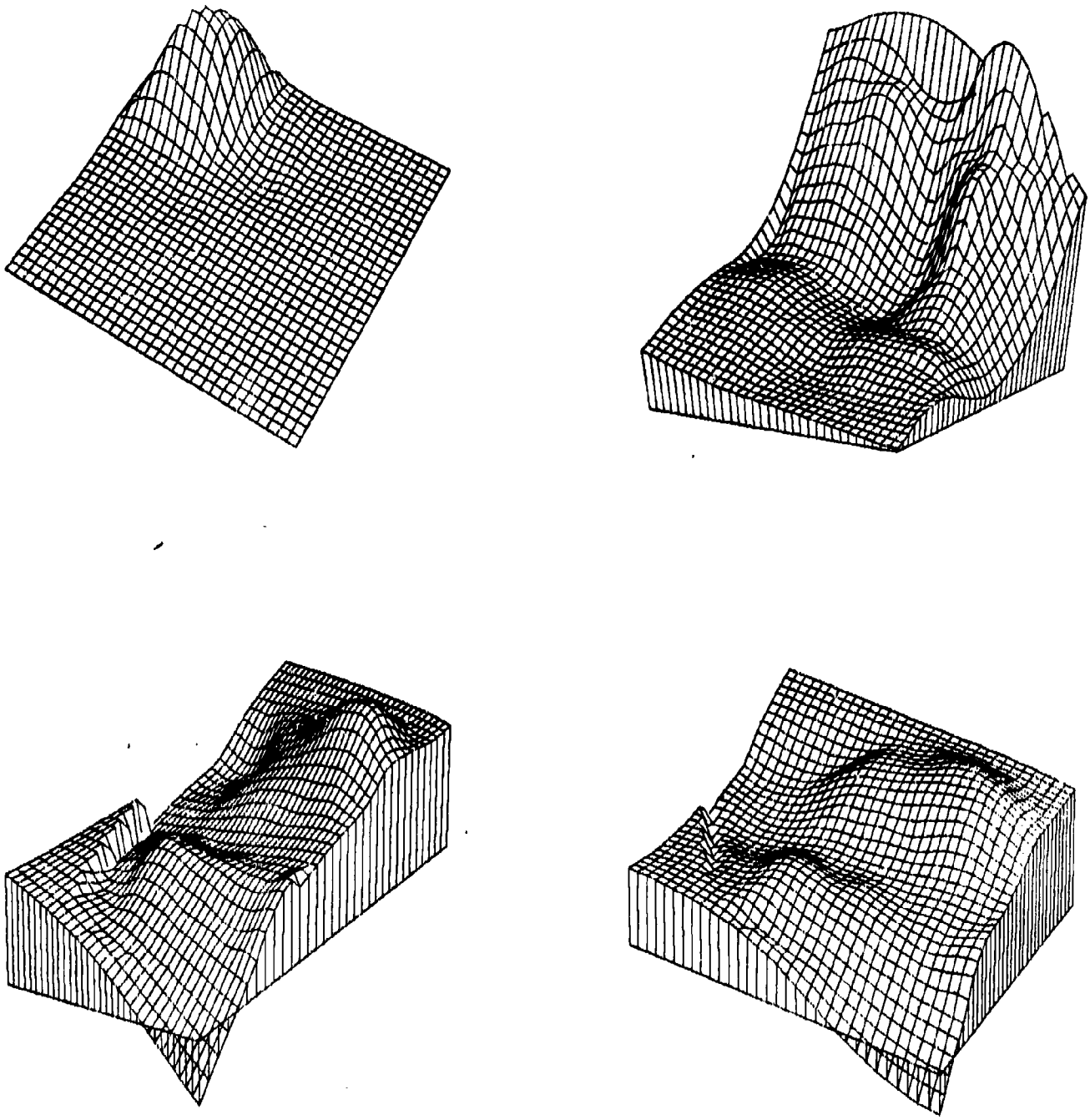
Modified McLain's Inverse Distance Weighted Plane

Figure 2.0.0.8



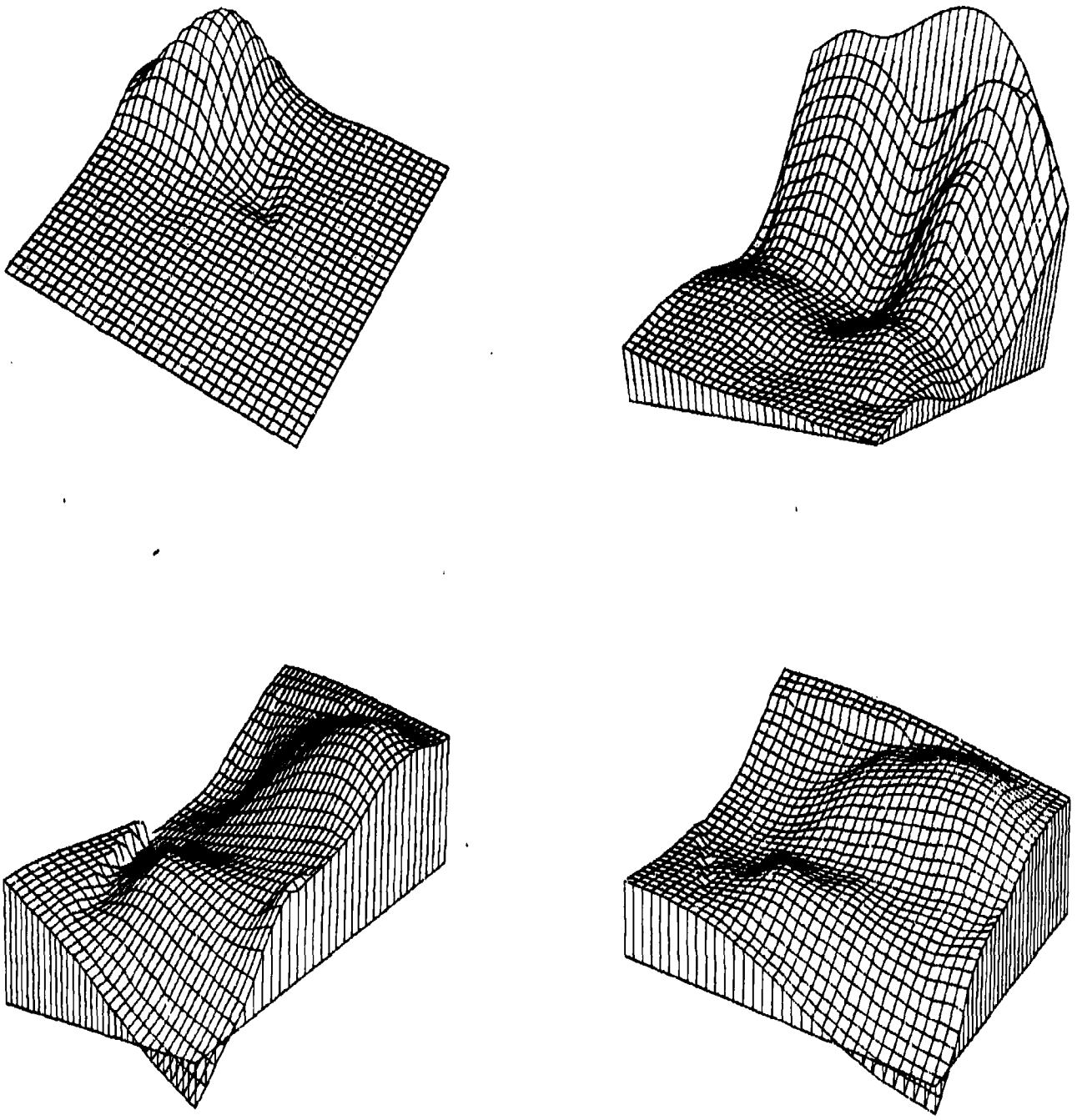
Akima's Method, Modification One

Figure 2.0.0.10



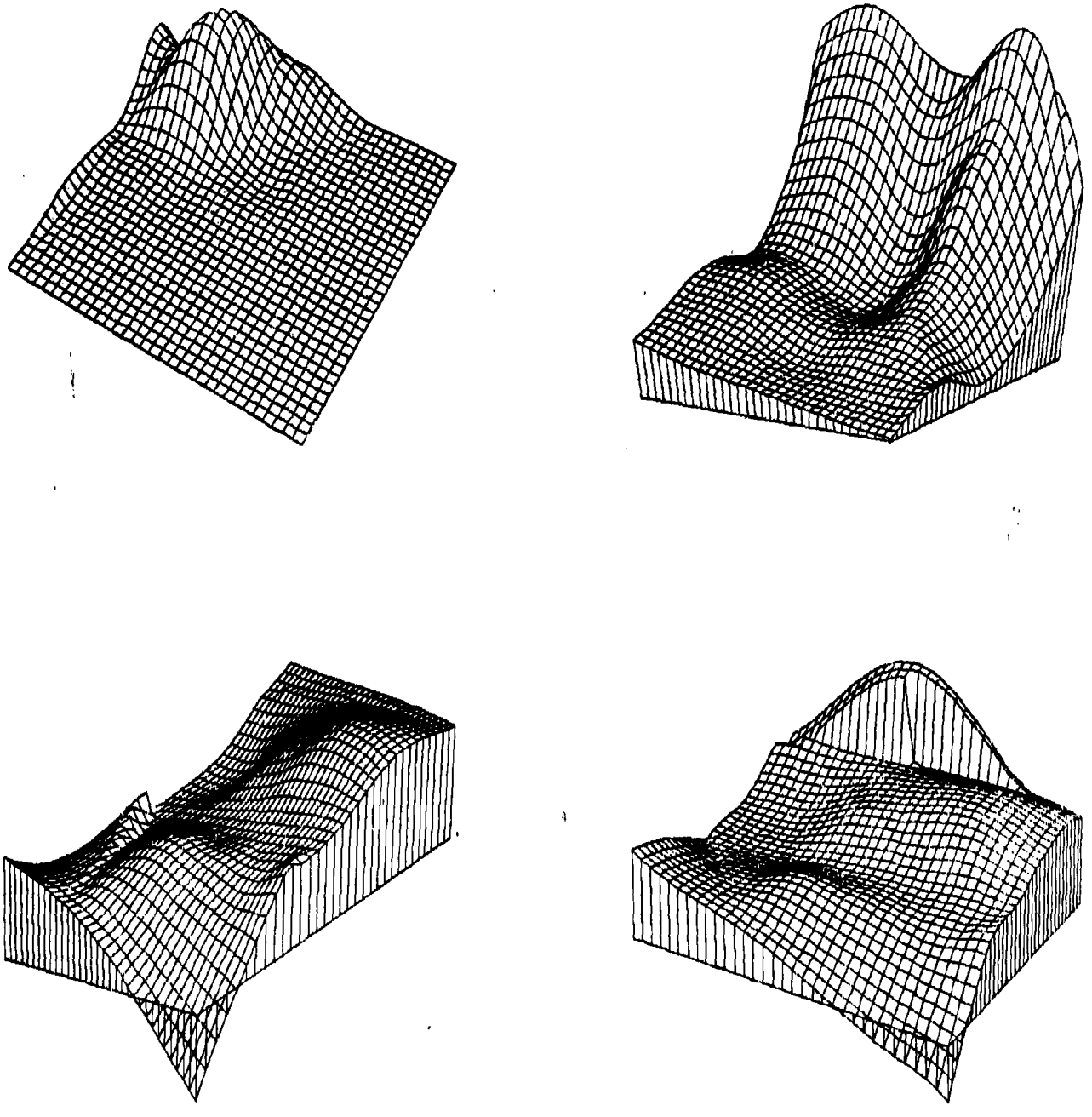
Akima's Method, Modification Two

Figure 2.0.0.11



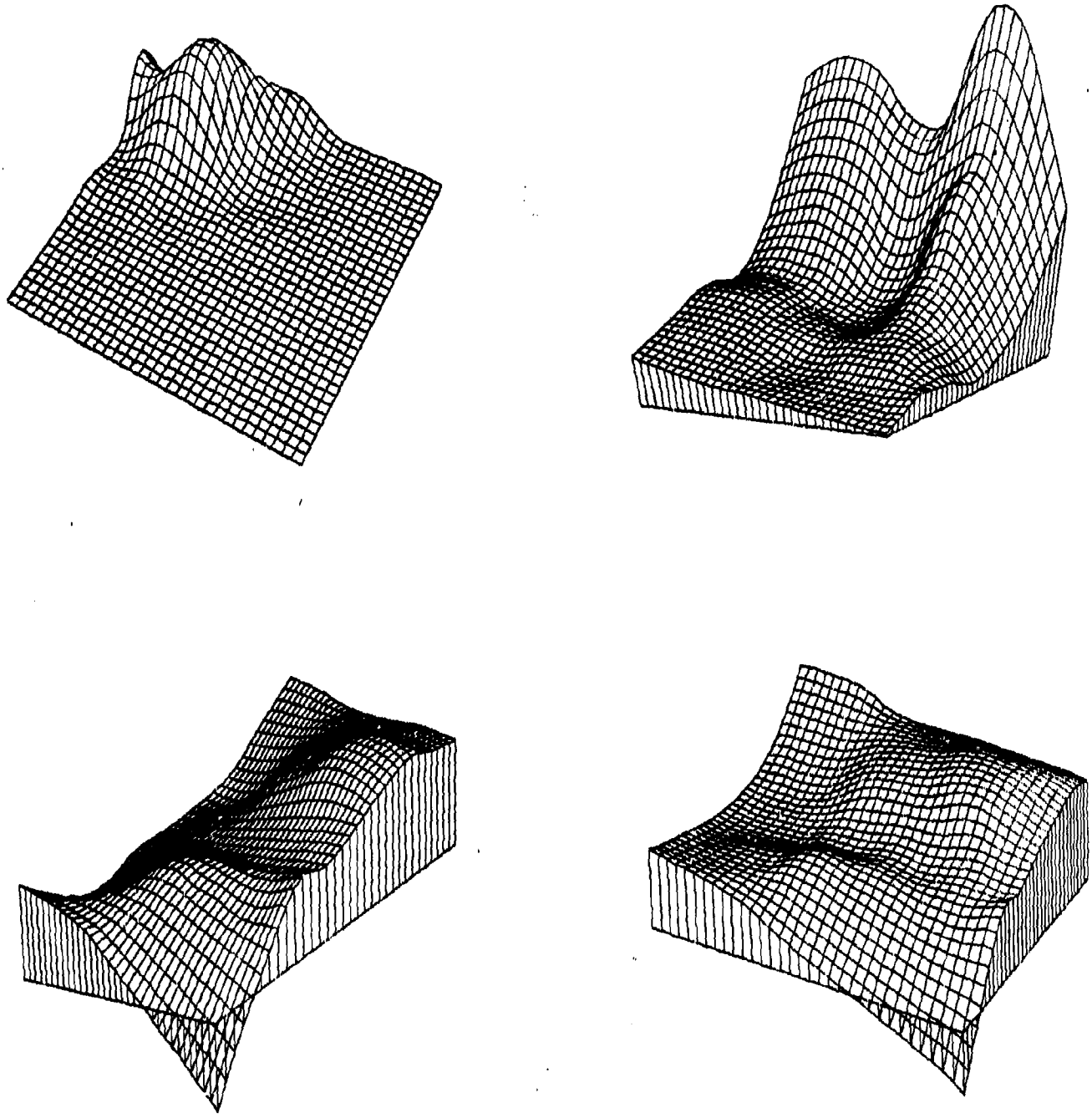
Nielson-Franke Linear Triangle Method

Figure 2.0.0.12



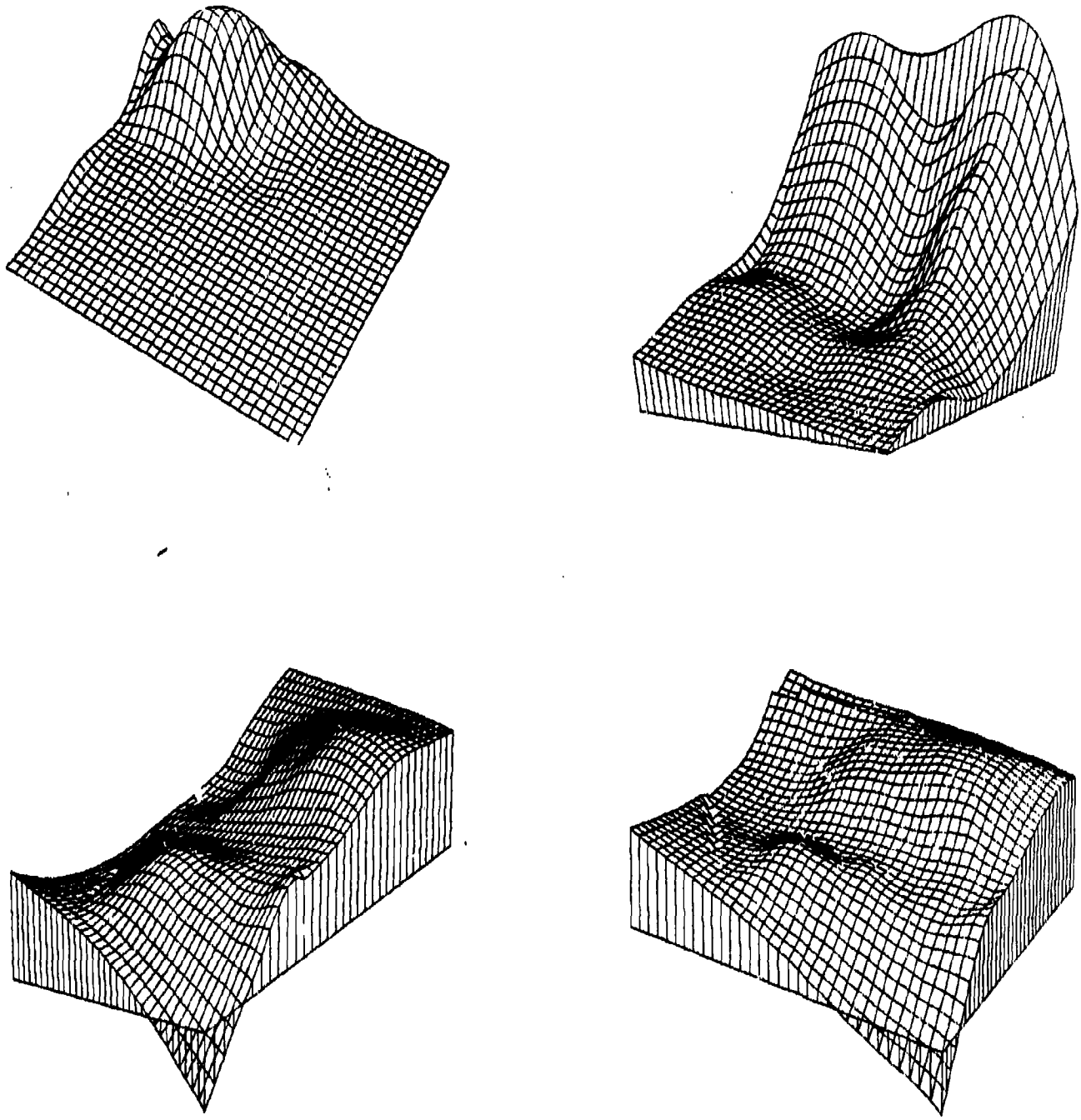
Nielson-Franke Quadratic Triangular Method

Figure 2.0.0.13



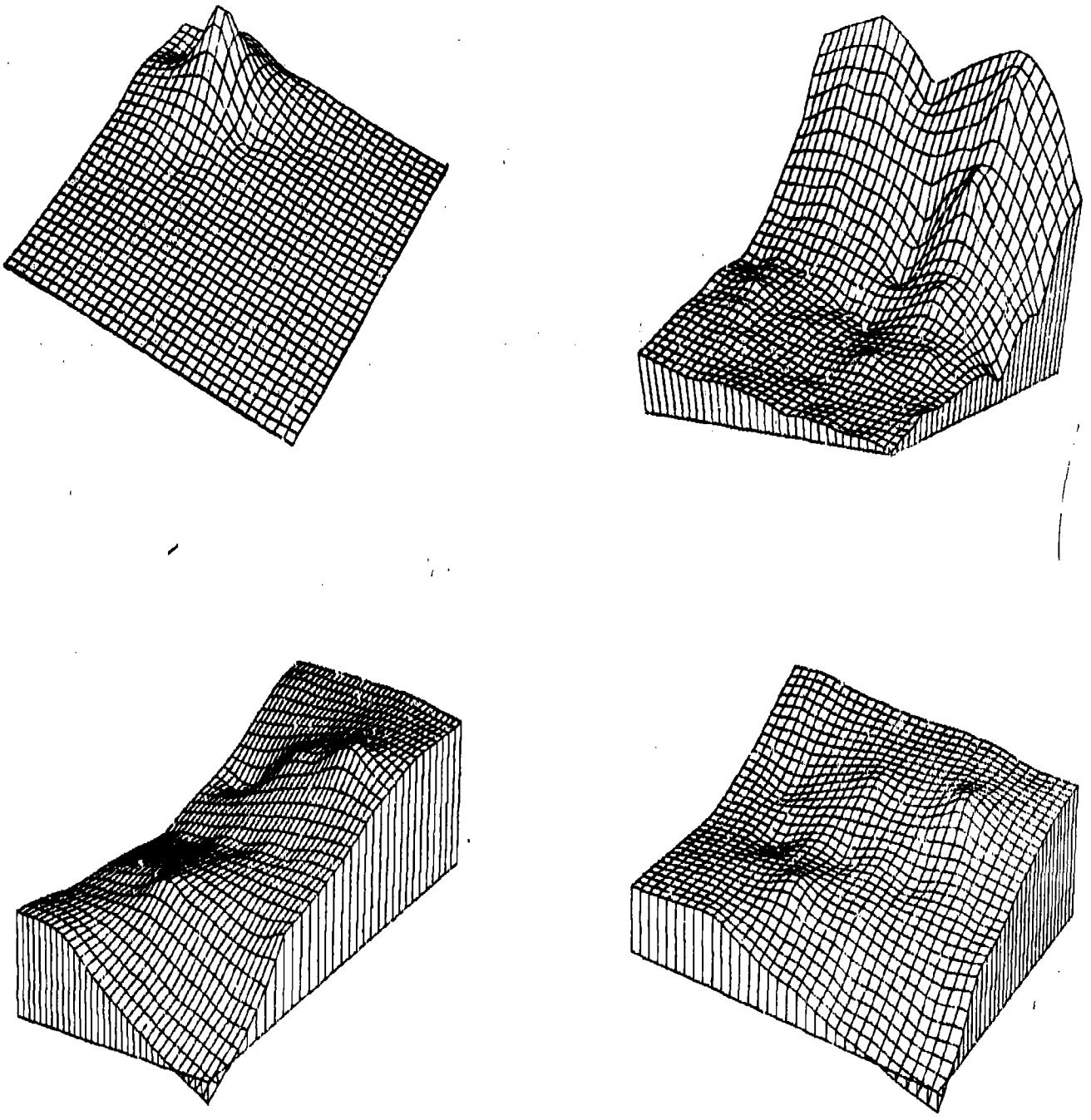
Modified Quadratic Shepard Method

Figure 2.0.0.14



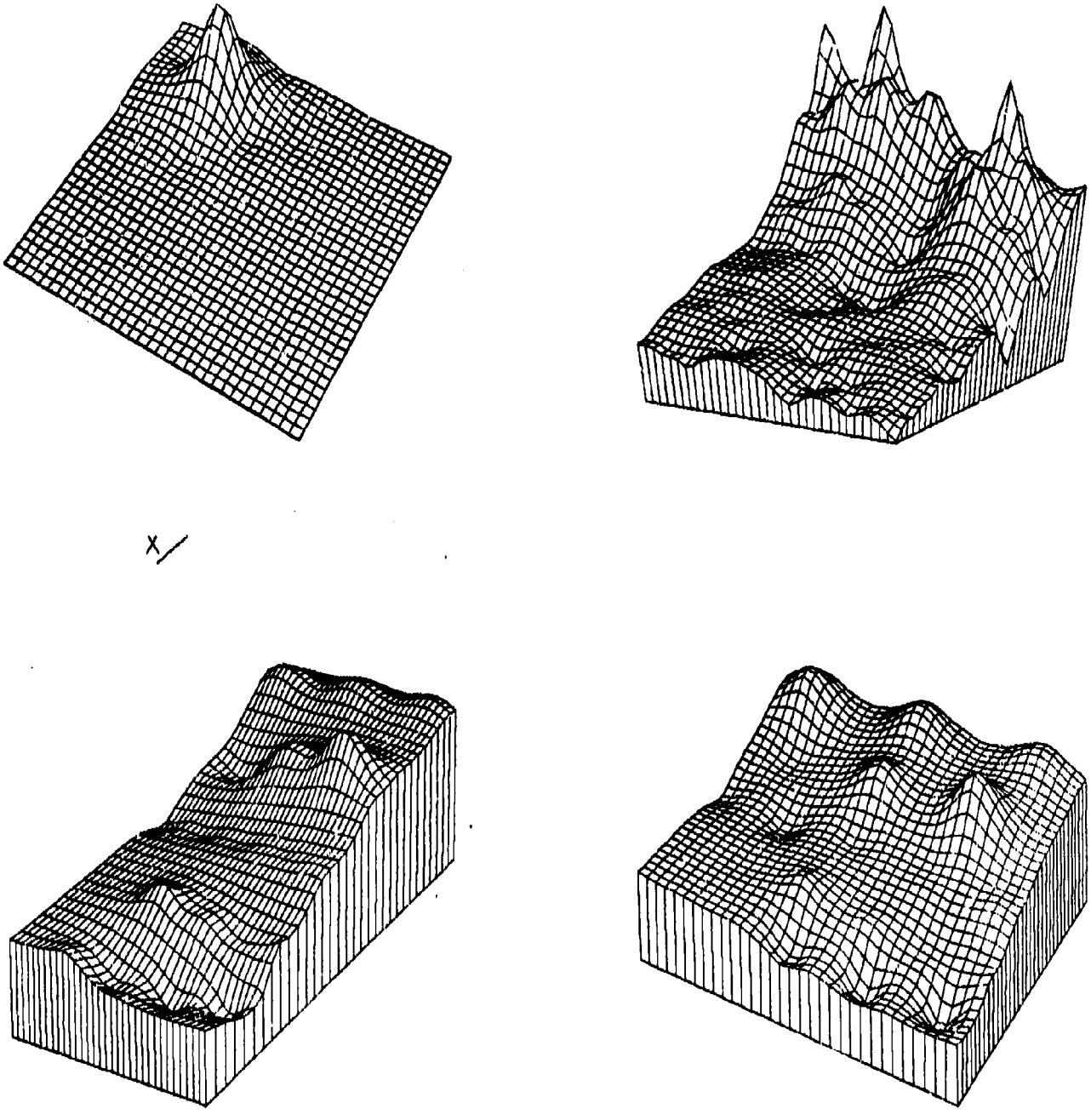
Akima's Method, Modification Three

Figure 2.0.0.16



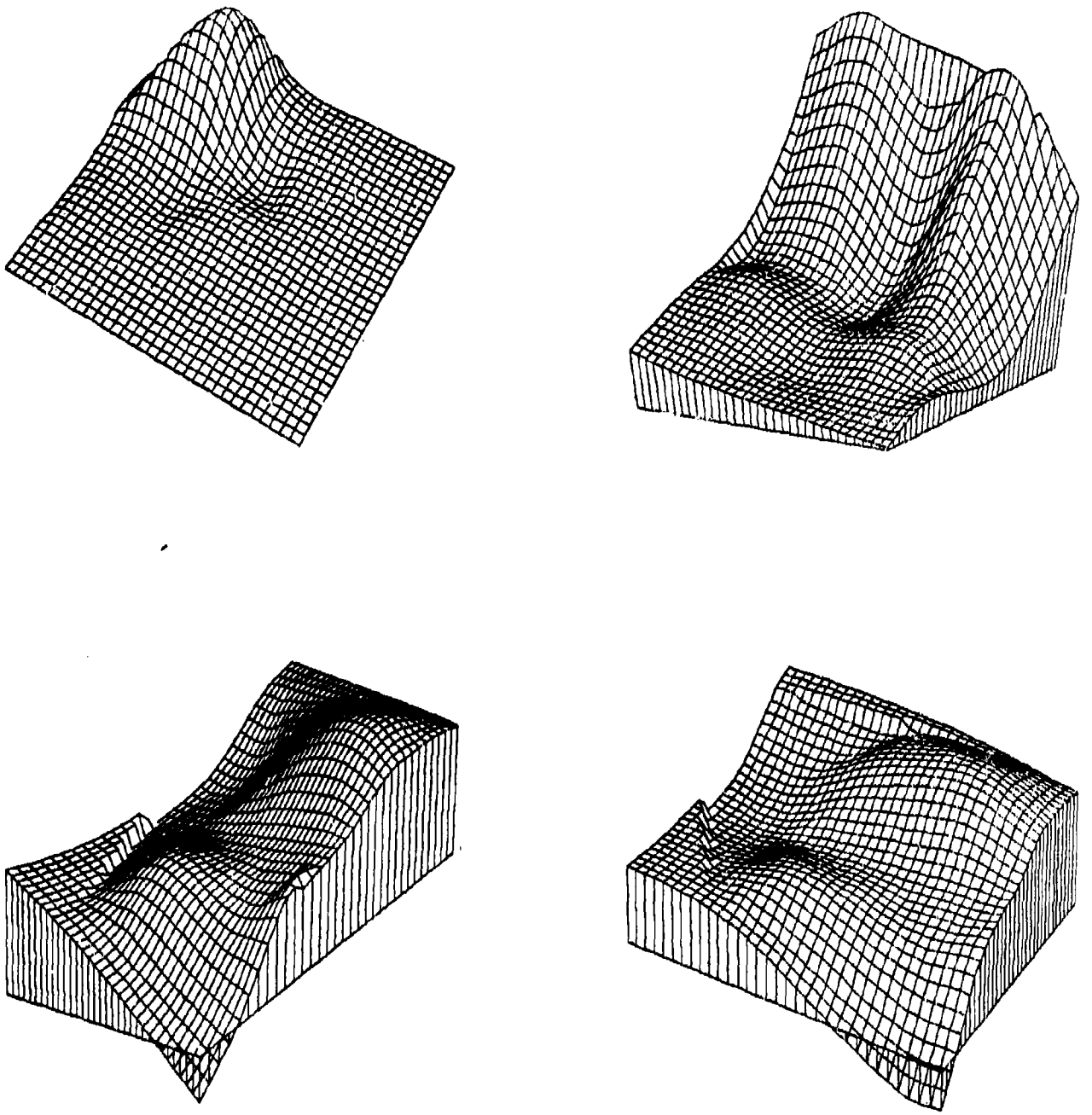
Quadratic Shepard's Method

Figure 2.0.0.17



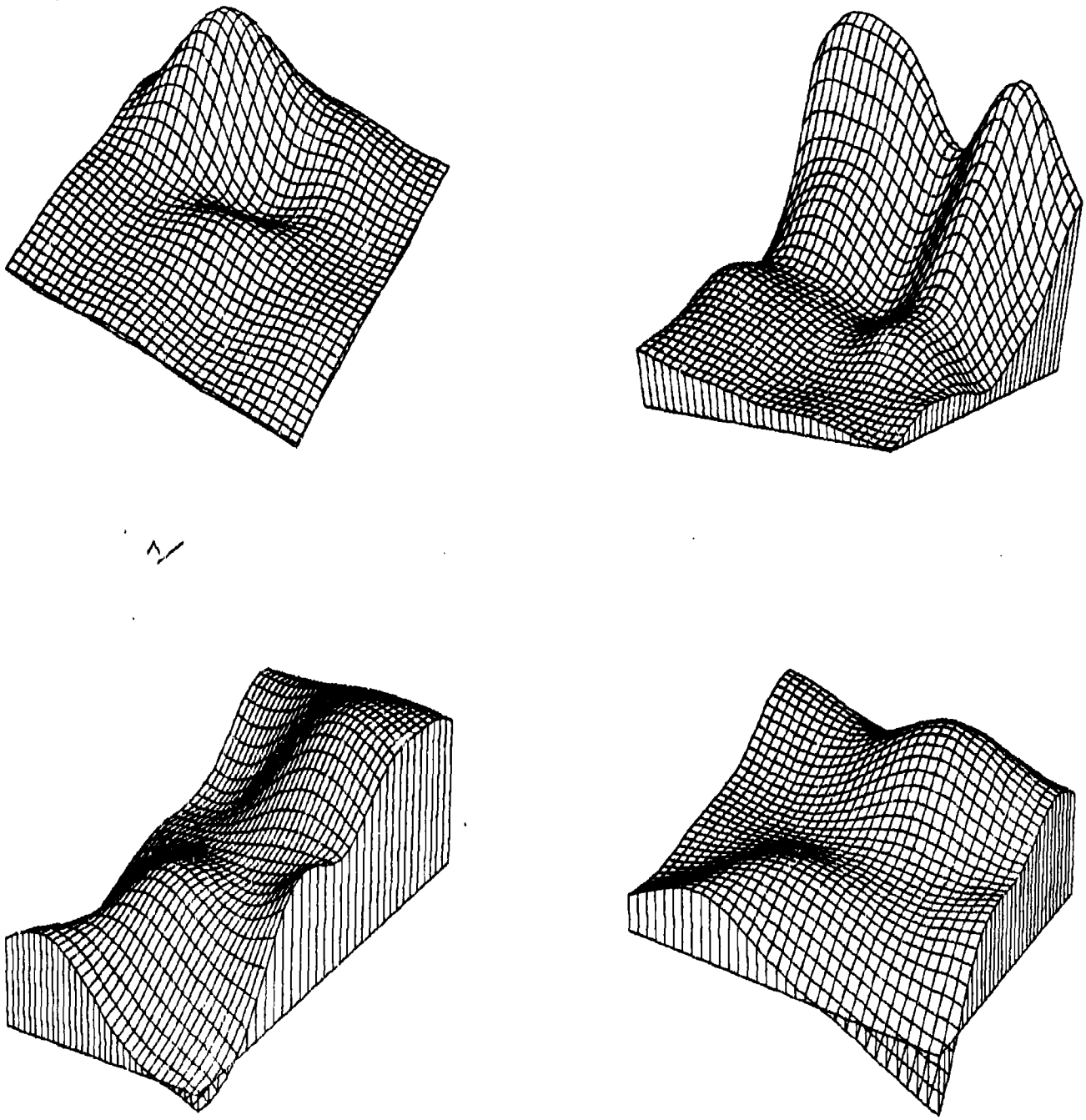
Shepard's Method

Figure 2.0.0.18



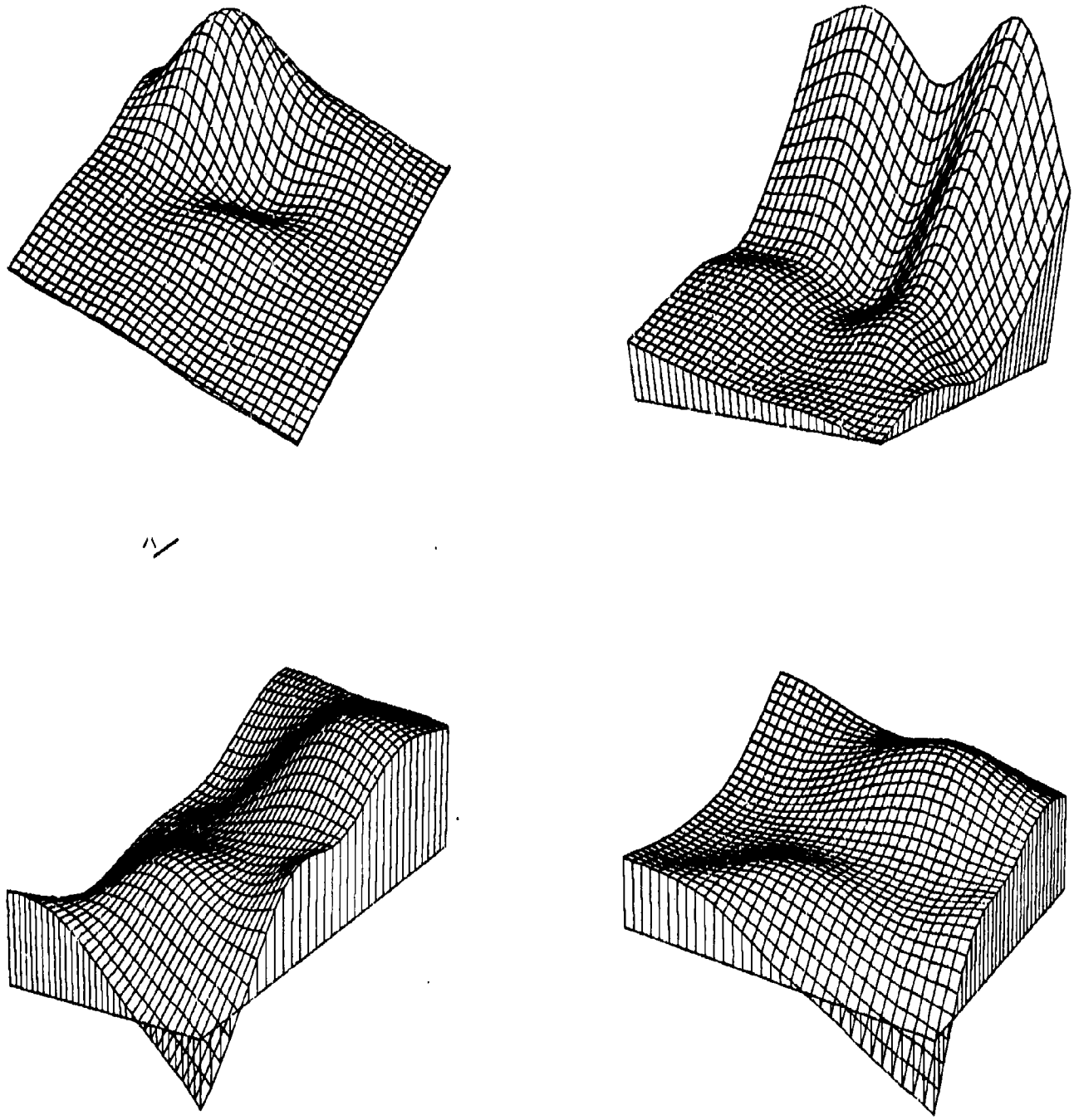
Nielson Minimum Norm Network

Figure 2.0.0.19



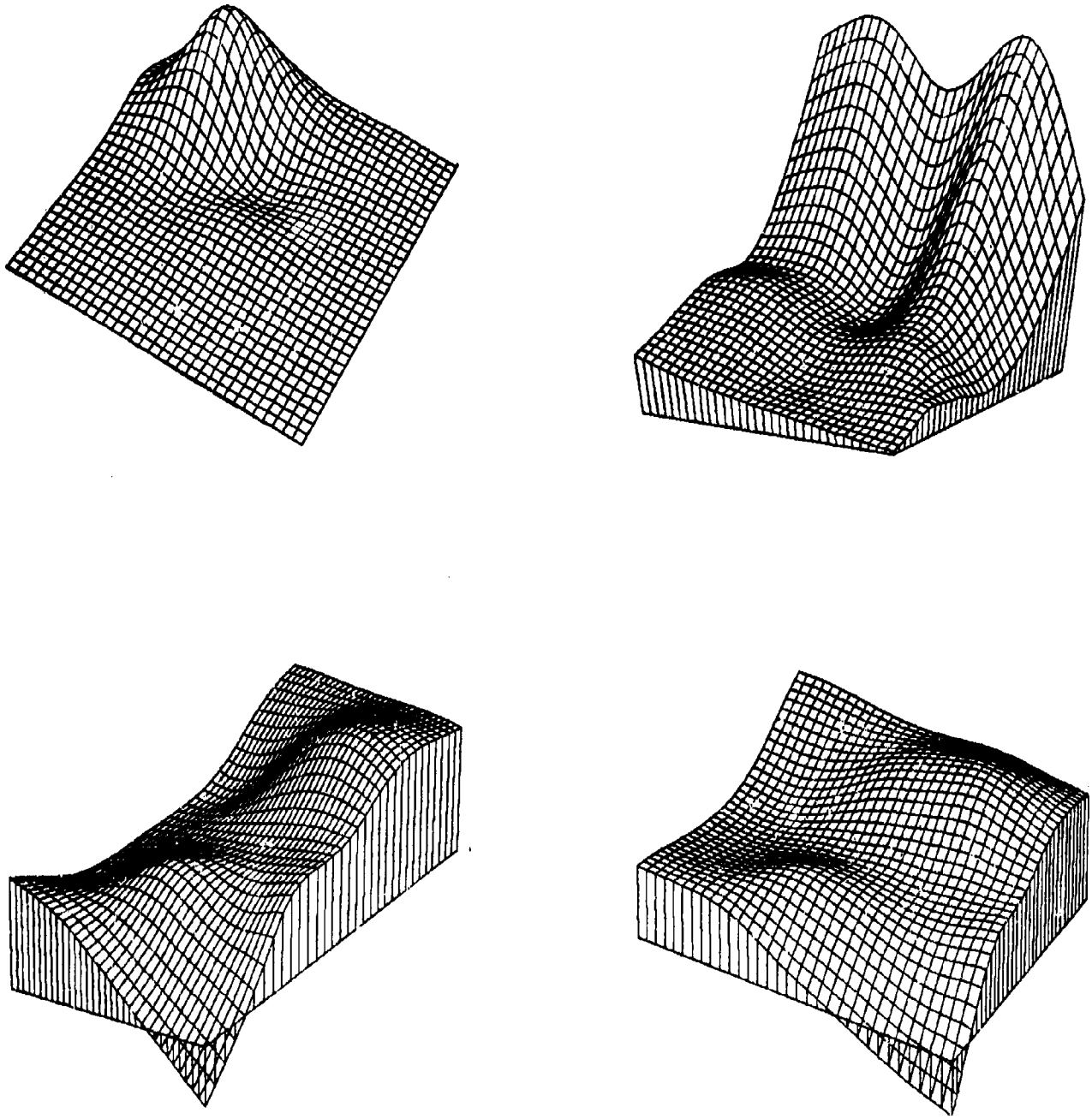
Rotated Gaussian Basis Functions

Figure 2.0.0.20



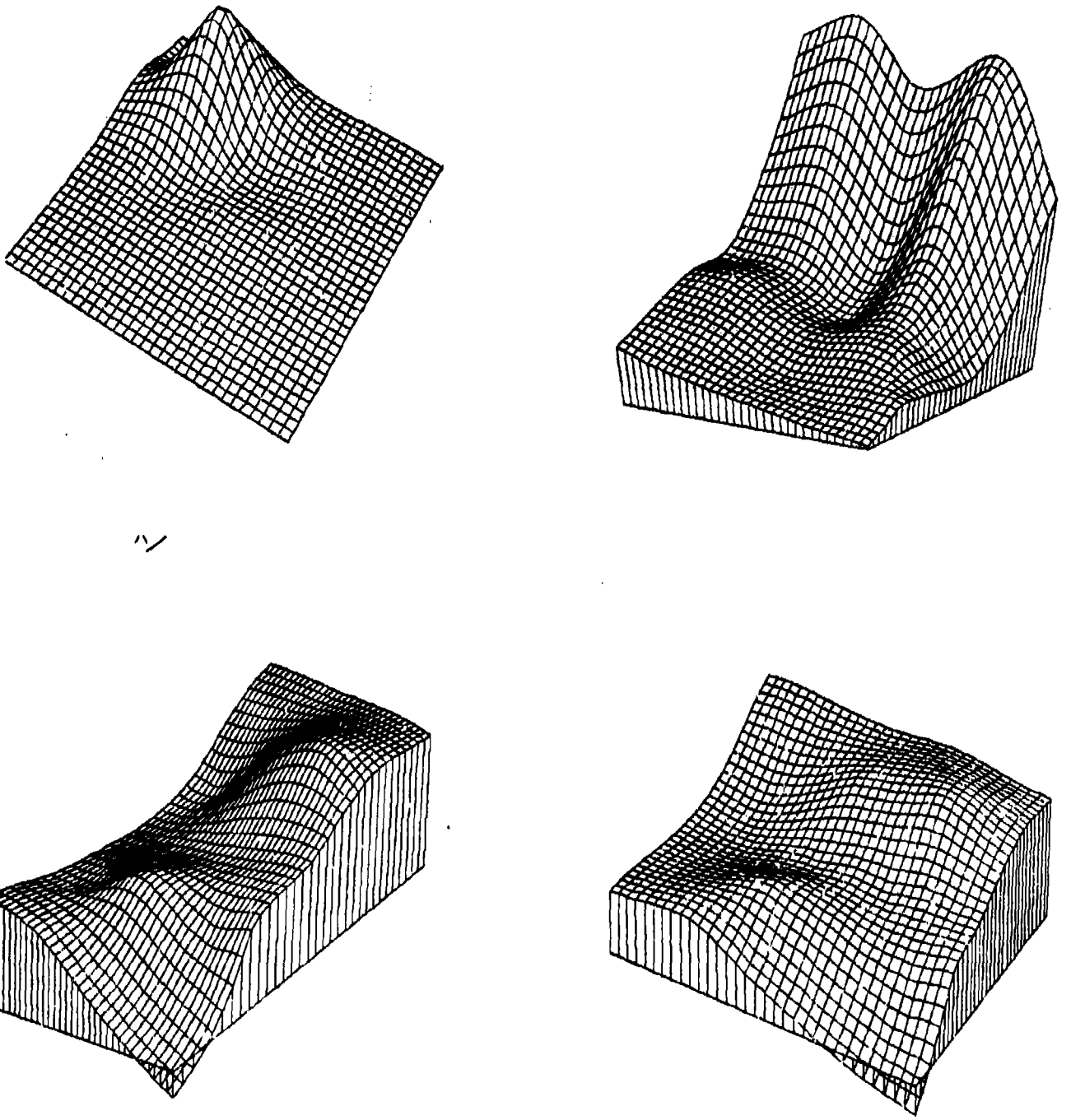
Hardy Multiquadratic Method

Figure 2.0.0.21



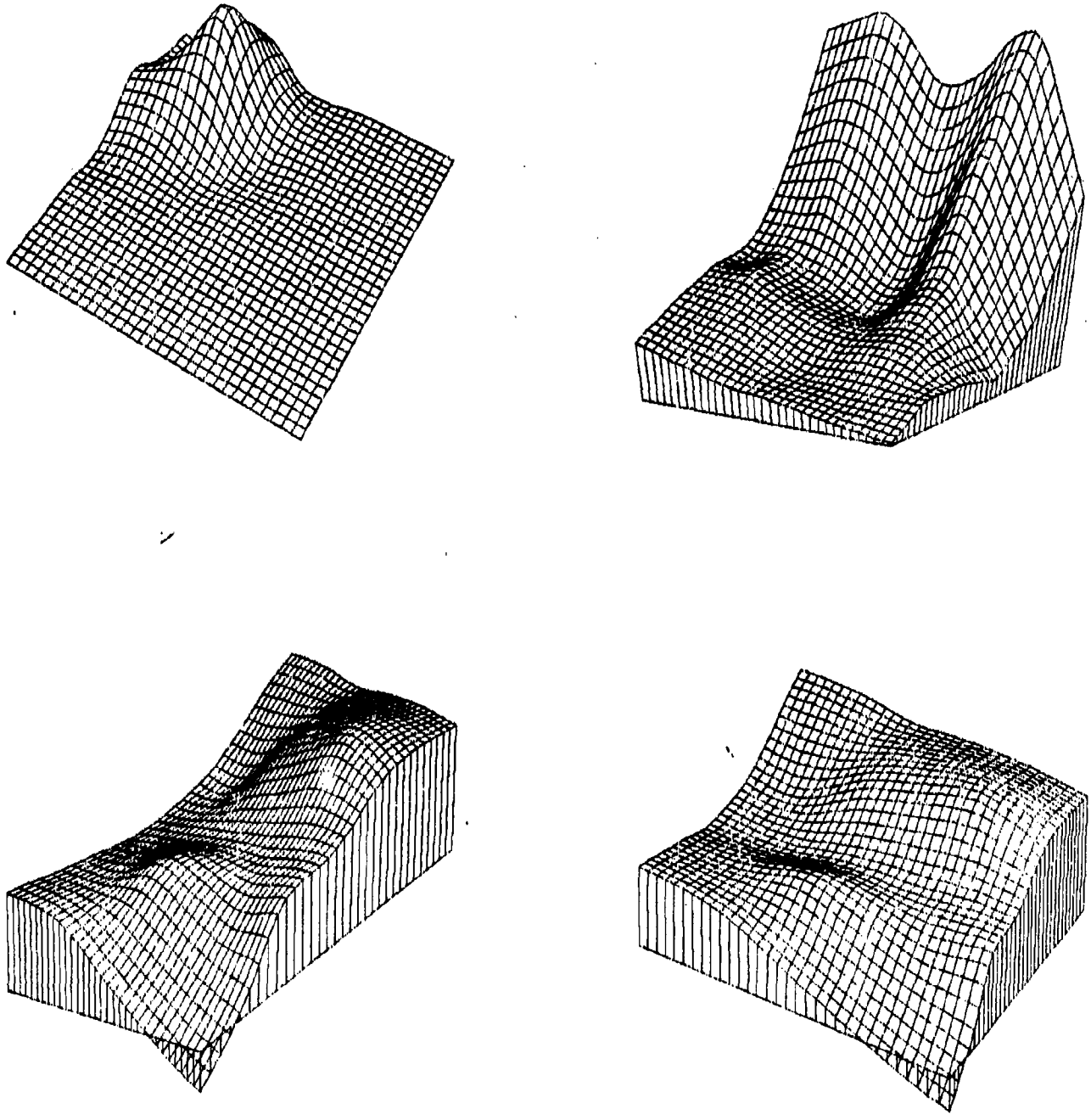
Duchon's Radial Cubics

Figure 2.0.0.22



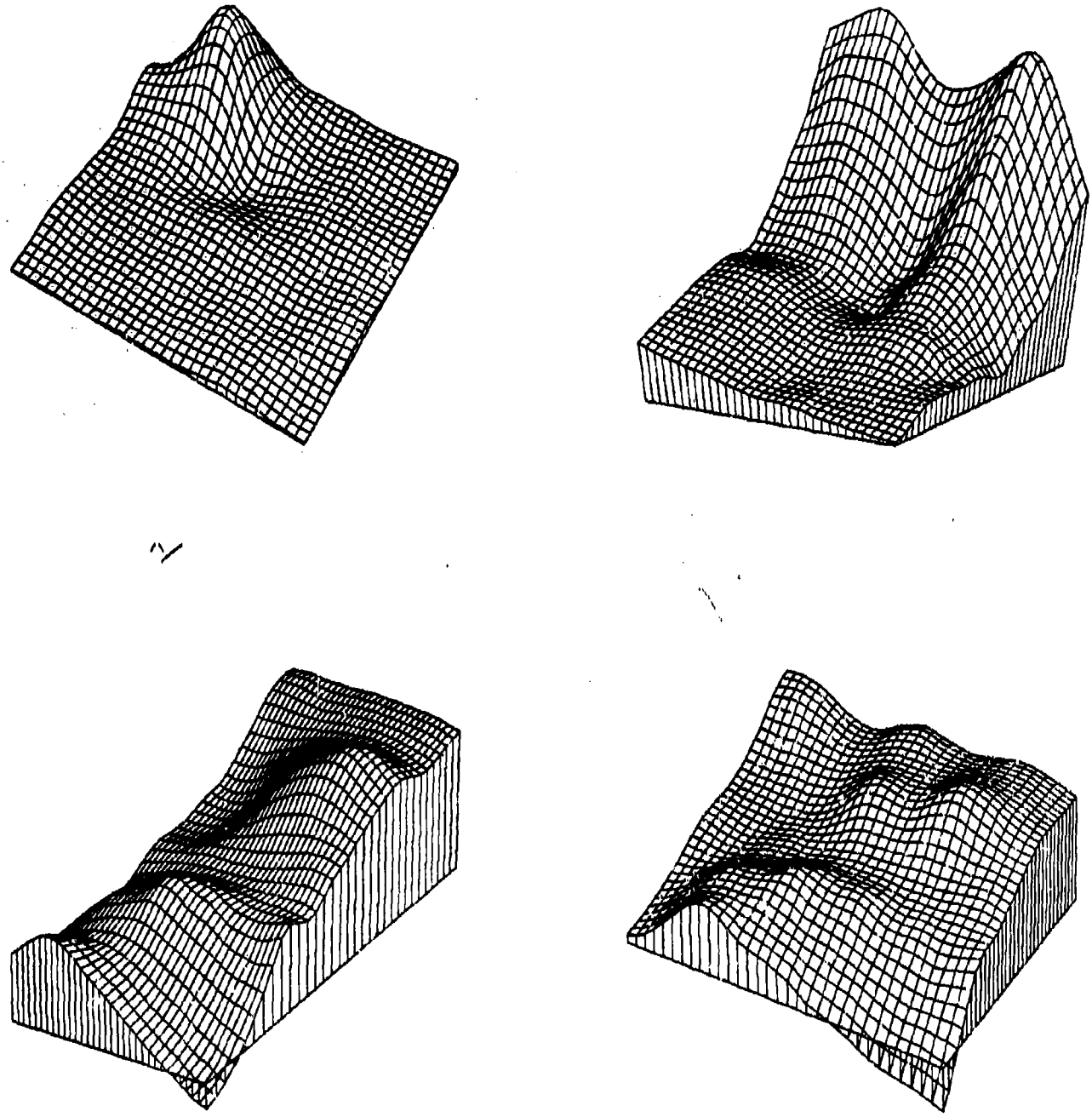
Duchon's Thin Plate Method

Figure 2.0.0.23



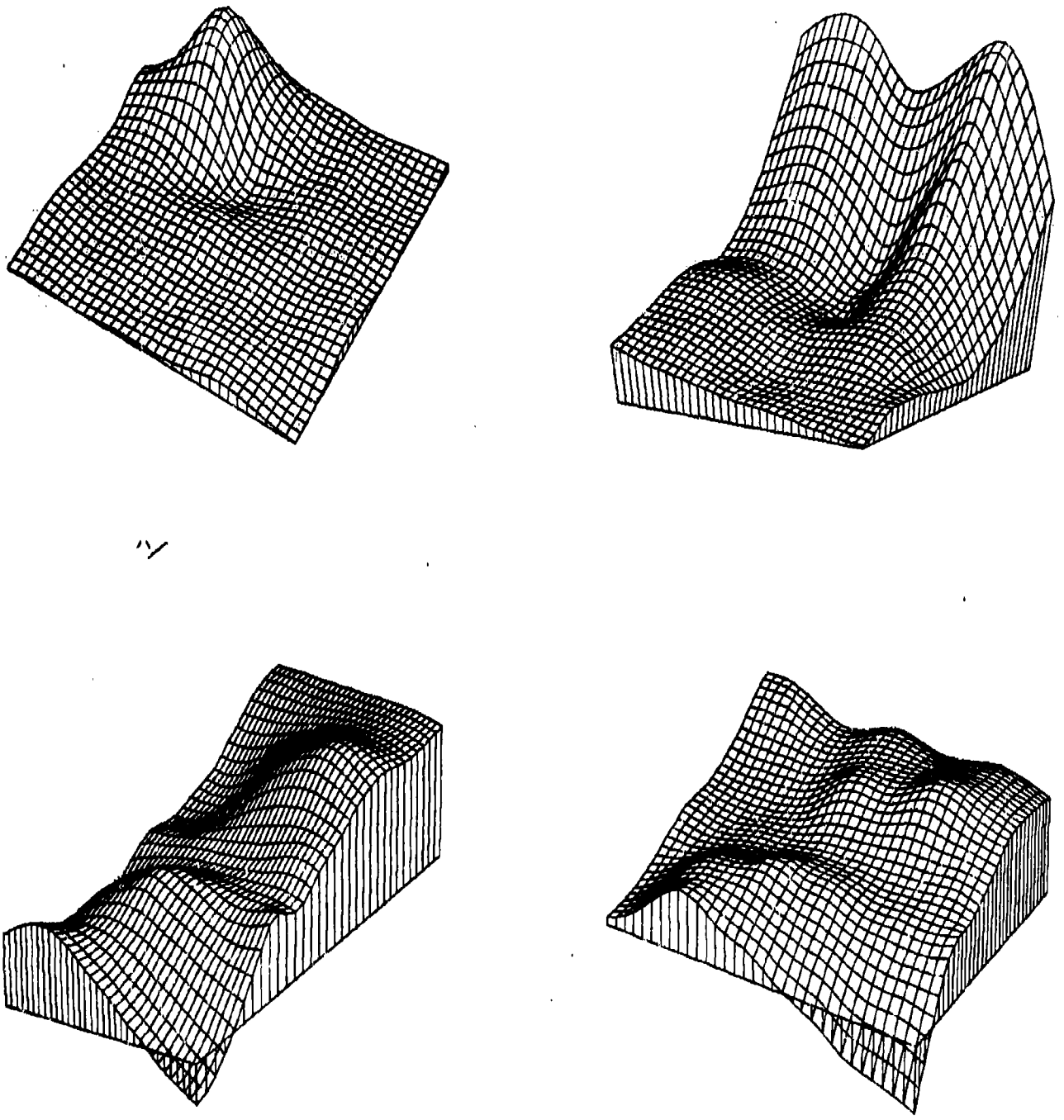
Franke's Method, Thin Plate Local Functions

Figure 2.0.0.24



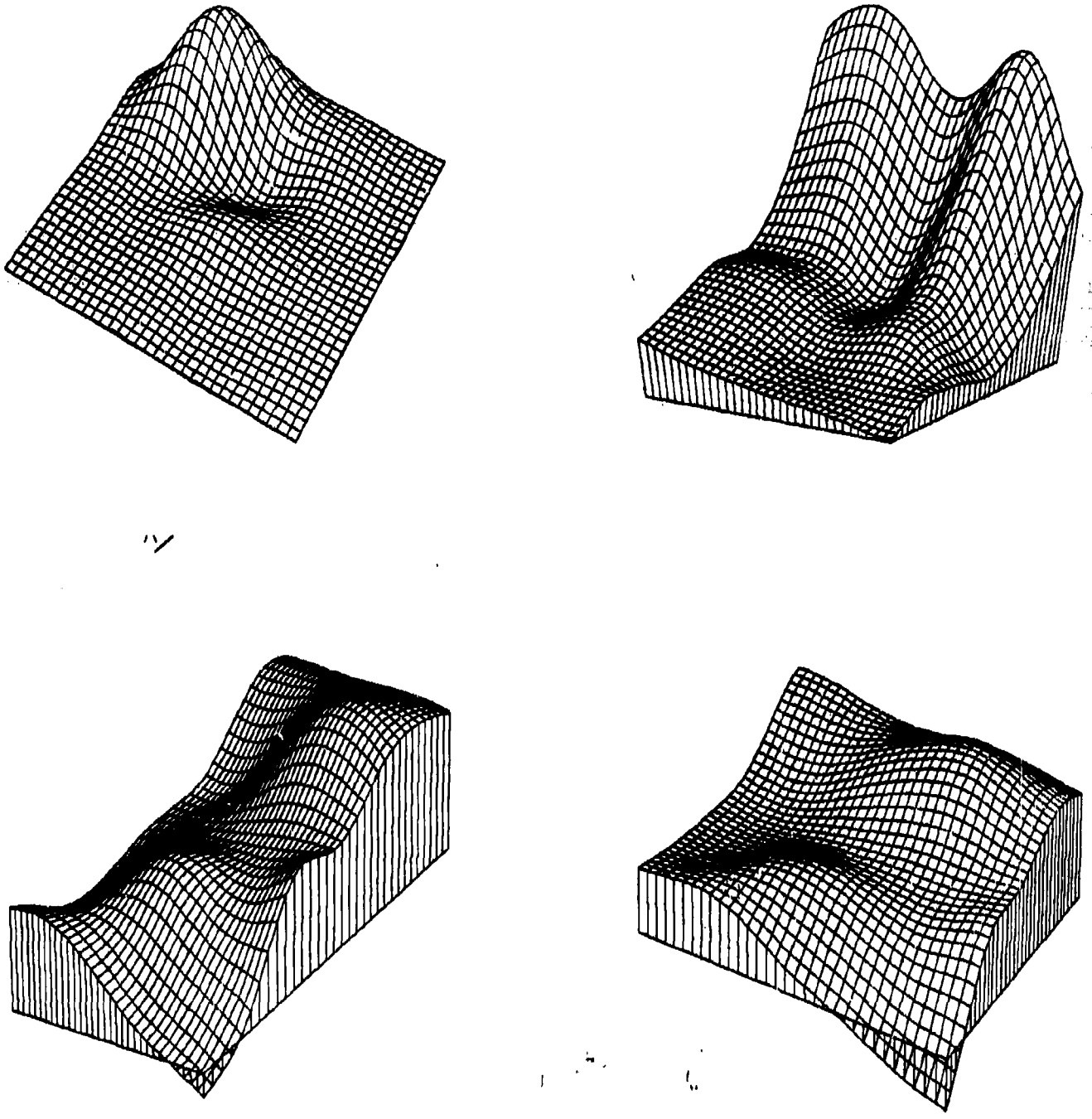
Foley's Generalized Newton Polynomial

Figure 2.0.0.25



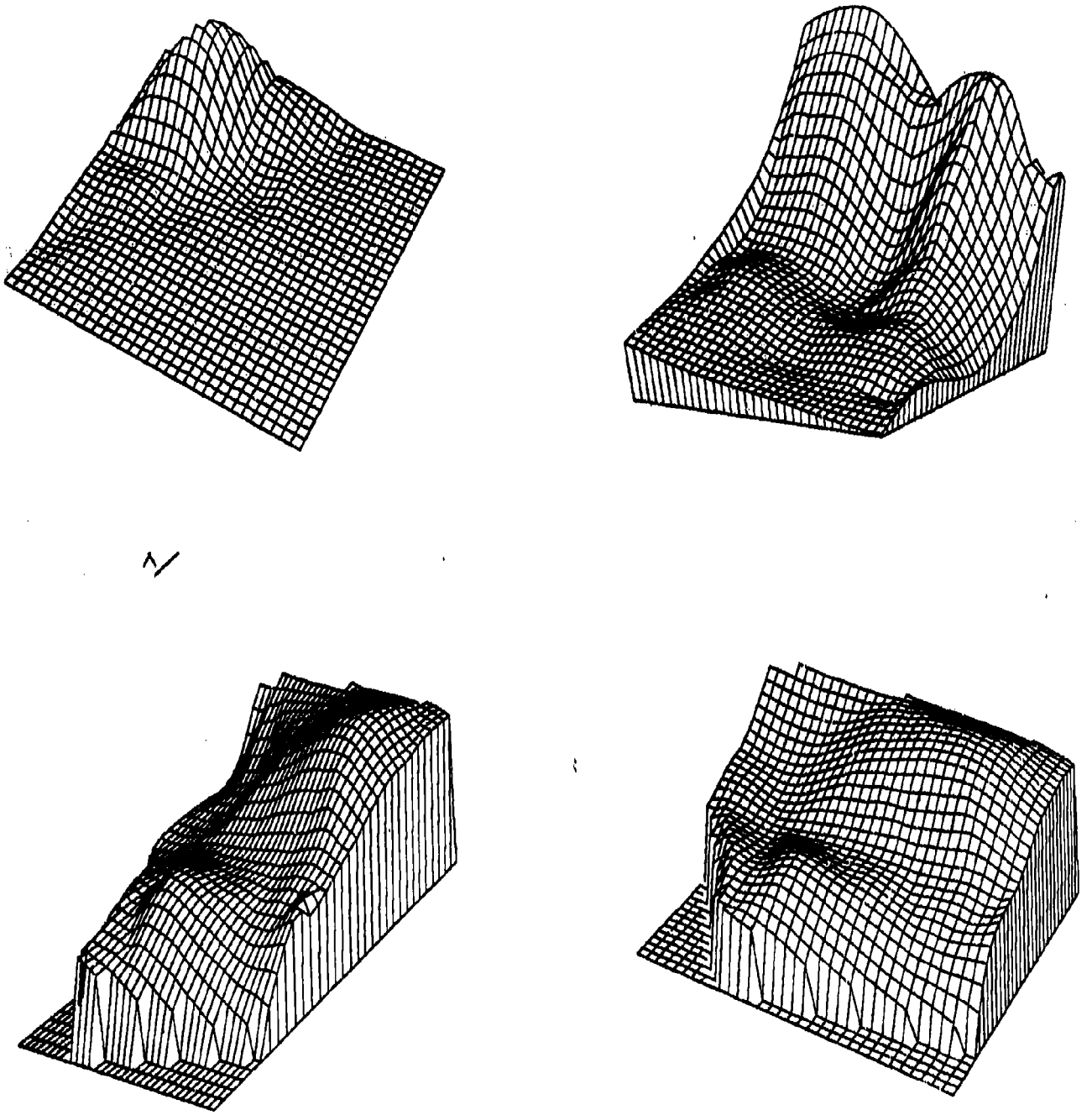
Foley's Generalized Newton Boolean Sum Bernstein - G. N.

Figure 2.0.0.26



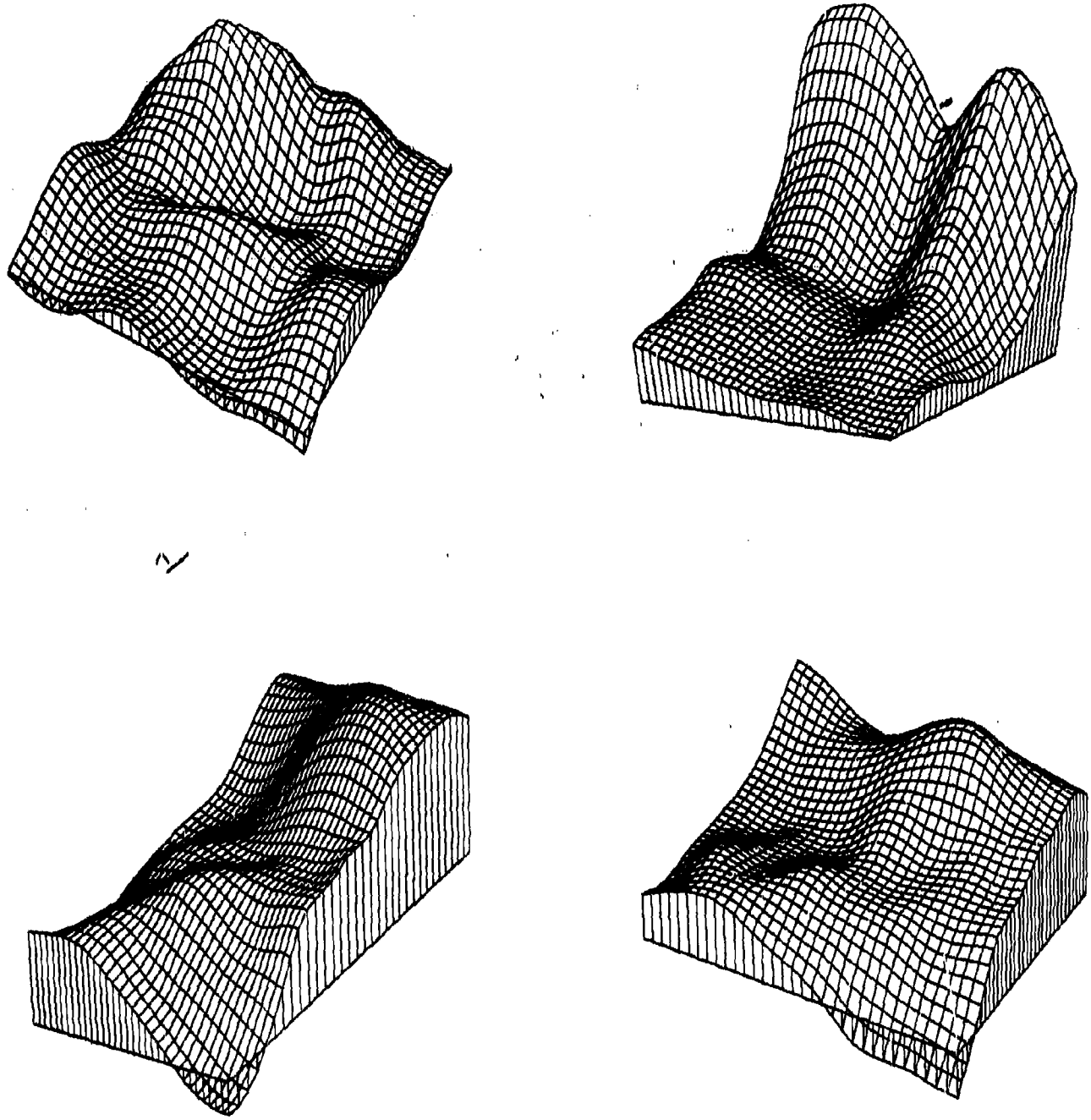
Hardy Reciprocal Multiquadric Method

Figure 2.0.0.27



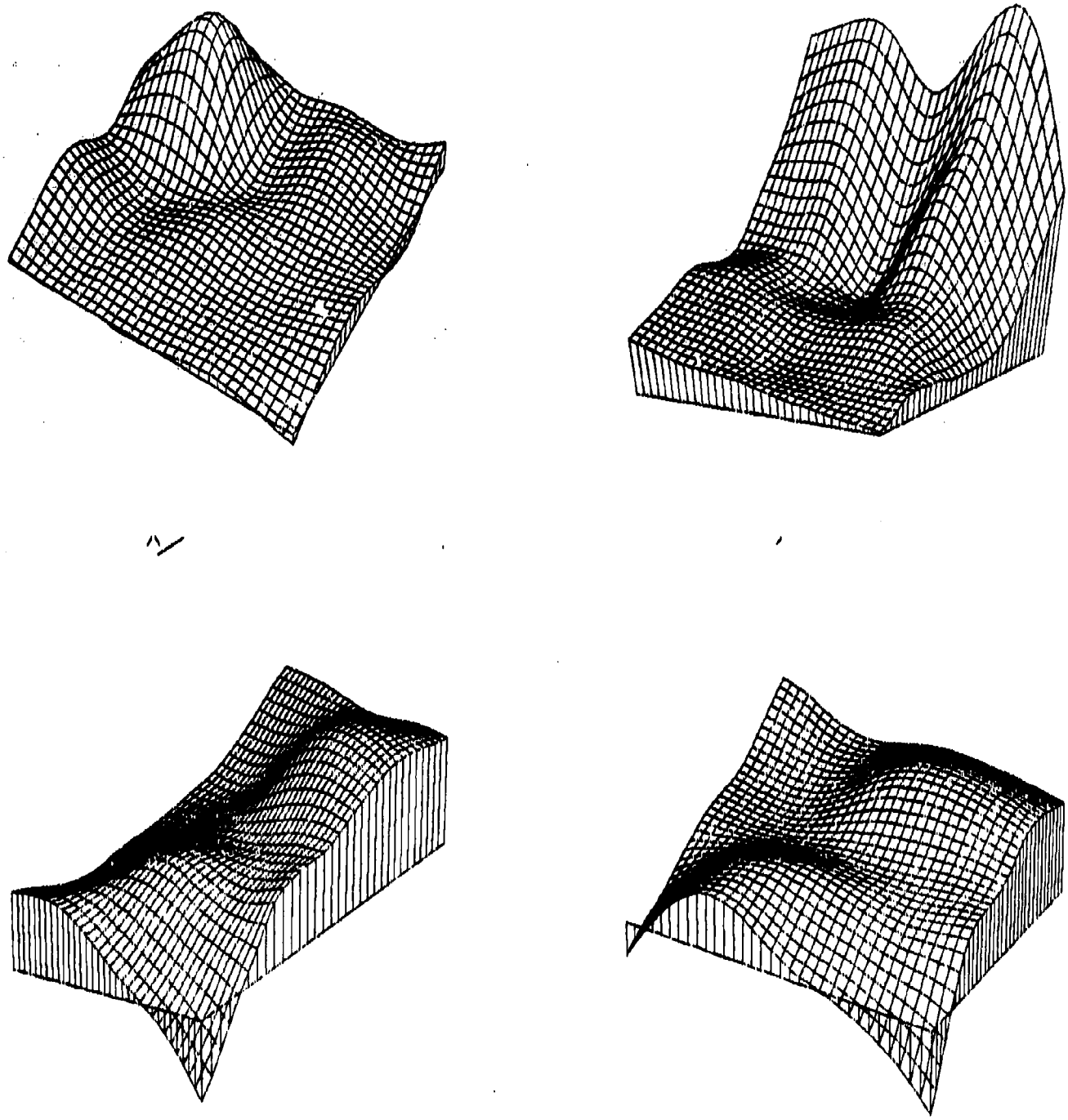
Lawson's Method

Figure 2.0.0.28



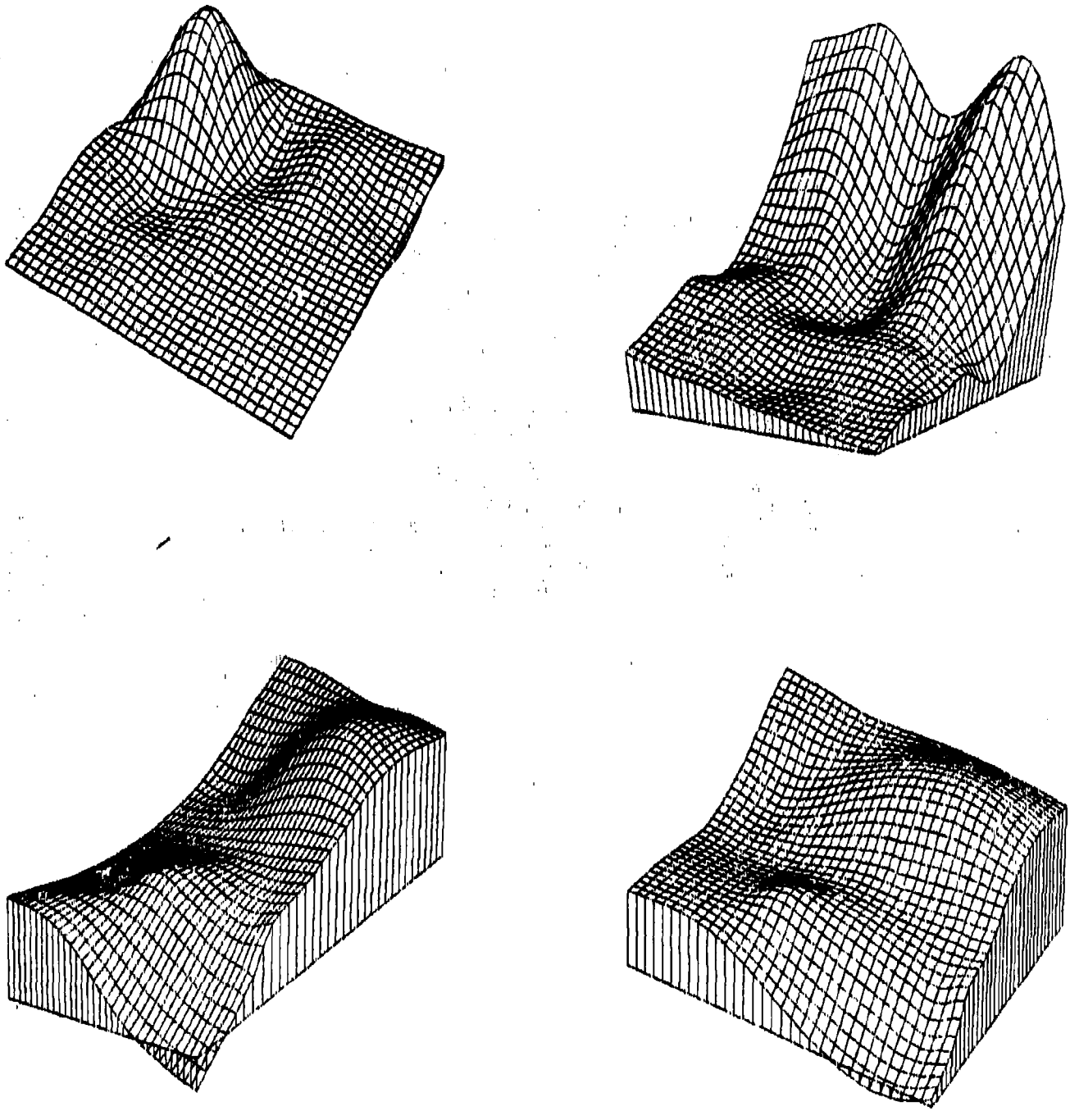
Rotated Cubic B-Spline Basis

Figure 2.0.0.29



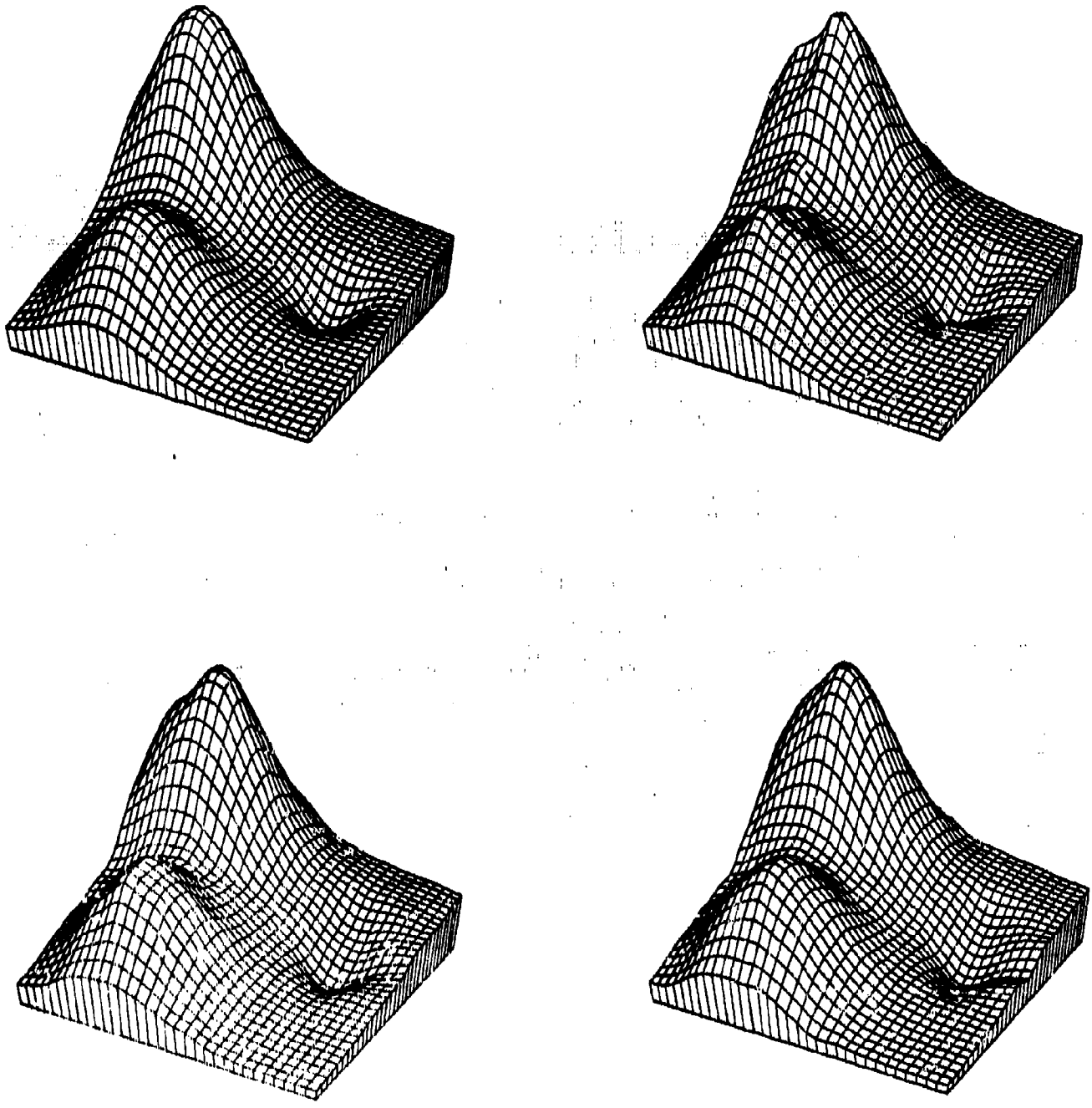
Foley's Iterated Generalized Newton
Delta Sum Bicubic Spline

Figure 2.0.0.30



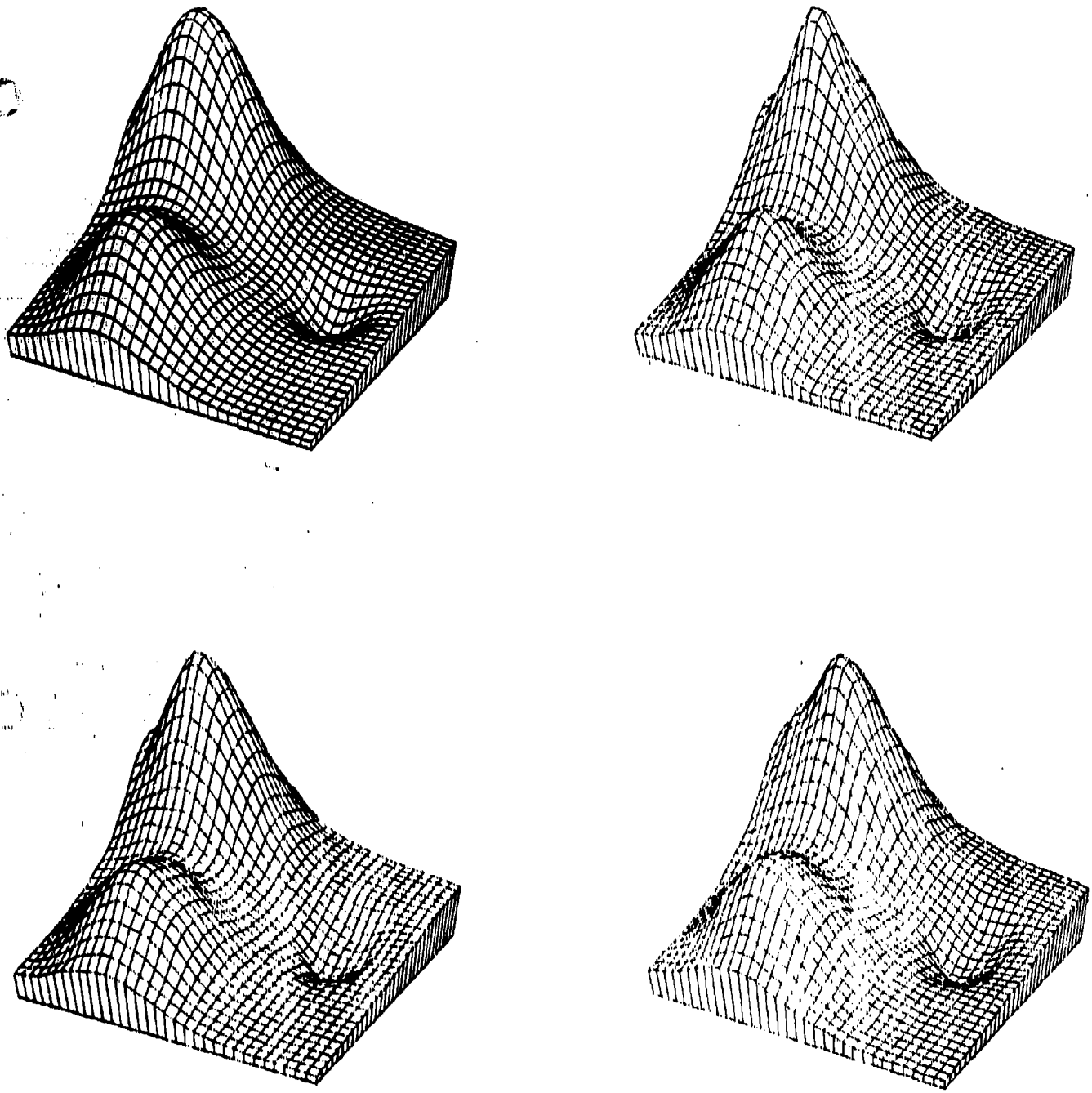
Foley's Iterated Modified Shepard
Delta Sum Bicubic Spline

Figure 2.0.0.31



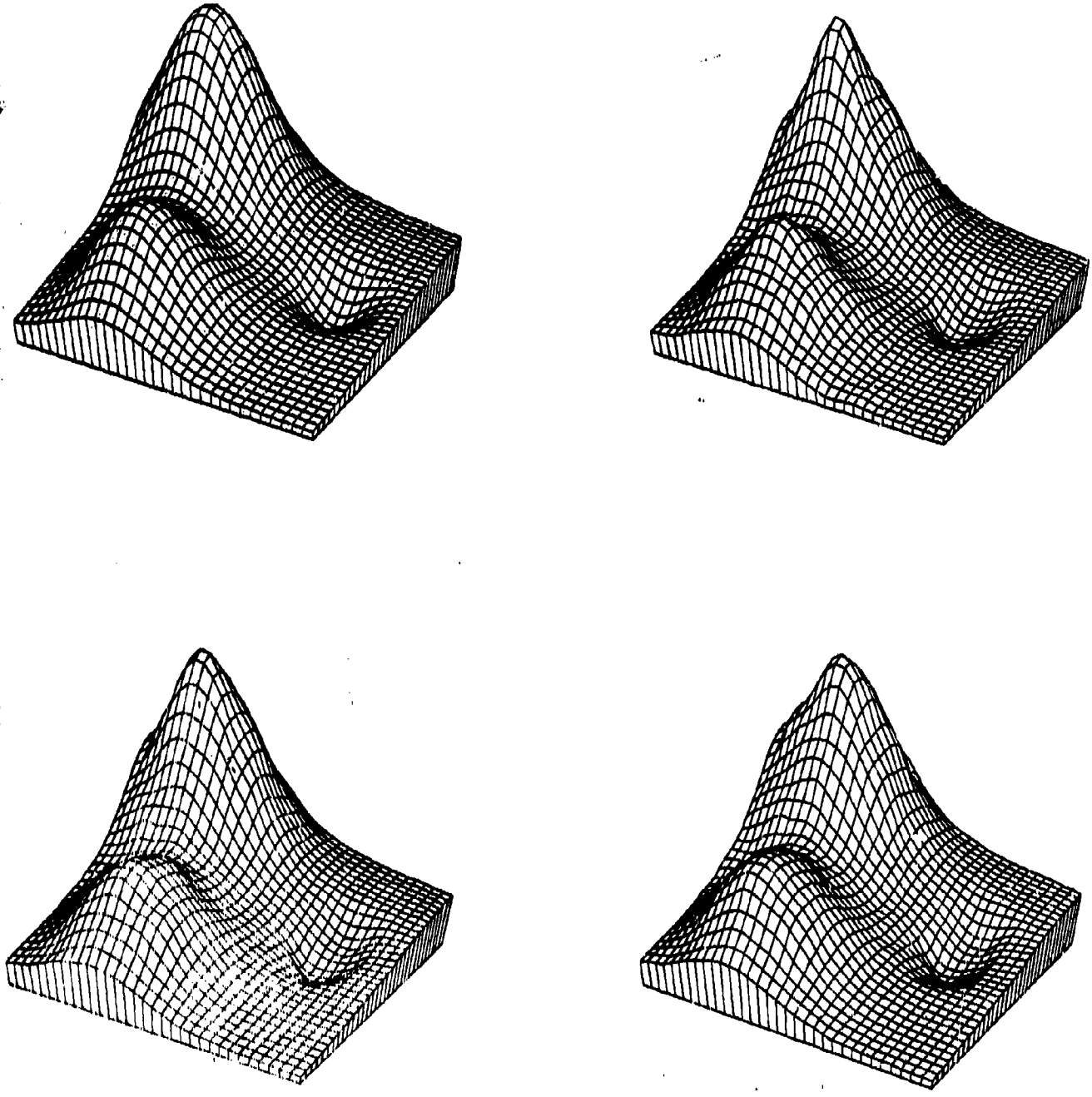
Franke's Method, Mode = 3

Figure 3.1.1.1



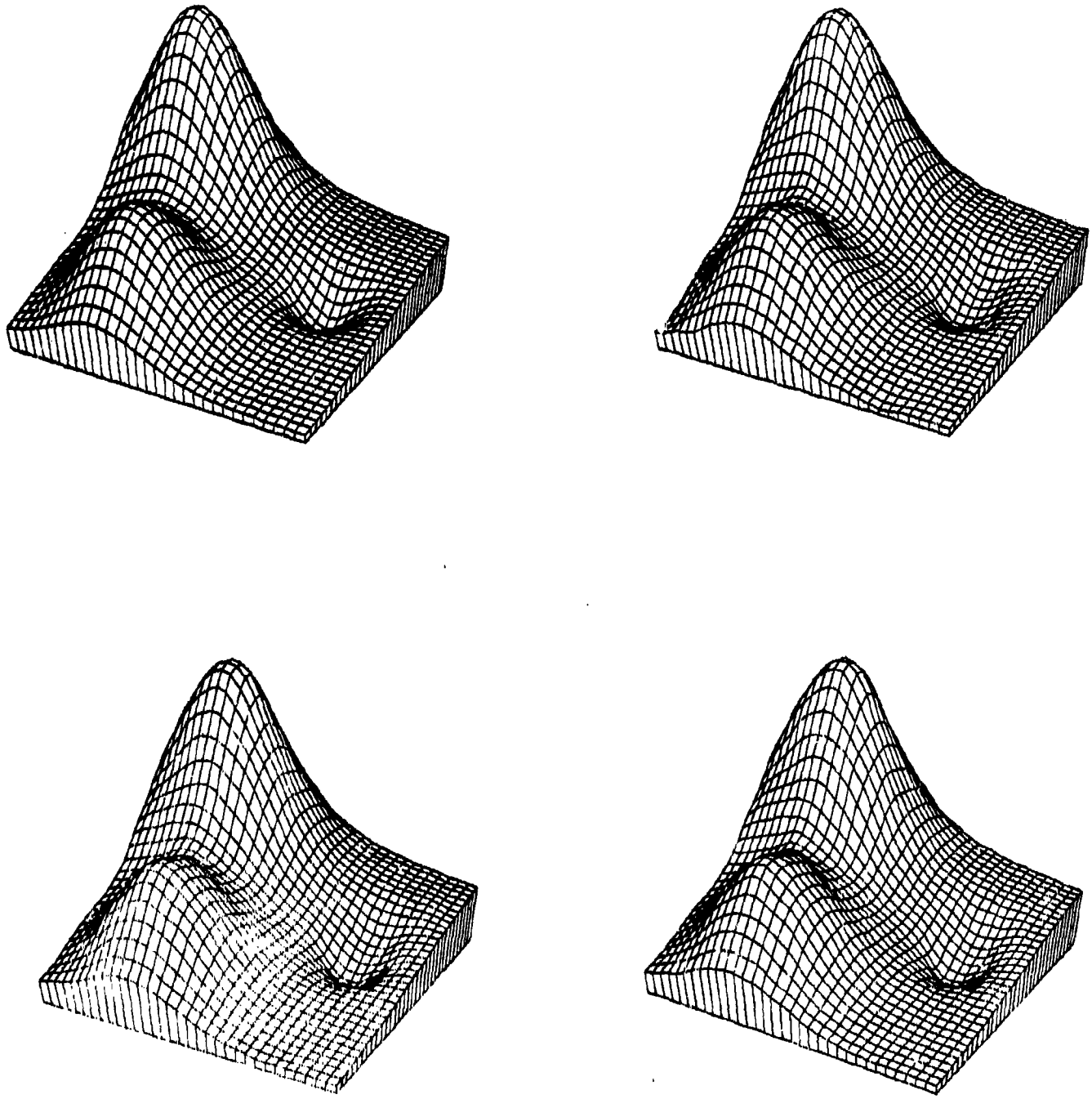
Akima's Method

Figure 3.1.1.4



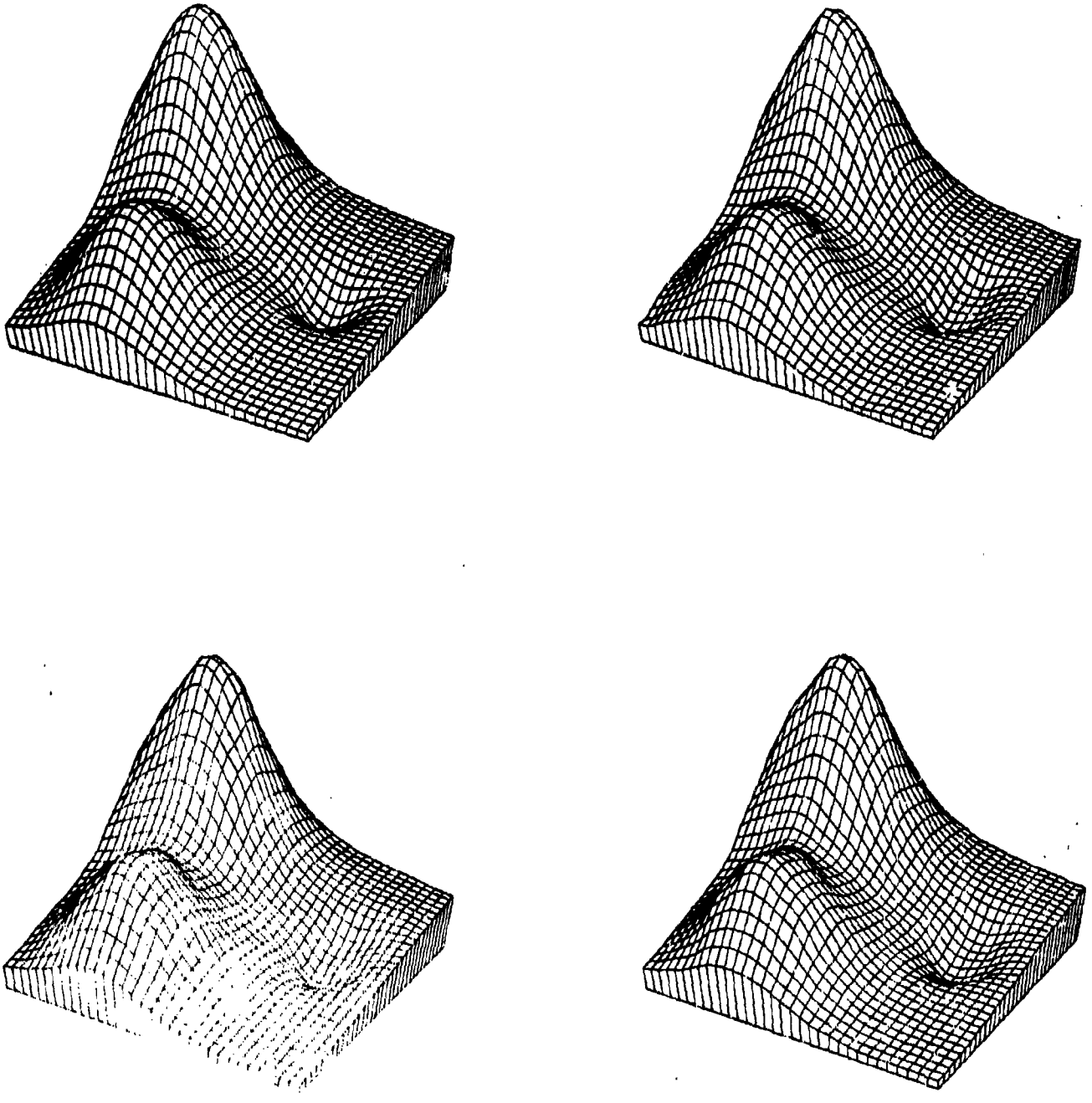
Akima's Method, Modification One

Figure 3.1.1.10



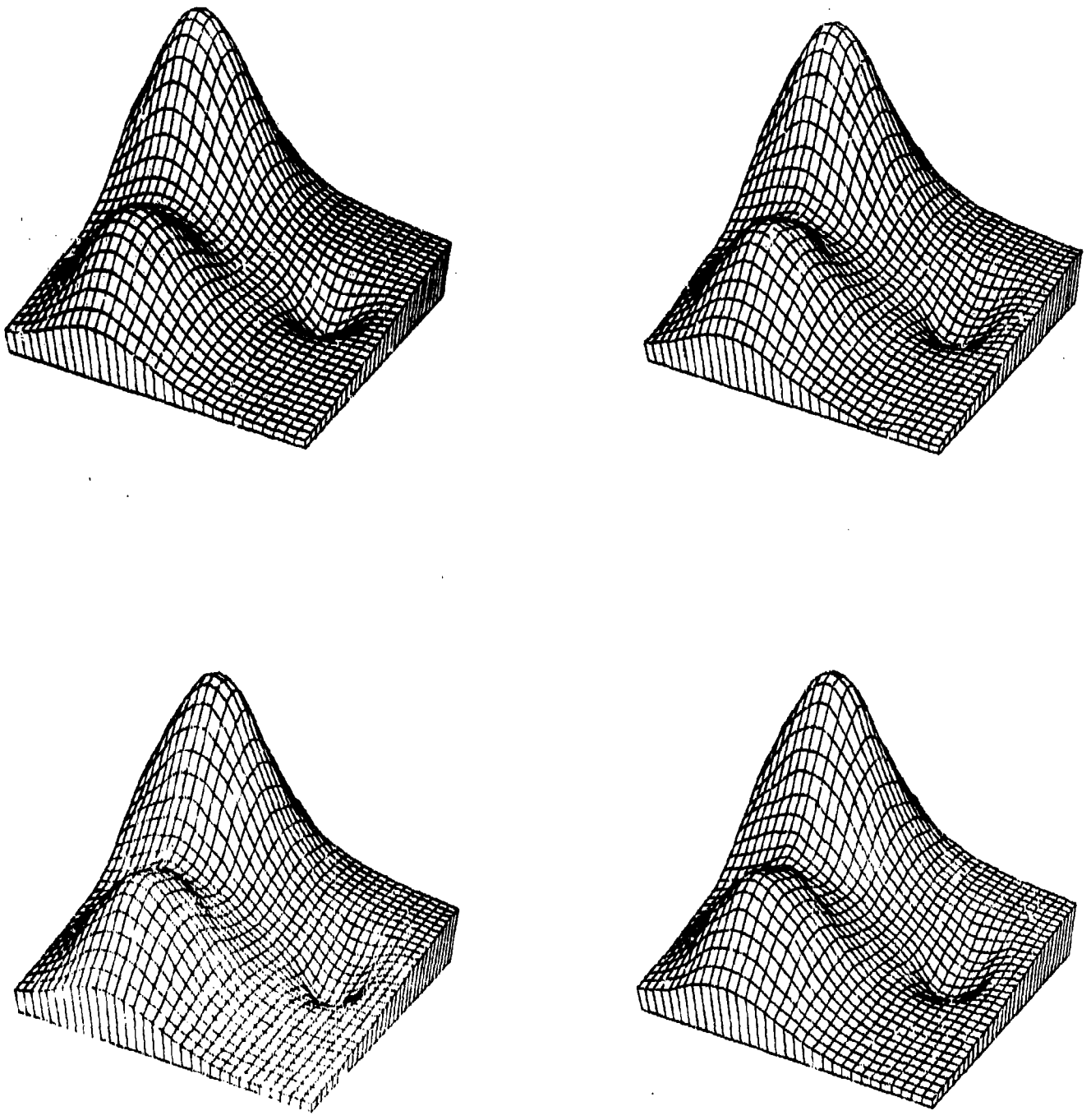
Nielson-Franke Quadratic Triangular Method

Figure 3.1.1.13



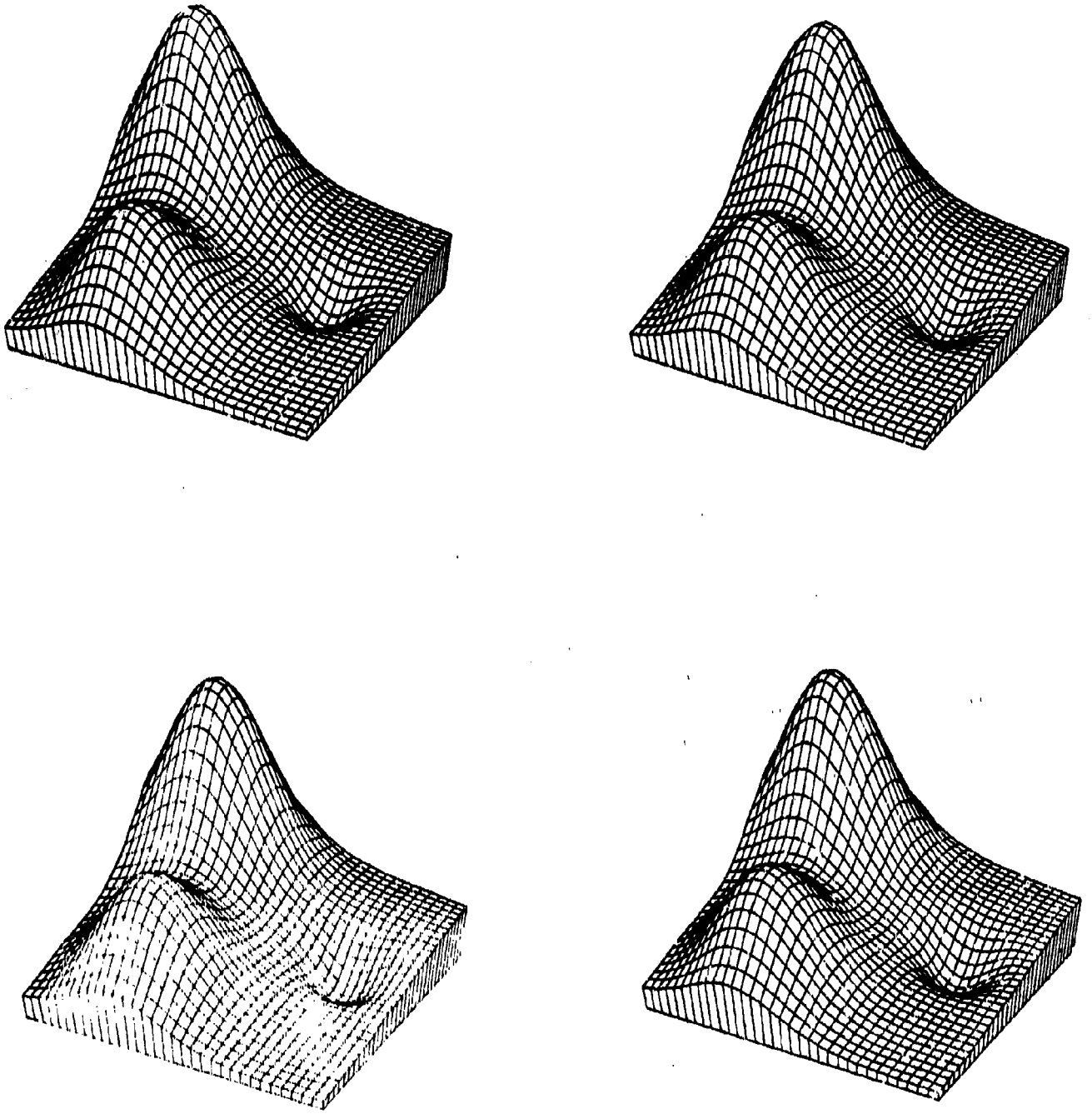
Modified Quadratic Shepard Method

Figure 3.1.1.14



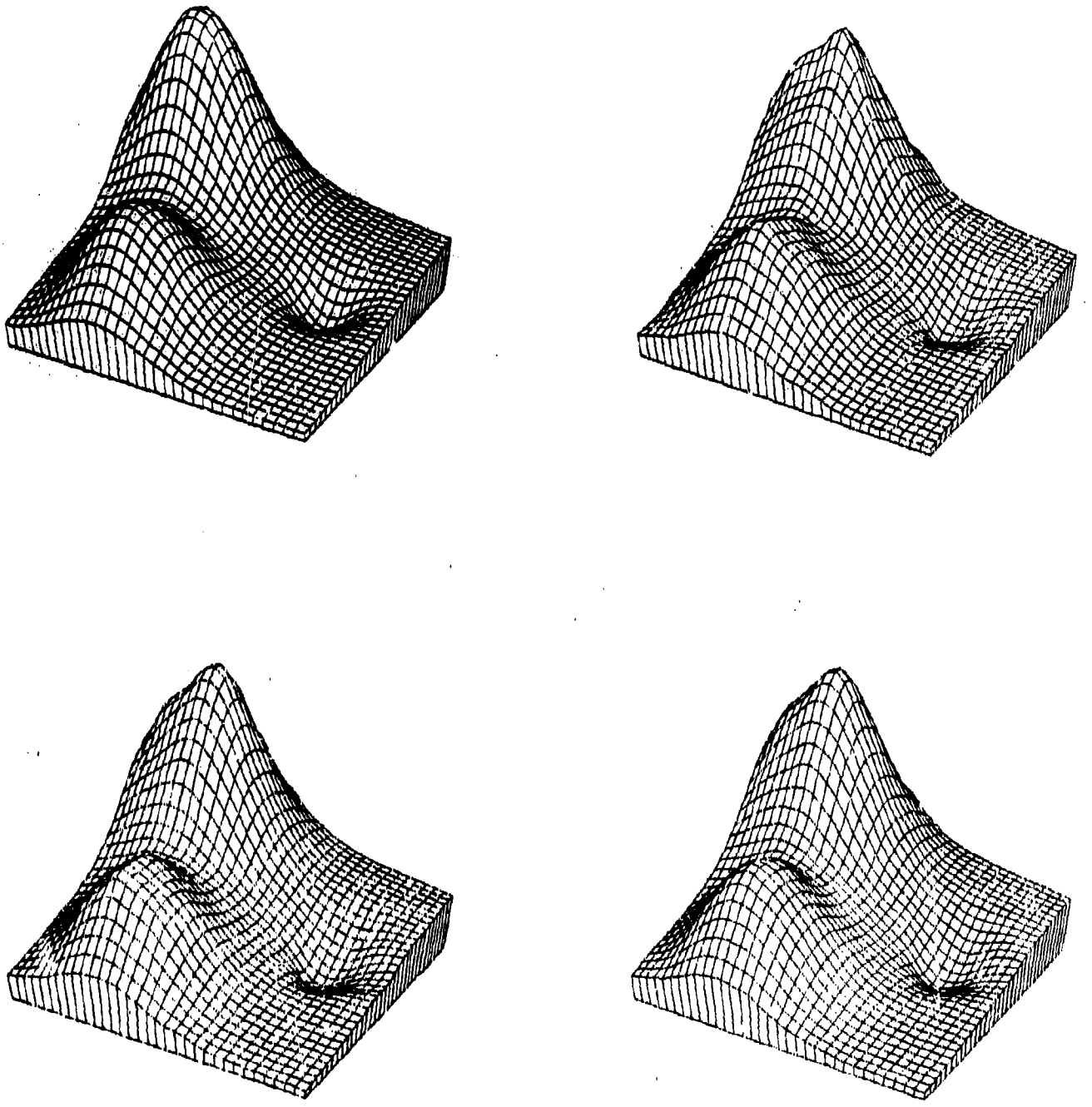
Akima's Method, Modification Three

Figure 3.1.1.16



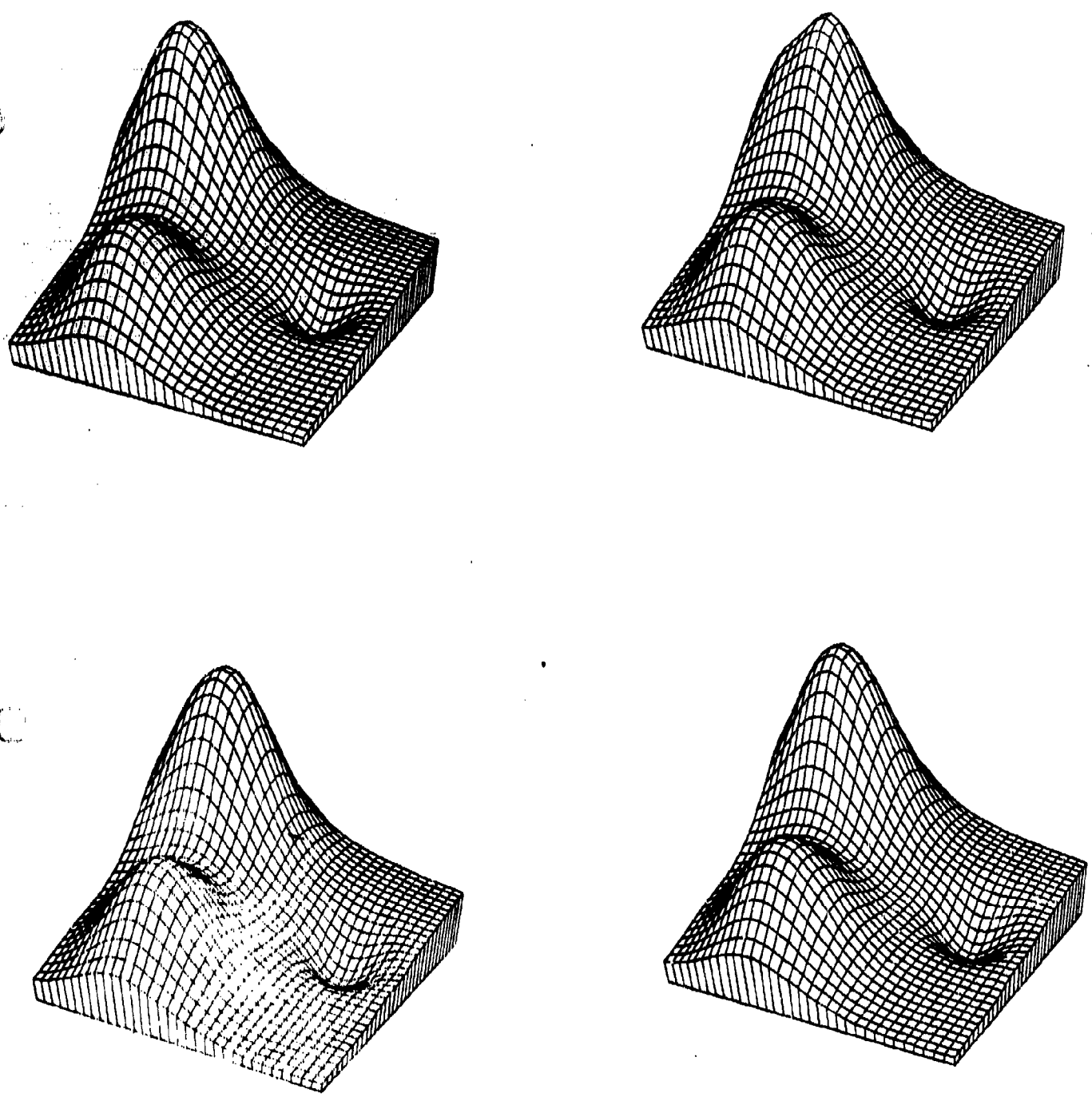
Hardy Multiquadric Method

Figure 3.1.1.21



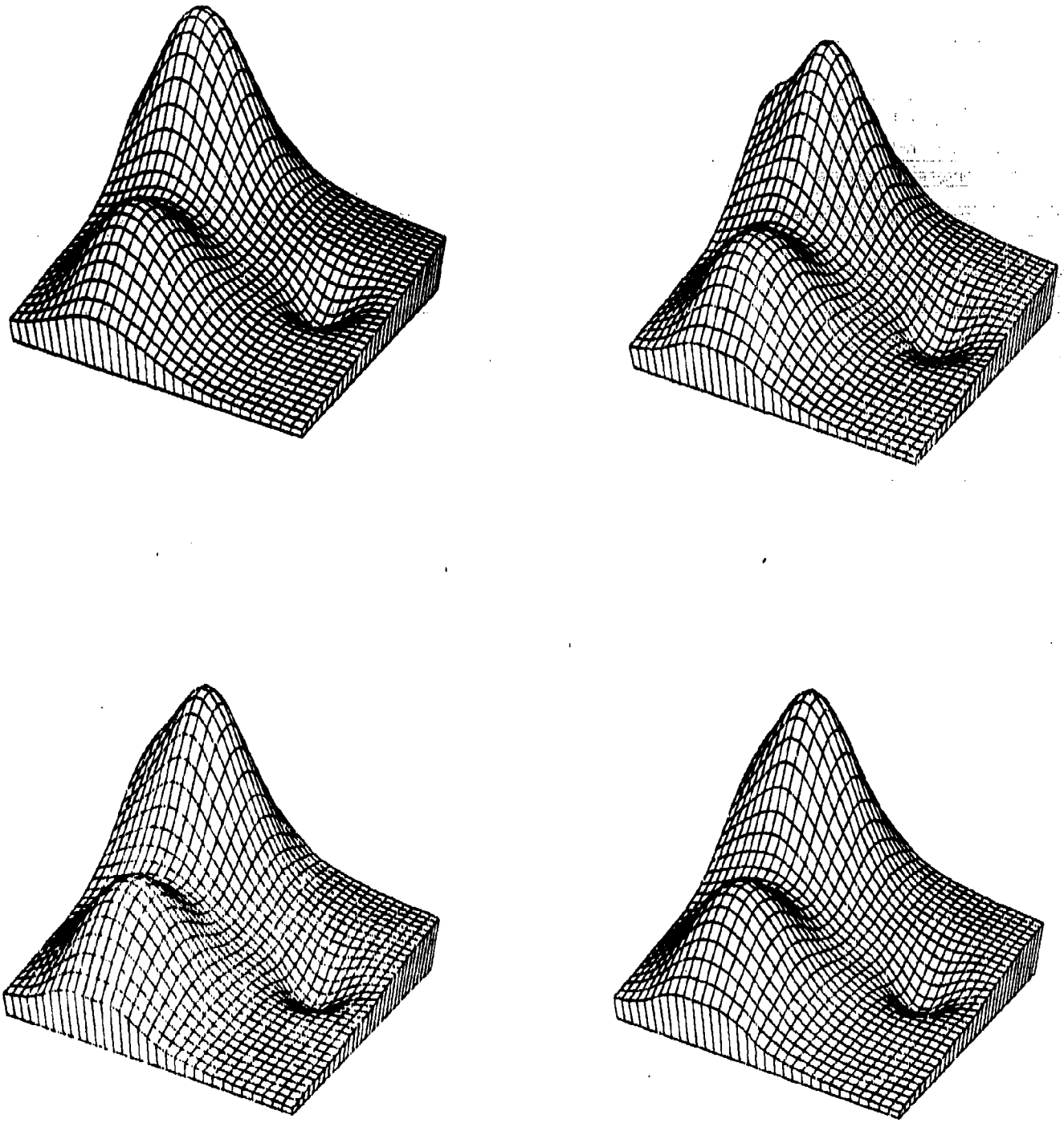
Franke's Method, Thin Plate Local Functions

Figure 3.1.1.24



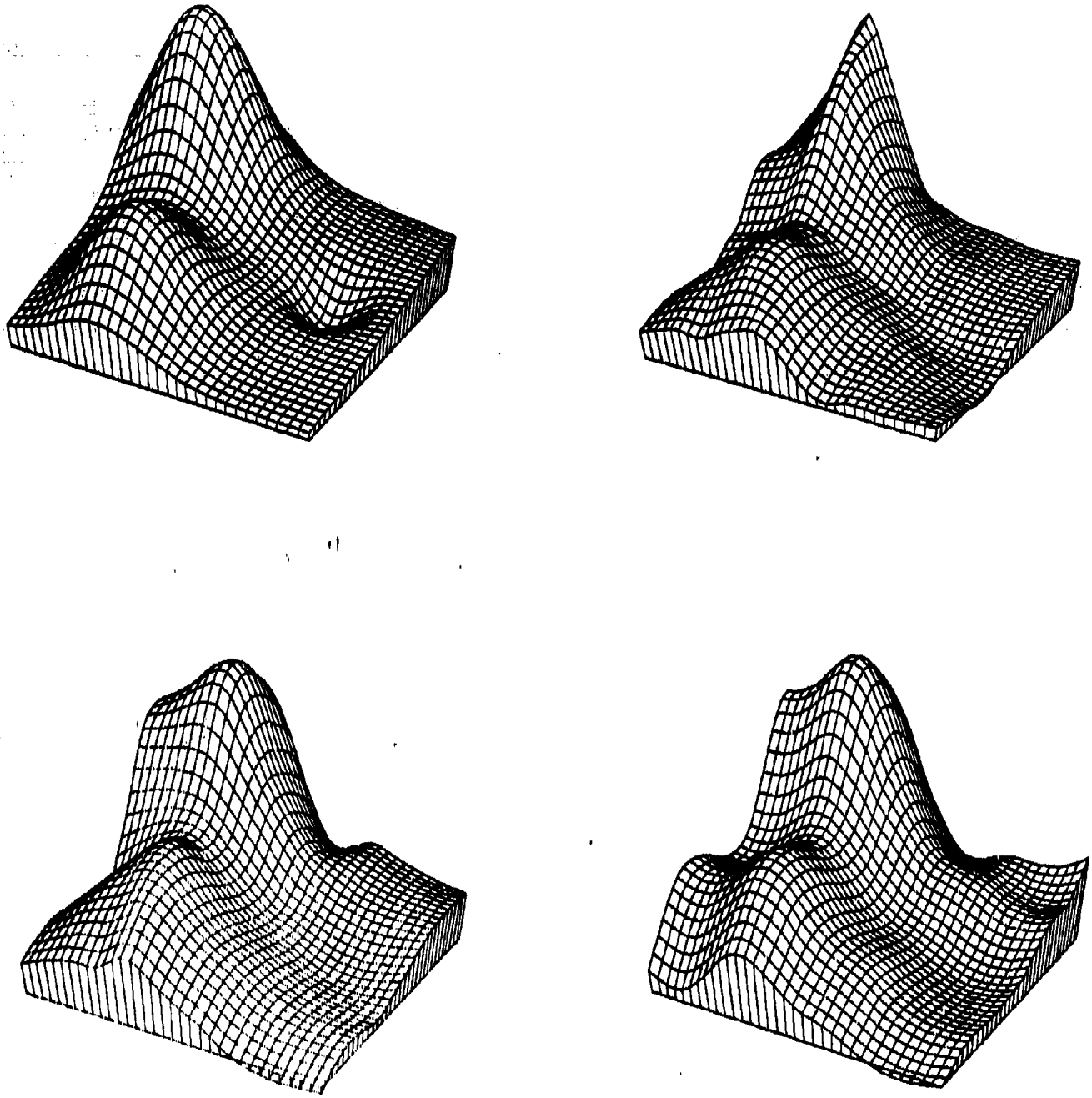
Hardy Reciprocal Multiquadric Method

Figure 3.1.1.27



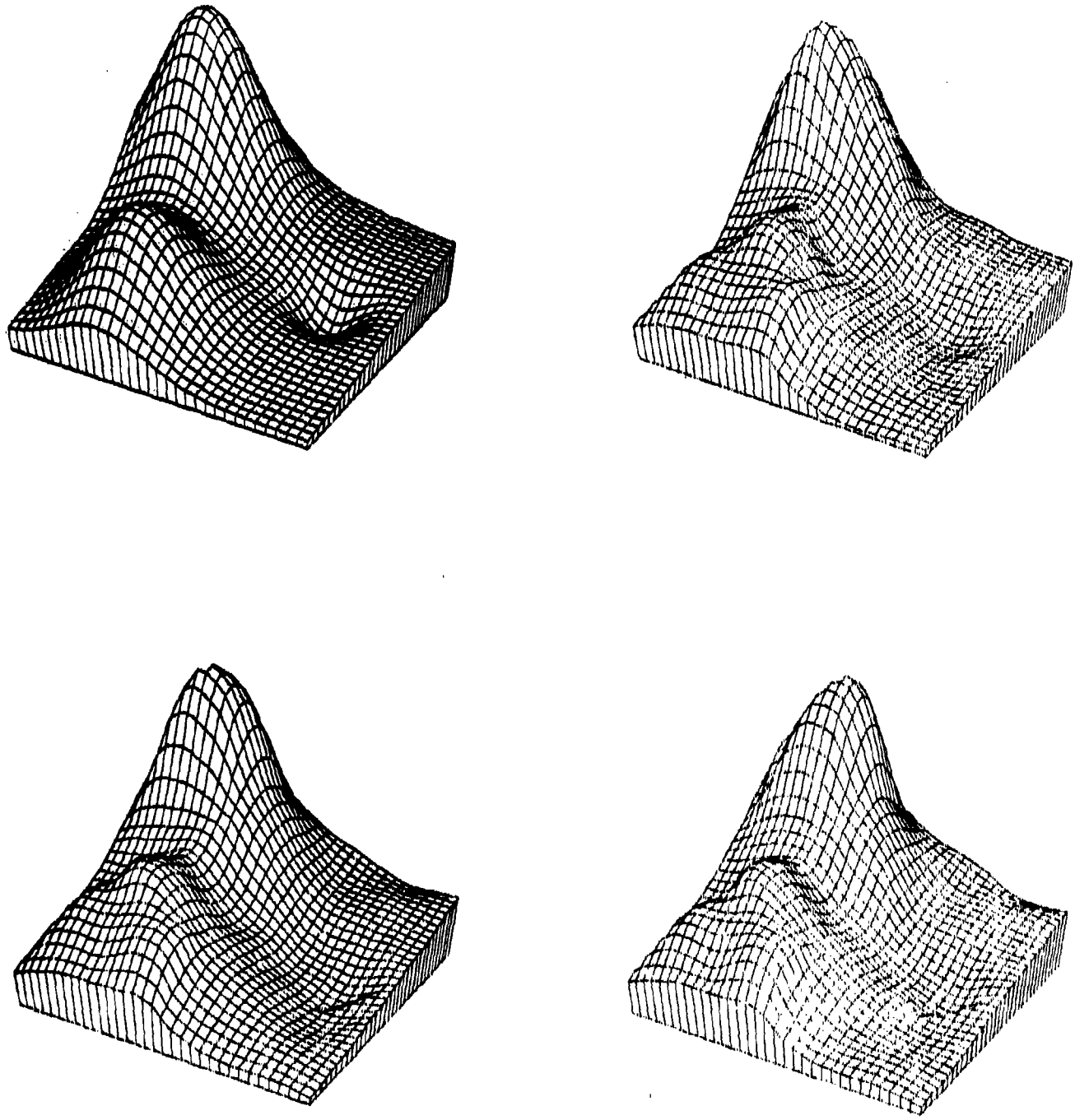
Foley's Iterated Generalized Newton
Delta Sum Bicubic Spline

Figure 3.1.1.30



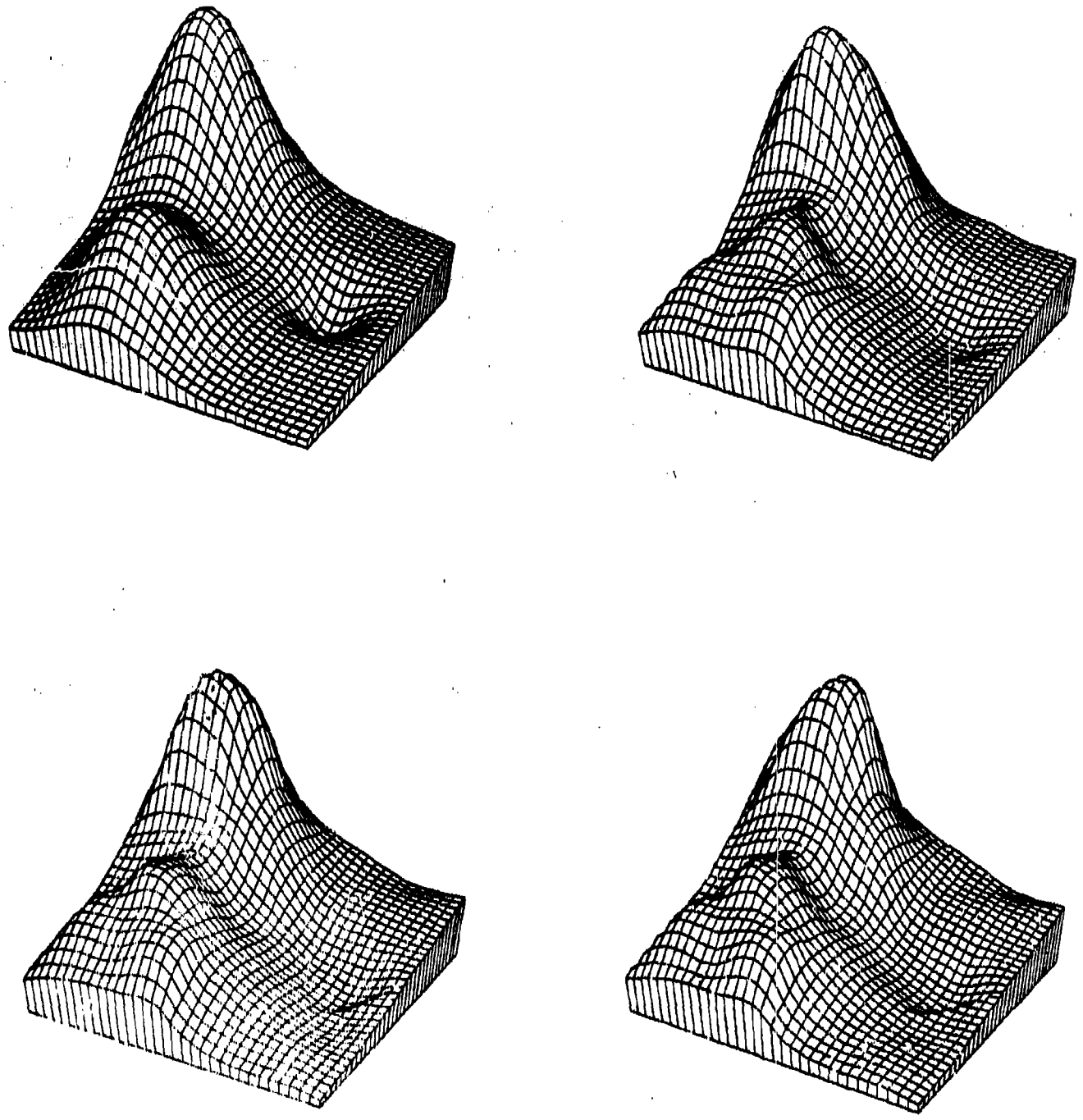
Franke's Method, Mode = 3

Figure 3.3.1.1



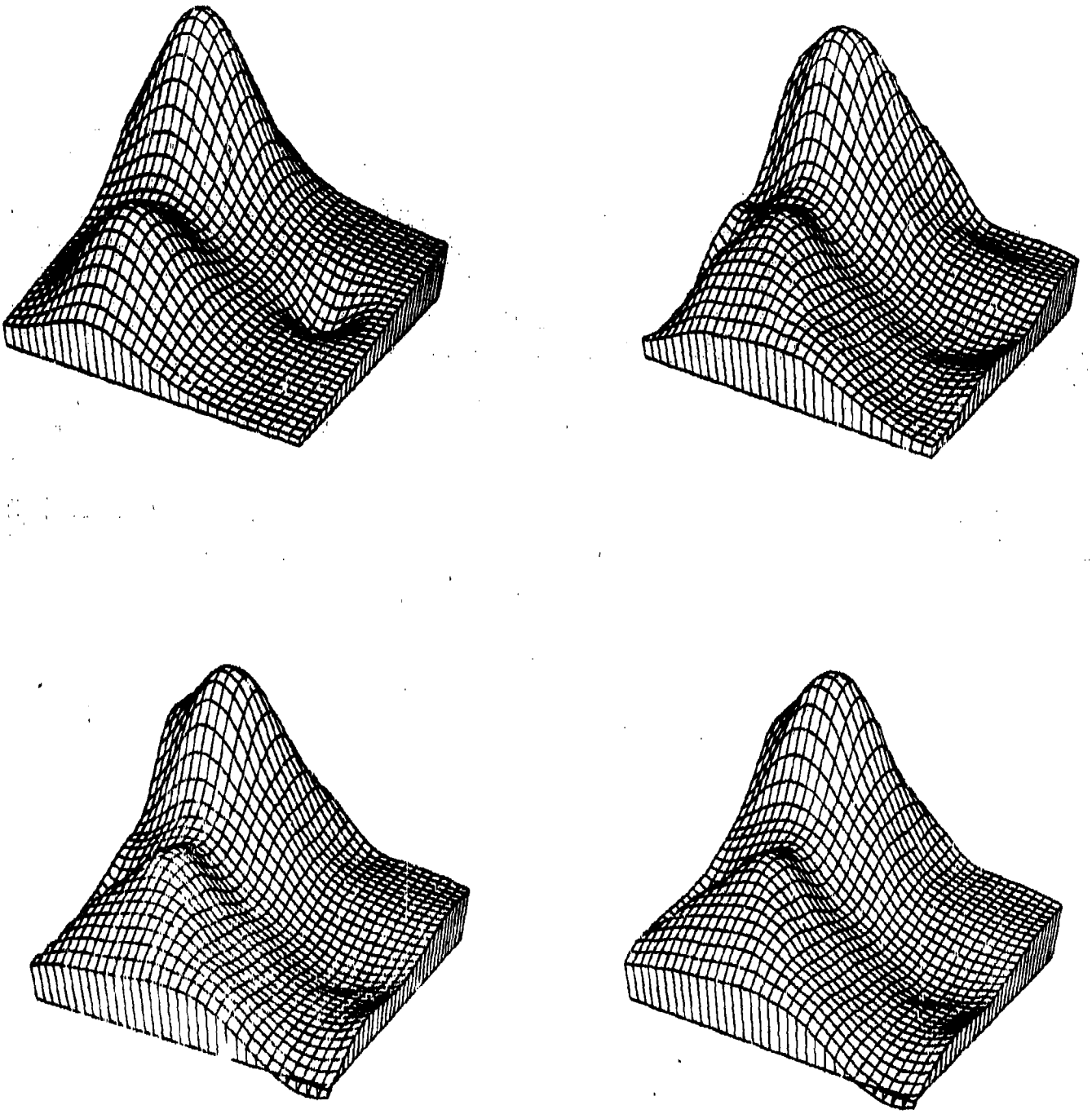
Akima's Method

Figure 3.3.1.4



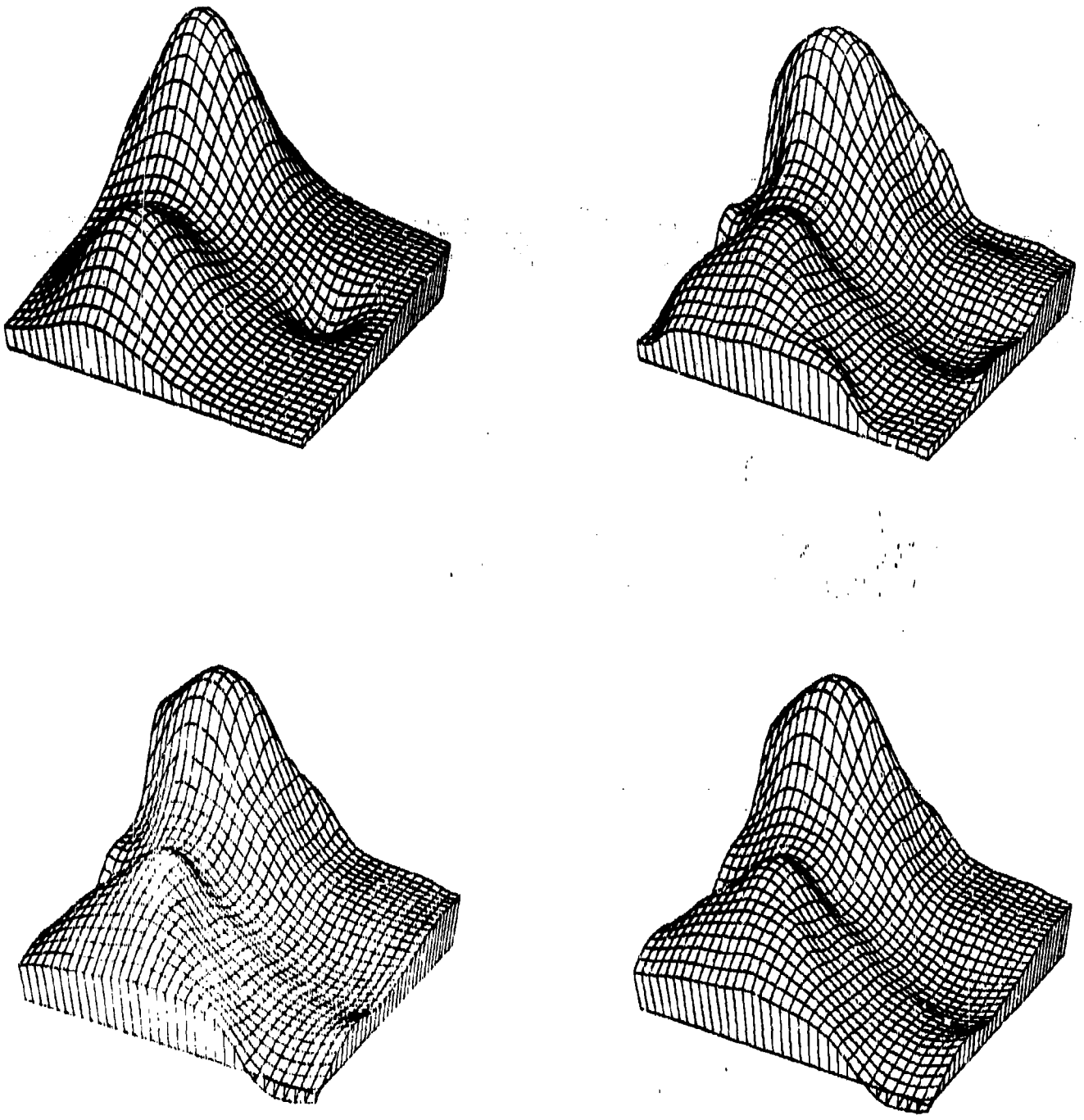
Akima's Method, Modification One

Figure 3.3.1.10



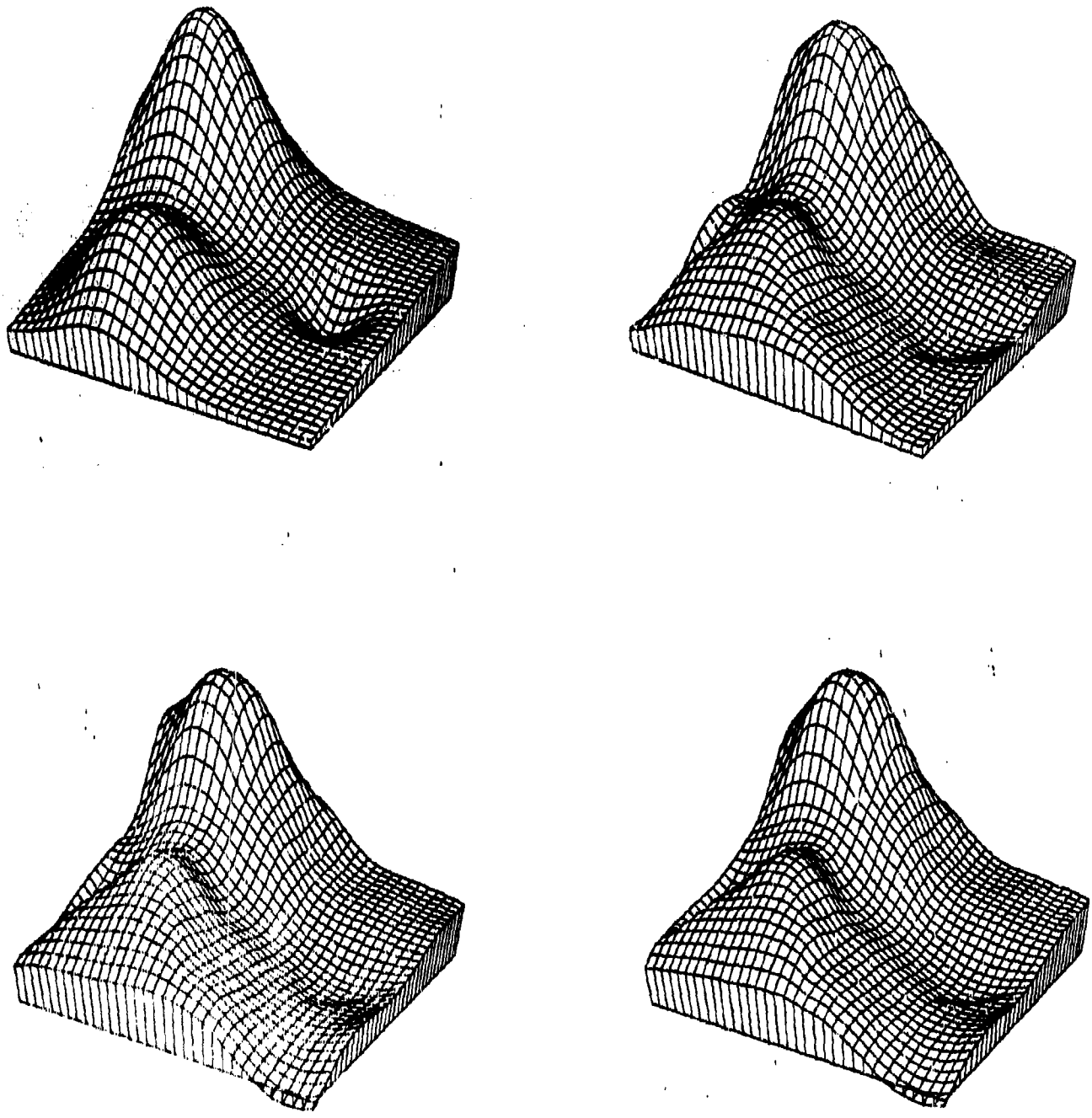
Nielson-Franke Quadratic Triangular Method

Figure 3.3.1.13



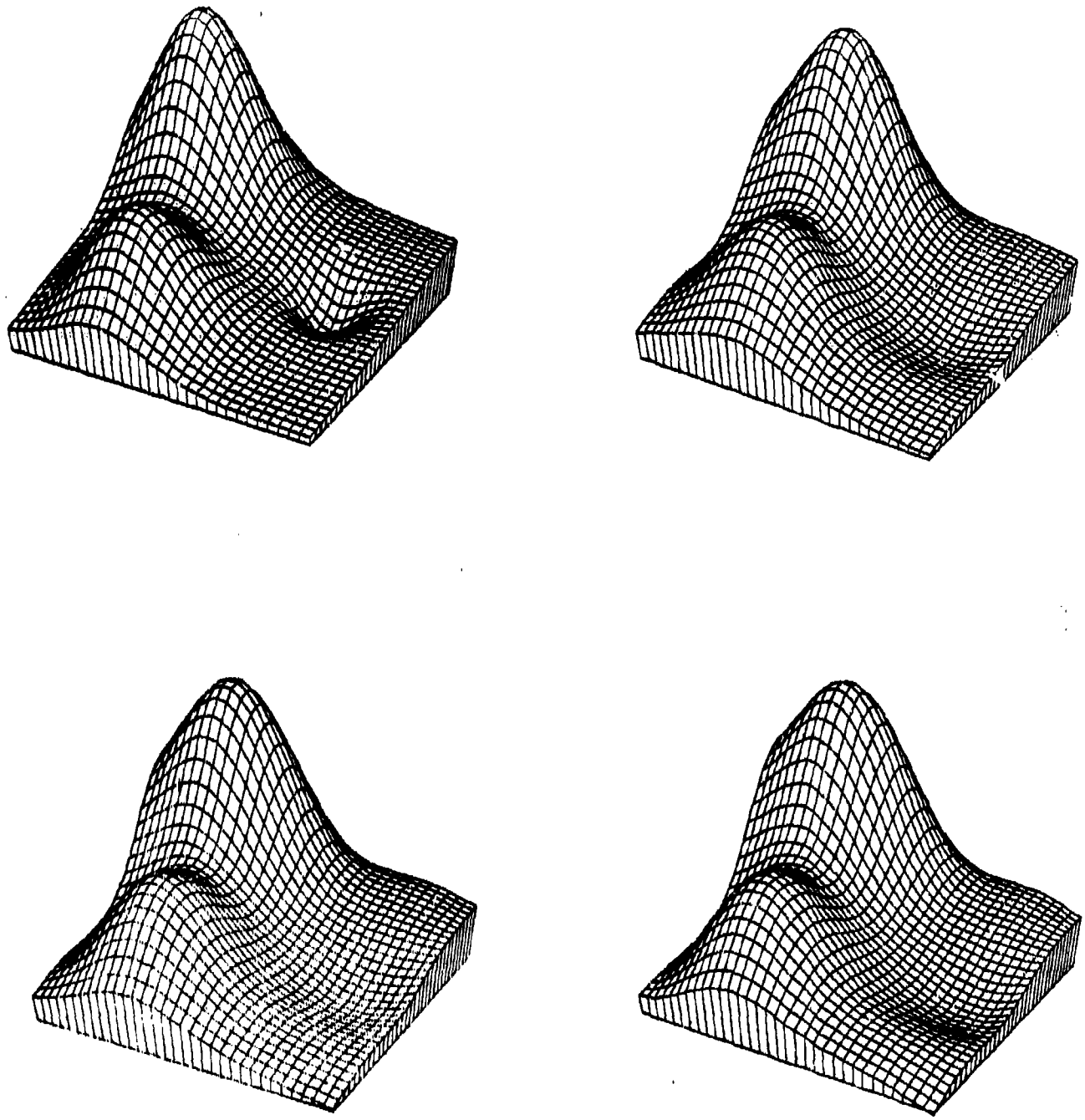
Modified Quadratic Shepard Method

Figure 3.3.1.14



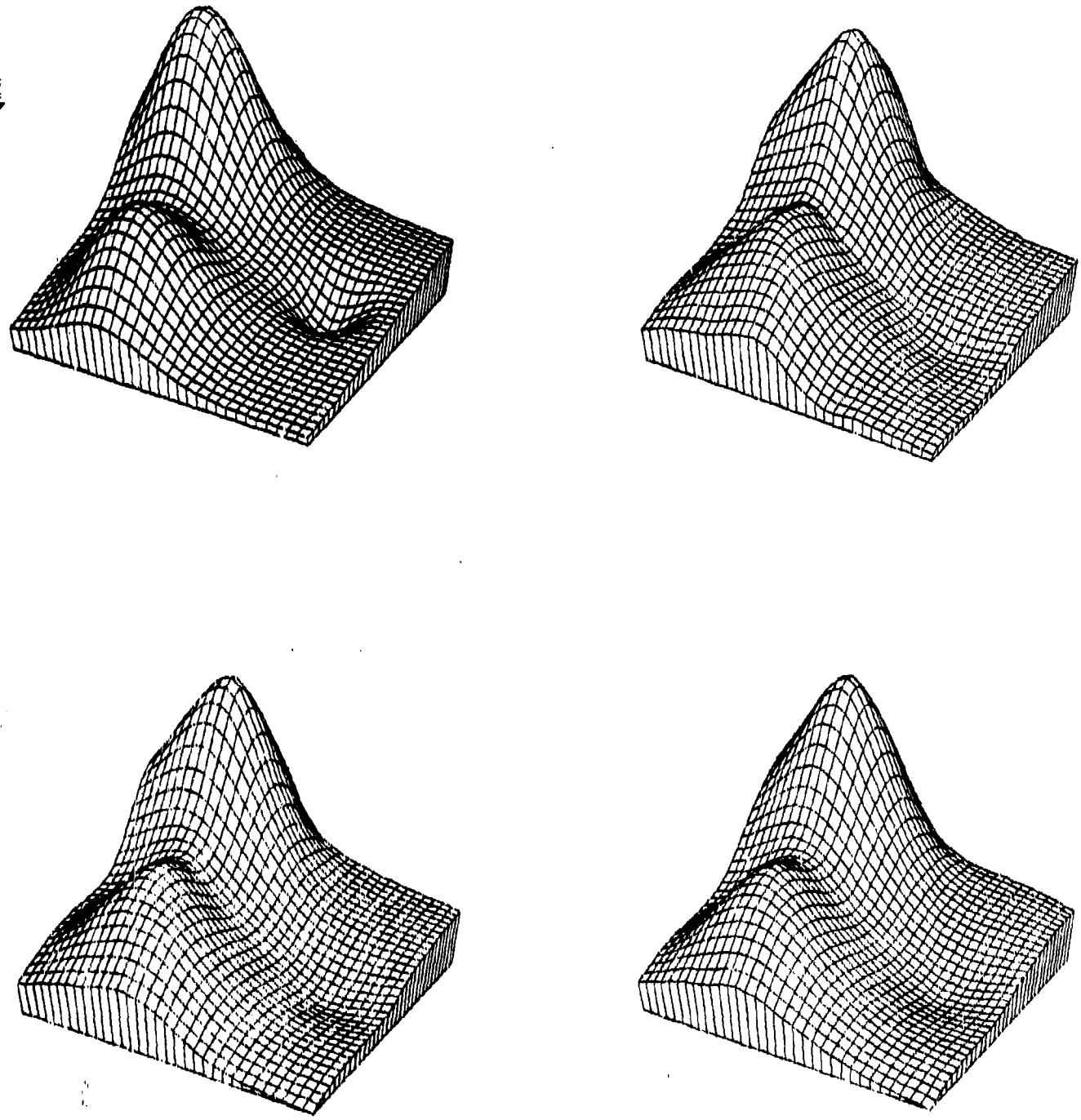
Akima's Method, Modification Three

Figure 3.3.1.16



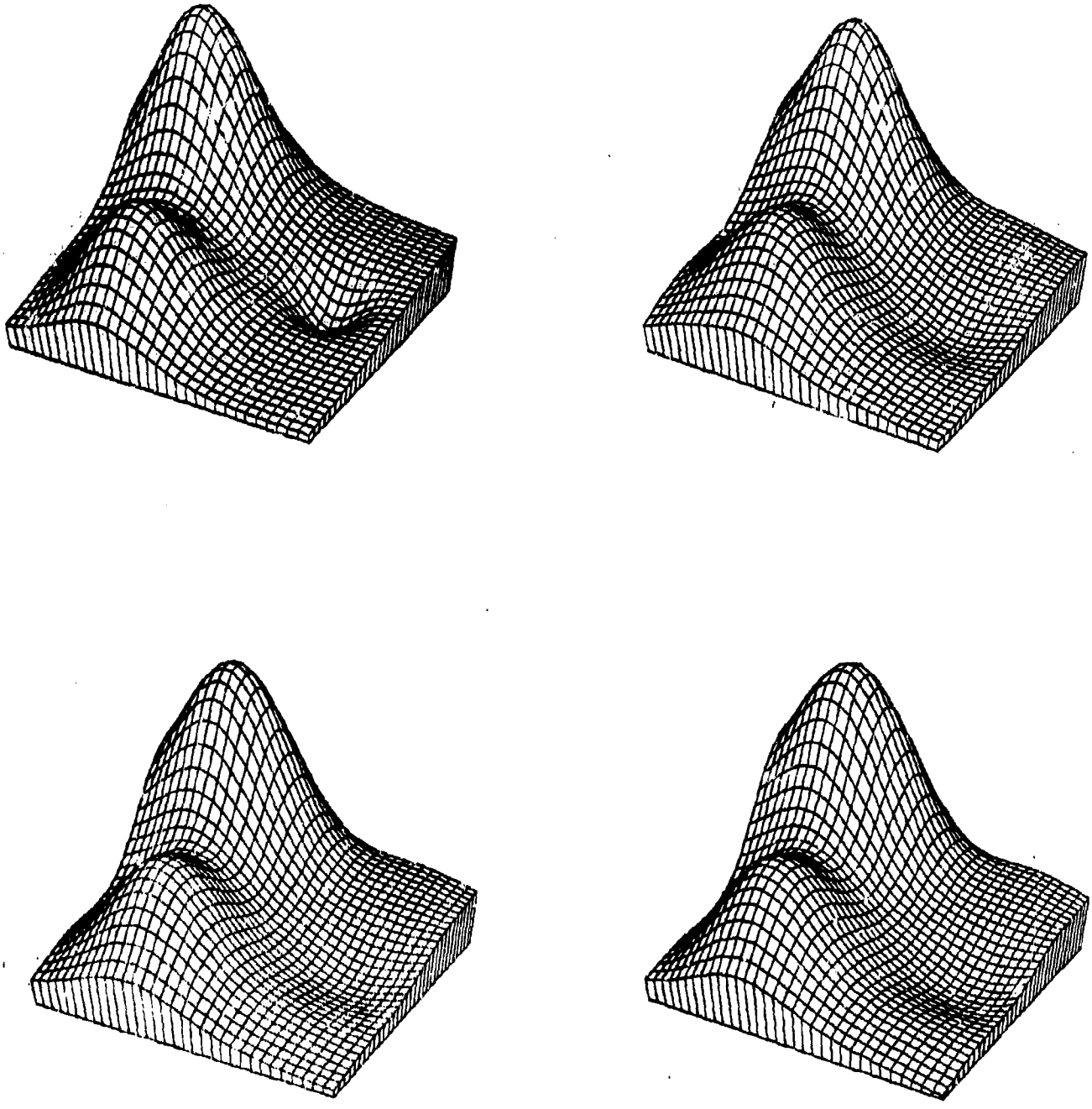
Hardy Multiquadric Method

Figure 3.3.1.21



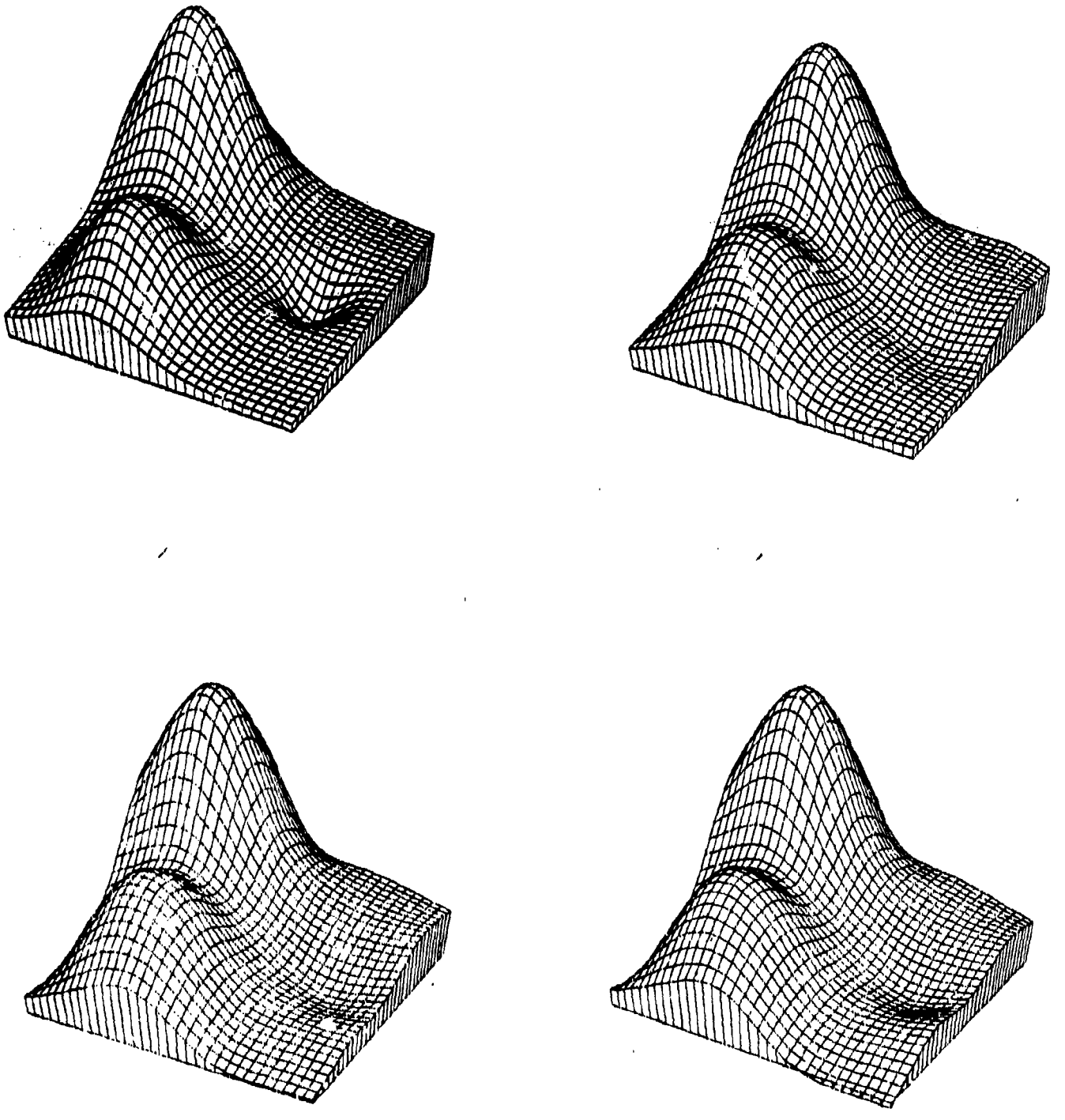
Franke's Method, Thin Plate Local Functions

Figure 3.3.1.24



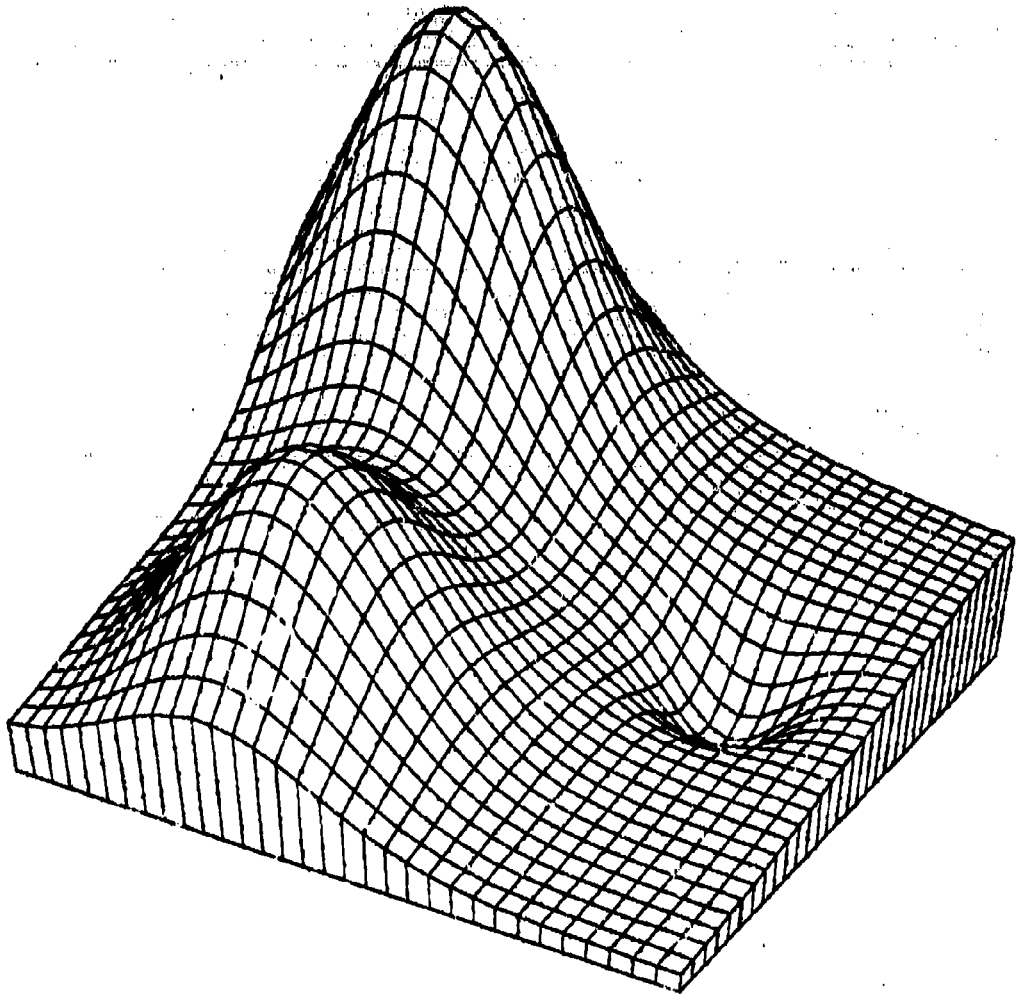
Hardy Reciprocal Multiquadric Method

Figure 3.3.1.27



Foley's Iterated Generalized Newton
Delta Sum Bicubic Spline

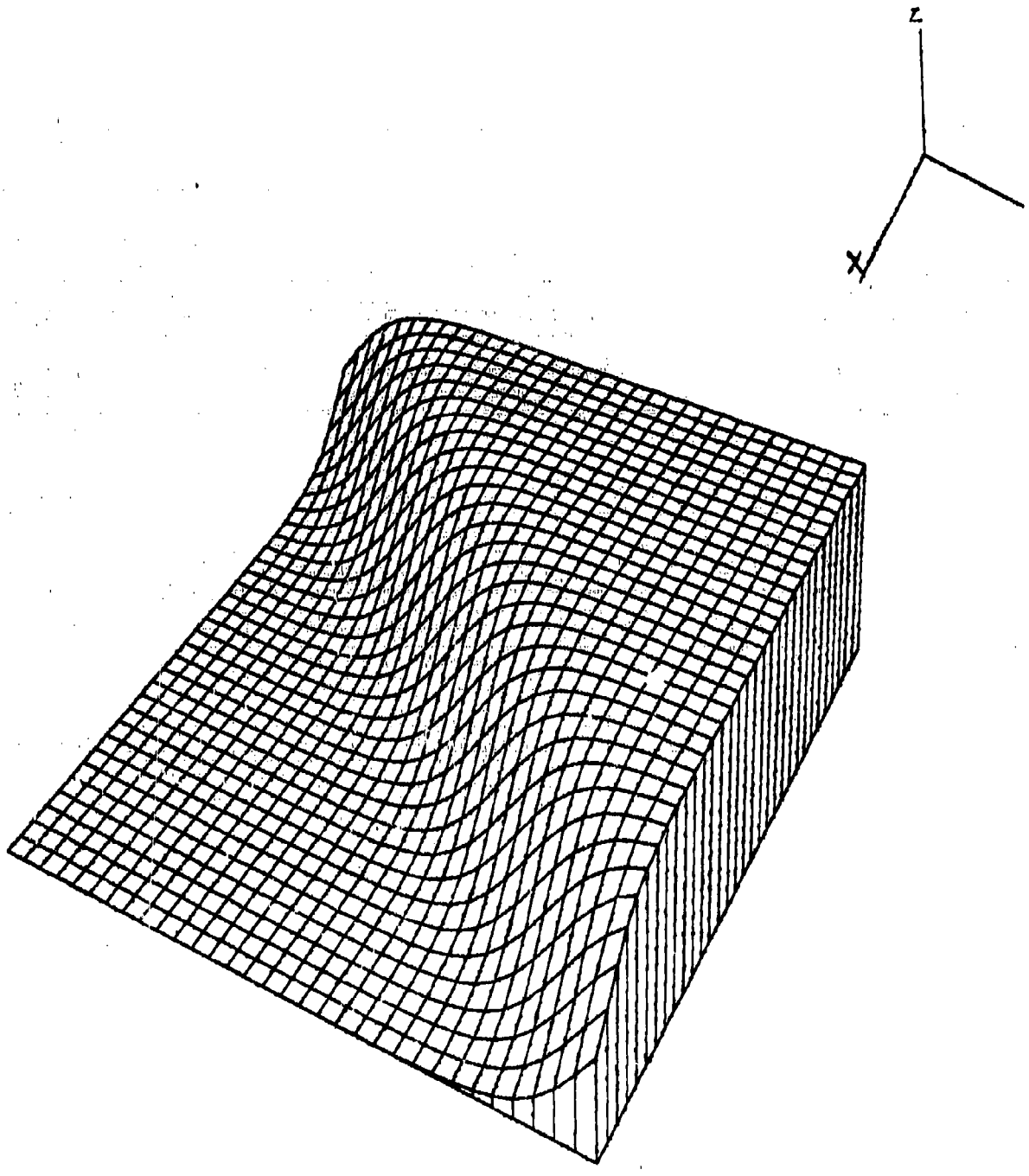
Figure 3.3.1.30



0 PT EXPONENTIAL HUMPS, DIP 30378 1052

TEST SURFACE PLOT

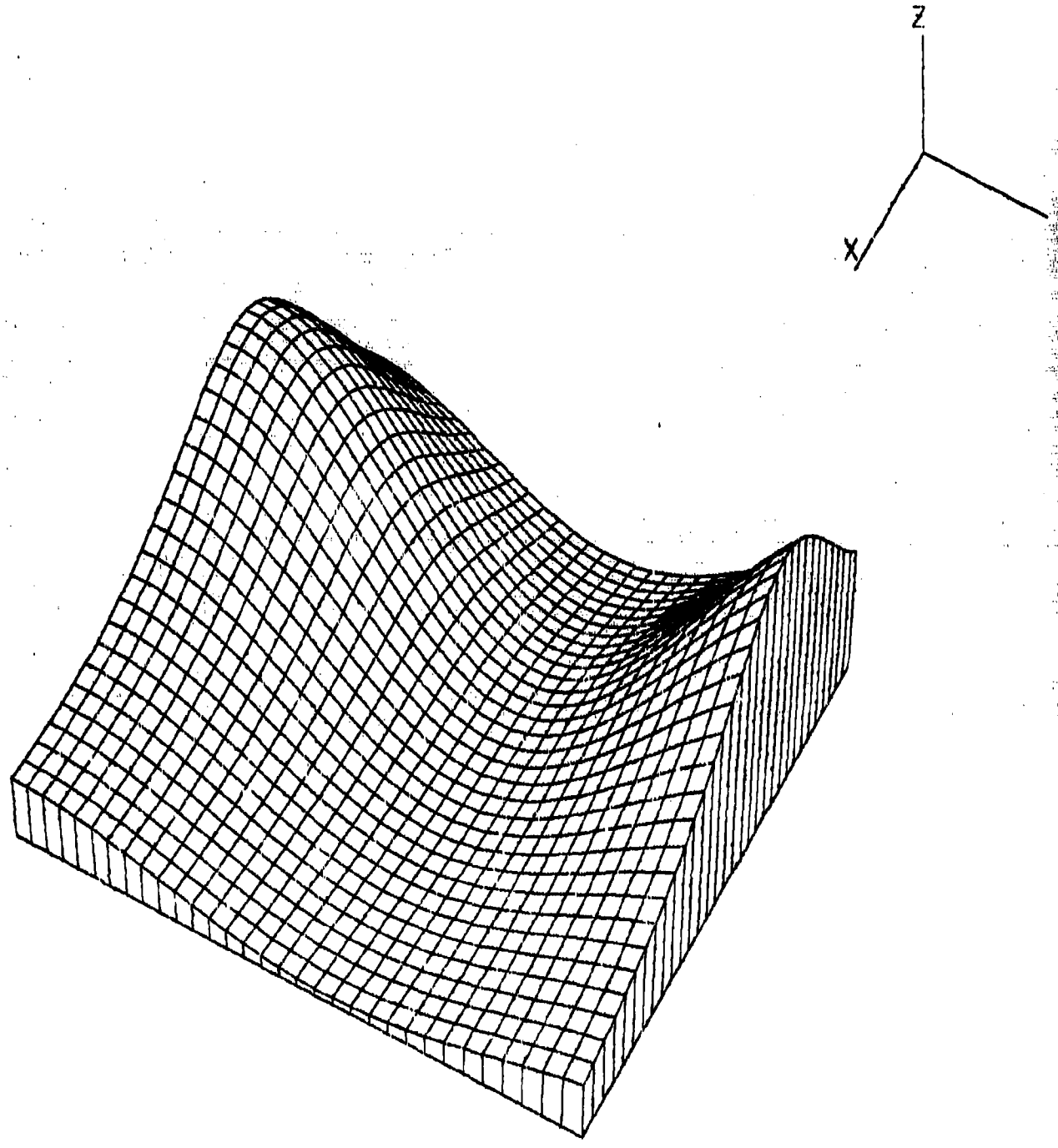
Figure 4.0.1.0.



0 PT CLIFF FUNCTION 30378 1054

TEST SURFACE PLOT

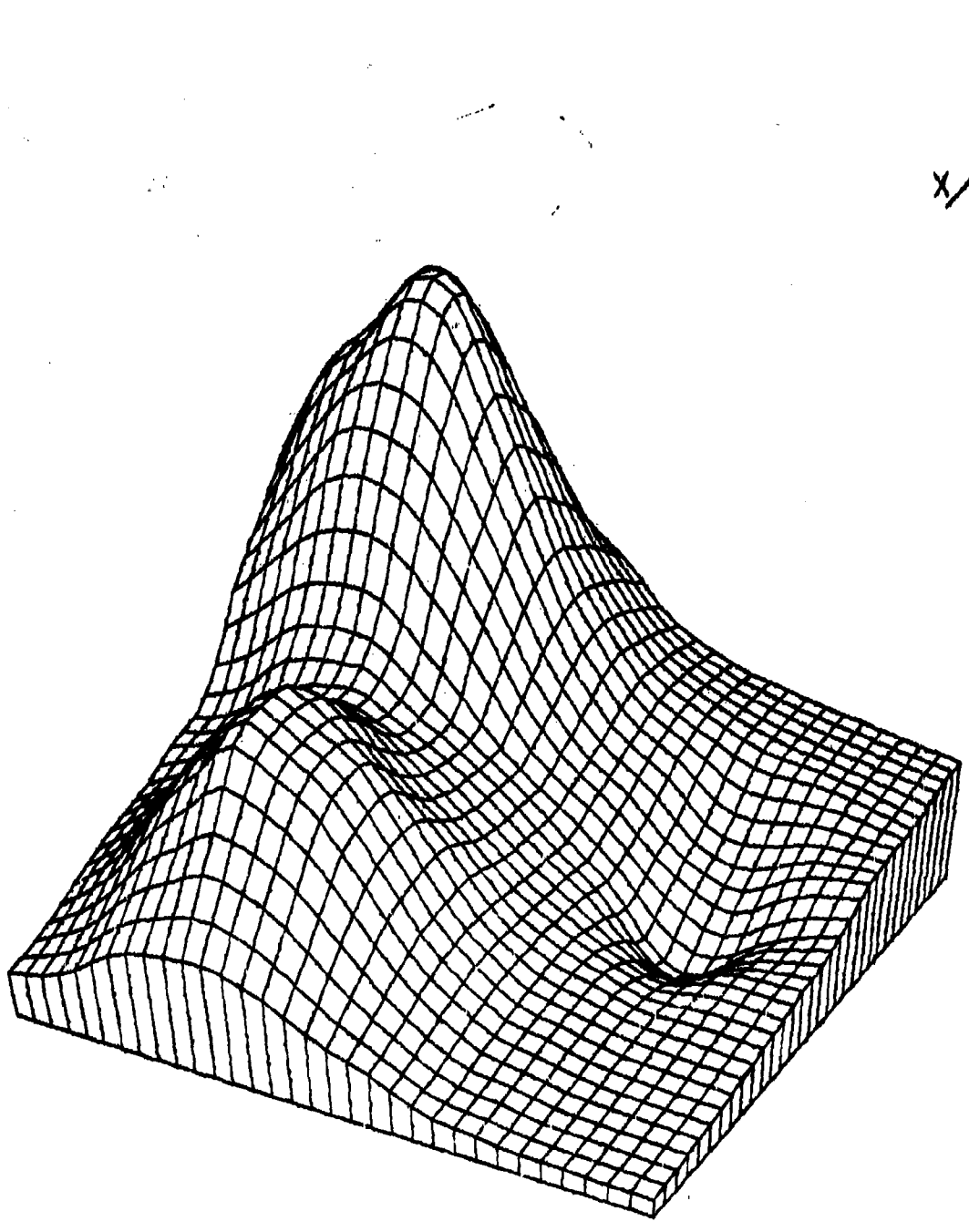
Figure 4.0.2.0.



0 PT SADDLE MOUNTAIN THREE 31778 1131

TEST SURFACE PLOT

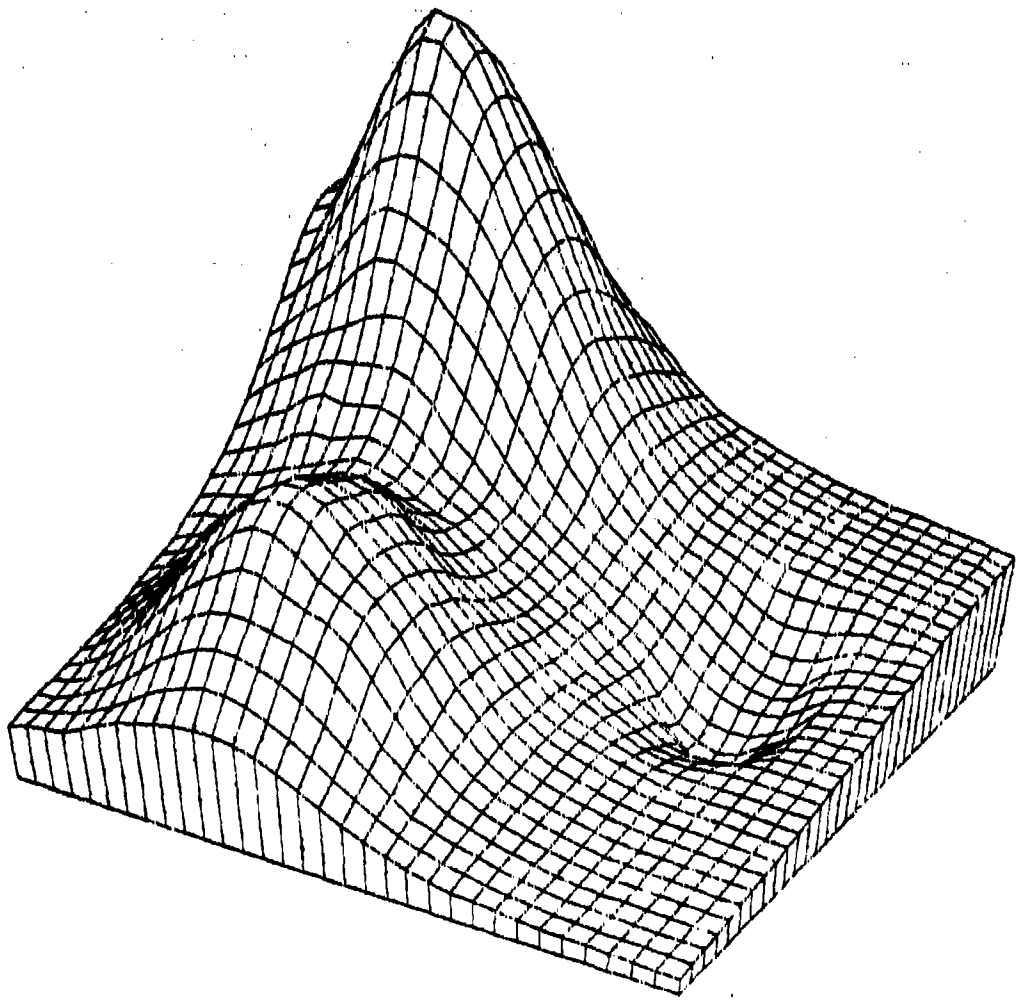
Figure 4.0.3.0.



100 PT EXPONENTIAL HUMPS, DIP 30878 1120

FRANKE - SARD BS PLANE NPPR = 6

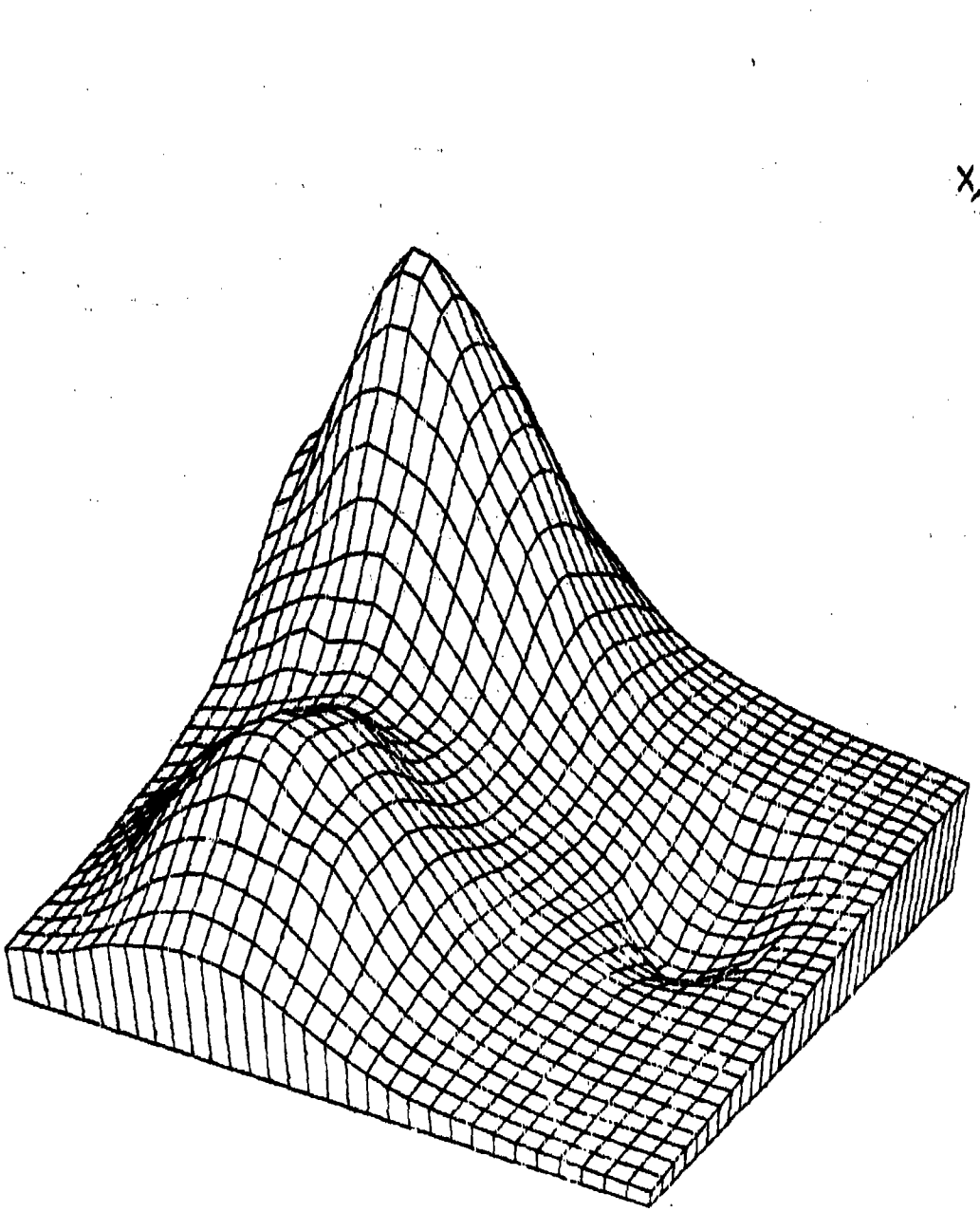
Figure 4.1.1.1



100 PT EXPONENTIAL HUMPS, DIP 32778 1643

AKIMA - FEM FIT NPPR = 6

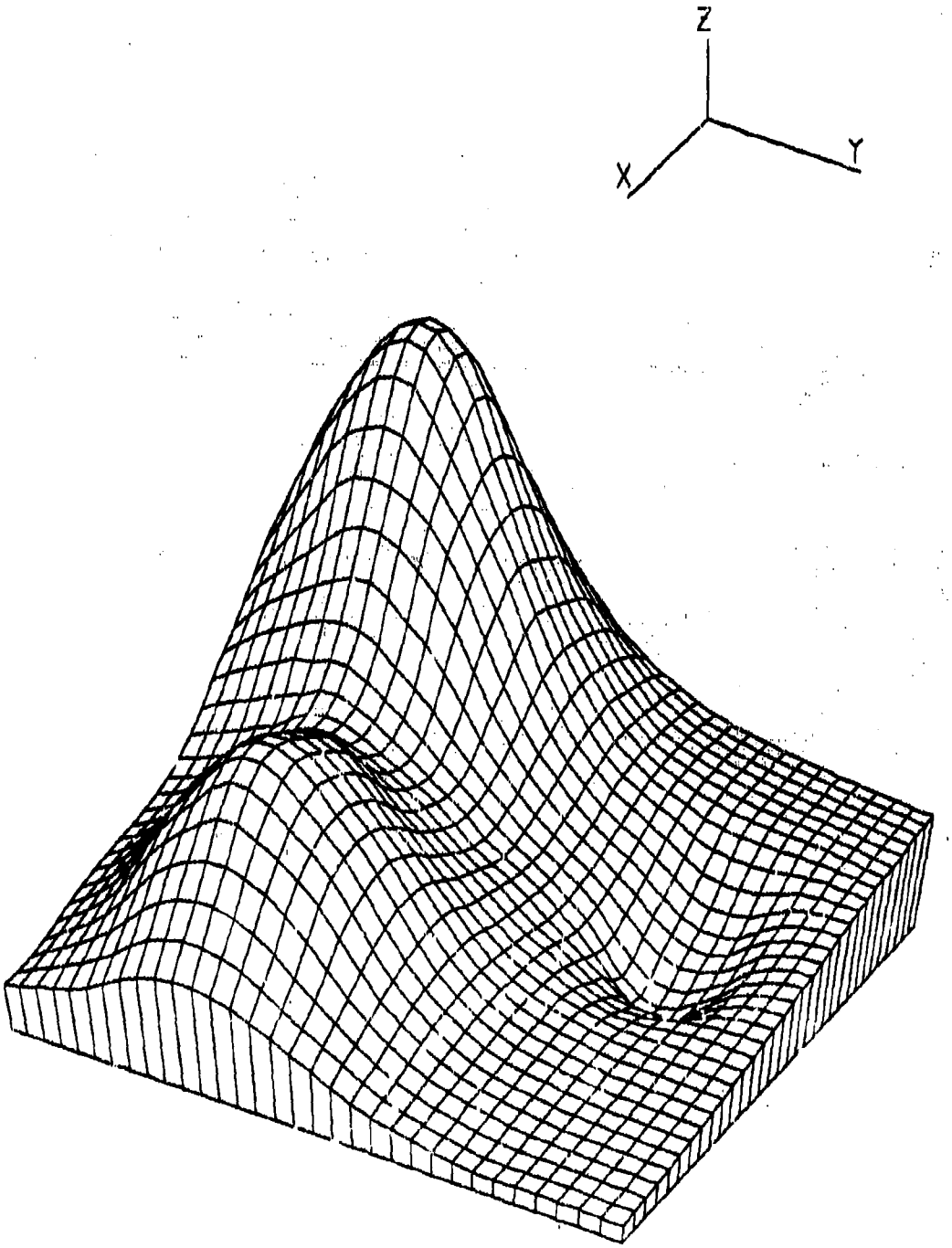
Figure 4.1.1.4



100 PT EXPONENTIAL HUMPS, DIP 32778 1543

MODIFIED AKIMA FEM FIT NPPR = 6

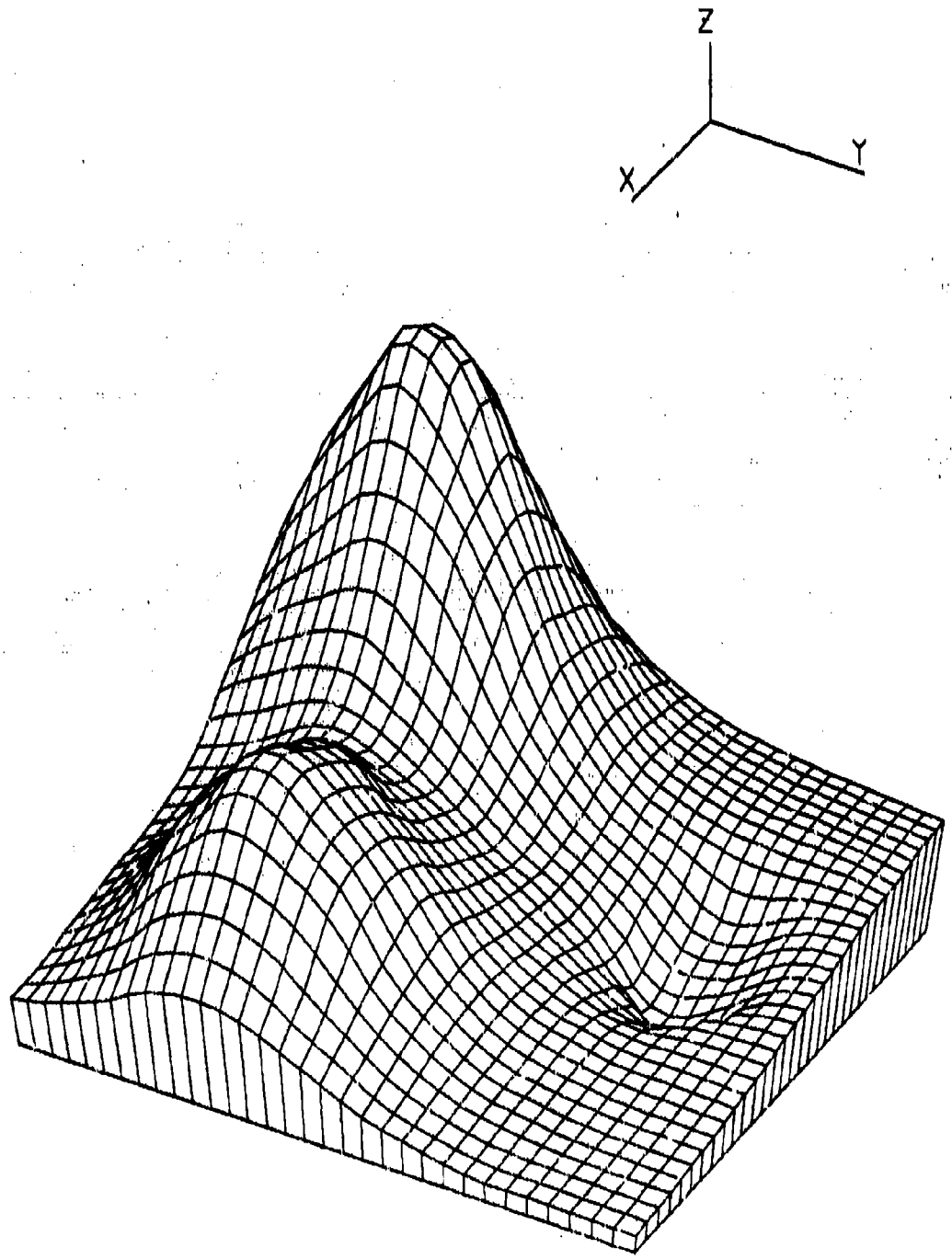
Figure 4.1.1.10



100PT EXPONENTIAL HUMPS, DIP 11579 1155

NIELSON-FRANKE QUAD-TRI NPPR = 18 R = 0.315

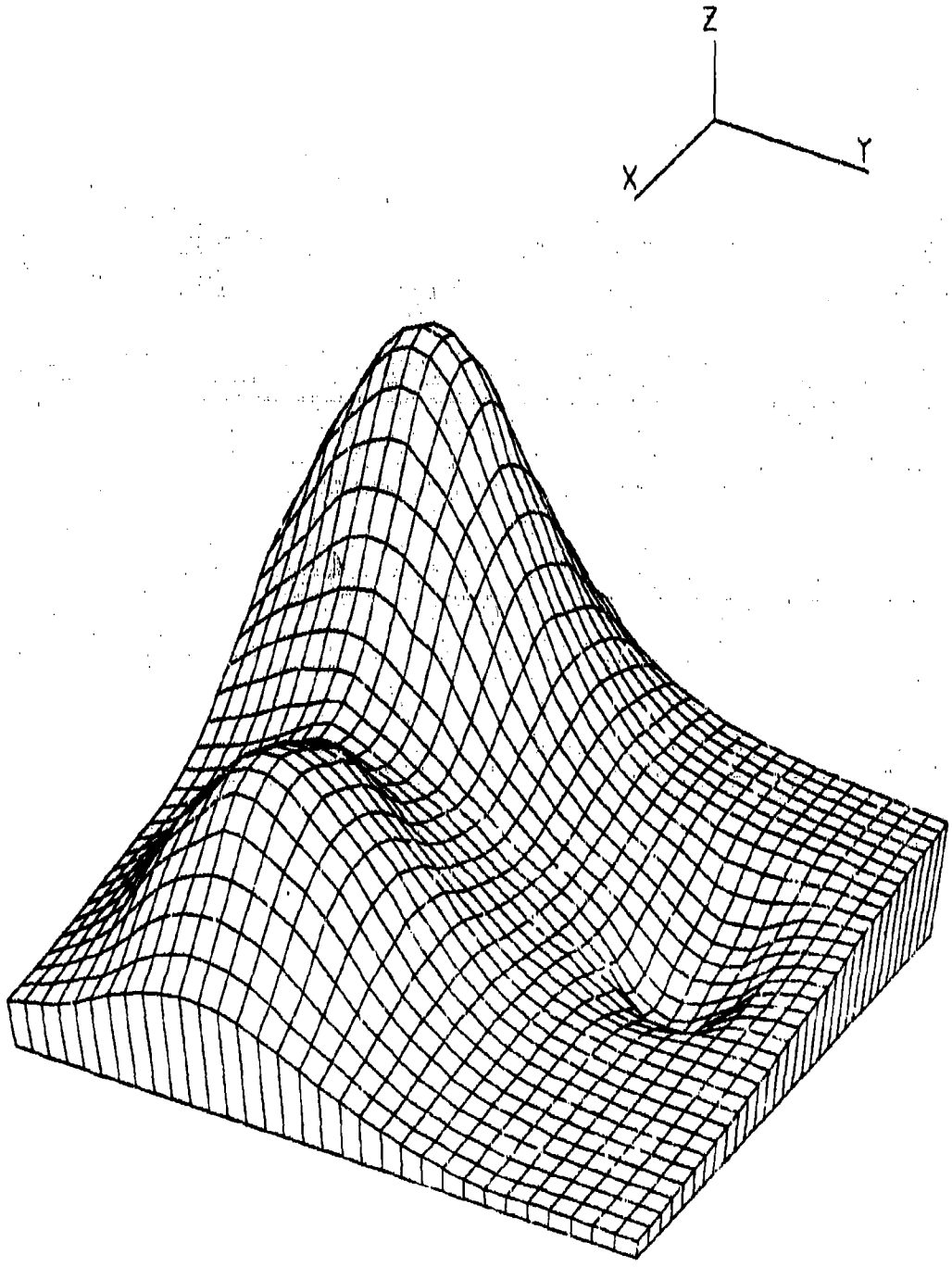
Figure 4.1.1.13



100PT EXPONENTIAL HUMPS, DIP 11579 1208

QUADRATIC SHEPARD METHOD NPPR = 9,18 R = 0.222

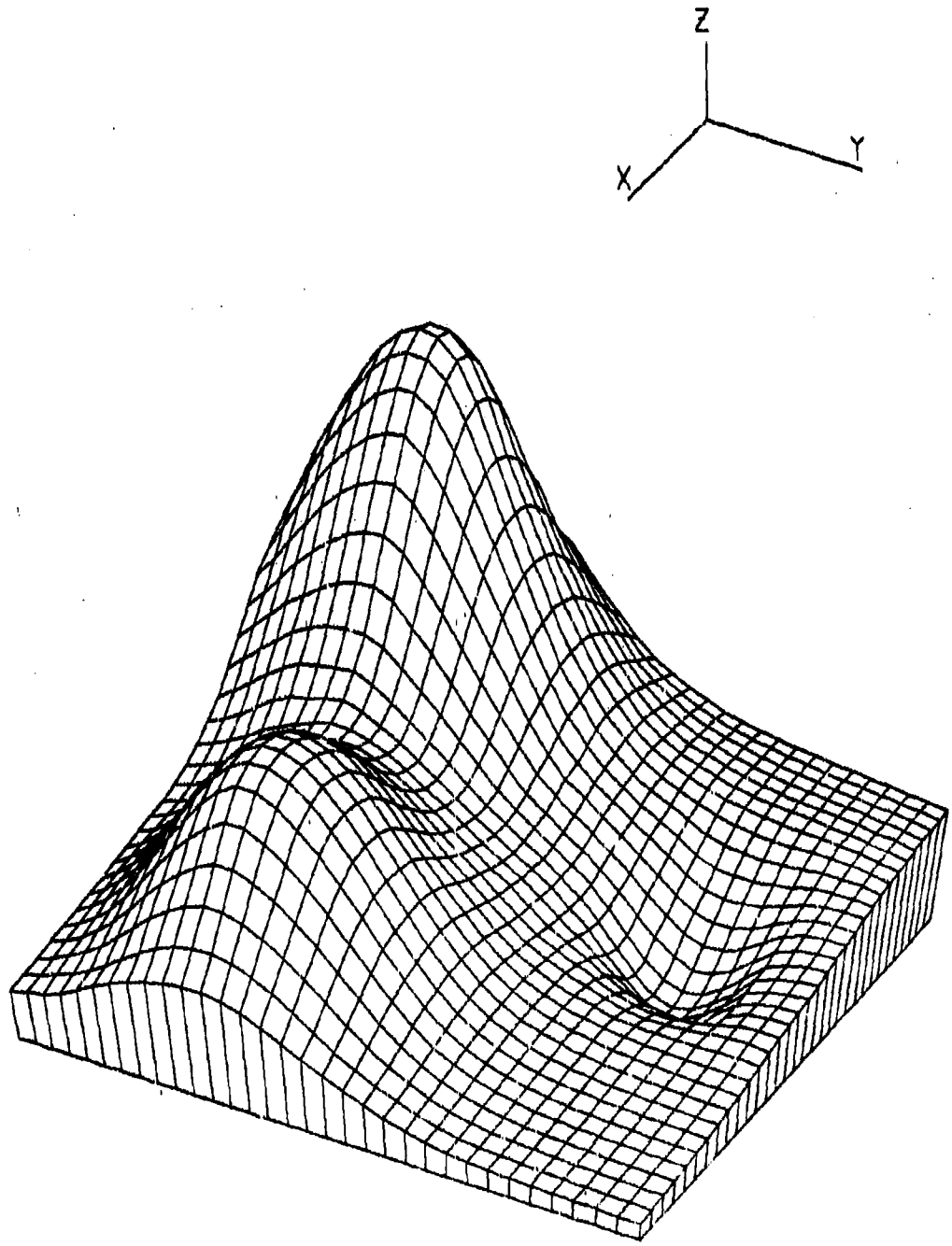
Figure 4.1.1.14



100PT EXPONENTIAL HUMPS, DIP 31079 1931

AKIMA - L2 QUAD DERIVS NPPR = 18 R = 0.315

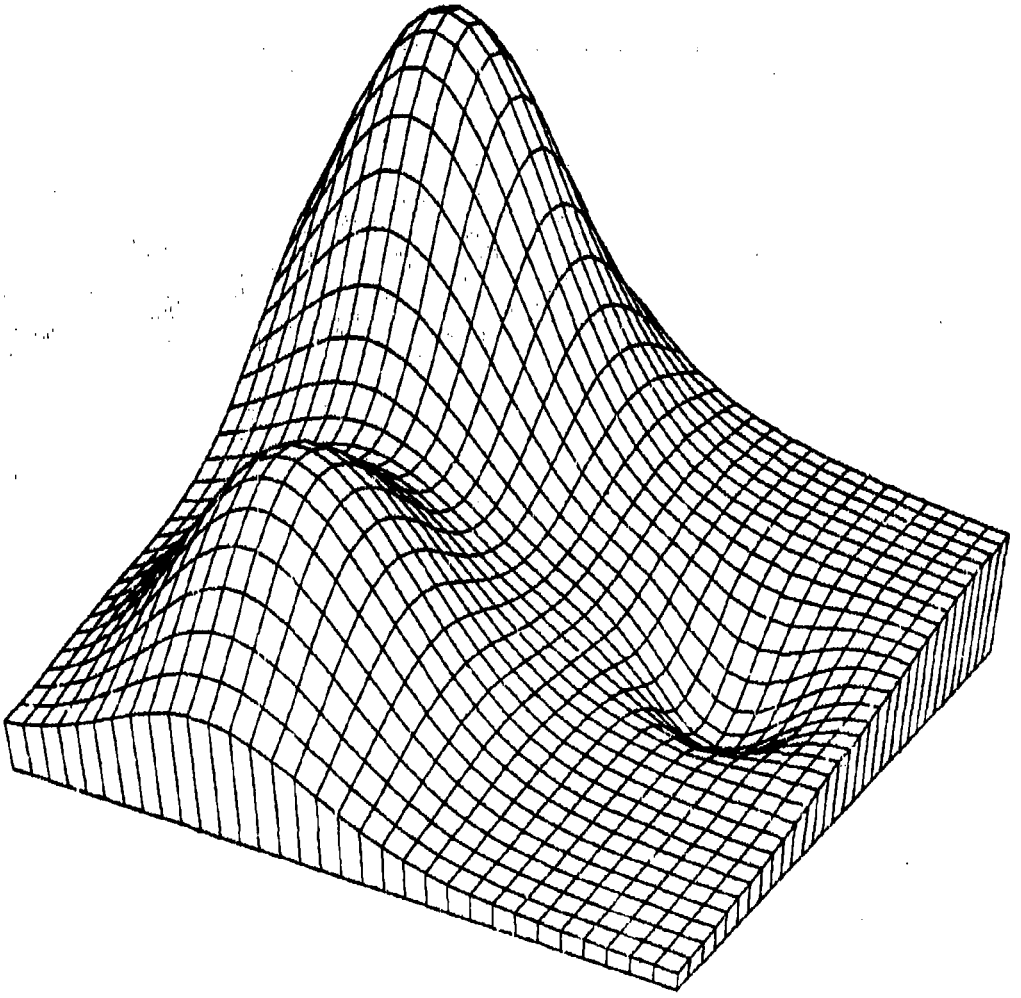
Figure 4.1.1.16



100PT EXPONENTIAL HUMPS, DIP 13179 930

NIELSON MIN NORM NETWORK

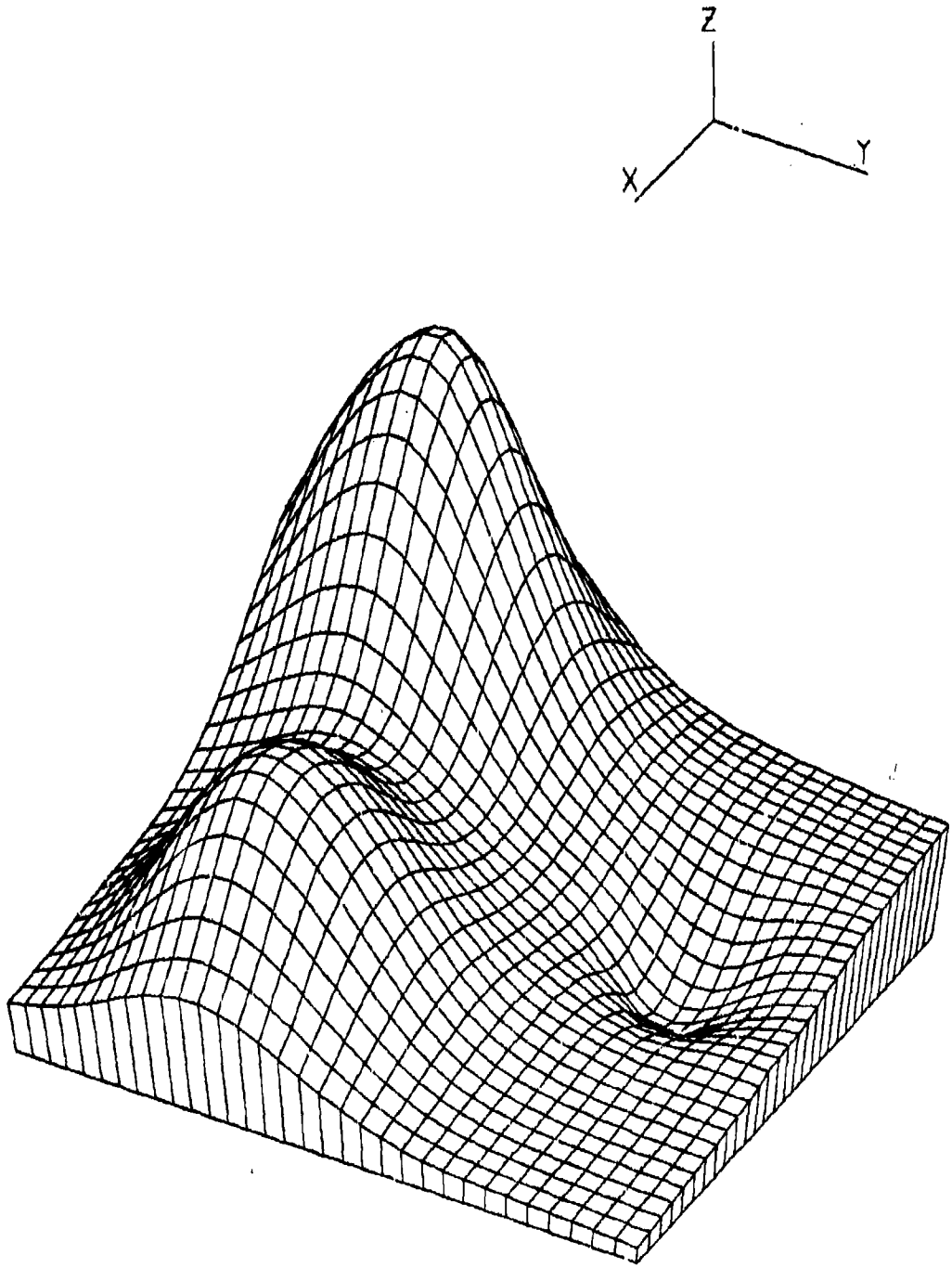
Figure 4.1.1.19



100PT EXPONENTIAL HUMPS, DIP 32279 1715

HARDY MQ SURFACE NPPR = 25 R = 0.185

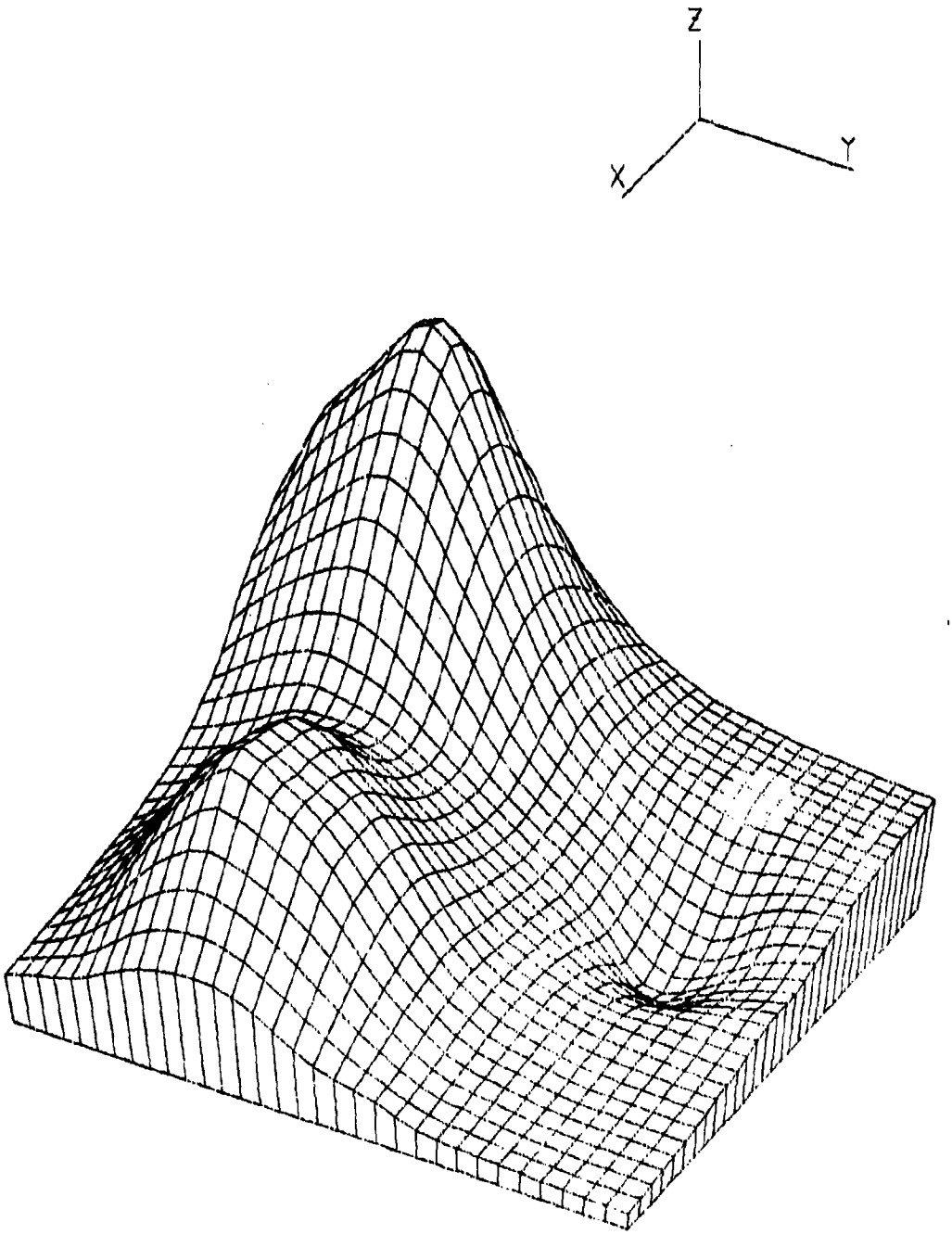
Figure 4.1.1.21



100PT EXPONENTIAL HUMPS, DIP 20679 1532

DUCHON S THIN PLATE

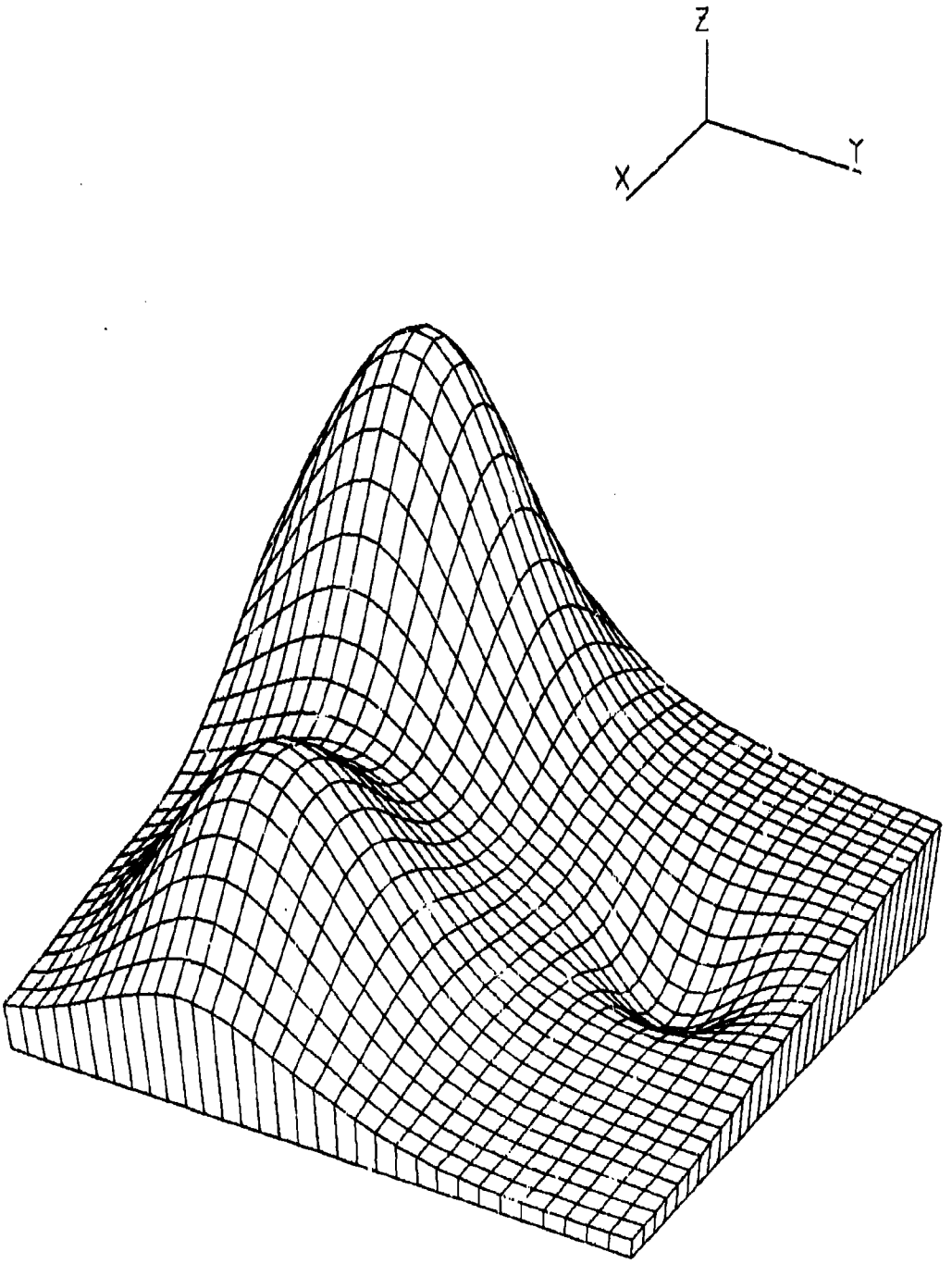
Figure 4.1.1.23



100PT EXPONENTIAL HUMPS, DIP 80879 910

FRANKE W/ THIN PLATE NPPR = 6

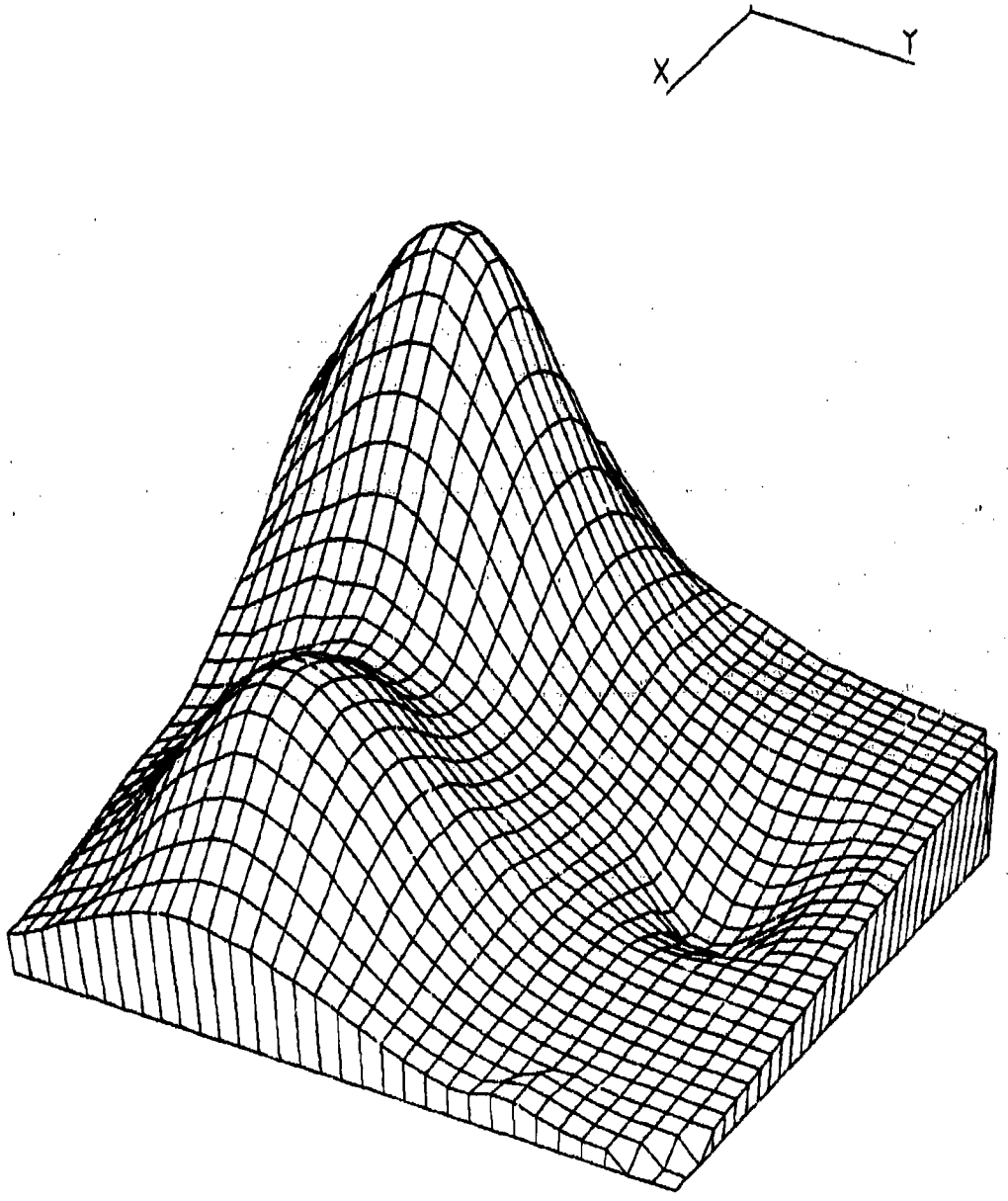
Figure 4.1.1.24



100PT EXPONENTIAL HUMPS, DIP 41079 1825

HARDY RECIP MQ SURFACE NPPR = 25 R = 0.185

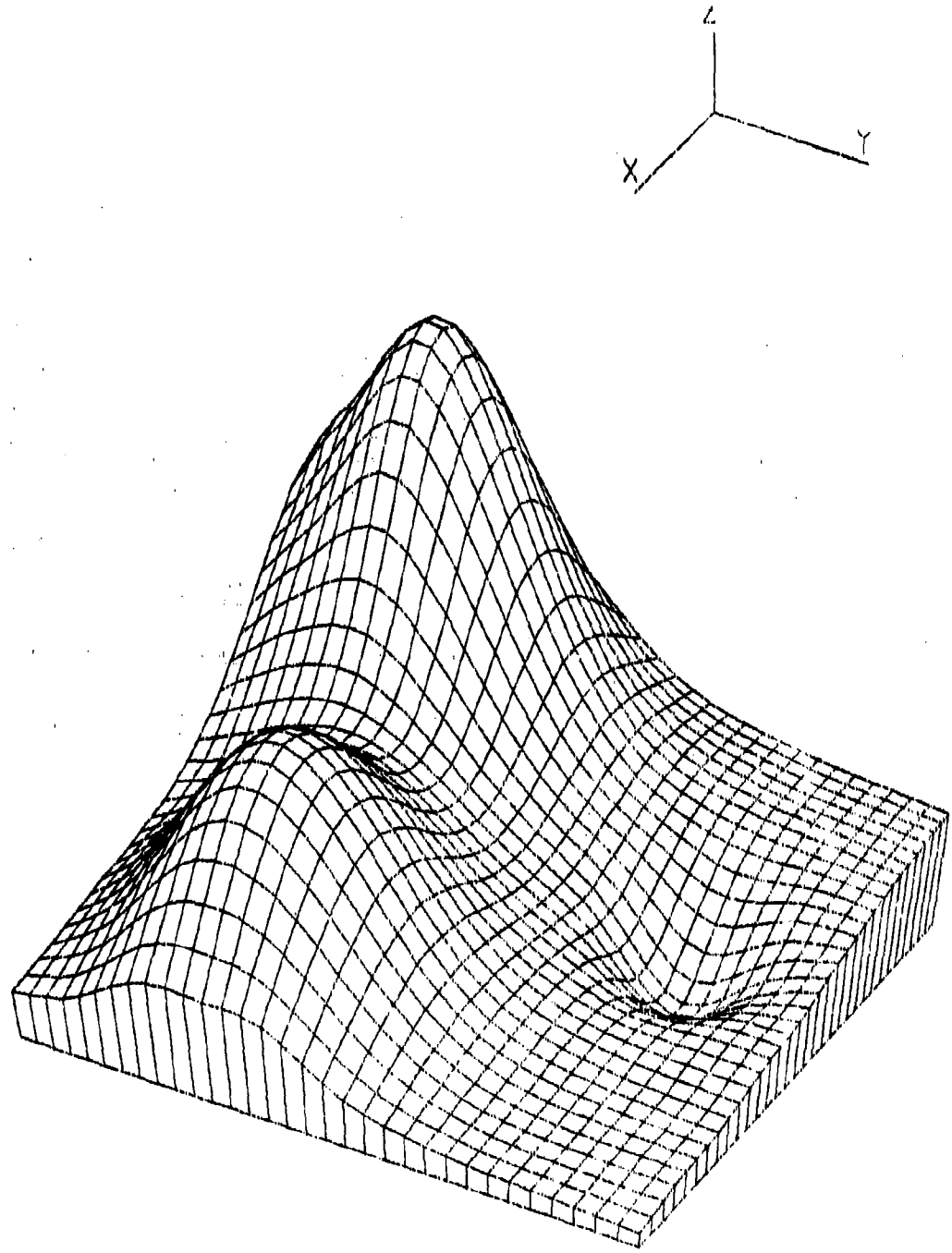
Figure 4.1.1.27



100PT EXPONENTIAL HUMPS, DIP 32379 1845

LAWSON TRIANGLE METHOD

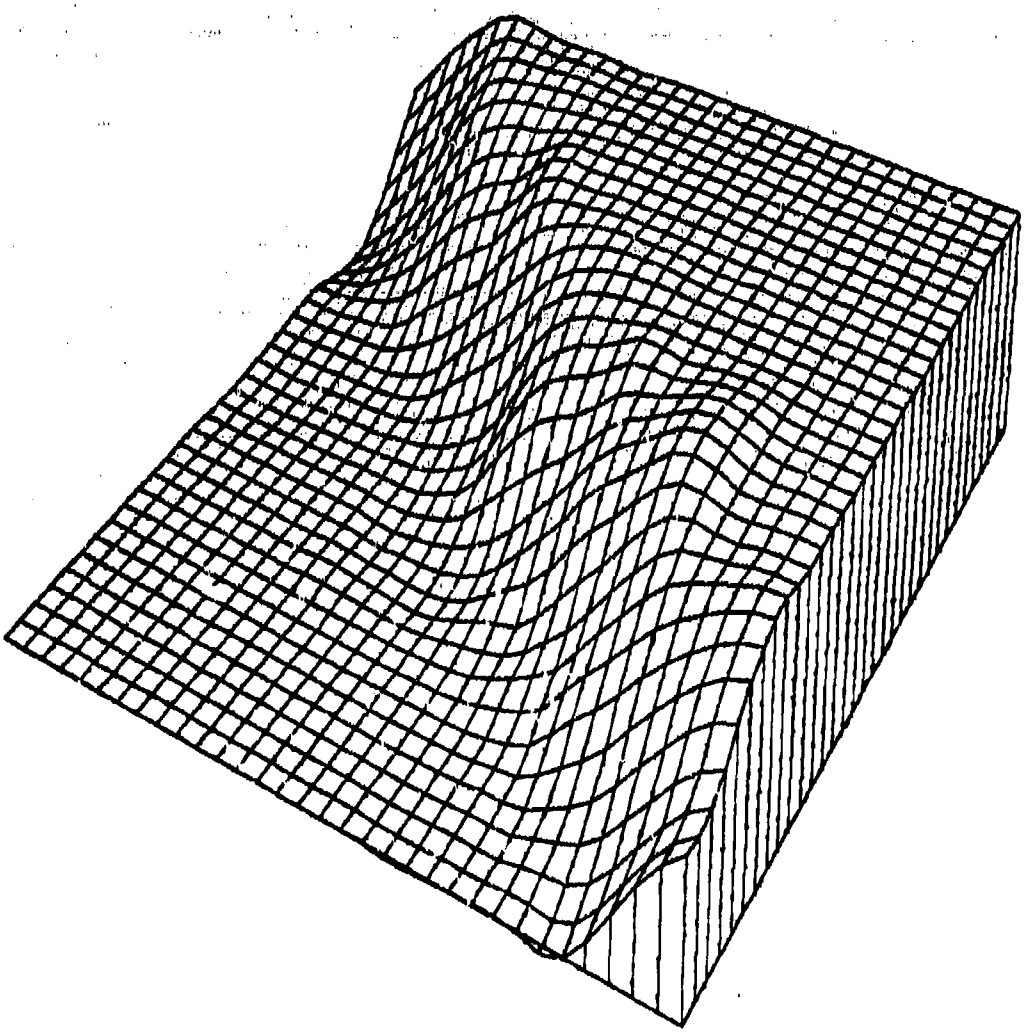
Figure 4.1.1.28



100PT EXPONENTIAL HUMPS, DIP 60279 1052

FOLEY ITERATED BICUBICS NPPR = 3

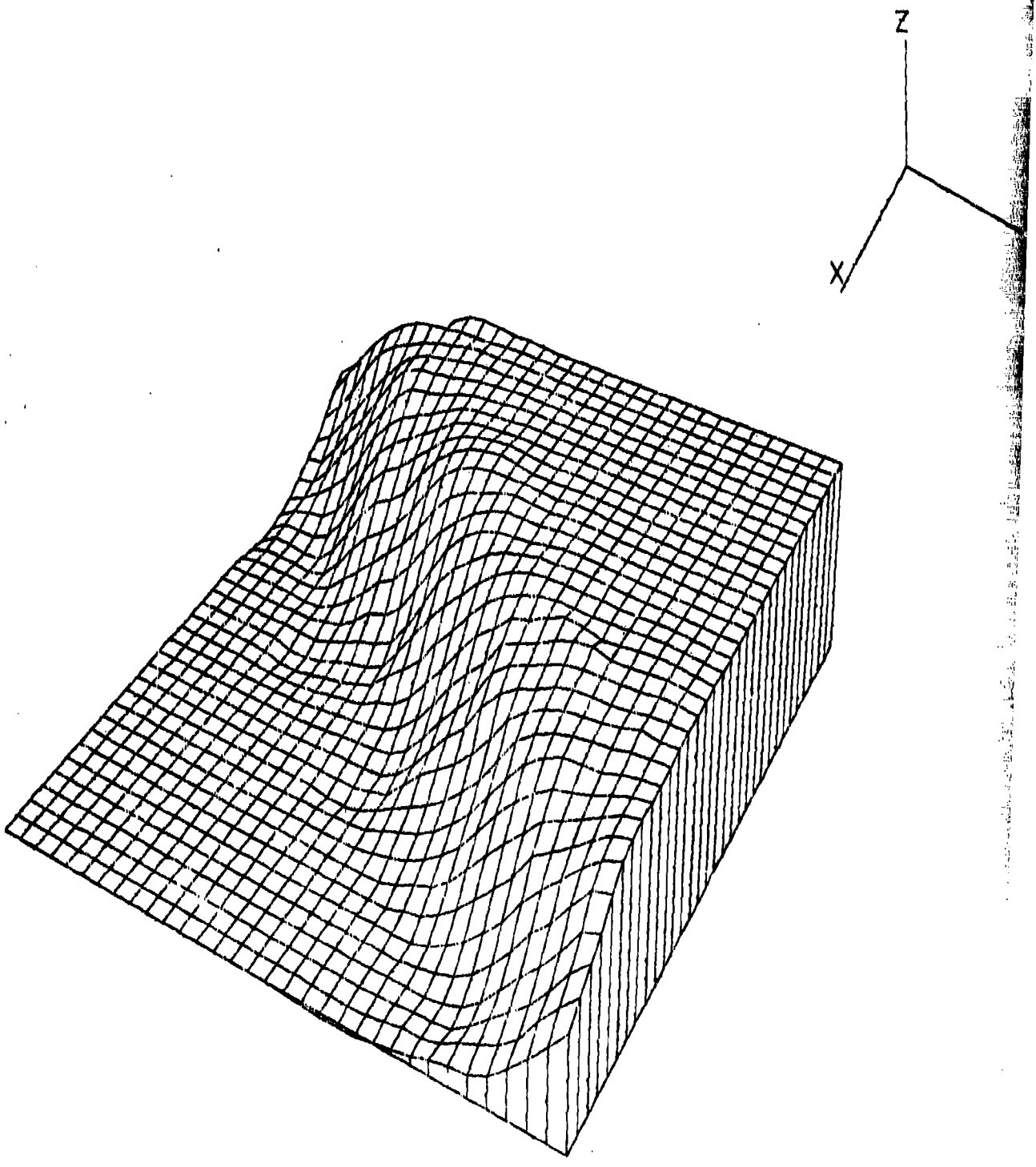
Figure 4.1.1.30



100 PT CLIFF FUNCTION 30878 1121

'RANKE - SARD BS PLANE NPPR = 6

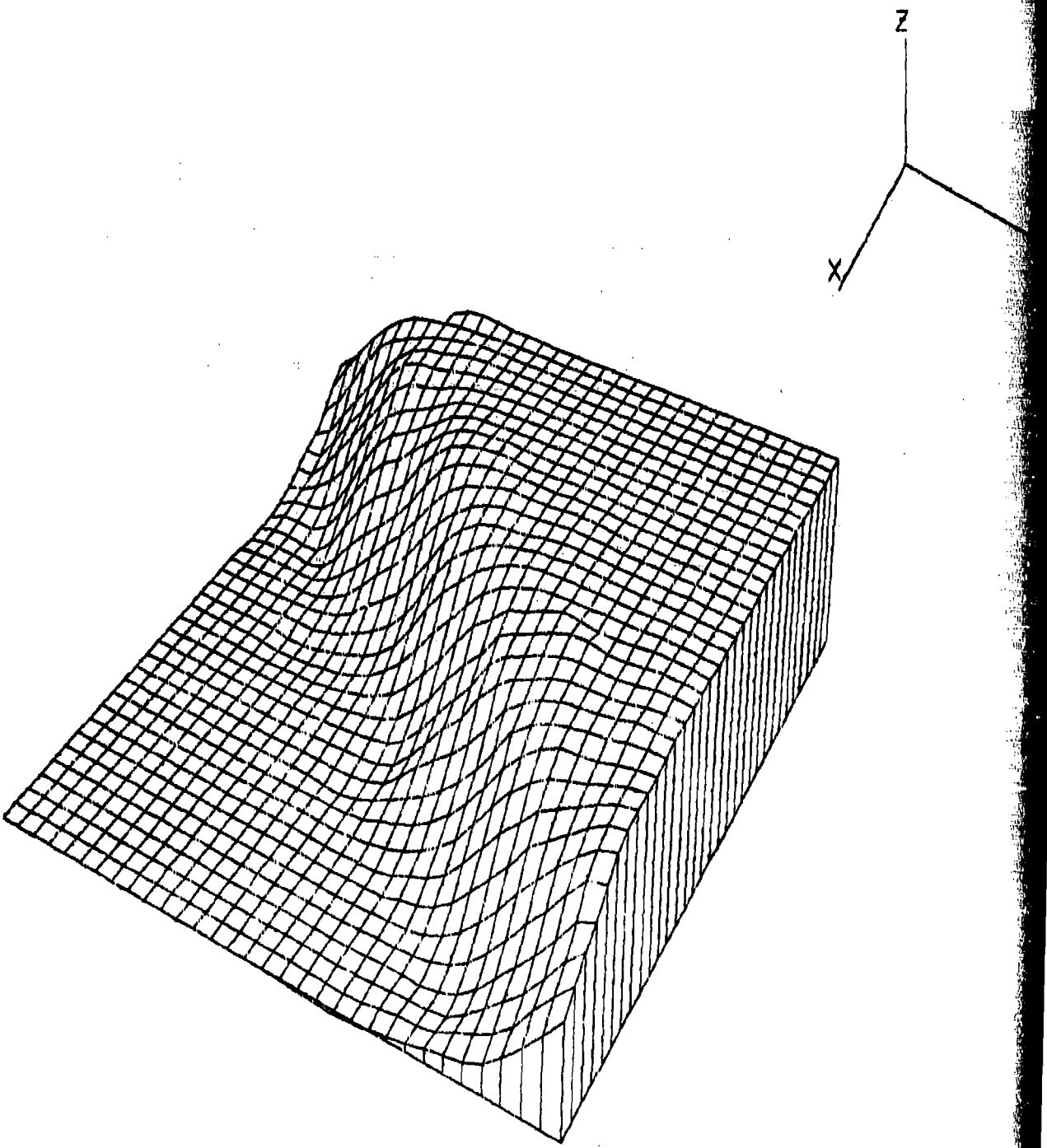
Figure 4.1.2.1



100 PT CLIFF FUNCTION 33078 1400

AKIMA - FEM FIT NPPR = 6

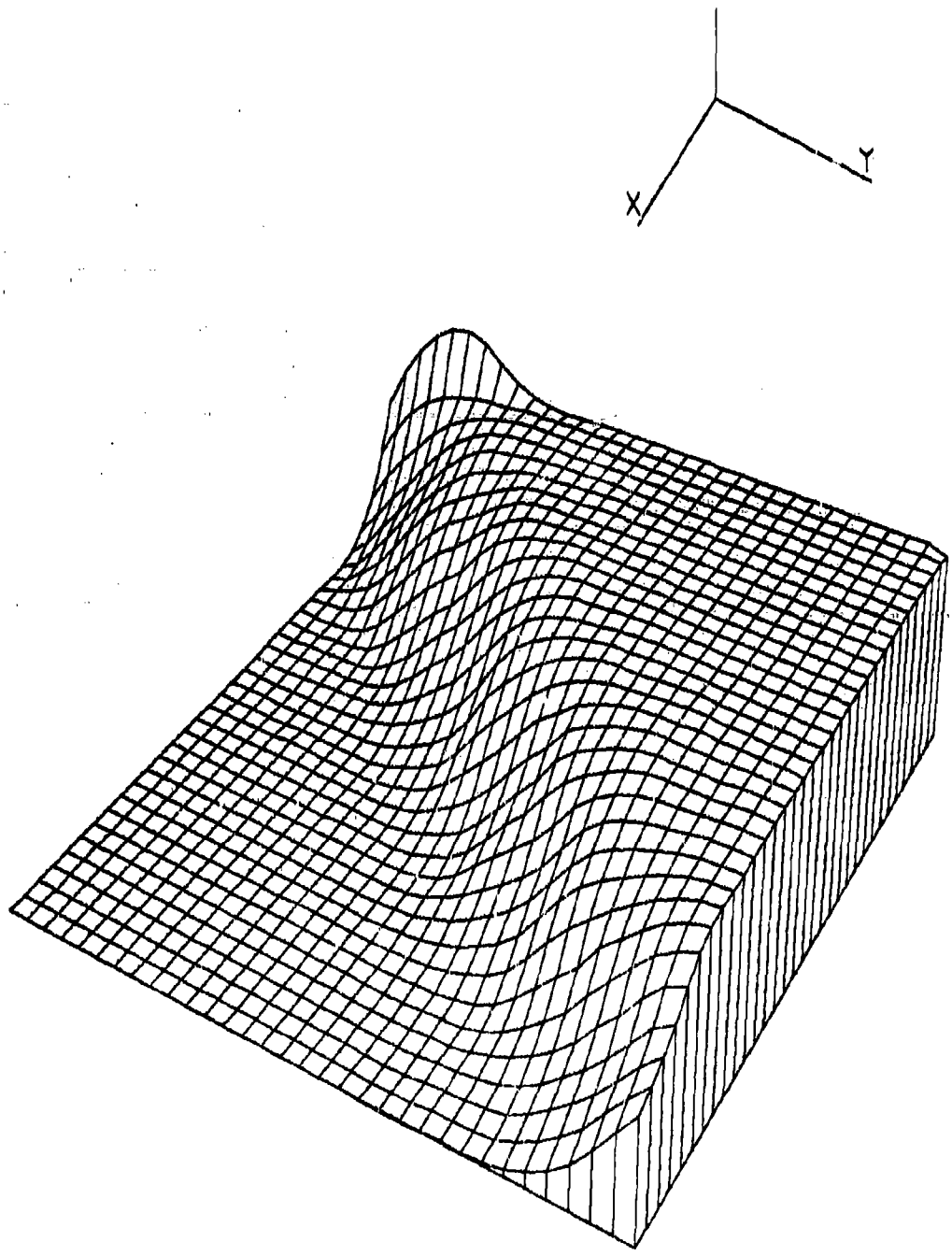
Figure 4.1.2.4



100 PT CLIFF FUNCTION 33078 1350

MODIFIED AKIMA FEM FIT NPPR = 6

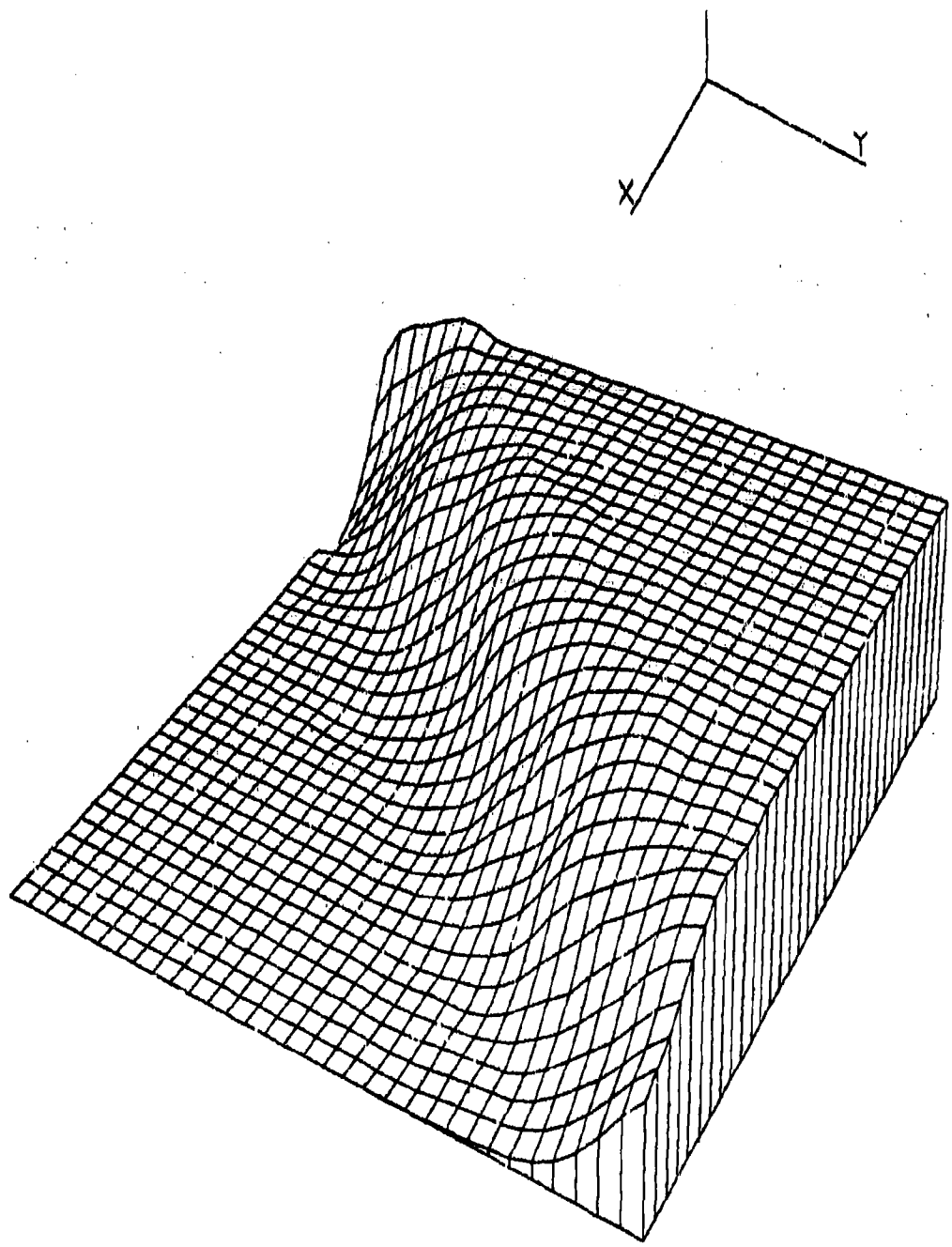
Figure 4.1.2.10



100PT CLIFF FUNCTION 31079 1919

NIELSON-FRANKE QUAD-TRI NPPR = 18 R = 0.315

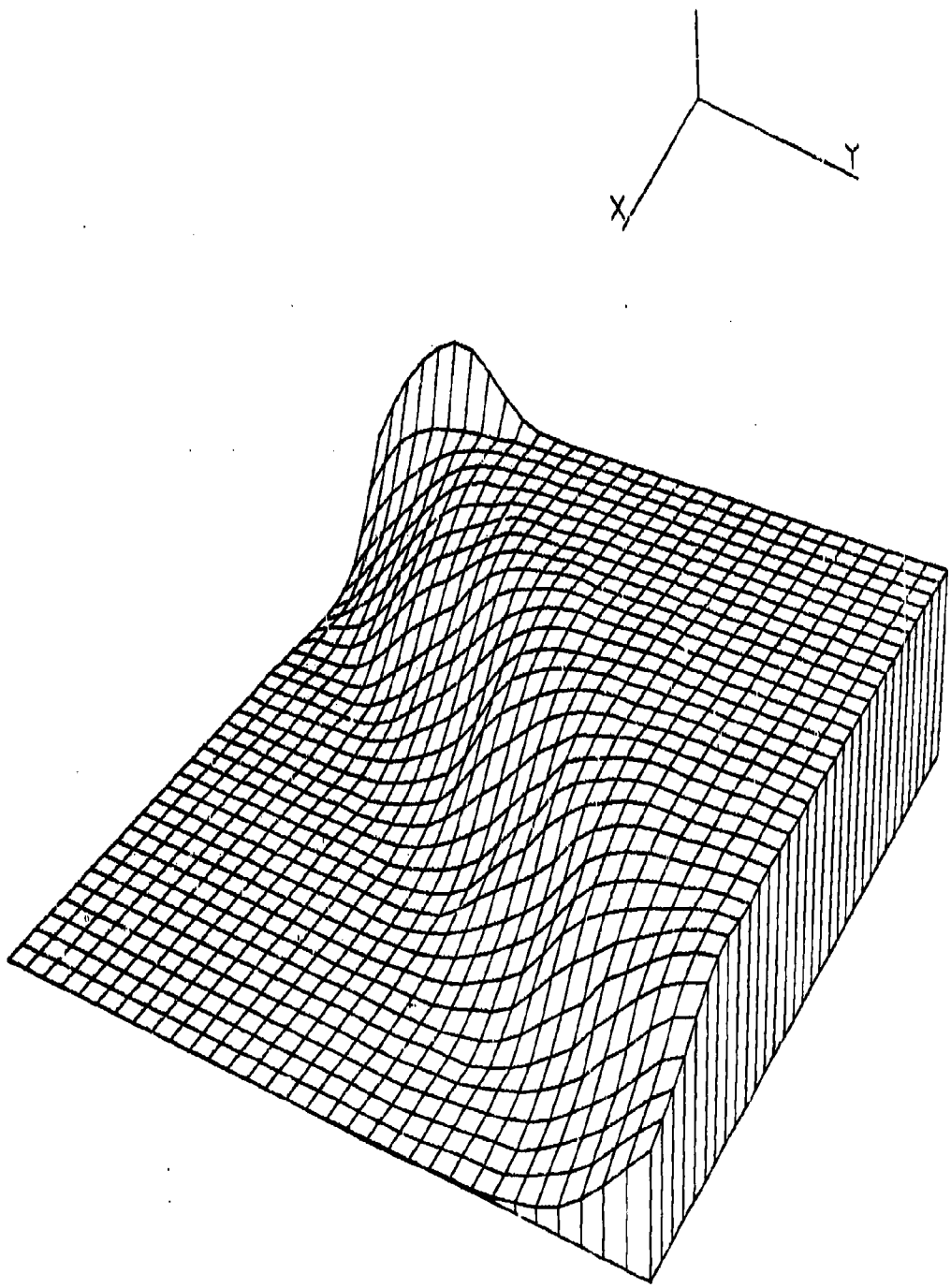
Figure 4.1.2.13



100PT CLIFF FUNCTION 31179 1541

QUADRATIC SHEPARD METHOD NPPR = 918 R = 0.222

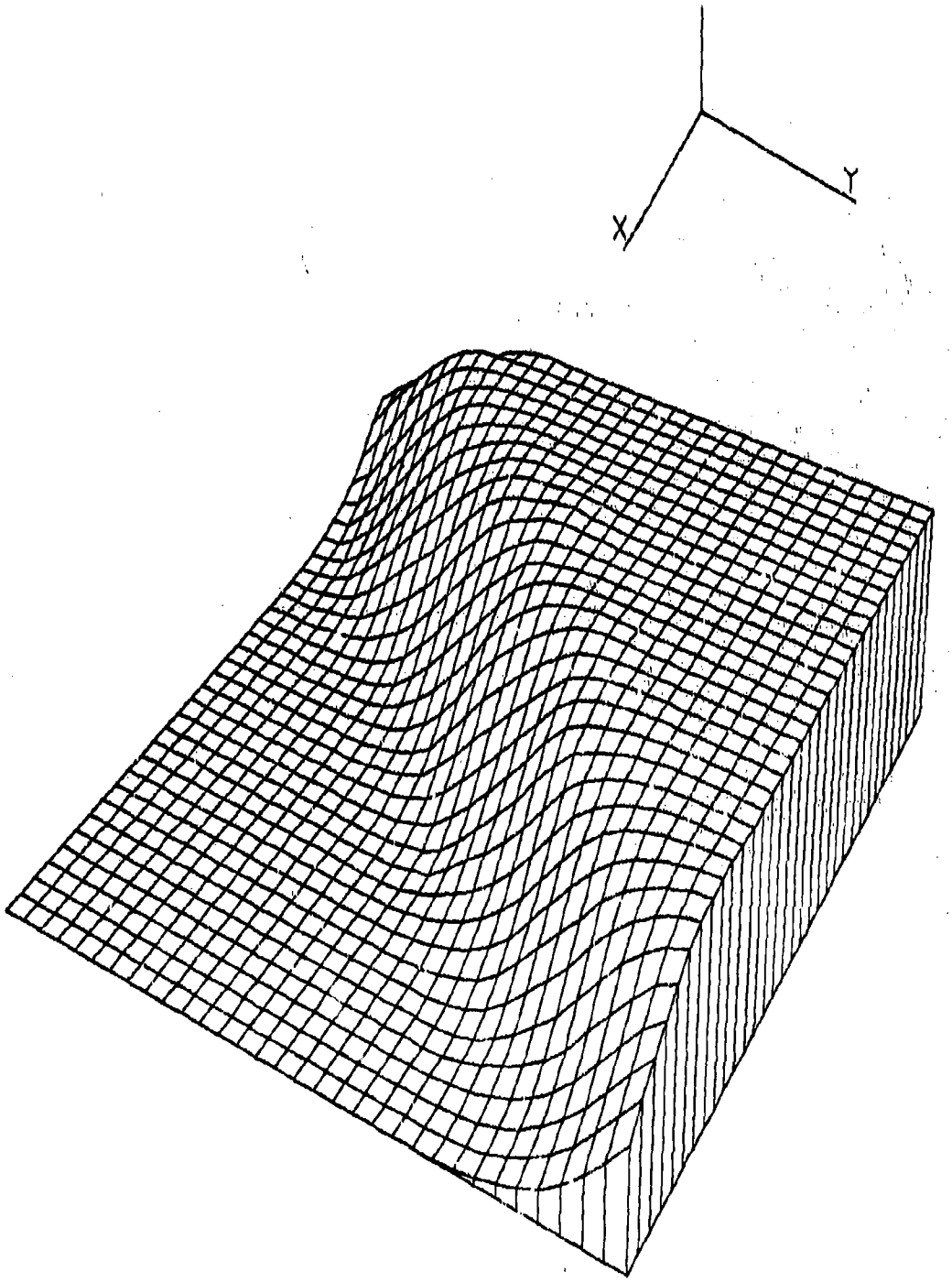
Figure 4.1.2.14



100PT CLIFF FUNCTION 31079 1933

AKIMA - L2 QUAD DERIVS NPPR = 18 R = 0.315

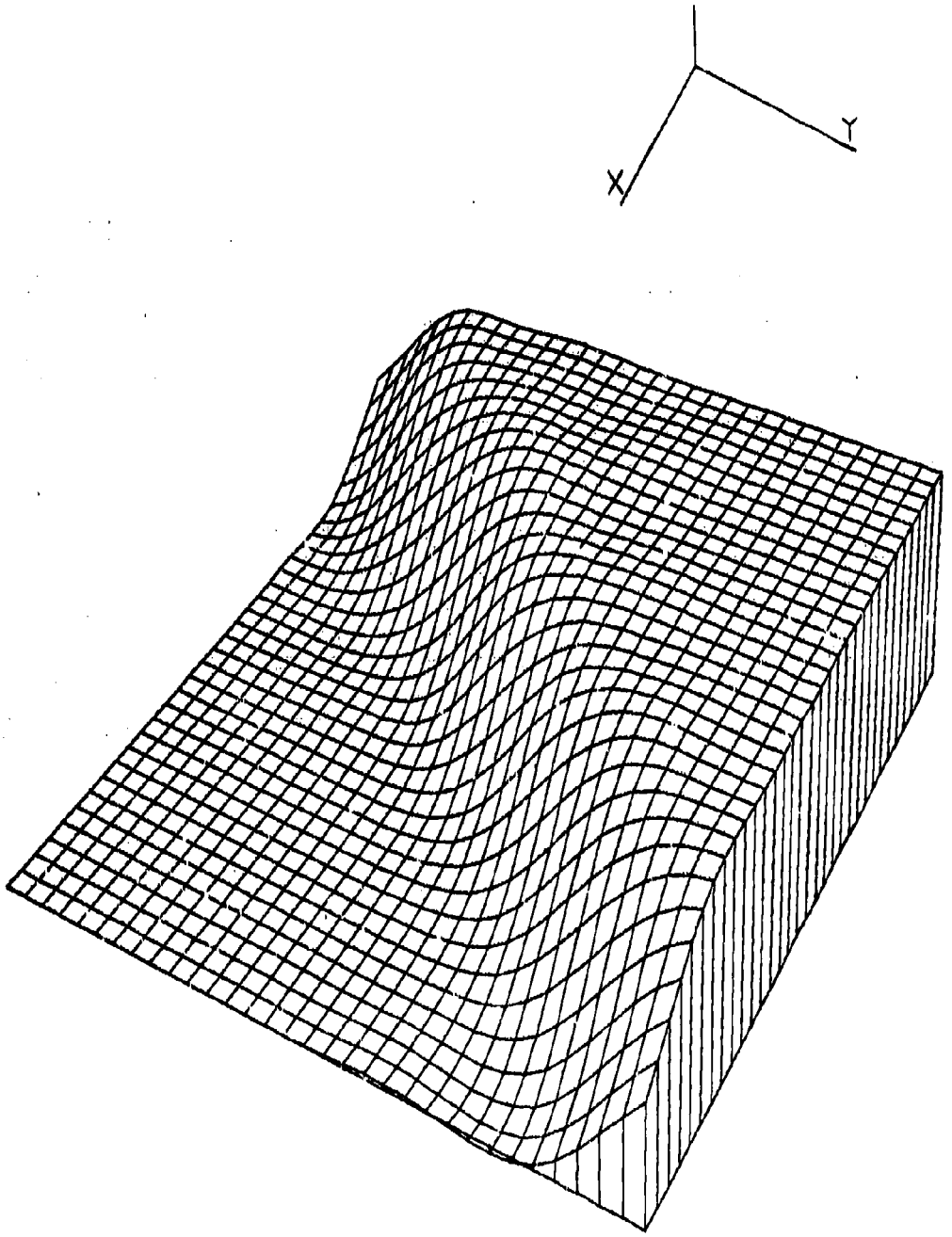
Figure 4.1.2.16



100PT CLIFF FUNCTION 13179 1215

NIELSON MIN NORM NETWORK

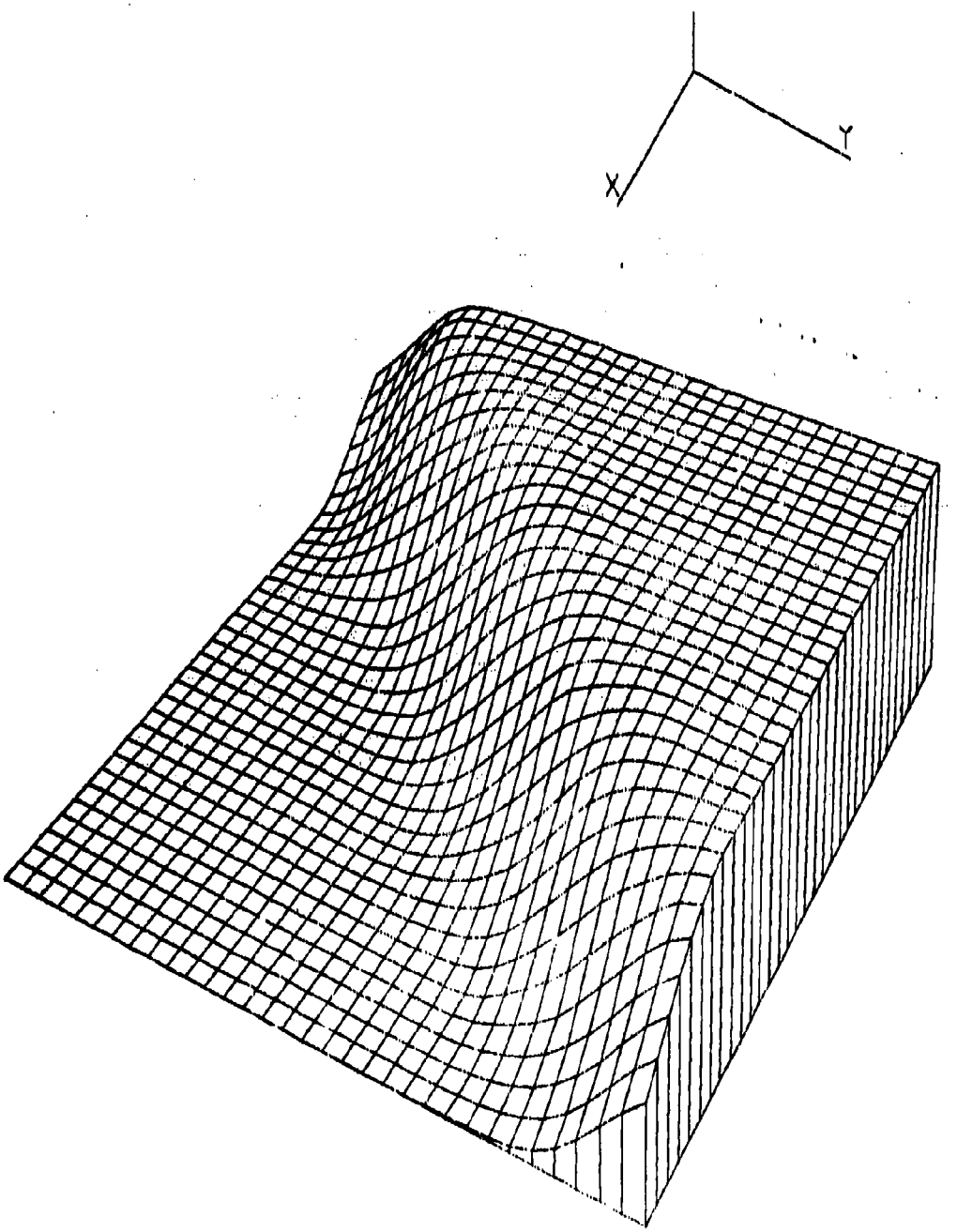
Figure 4.1.2.19



100PT CLIFF FUNCTION 31579 2015

HARDY MQ SURFACE NPPR = 25 R = 0.185

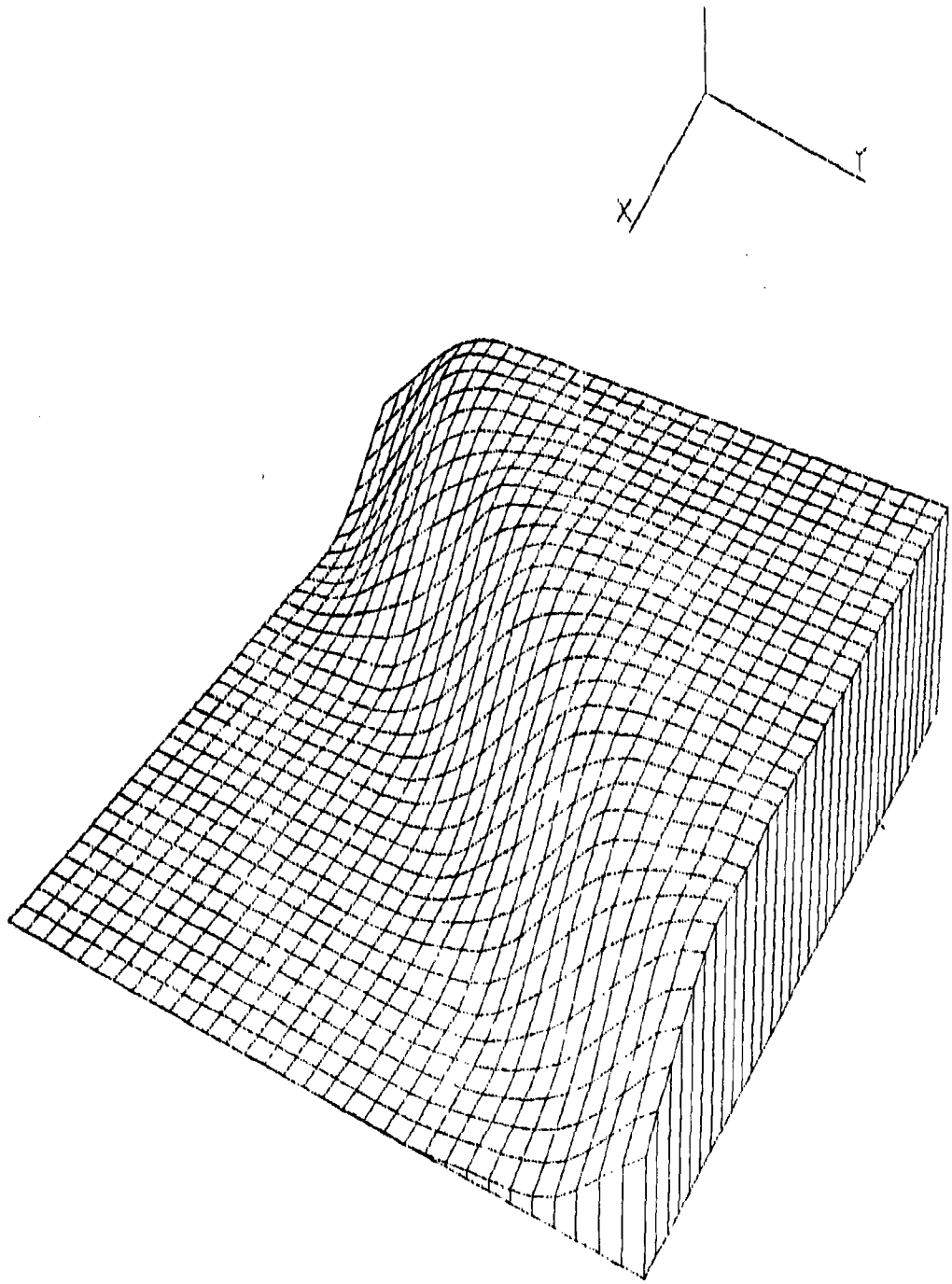
Figure 4.1.2.21



100PT CLIFF FUNCTION 20679 1535

DUCHON S THIN PLATE

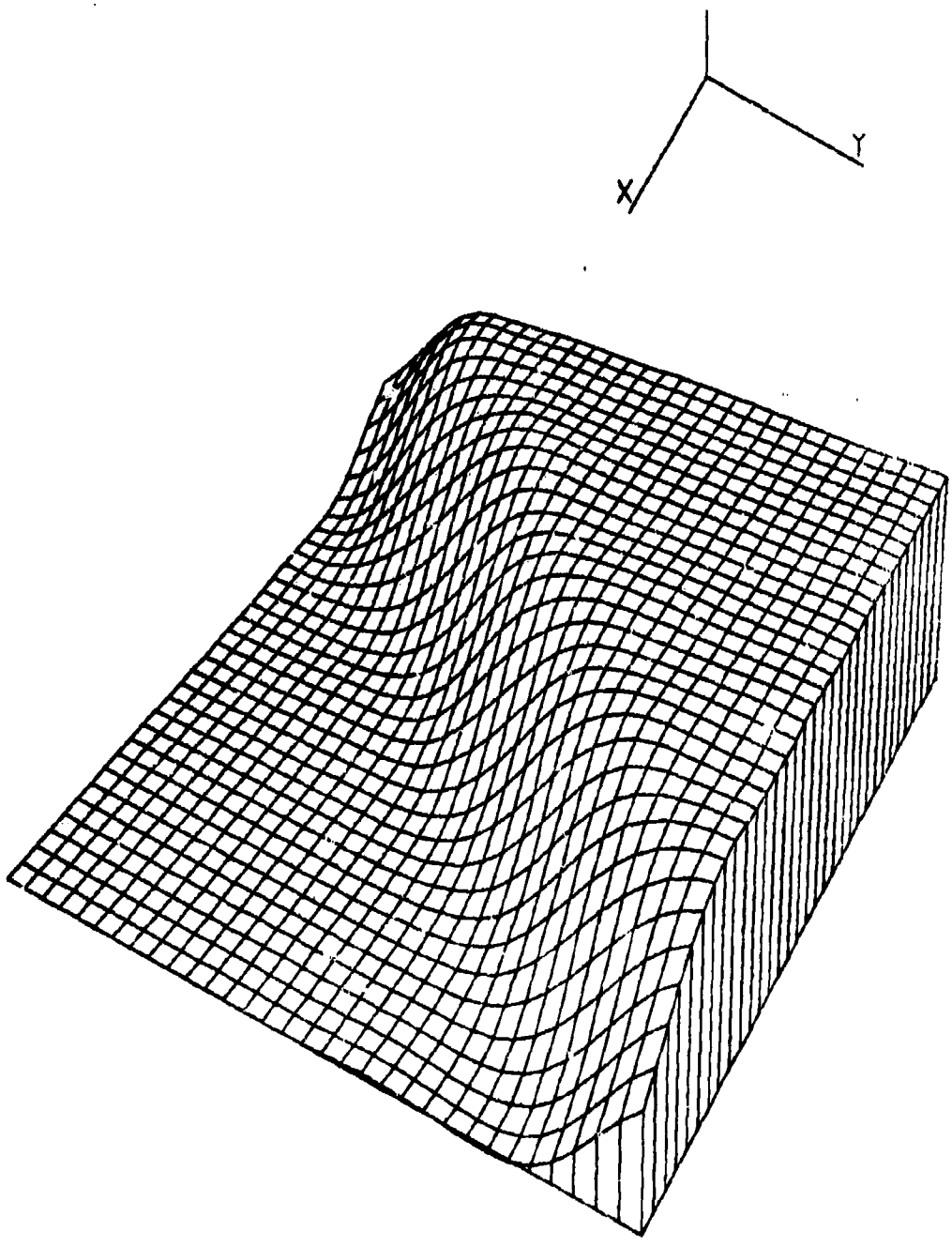
Figure 4.1.2.23



100PT CLIFF FUNCTION 80879 837

FRANKE W/ THIN PLATE NPPR = 6

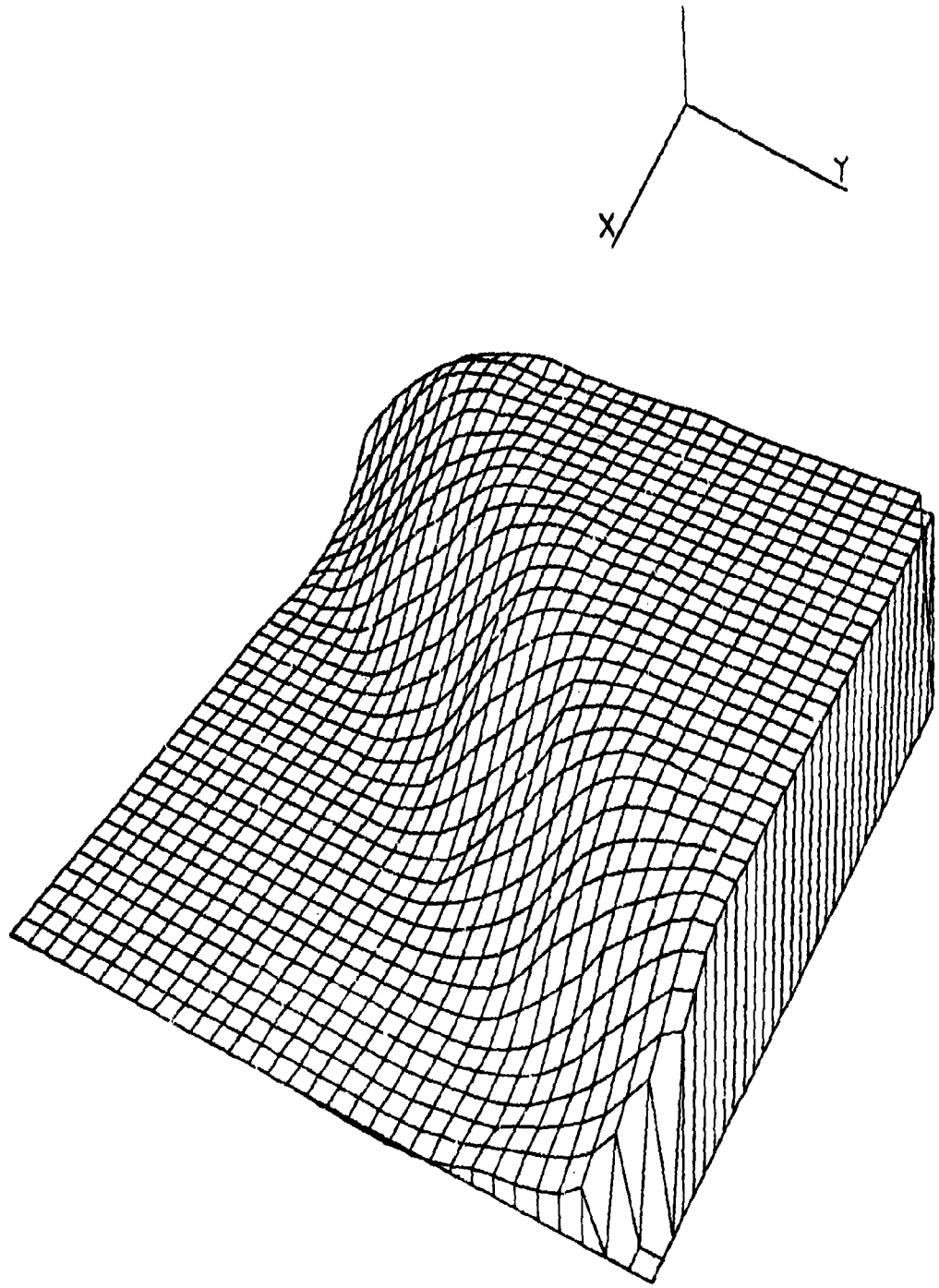
Figure 4.1.2.24



100PT CLIFF FUNCTION 40679 1525

HARDY RECIP MQ SURFACE NPPR = 25 R = 0.185

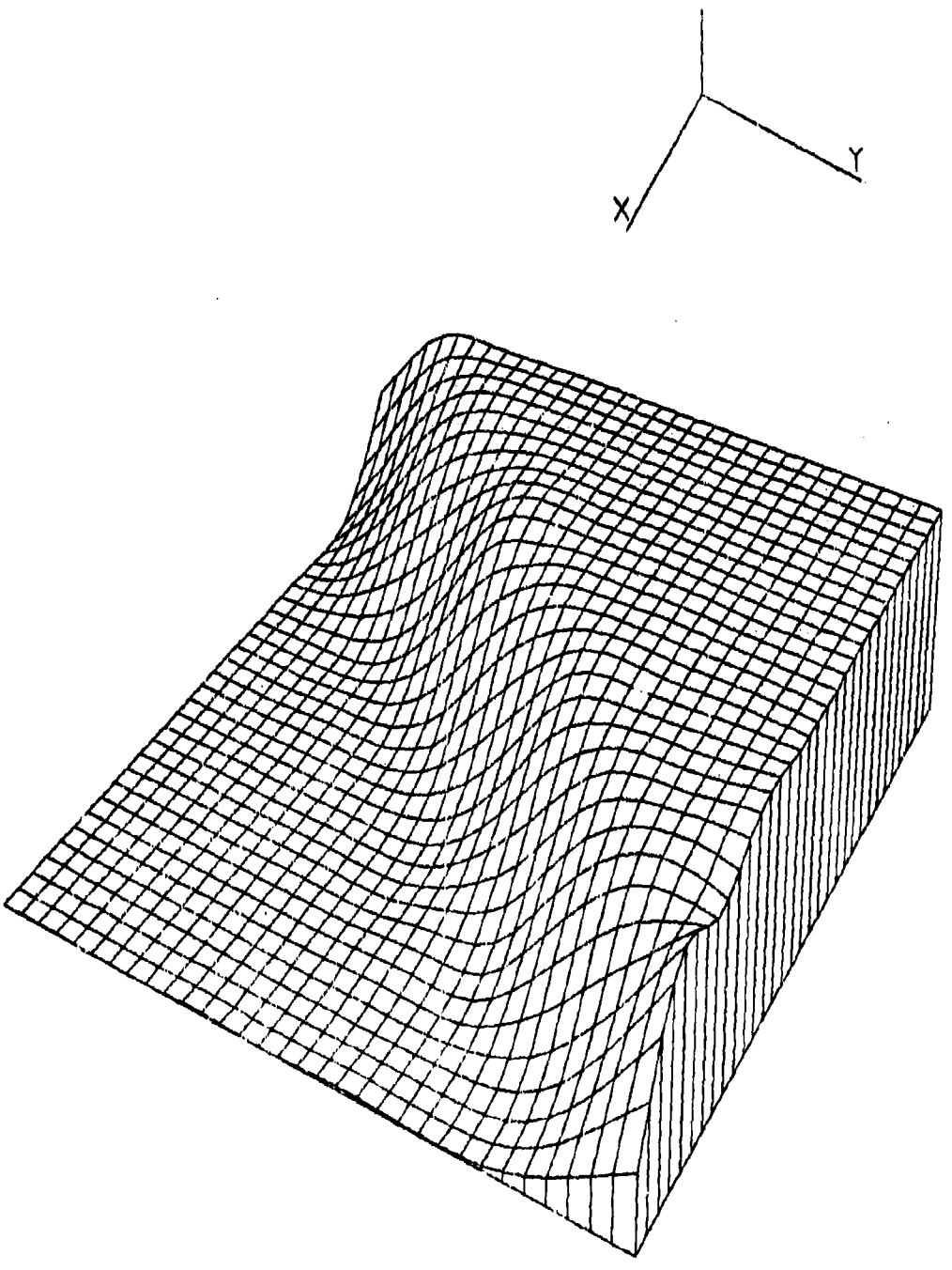
Figure 4.1.2.27



100PT CLIFF FUNCTION 32379 1422

LAWSON TRIANGLE METHOD

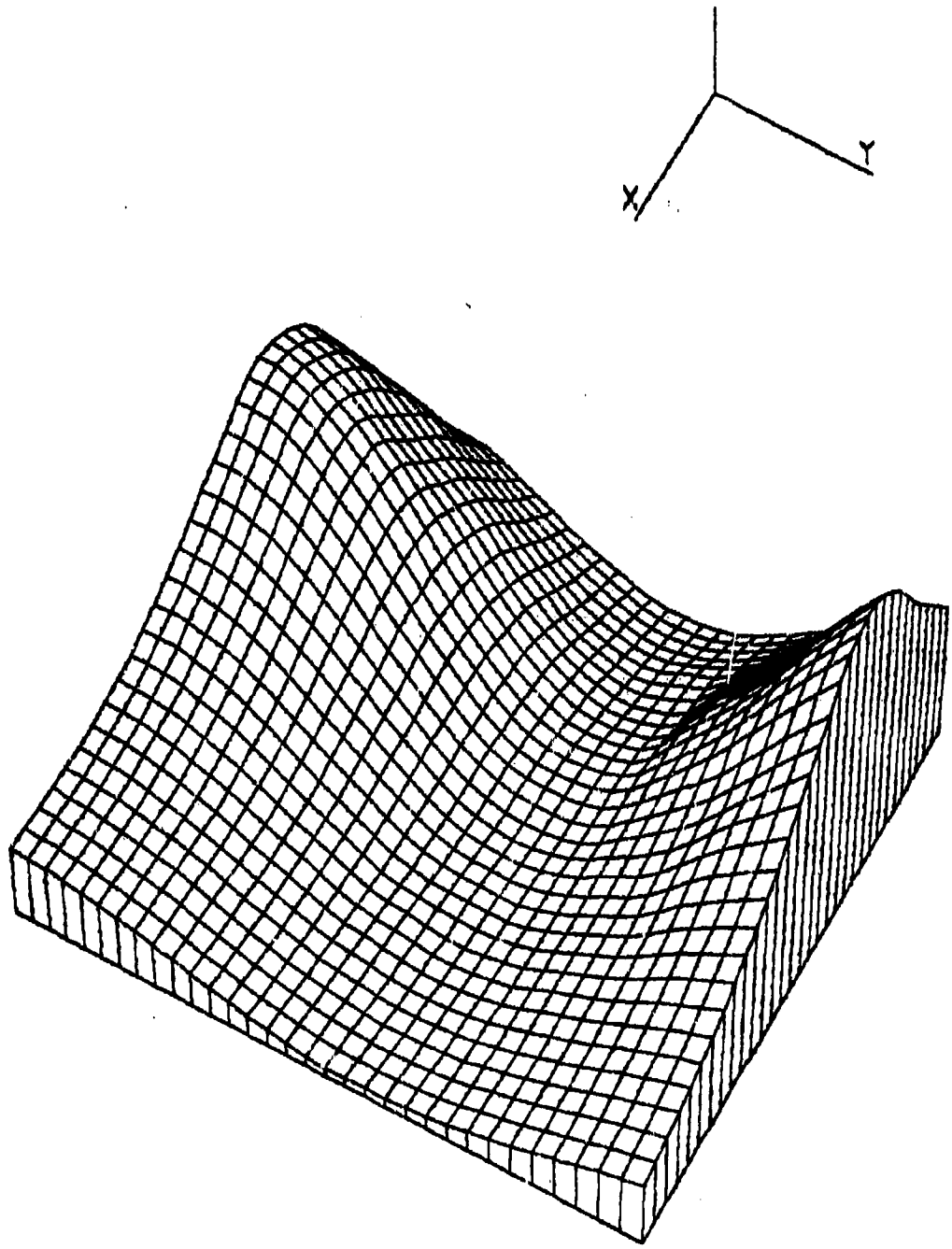
Figure 4.1.2.28



100PT CLIFF FUNCTION 80279 1104

FOLEY ITERATED BICUBICS NPPR = 3

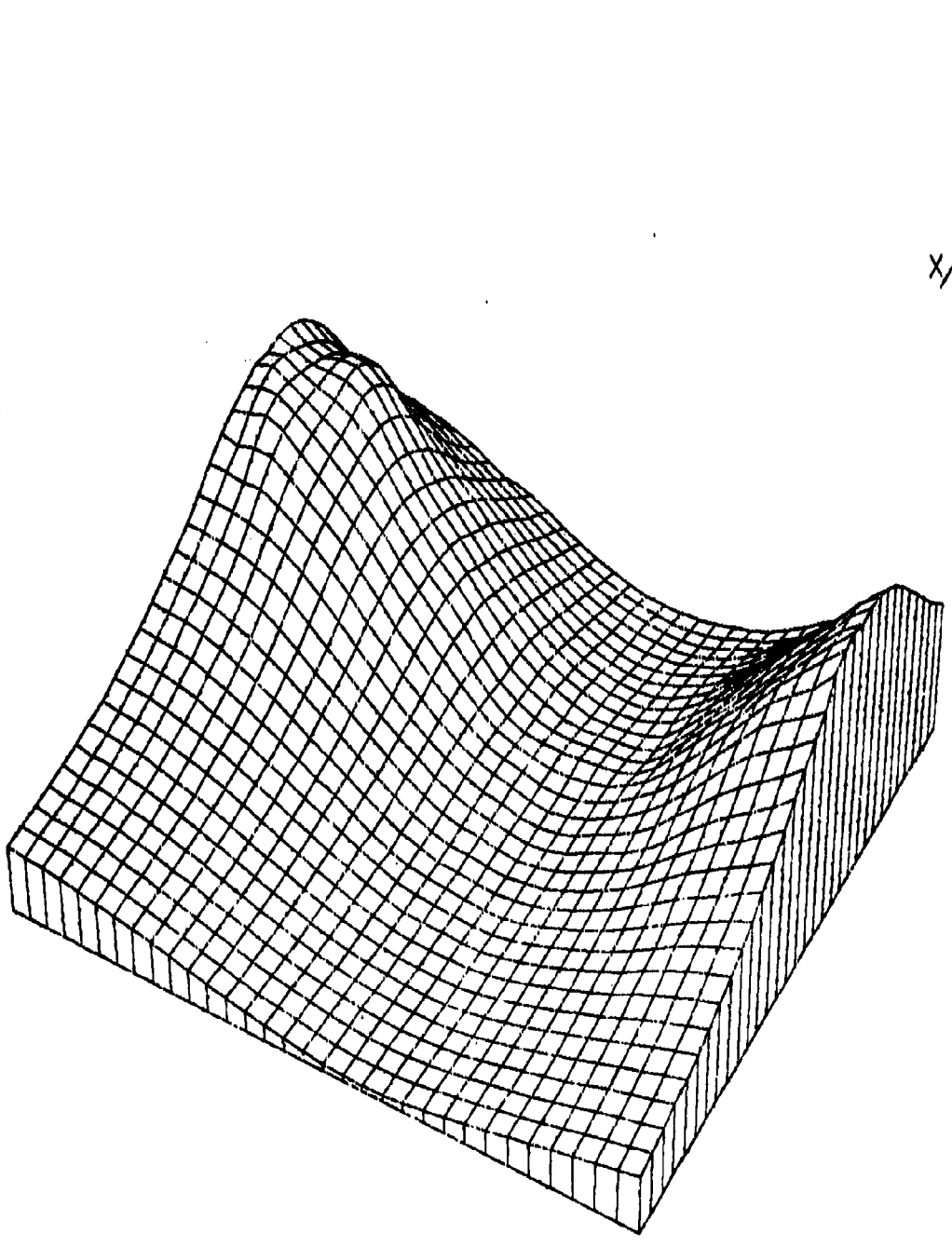
Figure 4.1.2.30



100 PT SADDLE MOUNTAIN THREE 50278 1502

FRANKE - SARD BS PLANE NPPR = 6

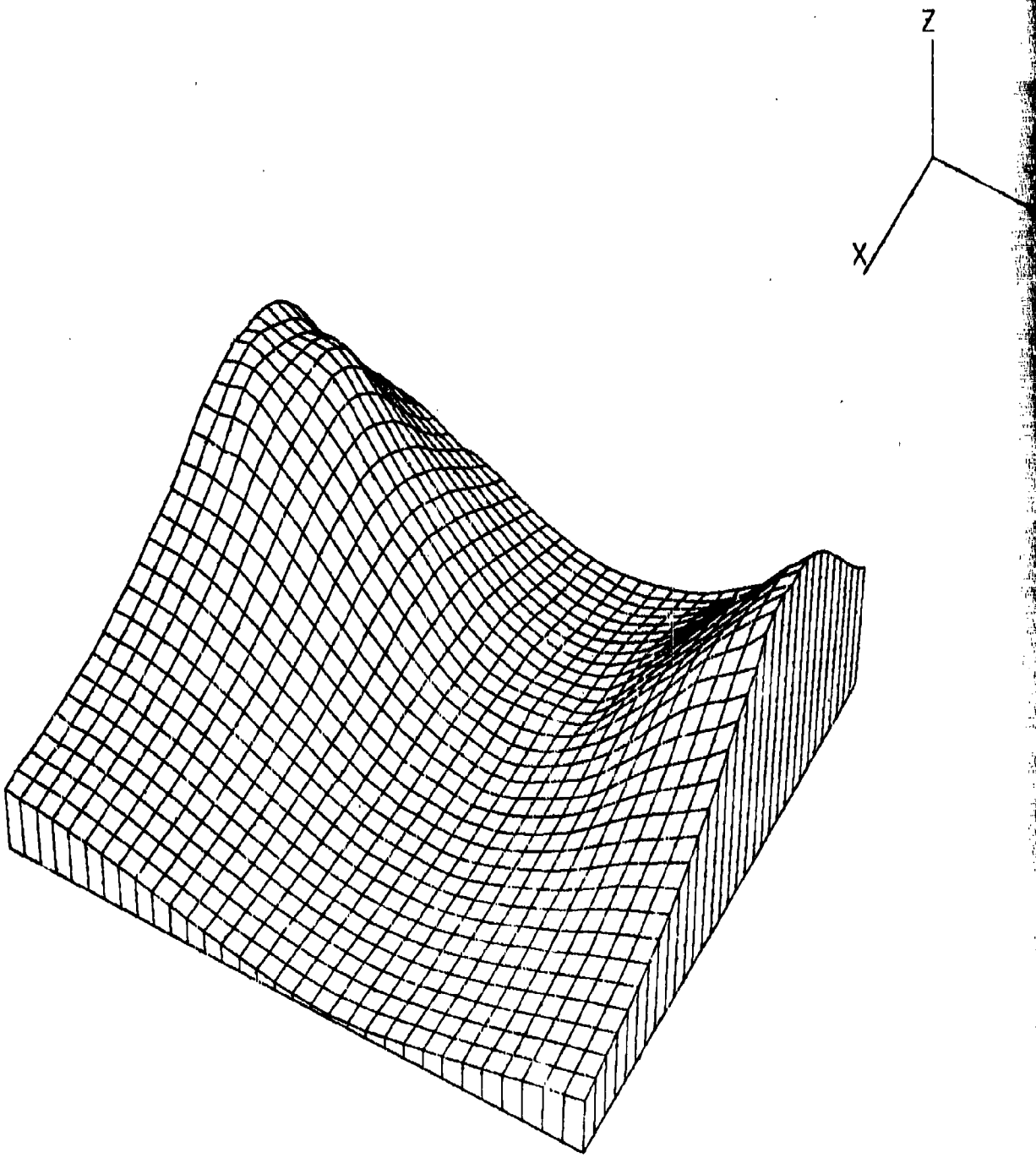
Figure 4.1.3.1



100 PT SADDLE MOUNTAIN THREE 32778 1642

AKIMA - FEM FIT NPPR = 6

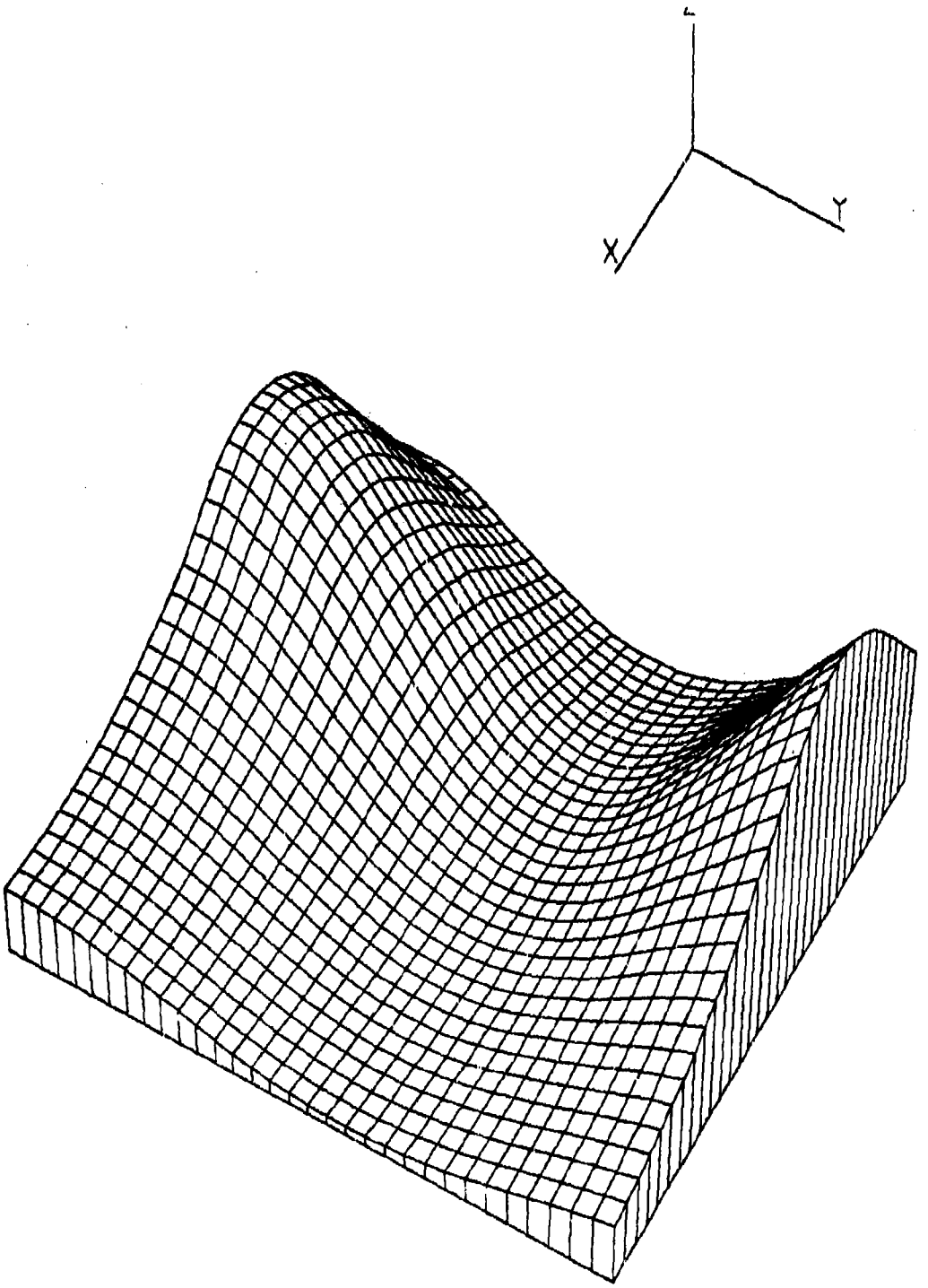
Figure 4.1.3.4



100 PT SADDLE MOUNTAIN THREE 32778 1542

MODIFIED AKIMA FEM FIT NPPR = 6

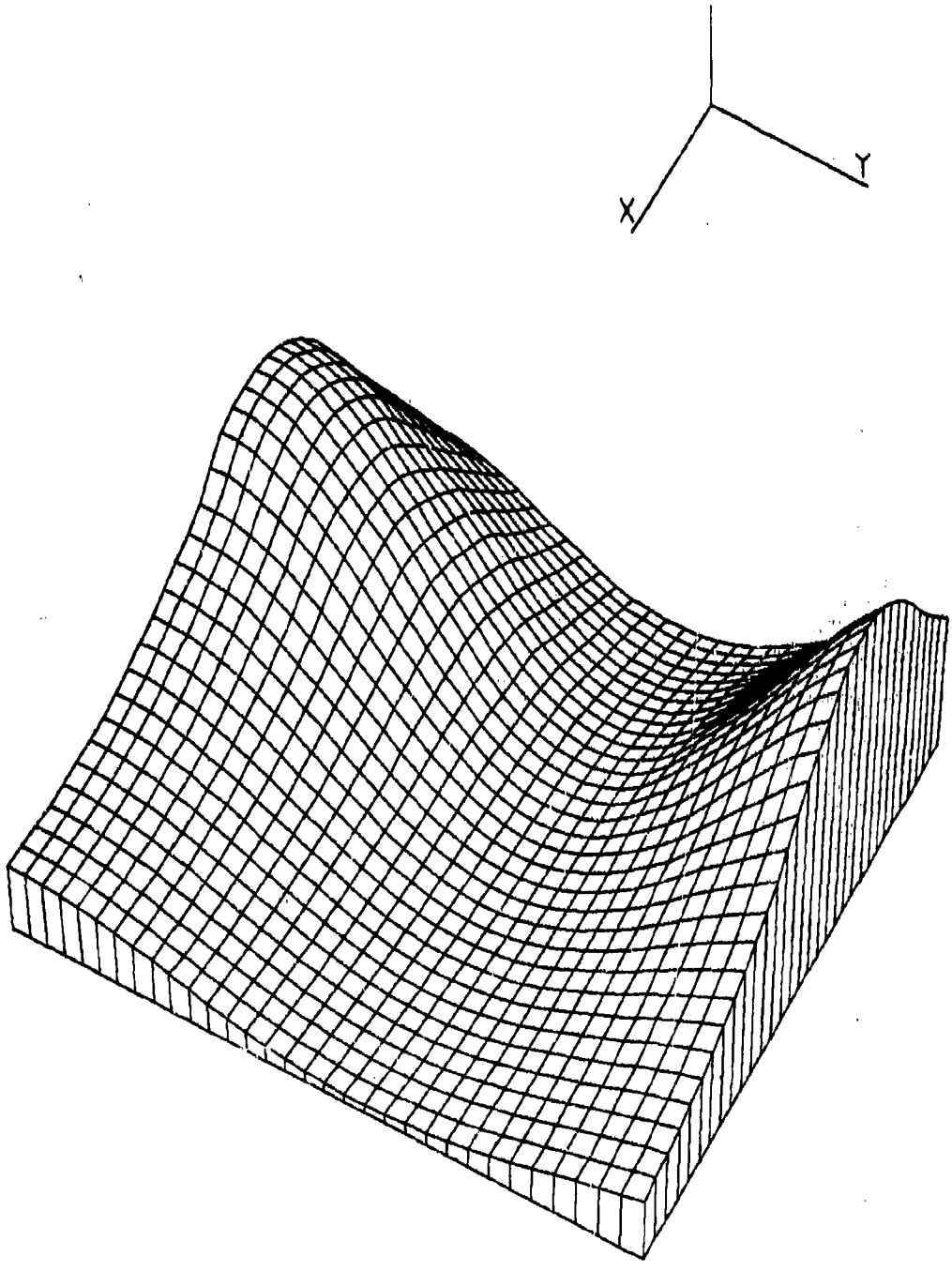
Figure 4.1.3.10



100PT SADDLE MOUNTAIN THREE 31079 1918

NIELSON-FRANKE QUAD-TRI NPPR = 18 R = 0.315

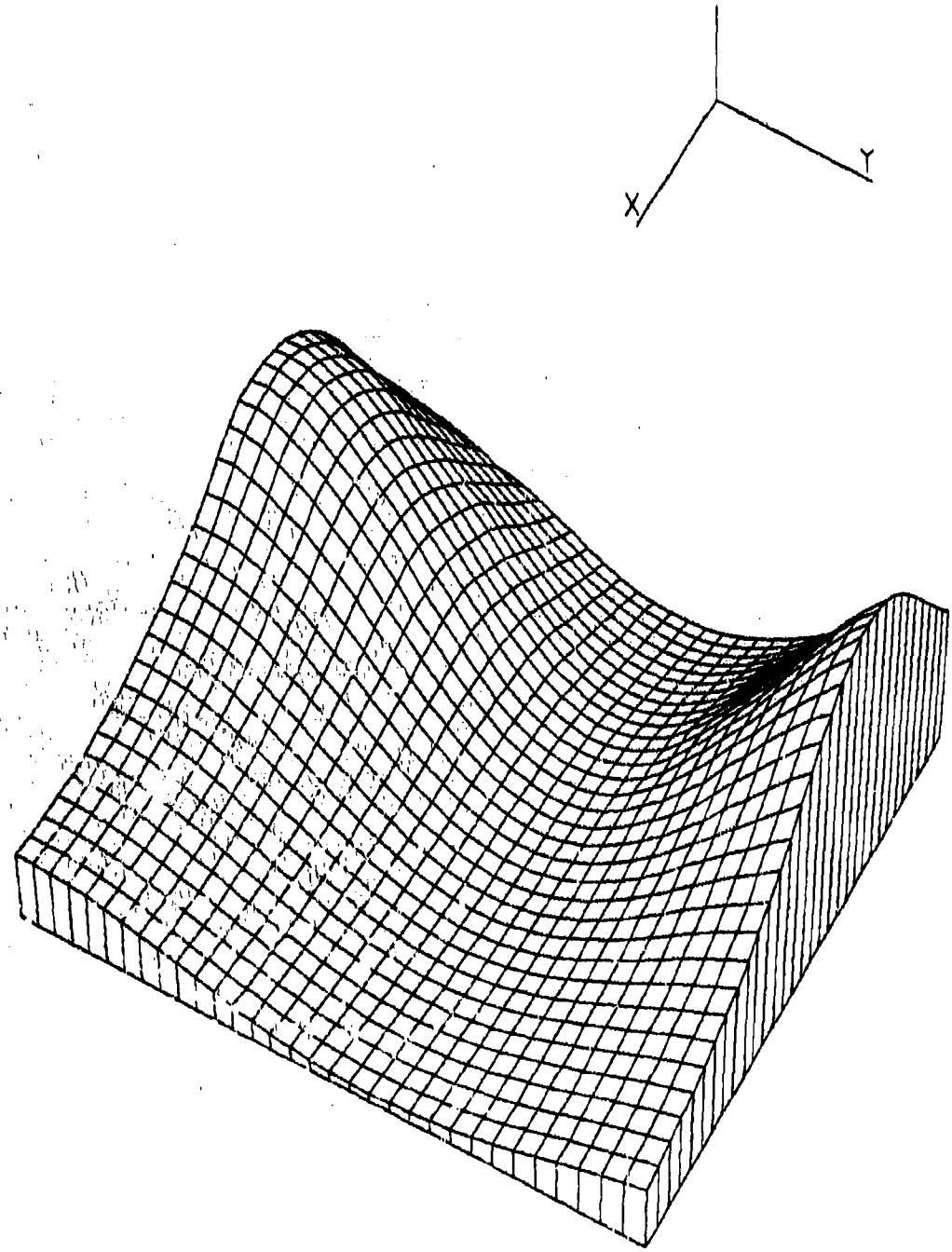
Figure 4.1.3.13



100PT SADDLE MOUNTAIN THREE 31179 1540

QUADRATIC SHEPARD METHOD NPPR = 918 R = 0.222

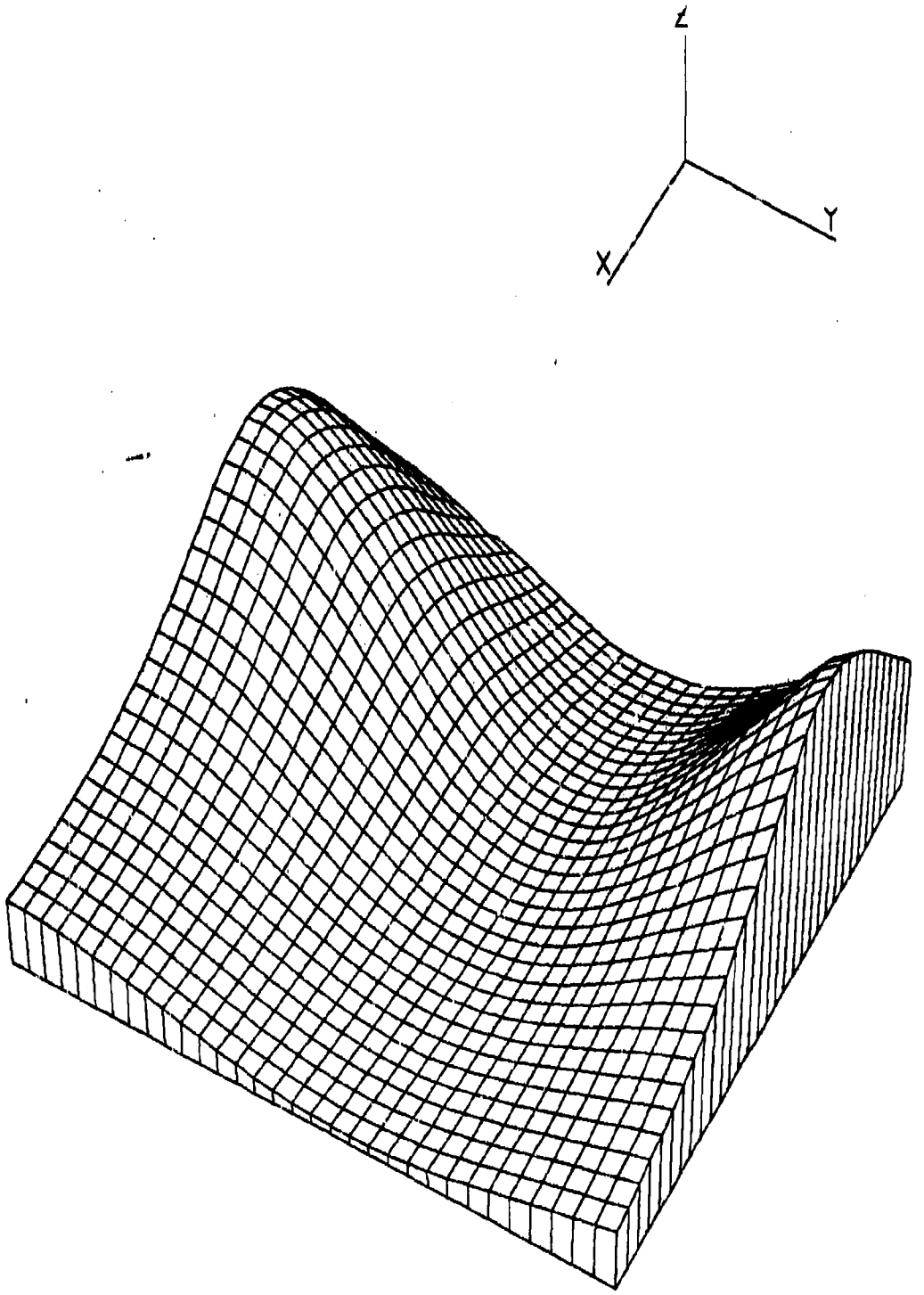
Figure 4.1.3.14



100PT SADDLE MOUNTAIN THREE 31079 1932

AKIMA - L2 QUAD DERIVS NPPR = 18 R = 0.315

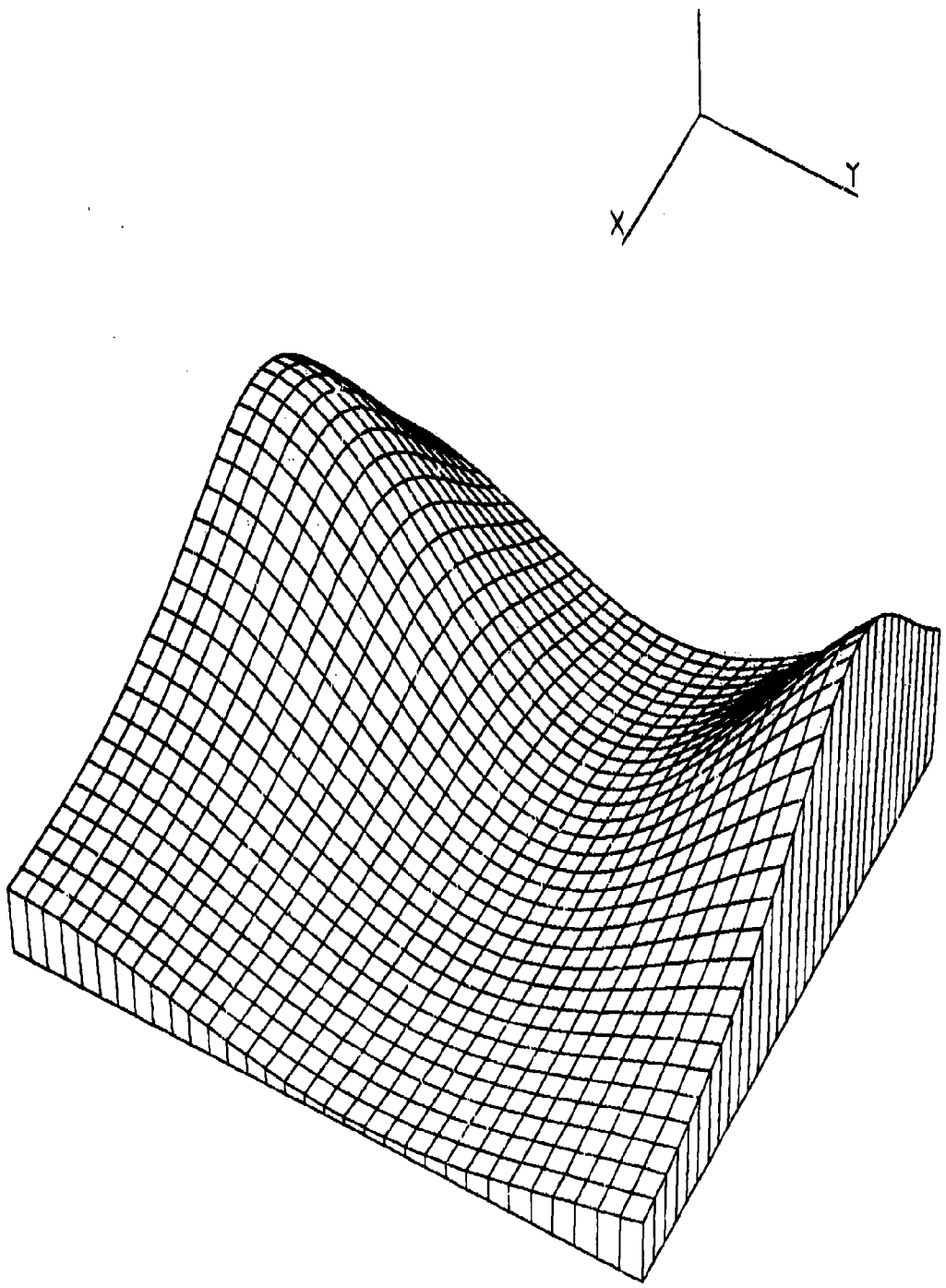
Figure 4.1.3.16



100PT SADDLE MOUNTAIN THREE 13179 1216

NIELSON MIN NORM NETWORK

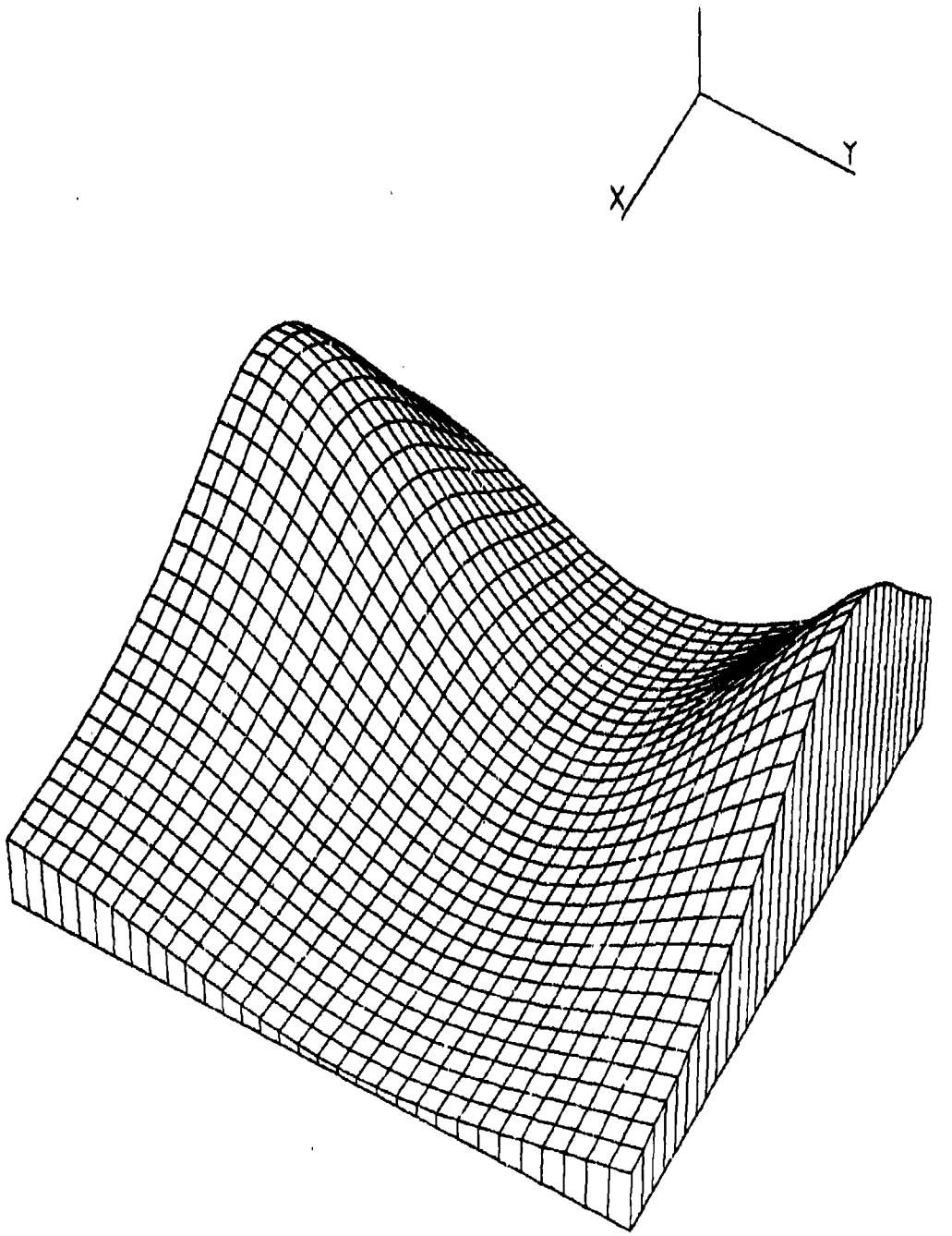
Figure 4.1.3.19



100PT SADDLE MOUNTAIN THREE 31579 2014

HARDY MQ SURFACE NPPR = 25 R = 0.185

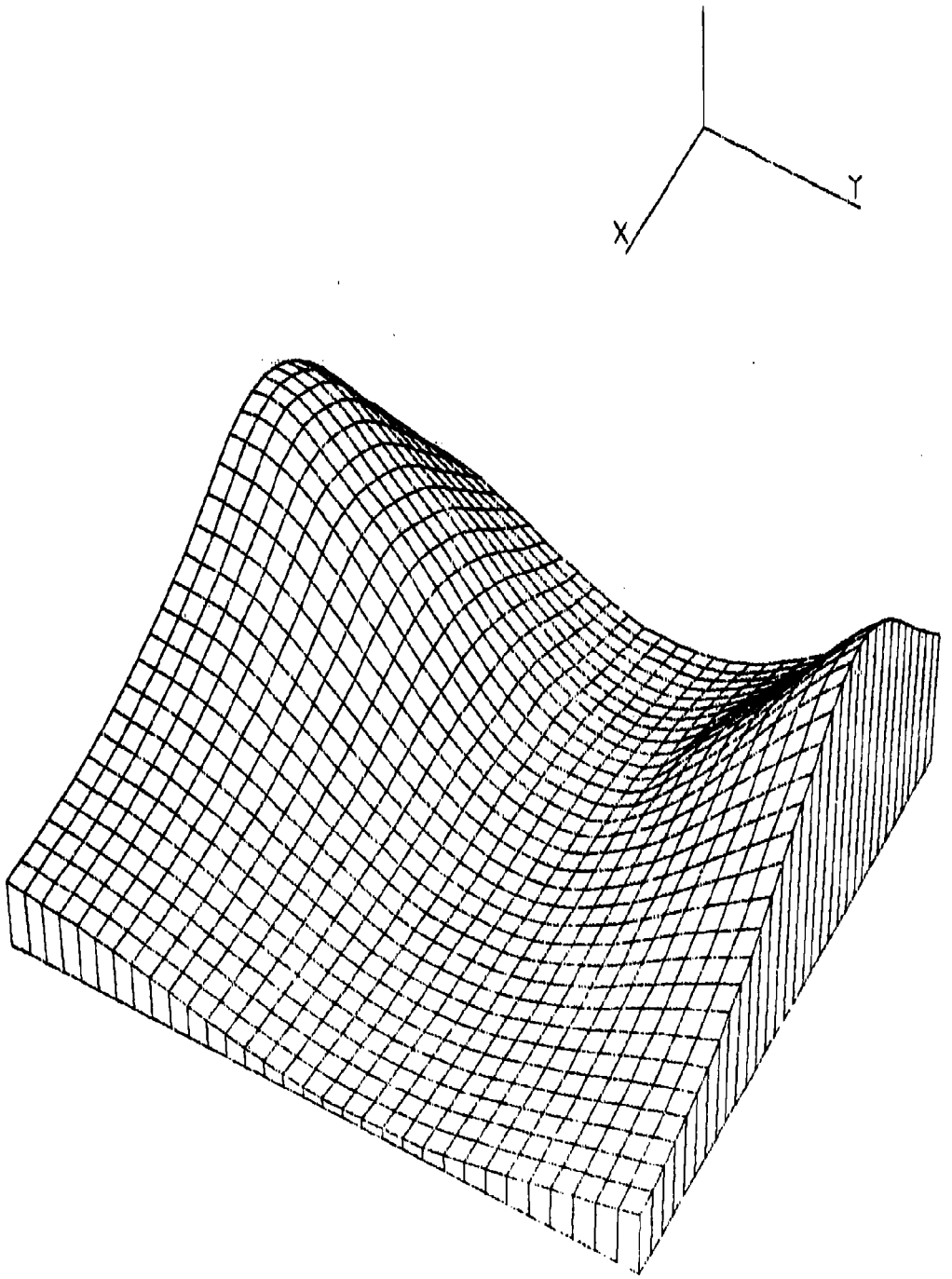
Figure 4.1.3.21



100PT SADDLE MOUNTAIN THREE 20779 1031

DUCHON S THIN PLATE

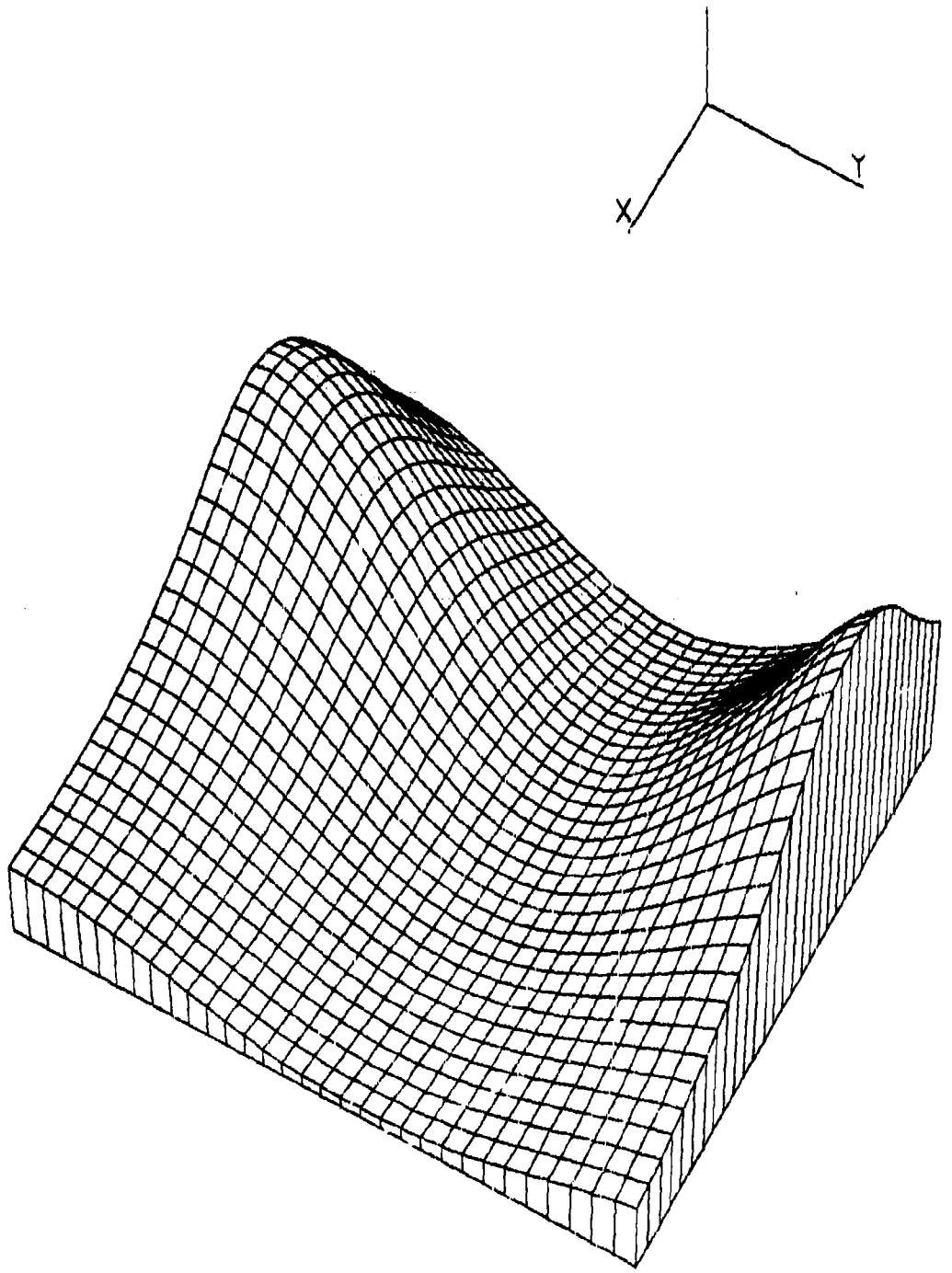
Figure 4.1.3.23



100PT SADDLE MOUNTAIN THREE 80879 836

FRANKE W/ THIN PLATE NPPR = 6

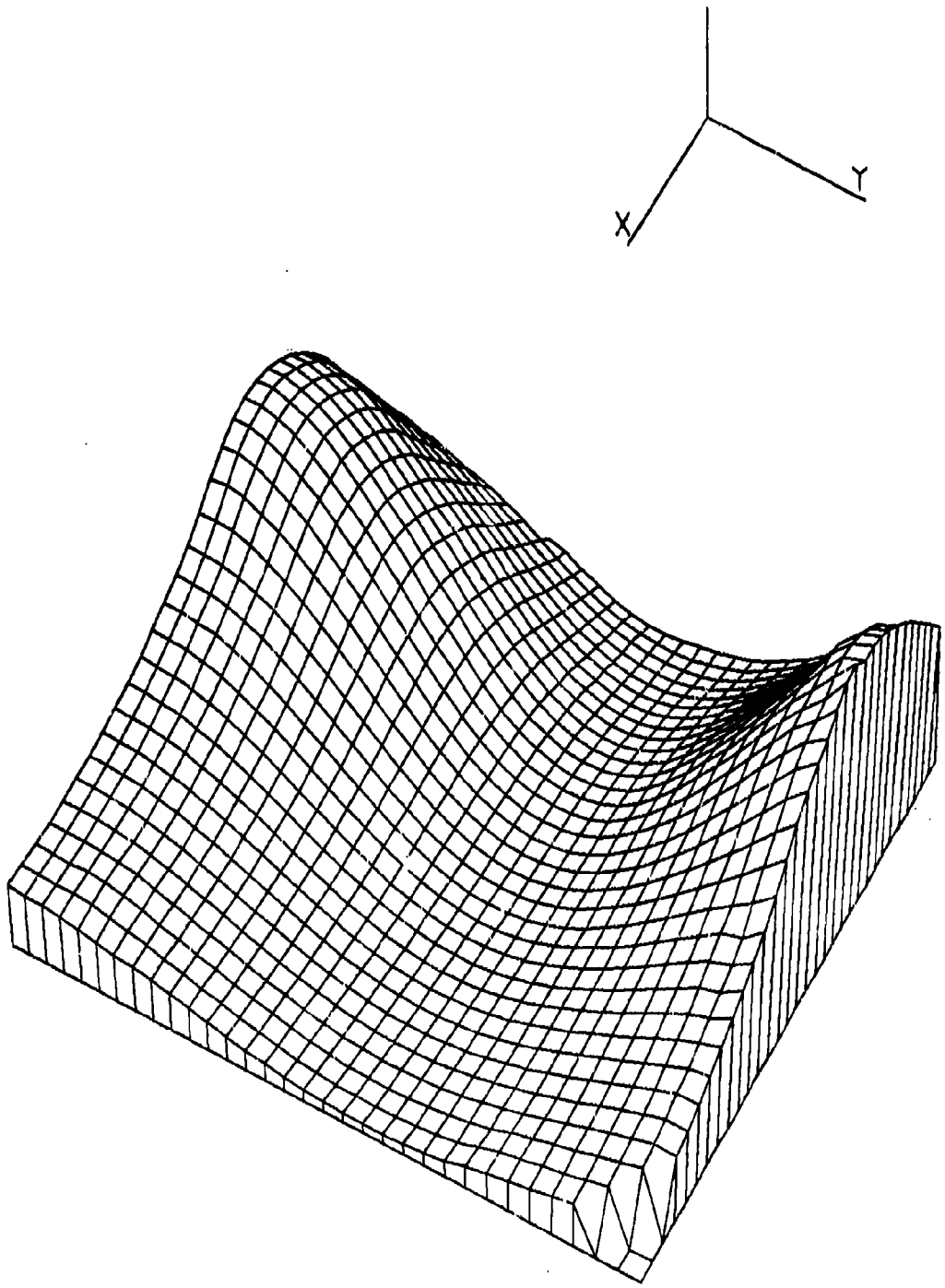
Figure 4.1.3.24



100PT SADDLE MOUNTAIN THREE 40679 1520

HARDY RECIP MQ SURFACE NPPR = 25 R = 0.185

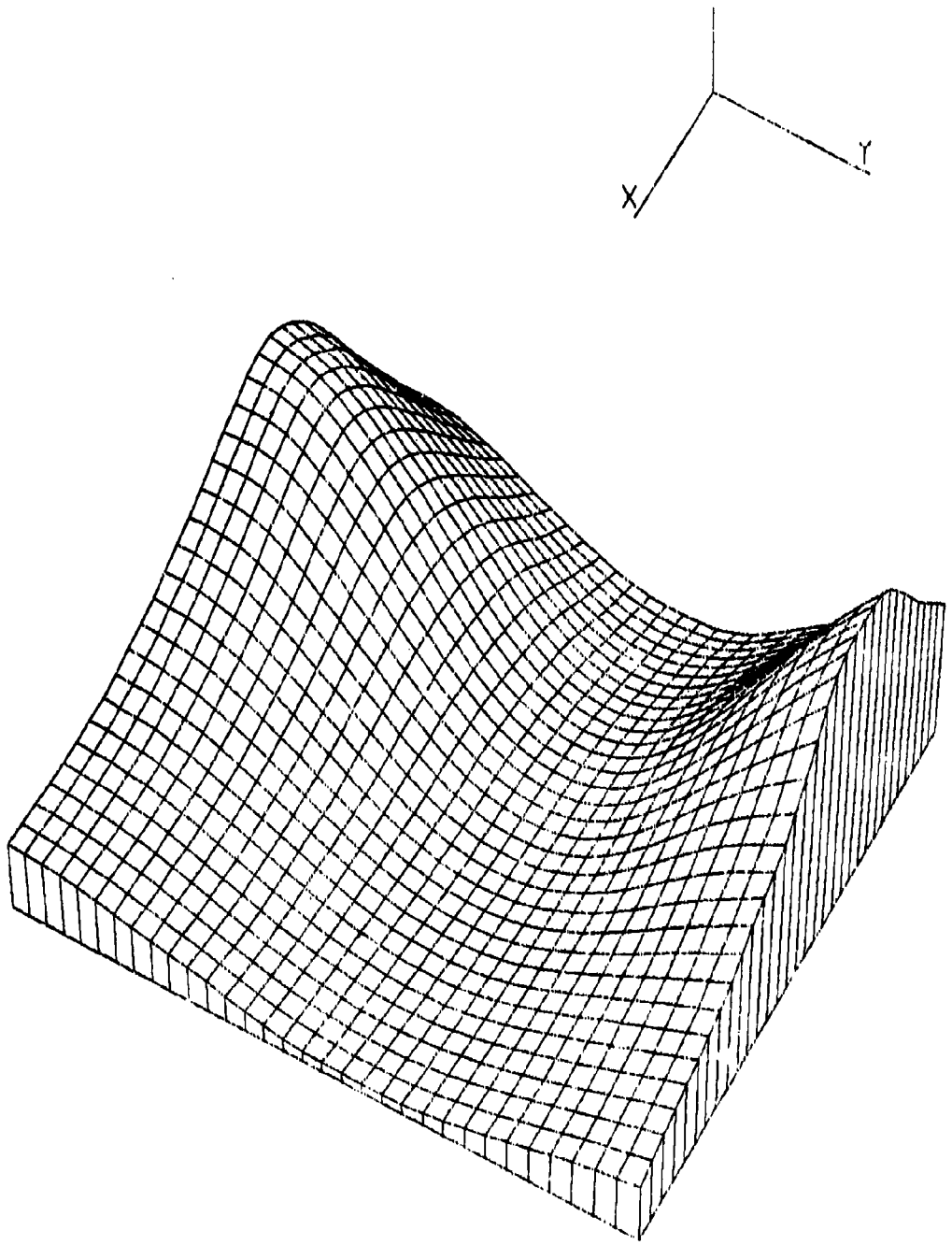
Figure 4.1.3.27



100PT SADDLE MOUNTAIN THREE 32379 1421

LAWSON TRIANGLE METHOD

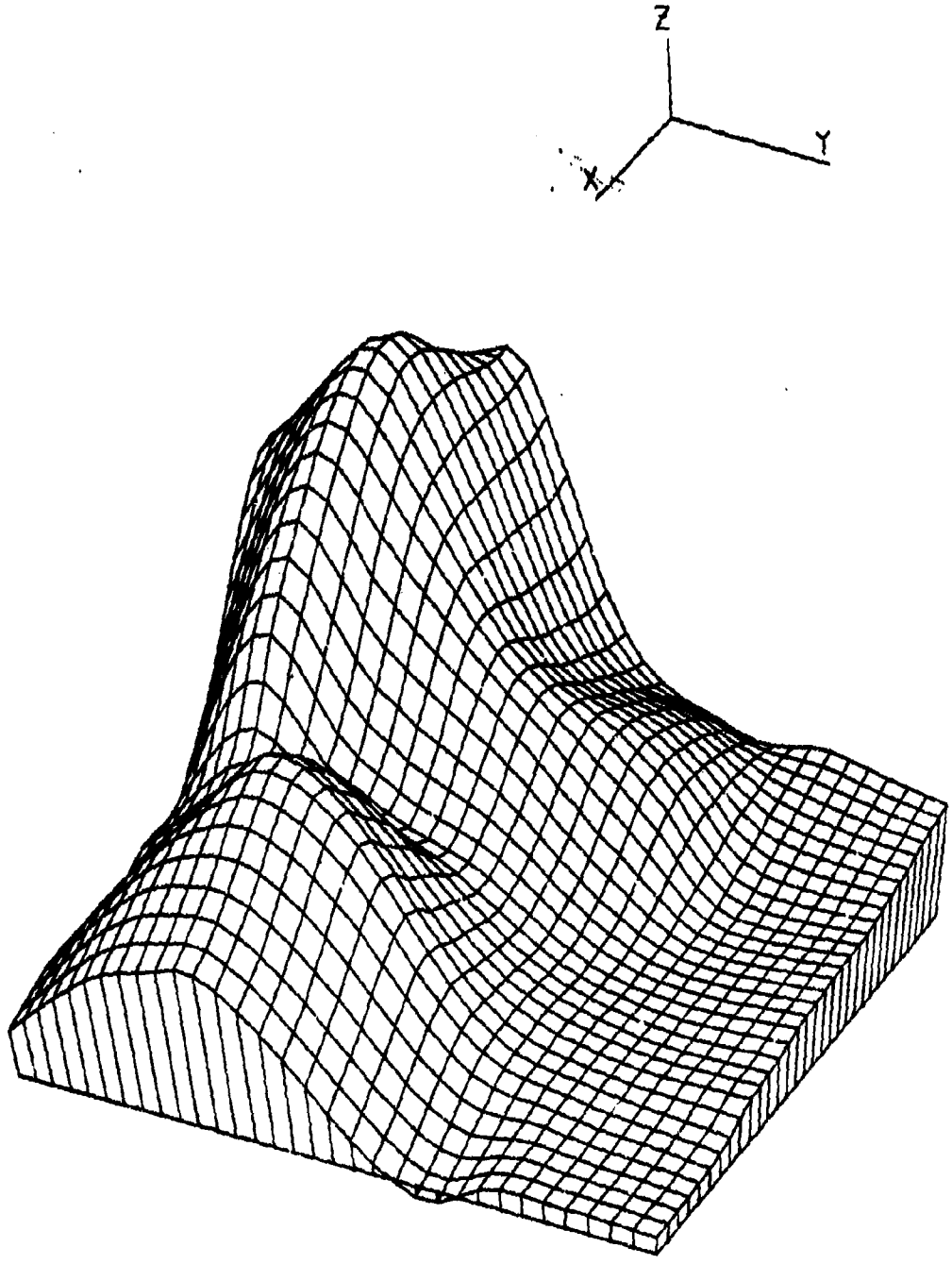
Figure 4.1.3.28



100PT SADDLE MOUNTAIN THREE 80279 1101

FOLEY ITERATED BICUBICS NPPR = 3

Figure 4.1.3.30



33PT EXP HUMPS, DIP ON SPARSE 21679 942

FRANKE - SARD BS PLANE NPPR = 6

Figure 4.2.1.1

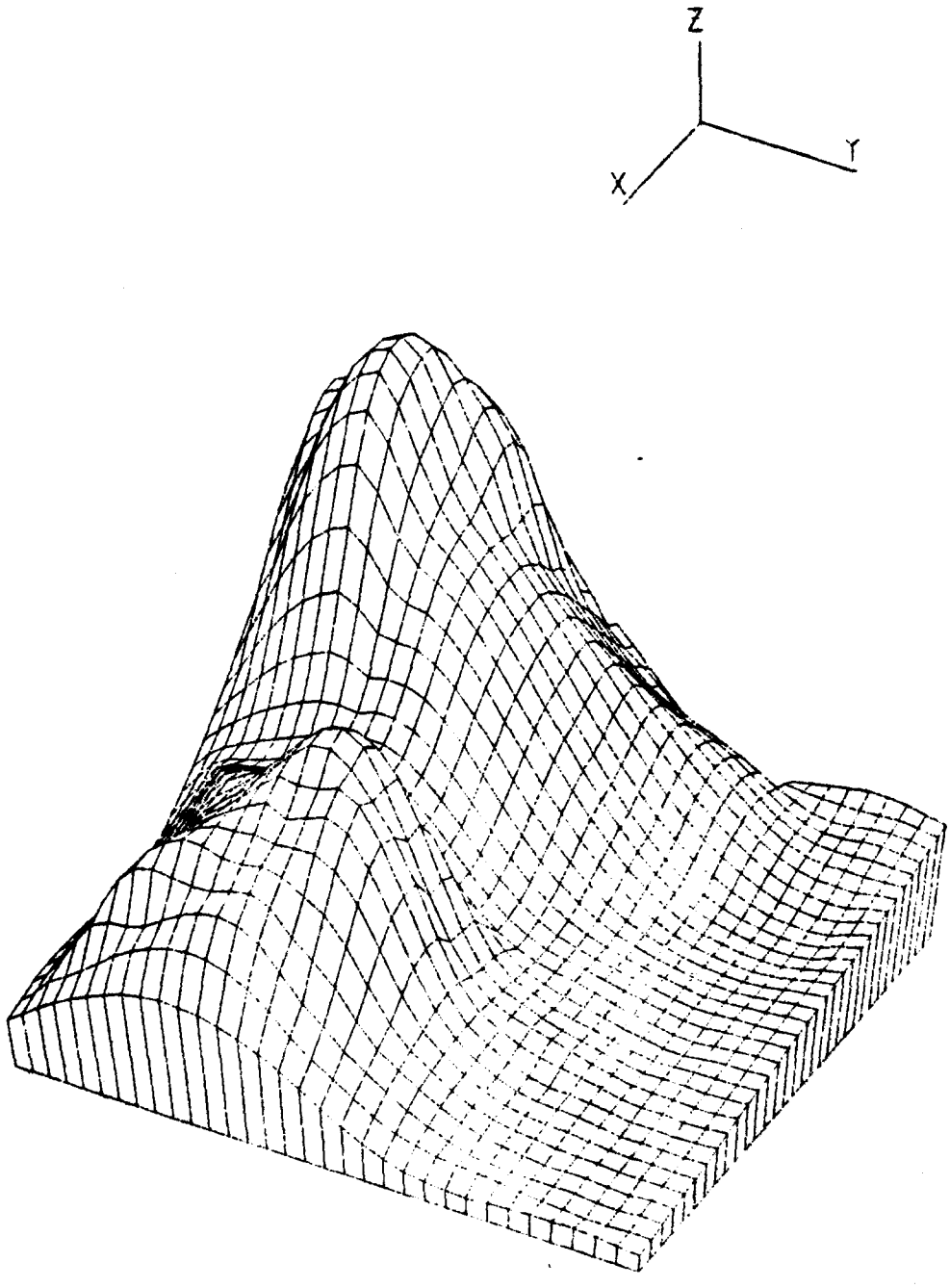
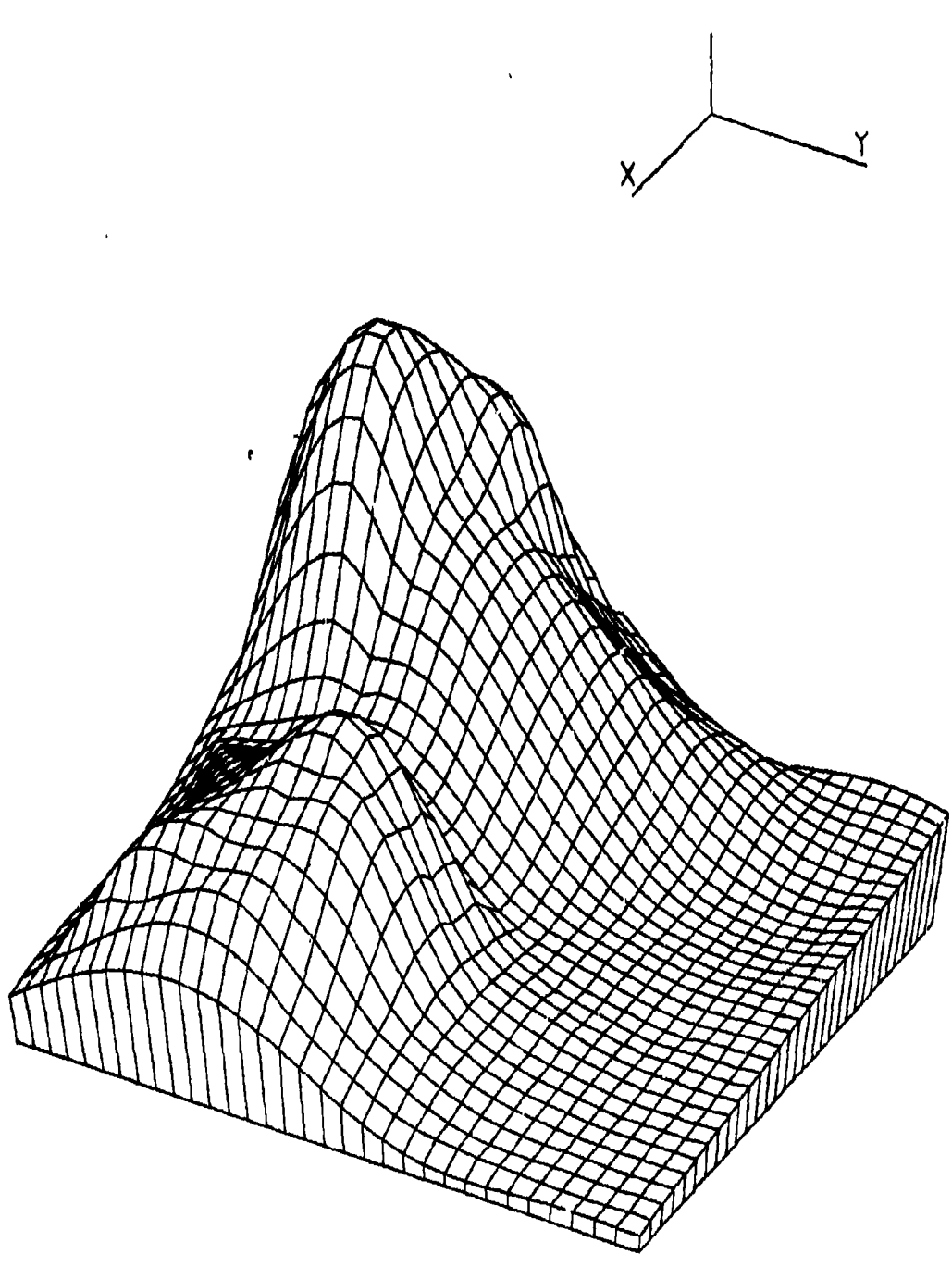


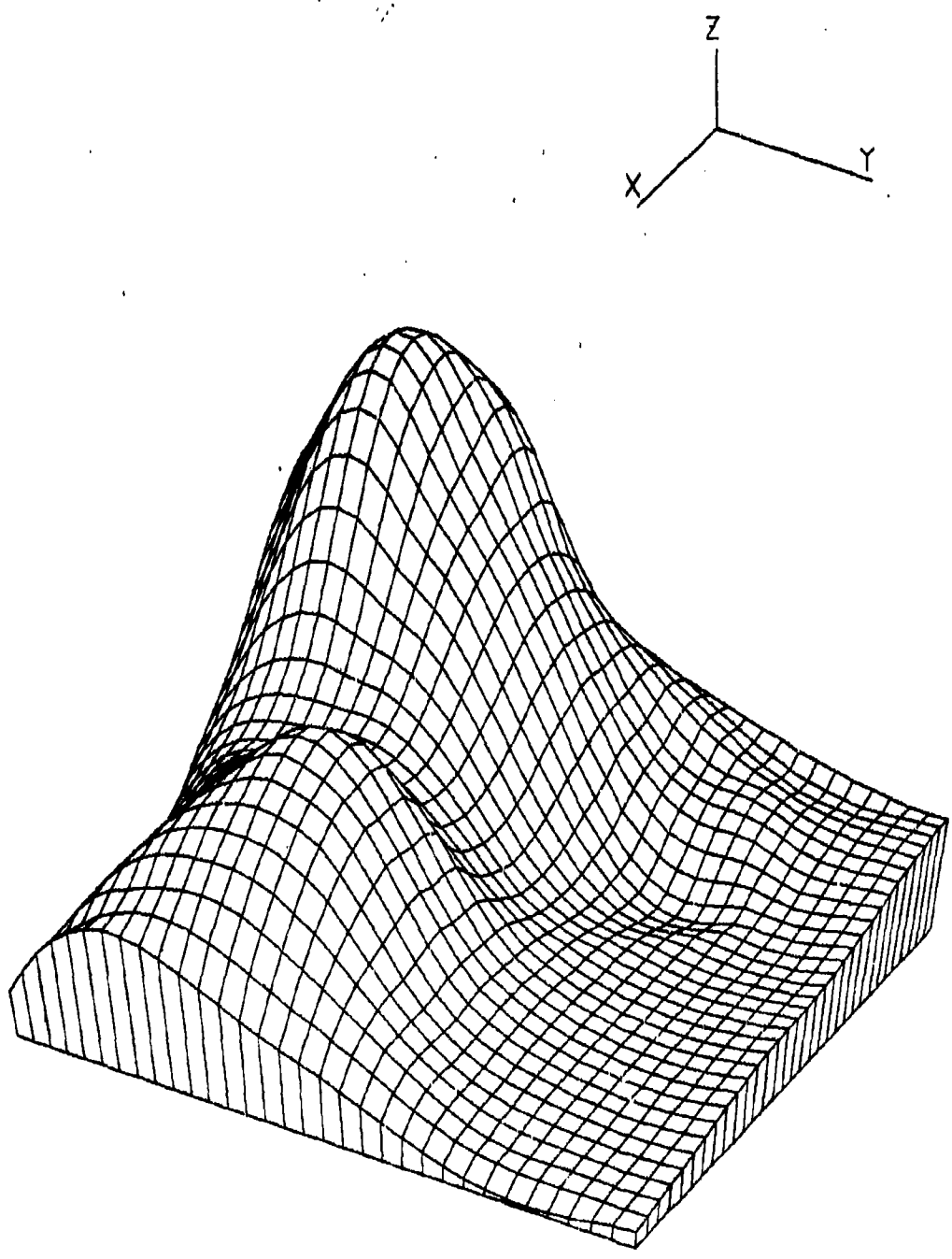
Figure 4.2.1.4



33PT EXP HUMPS, DIP ON SPARSE 30779 1333

MODIFIED AKIMA FEM FIT NPPR = 6

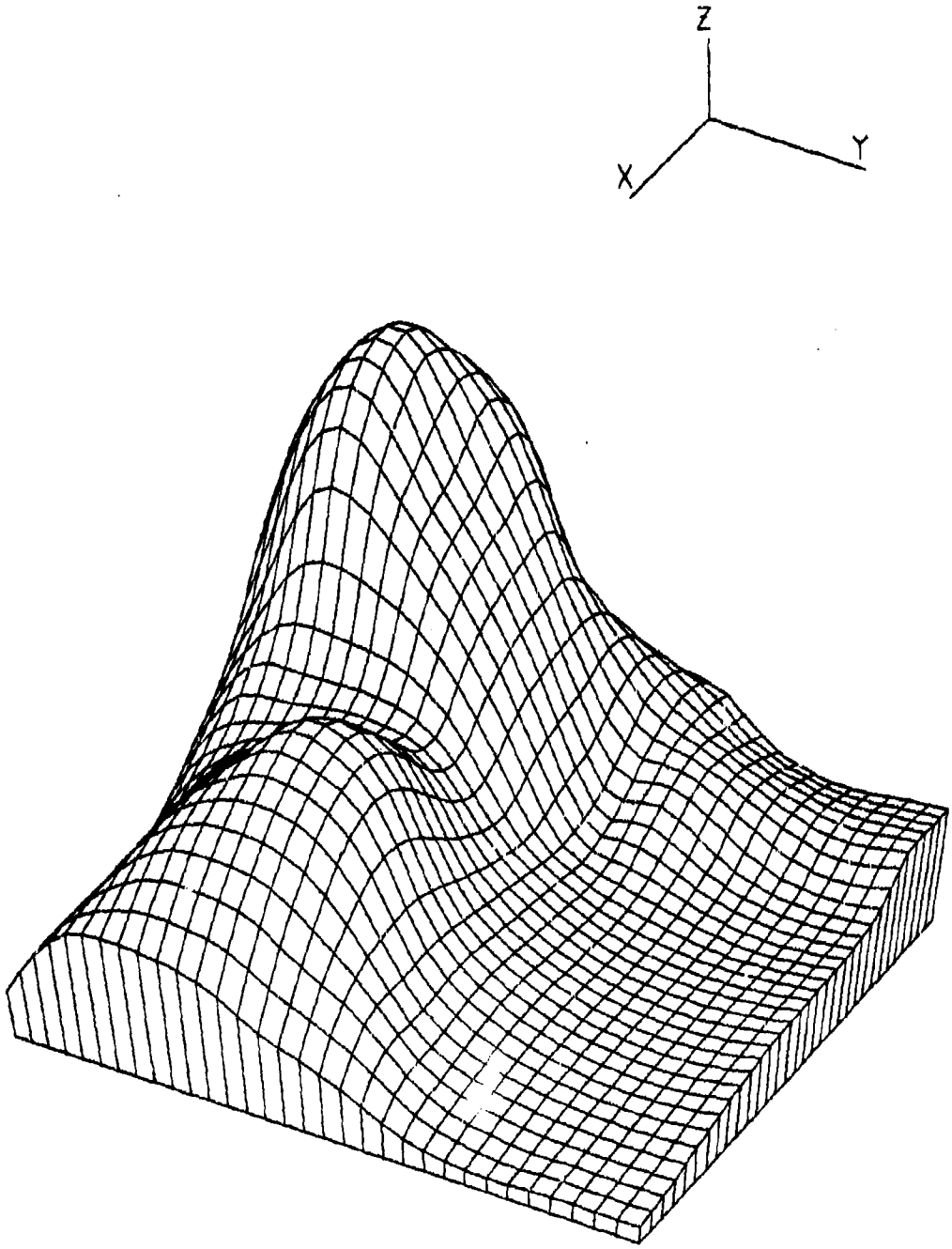
Figure 4.2.1.10



33PT EXPONENTIAL ON SPARSE 11979 1208

NIELSON-FRANKE QUAD-TRI NPPR = 18 R = 0.522

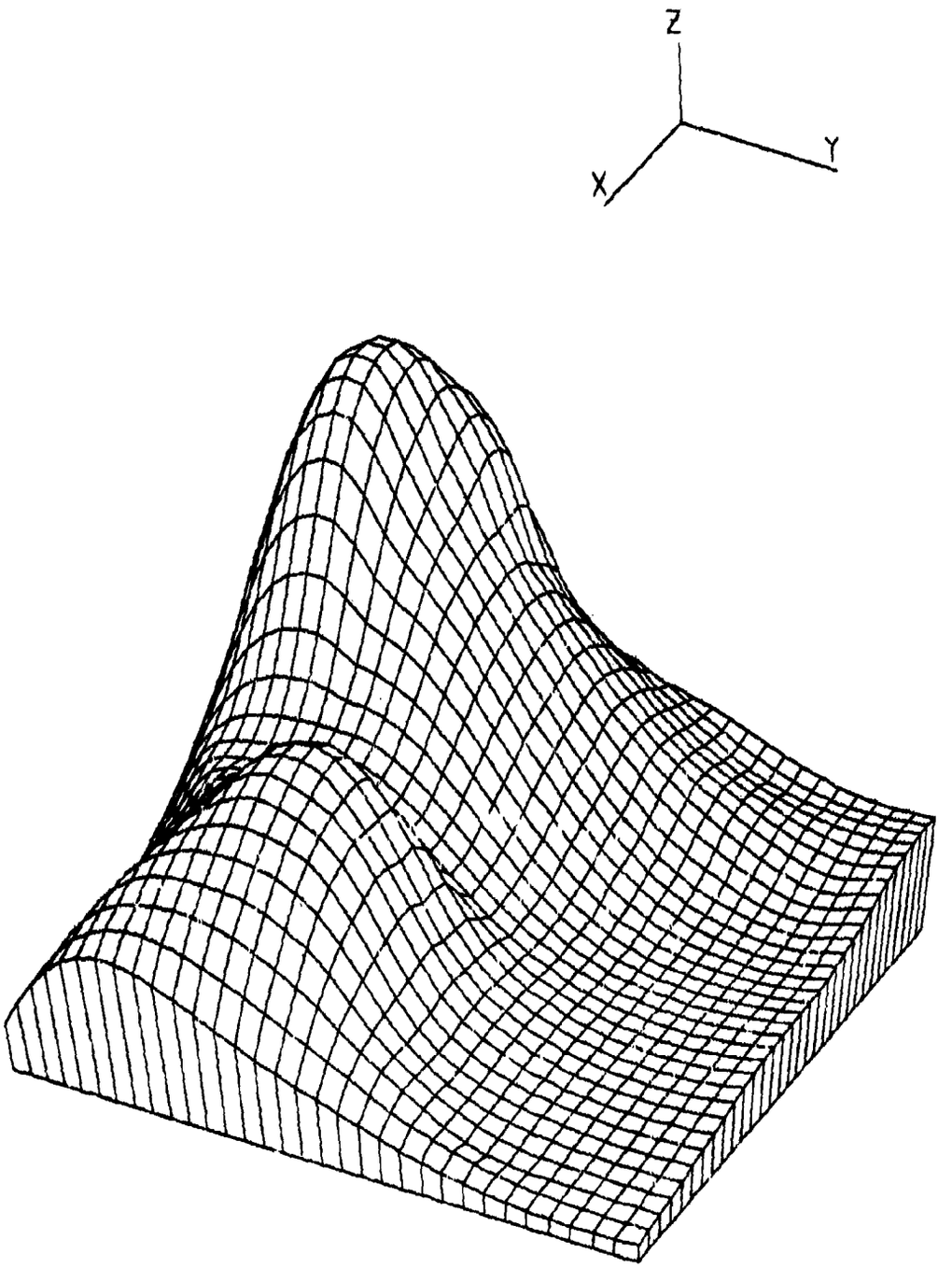
Figure 4.2.1.13



33PT EXPONENTIAL ON SPARSE 11979 1202

QUADRATIC SHEPARD METHOD NPPR = 918 R = 0.369

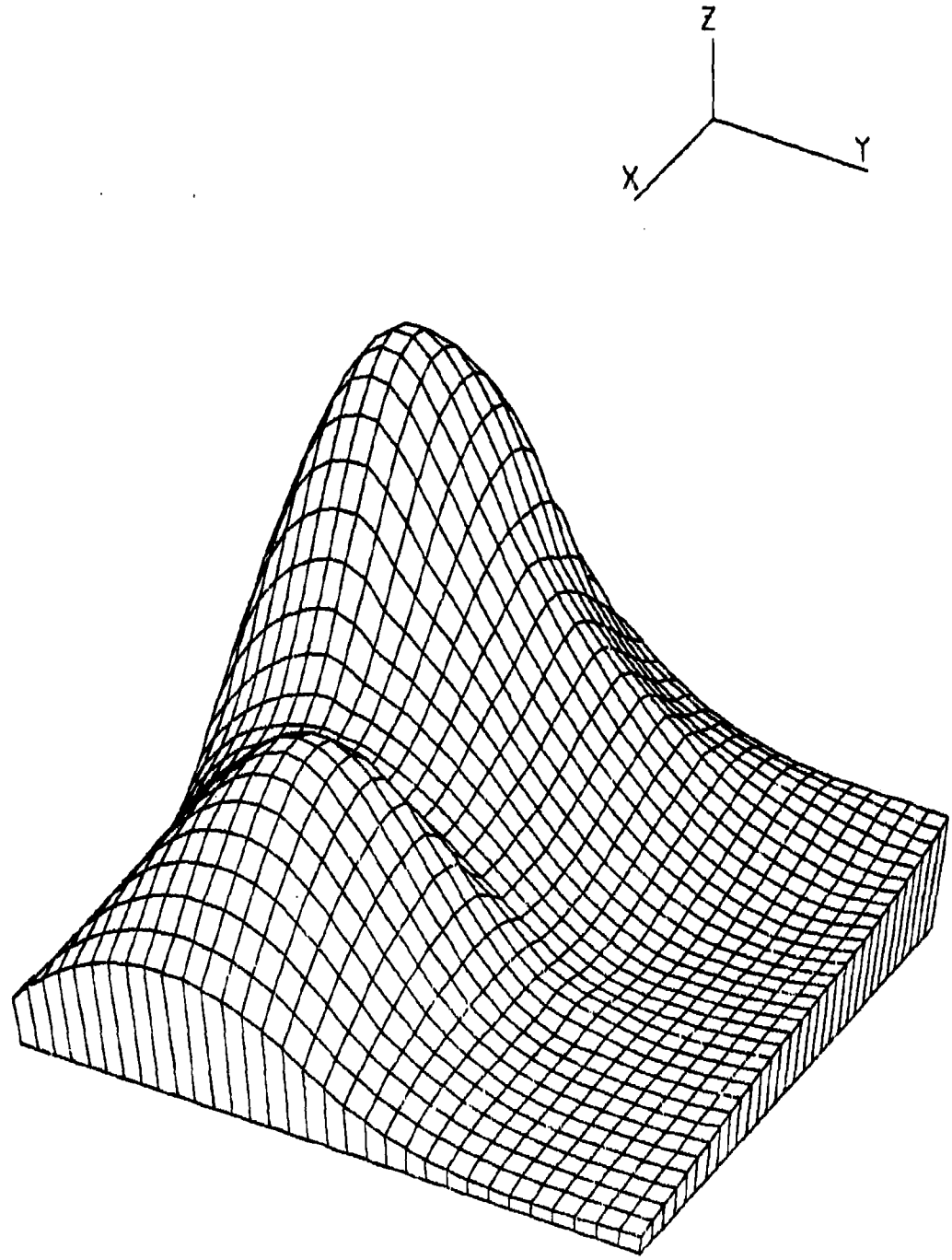
Figure 4.2.1.14



33PT EXP HUMPS, DIP ON SPARSE 30979 1138

AKIMA - L2 QUAD DERIVS NPPR = 18 R = 0.522

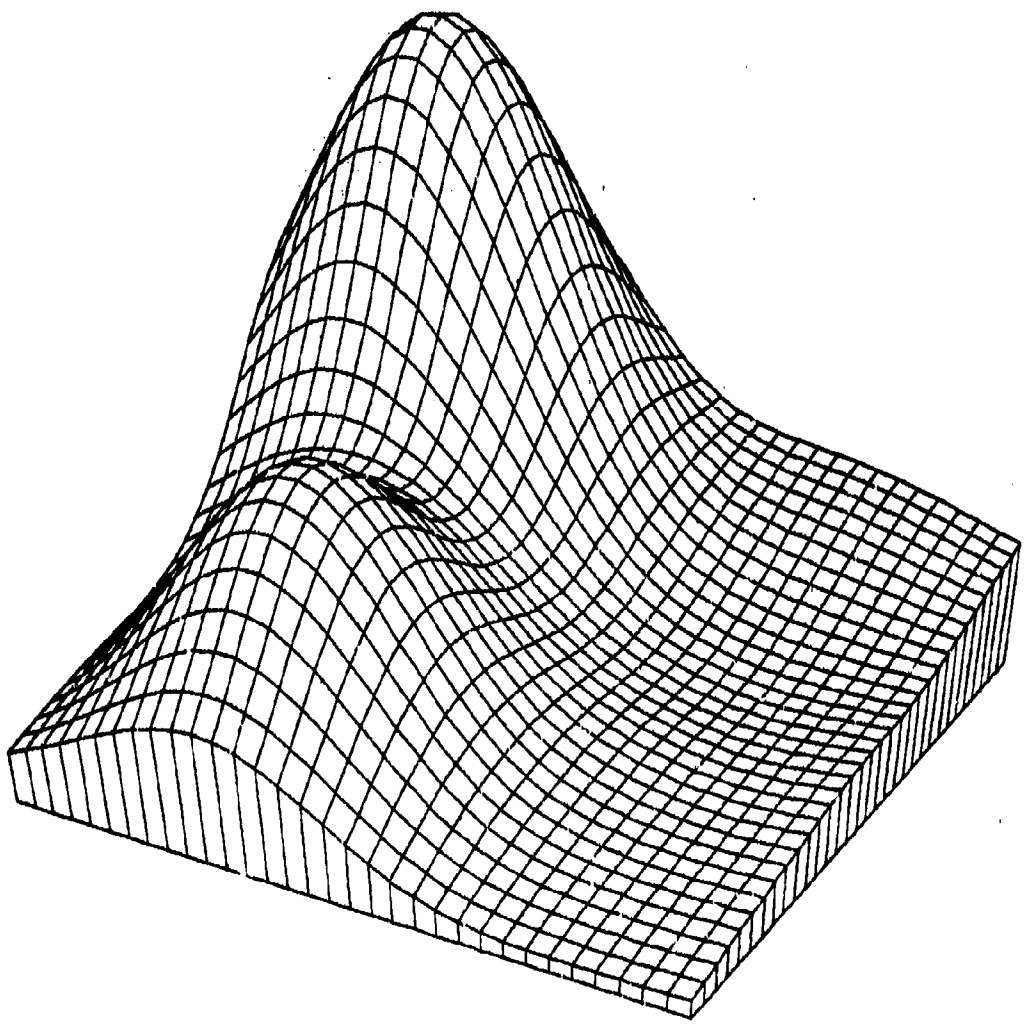
Figure 4.2.1.16



33PT EXP HUMPS, DIP ON SPARSE 12979 1351

NIELSON MIN NORM NETWORK

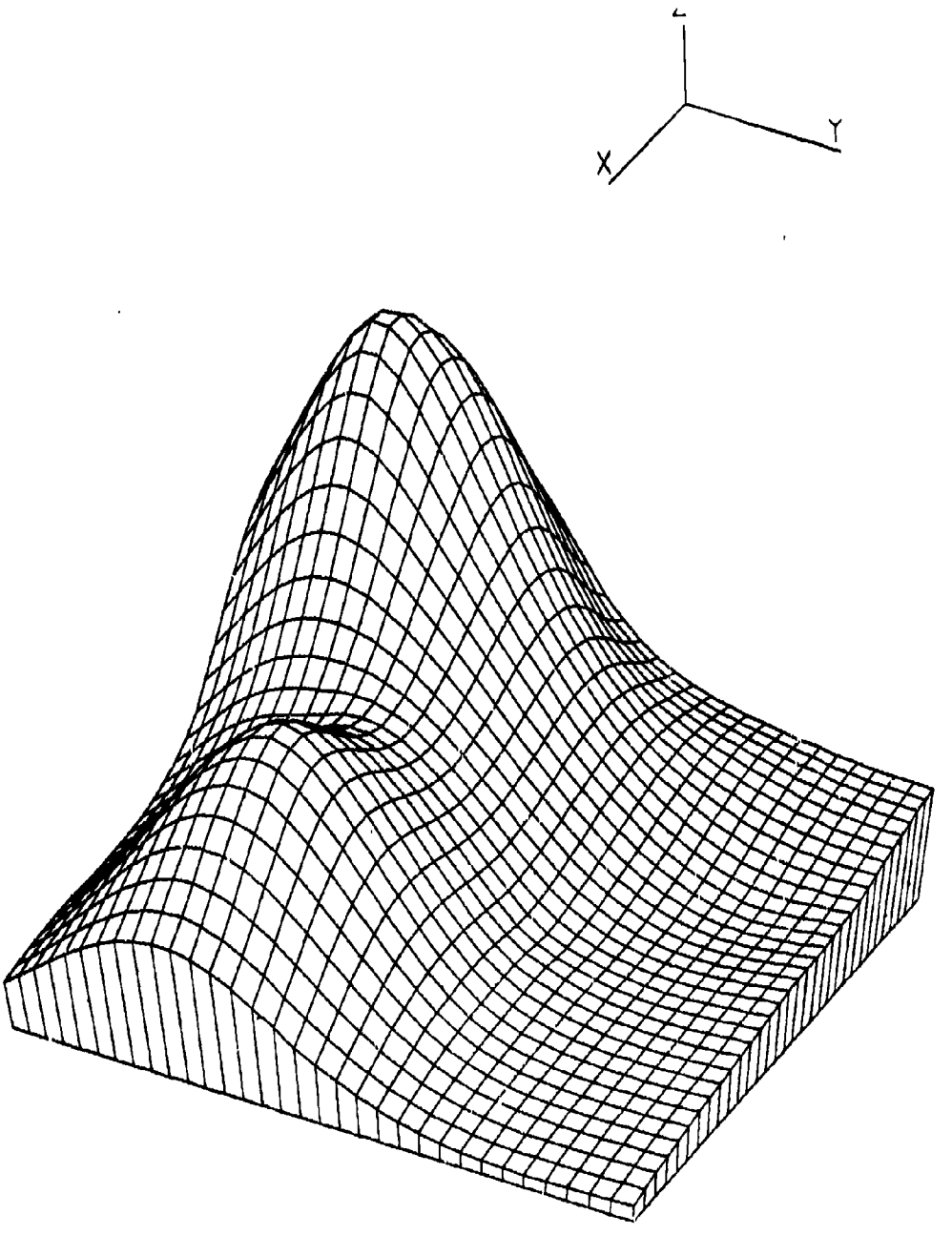
Figure 4.2.1.19



33PT EXP HUMPS, DIP ON SPARSE 32279 1716

HARDY MQ SURFACE NPPR = 25 R = 0.308

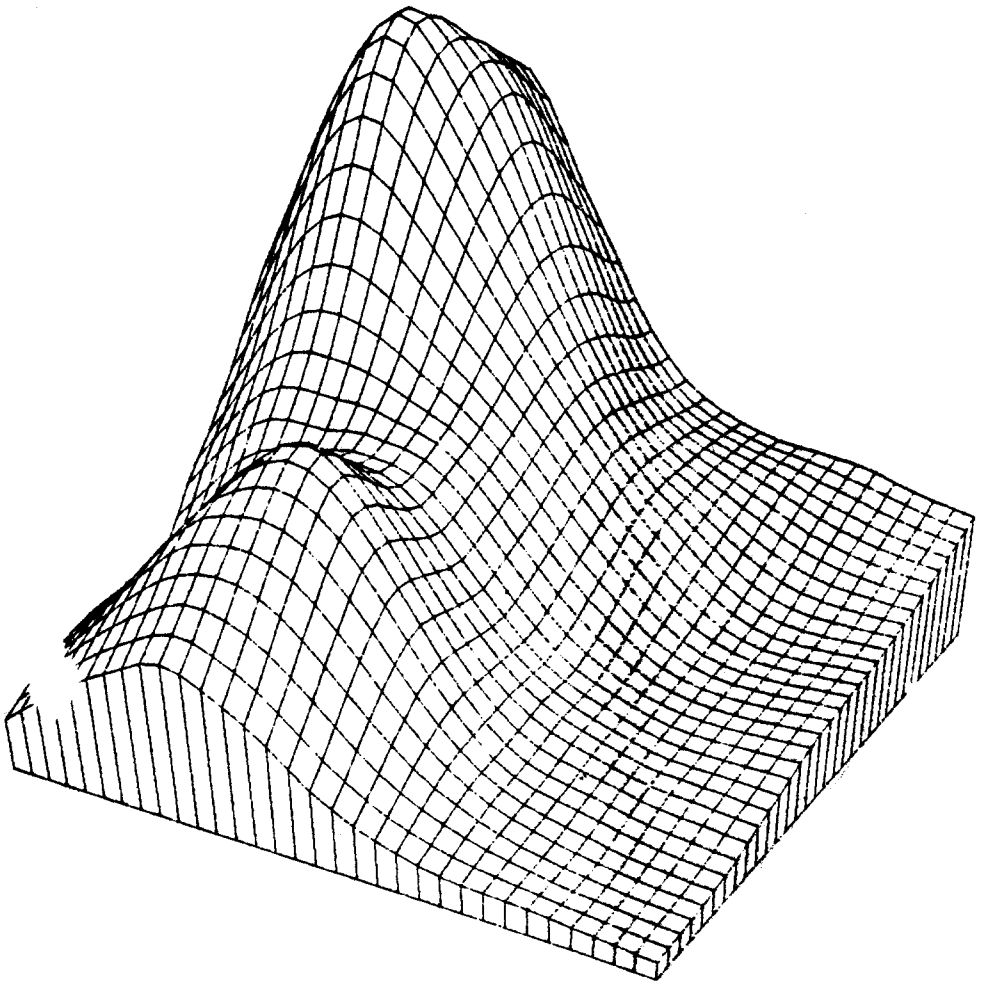
Figure 4.2.1.21



33PT EXP HUMPS, DIP ON SPARSE 20679 1534

DUCHON S THIN PLATE

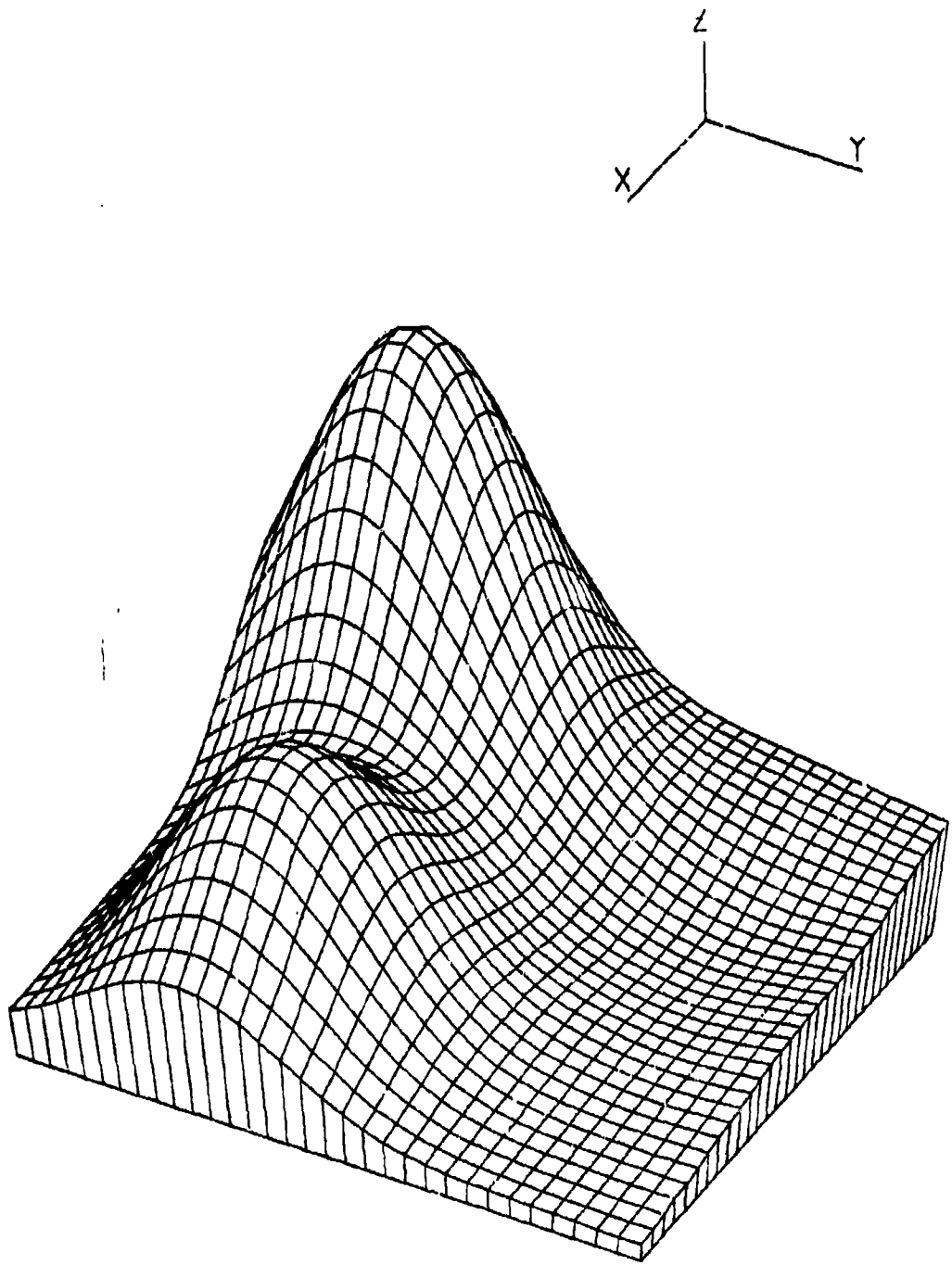
Figure 4.2.1.23



33PT EXP HUMPS, DIP ON SPARSE 80879 910

FRANKE W/ THIN PLATE NPPR = 6

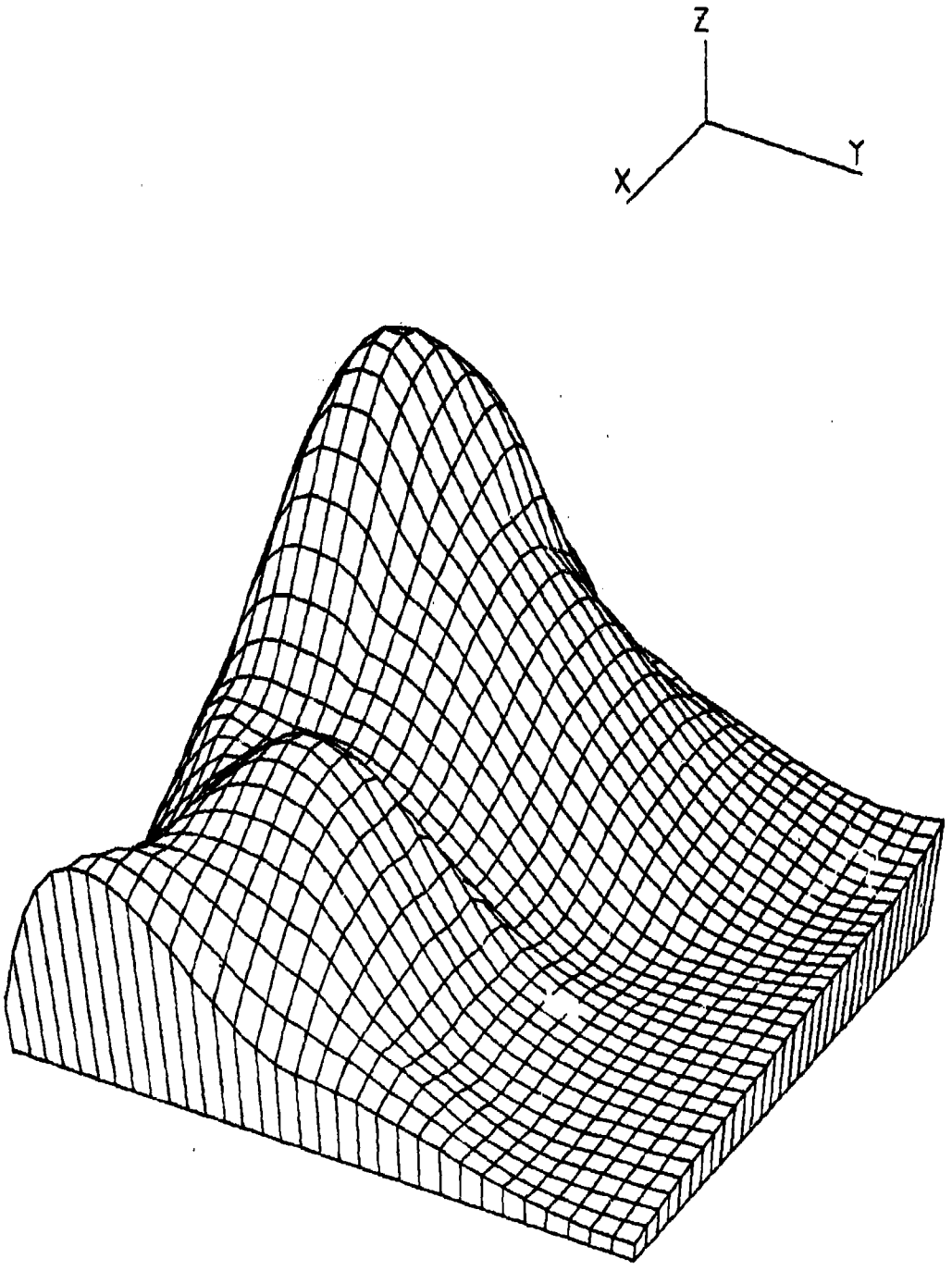
Figure 4.2.1.24



33PT EXP HUMPS, DIP ON SPARSE 41079 1824

HARDY RECIP MQ SURFACE NPPR = 25 R = 0.308

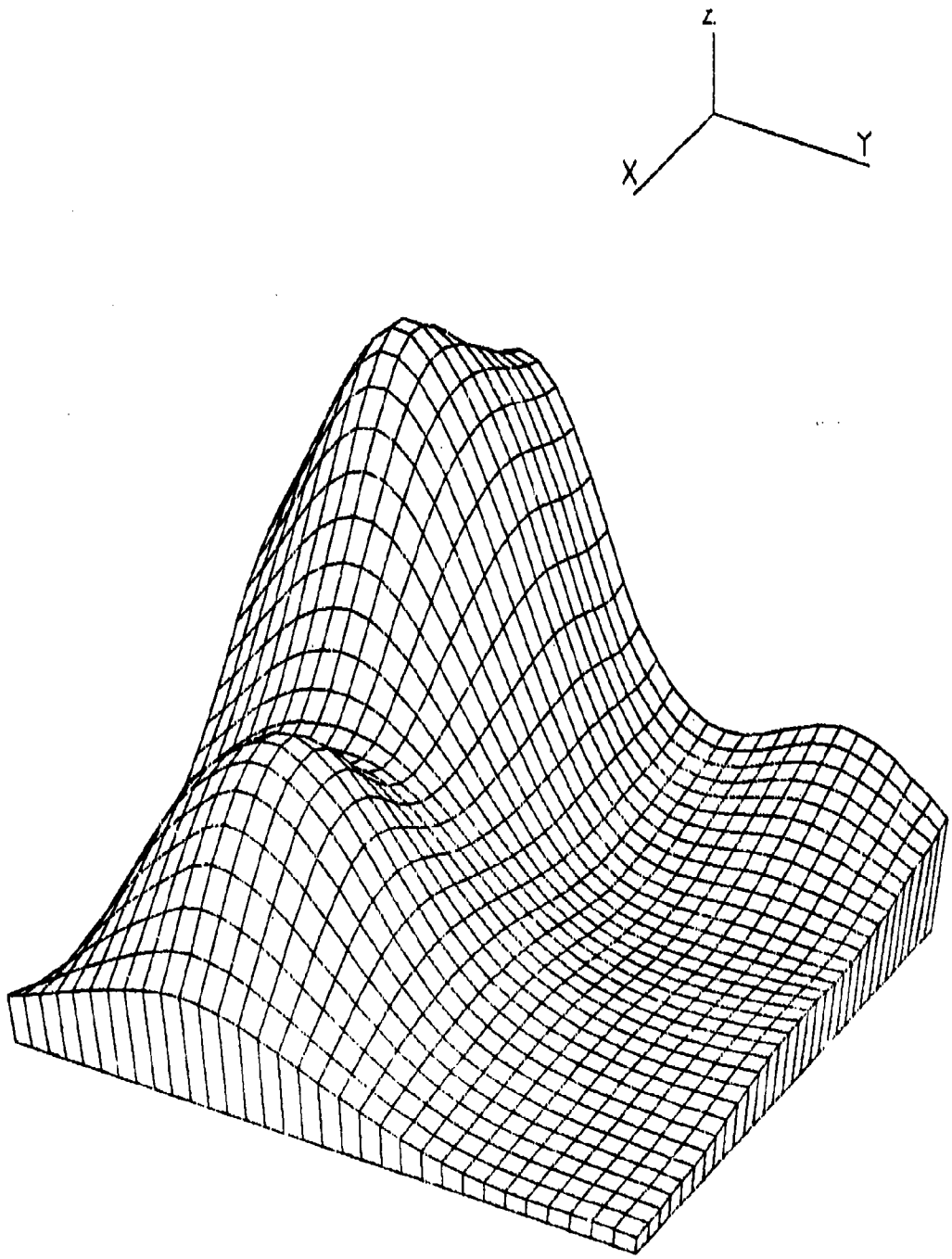
Figure 4.2.1.27



33PT EXP HUMPS, DIP ON SPARSE 32379 1846

LAWSON TRIANGLE METHOD

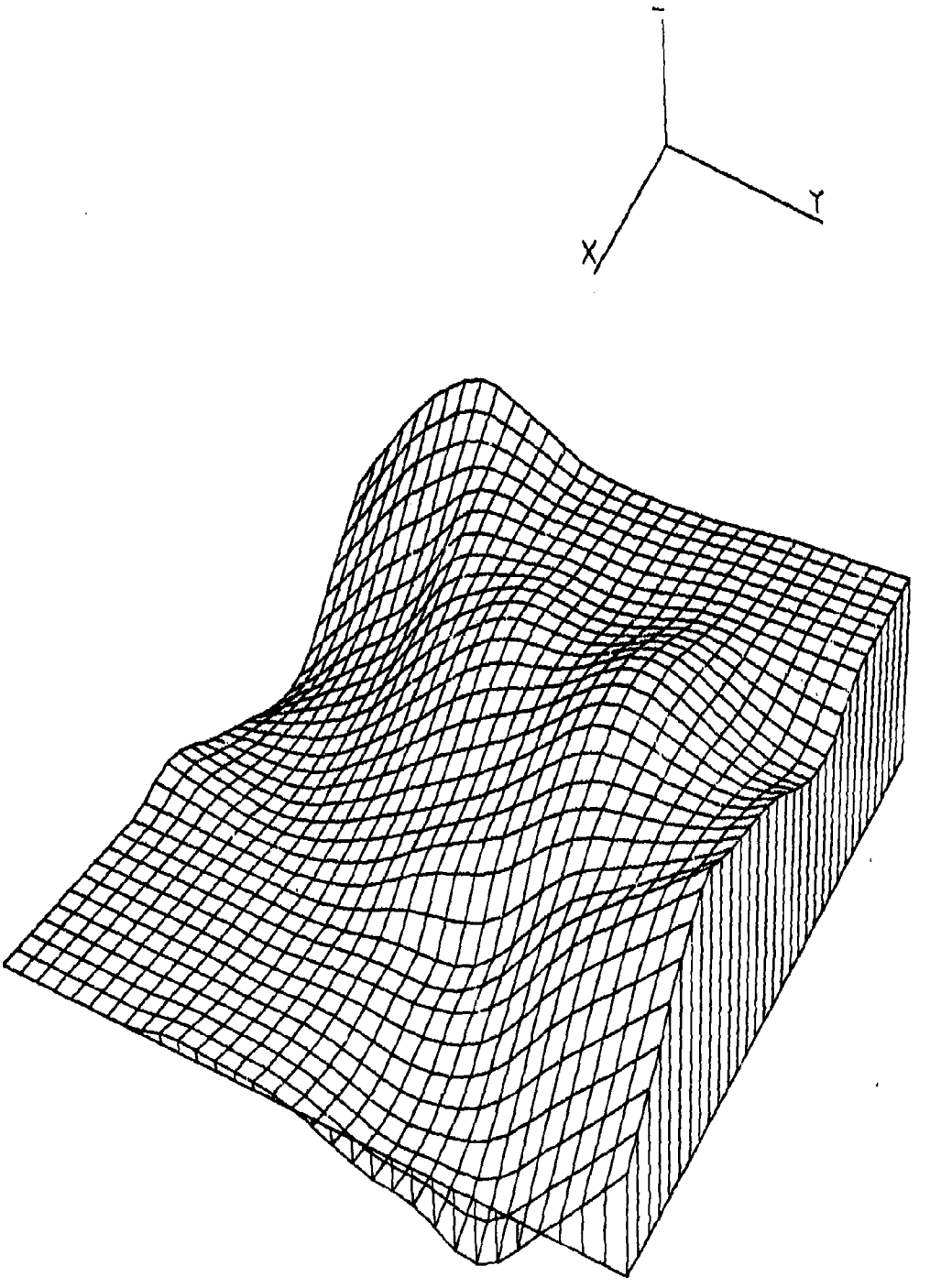
Figure 4.2.1.28



33PT EXP HUMPS, DIP ON SPARSE 80279 1051

FOLEY ITERATED BICUBICS NPPR = 3

Figure 4.2.1.30



33PT CLIFF FUNCTION ON SPARSE 21679 944

FRANKE - SARD BS PLANE NPPR = 6

Figure 4.2.2.1

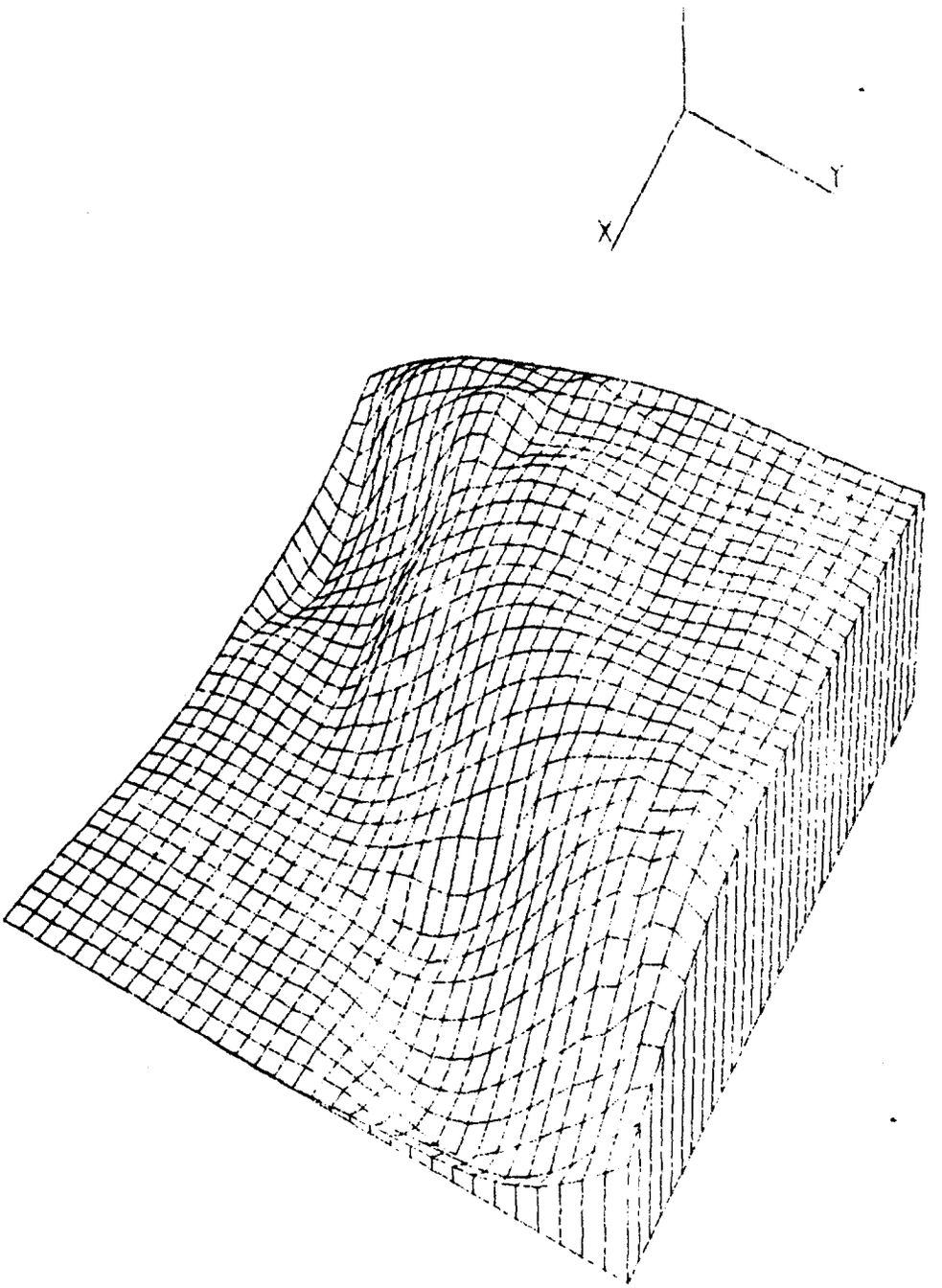
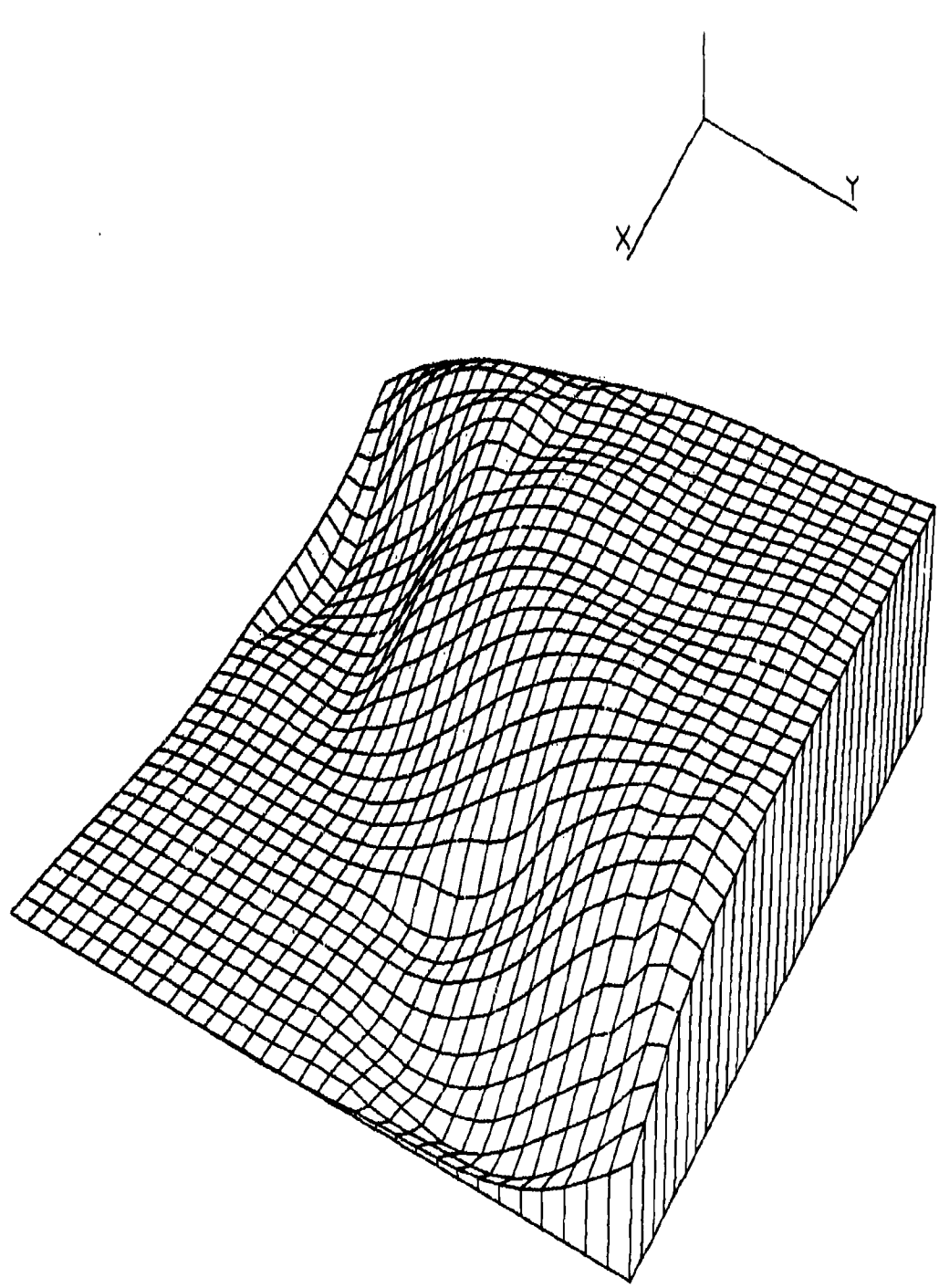


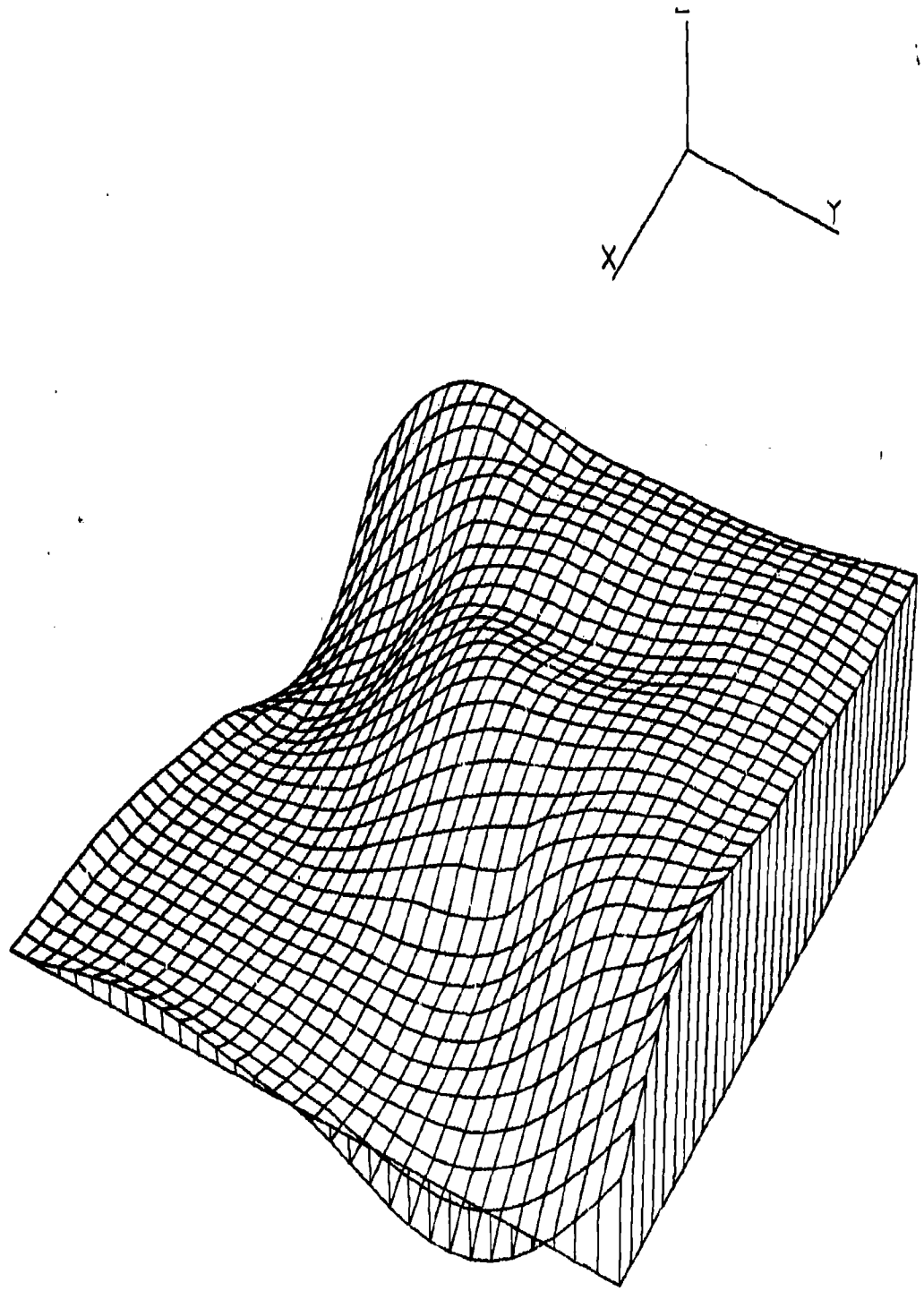
Figure 4.2.2.4



33PT CLIFF FUNCTION ON SPARSE 31079 1934

MODIFIED AKIMA FEM FIT NPPR = 6

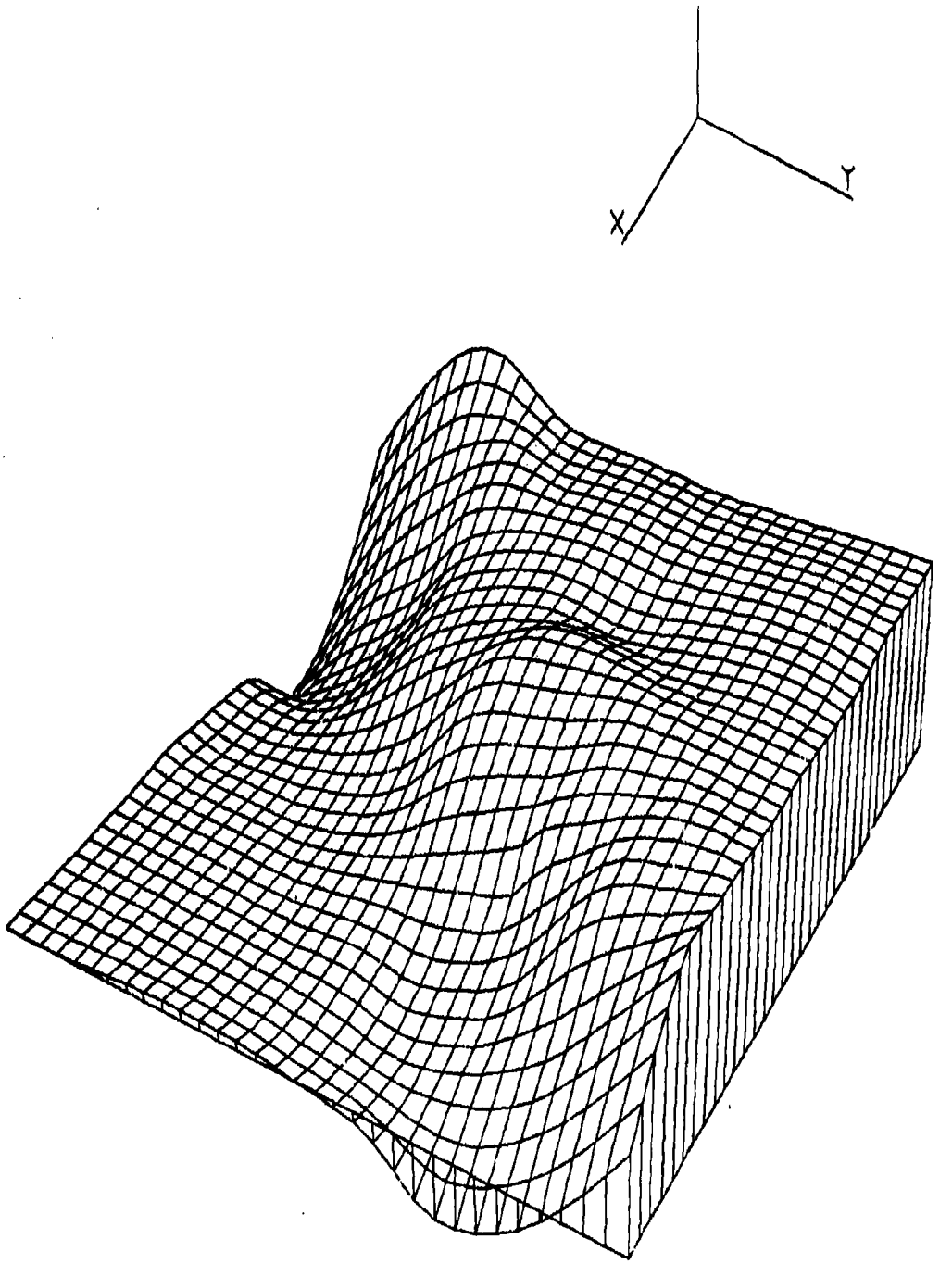
Figure 4.2.2.10



33PT CLIFF FUNCTION ON SPARSE 31079 1919

NIELSON-FRANKE QUAD-TRI NPPR = 18 R = 0.522

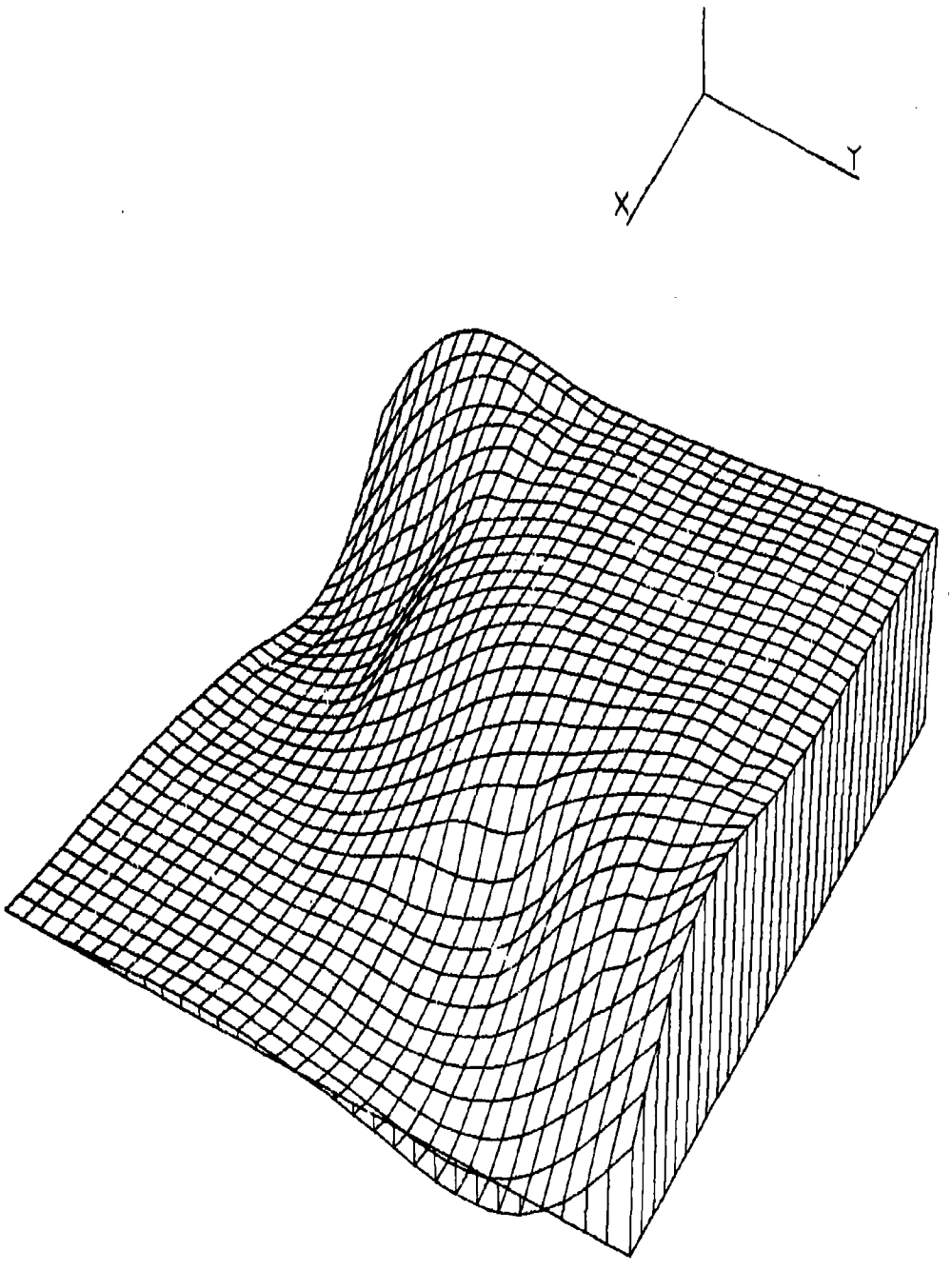
Figure 4.2.2.13



33PT CLIFF FUNCTION ON SPARSE 31179 1442

QUADRATIC SHEPARD METHOD NPPR = 918 R = 0.369

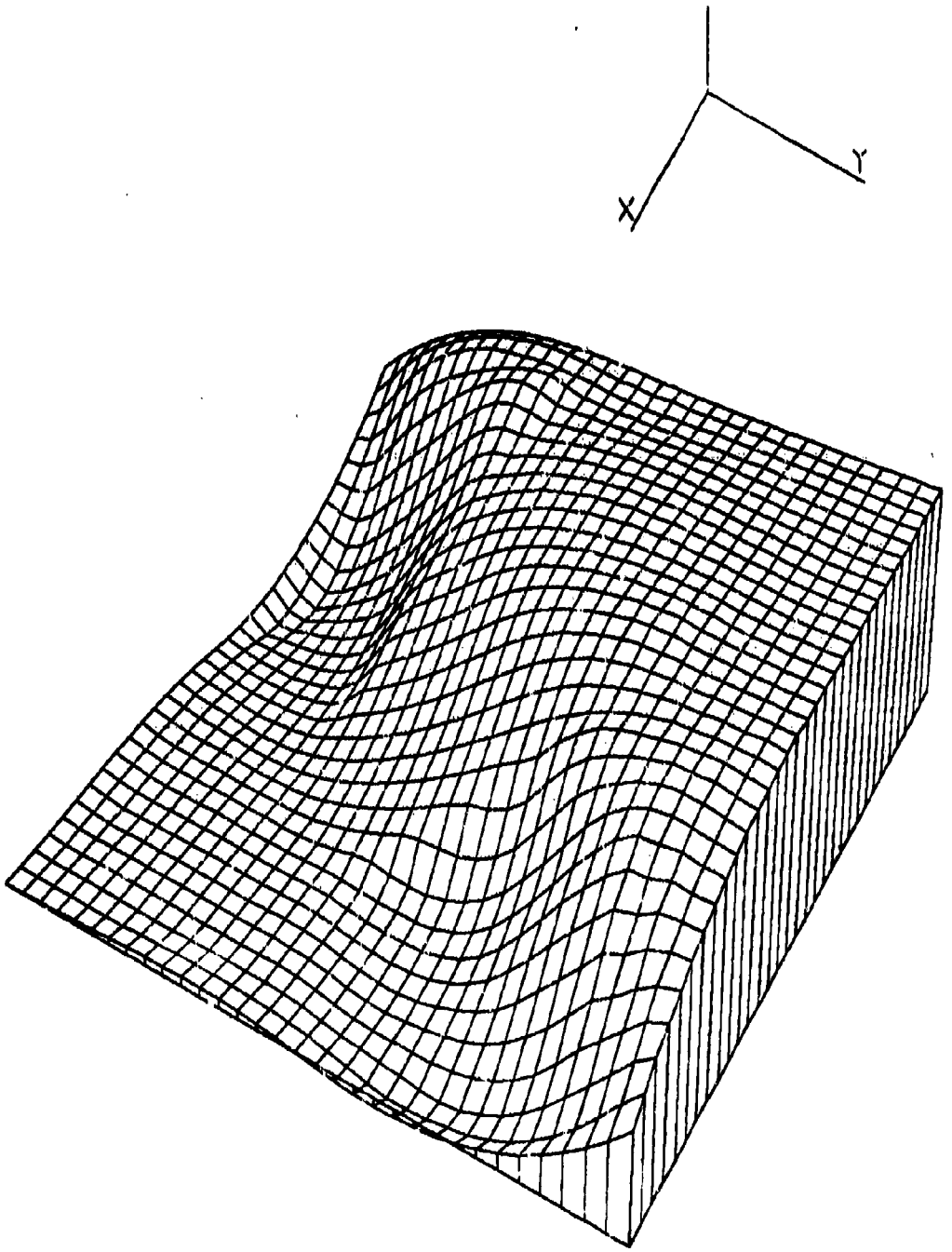
Figure 4.2.2.14



33PT CLIFF FUNCTION ON SPARSE 31079 1933

AKIMA - L2 QUAD DERIVS NPPR = 18 R = 0.522

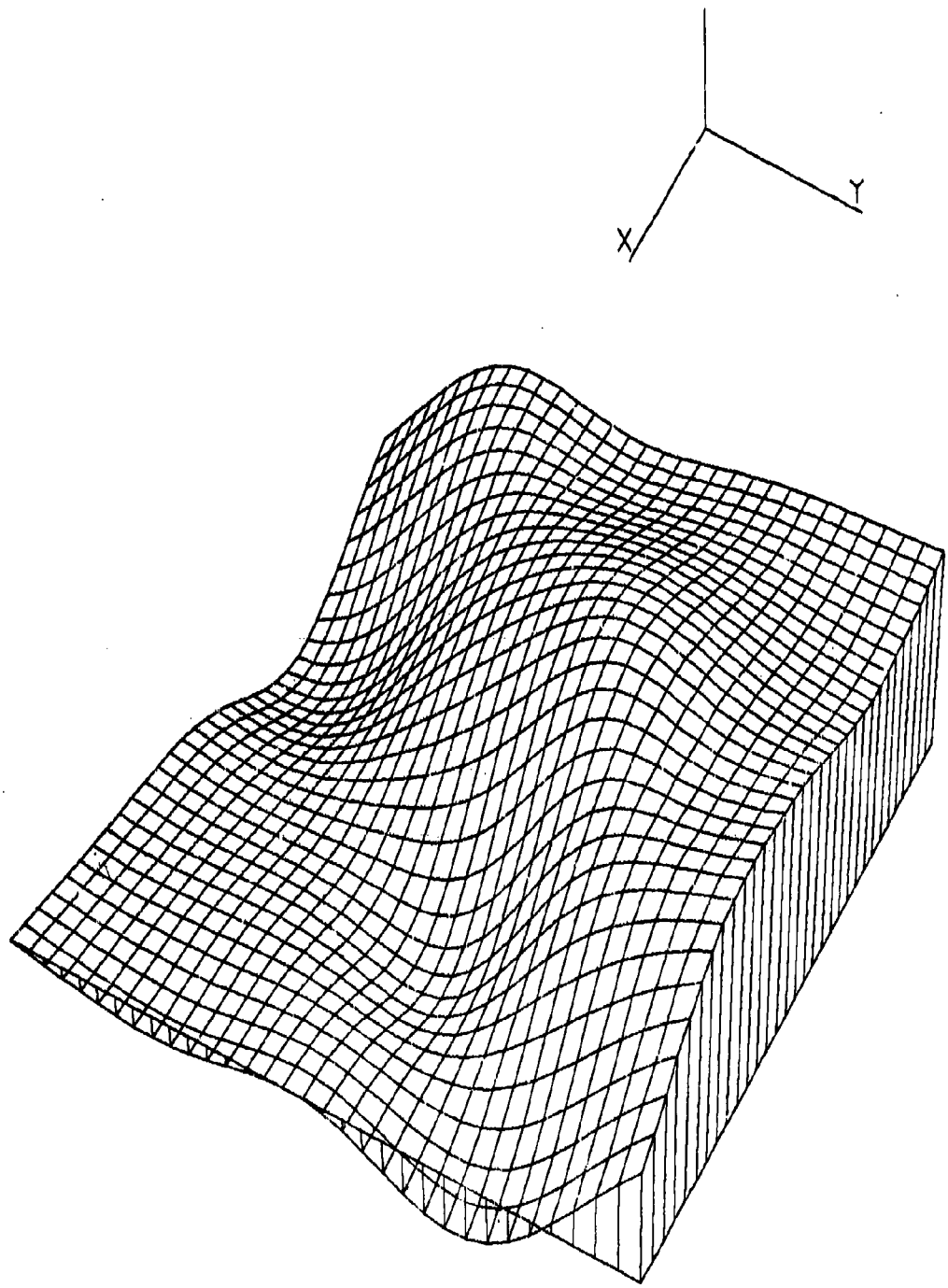
Figure 4.2.2.16



33PT CLIFF FUNCTION ON SPARSE 12979 1352

NIELSON MIN NORM NETWORK

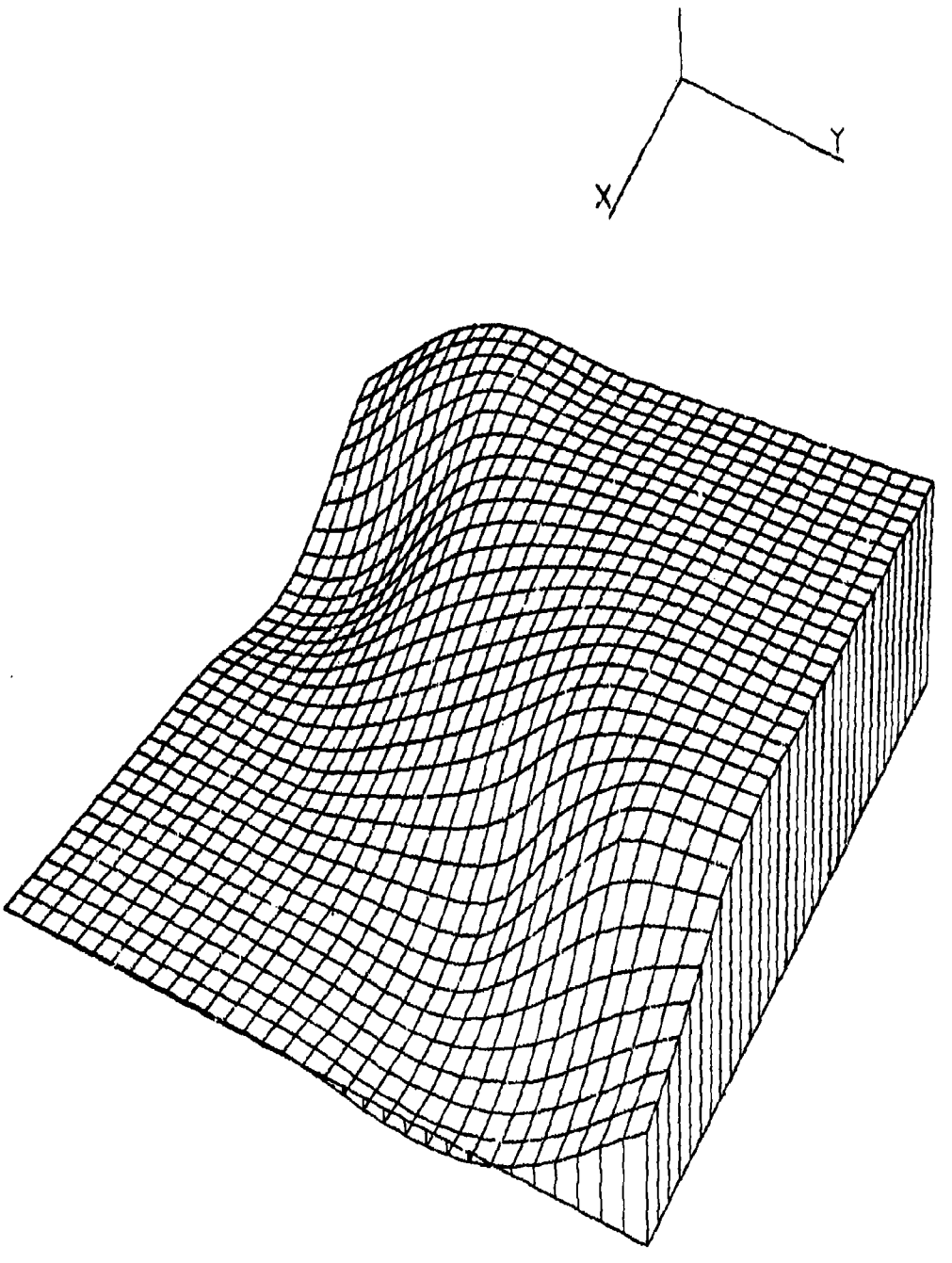
Figure 4.2.2.19



33PT CLIFF FUNCTION ON SPARSE 31579 2017

HARDY MQ SURFACE NPPR = 25 R = 0.308

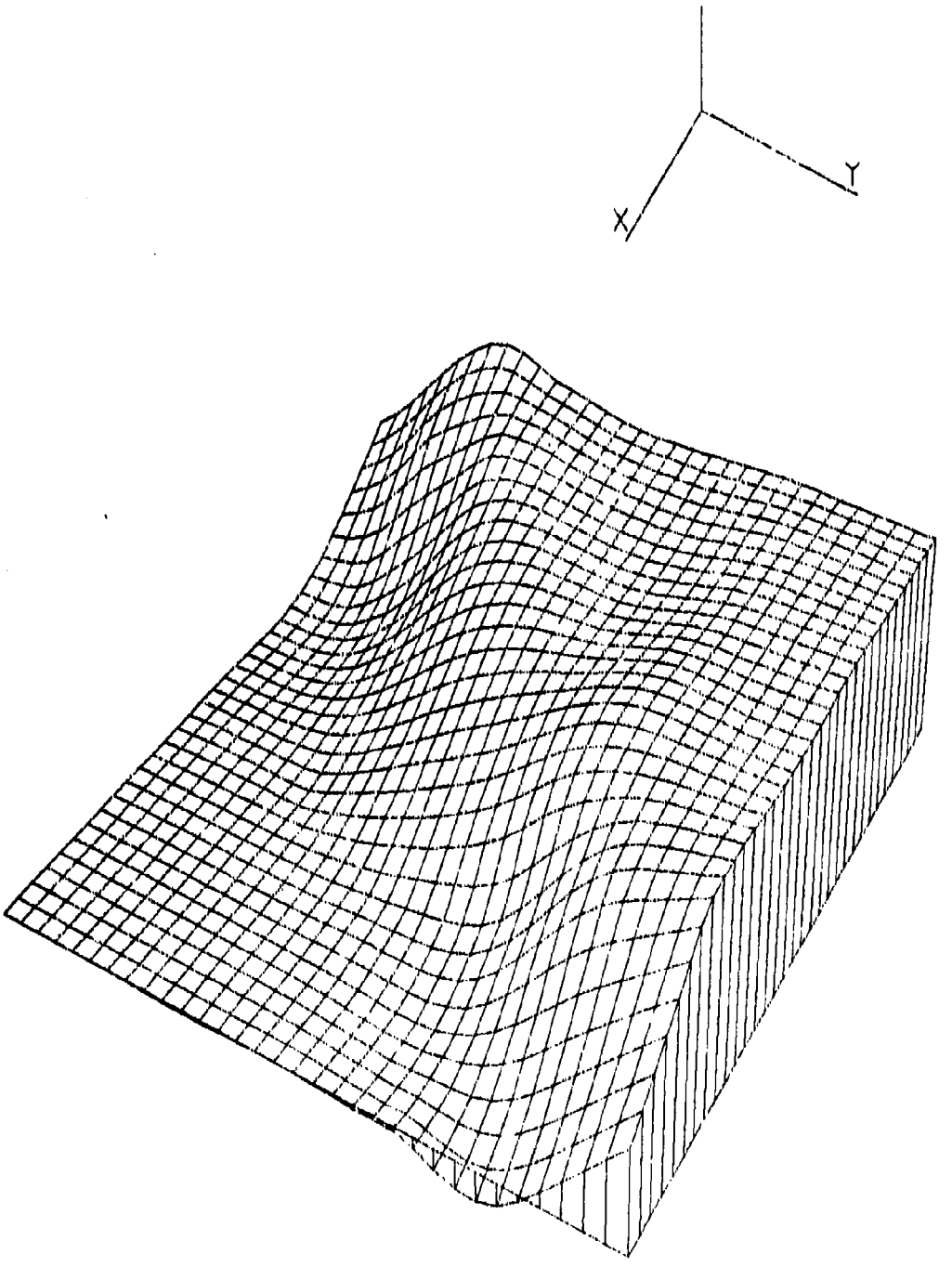
Figure 4.2.2.21



33PT CLIFF FUNCTION ON SPARSE 20779 1039

DUCHON S THIN PLATE

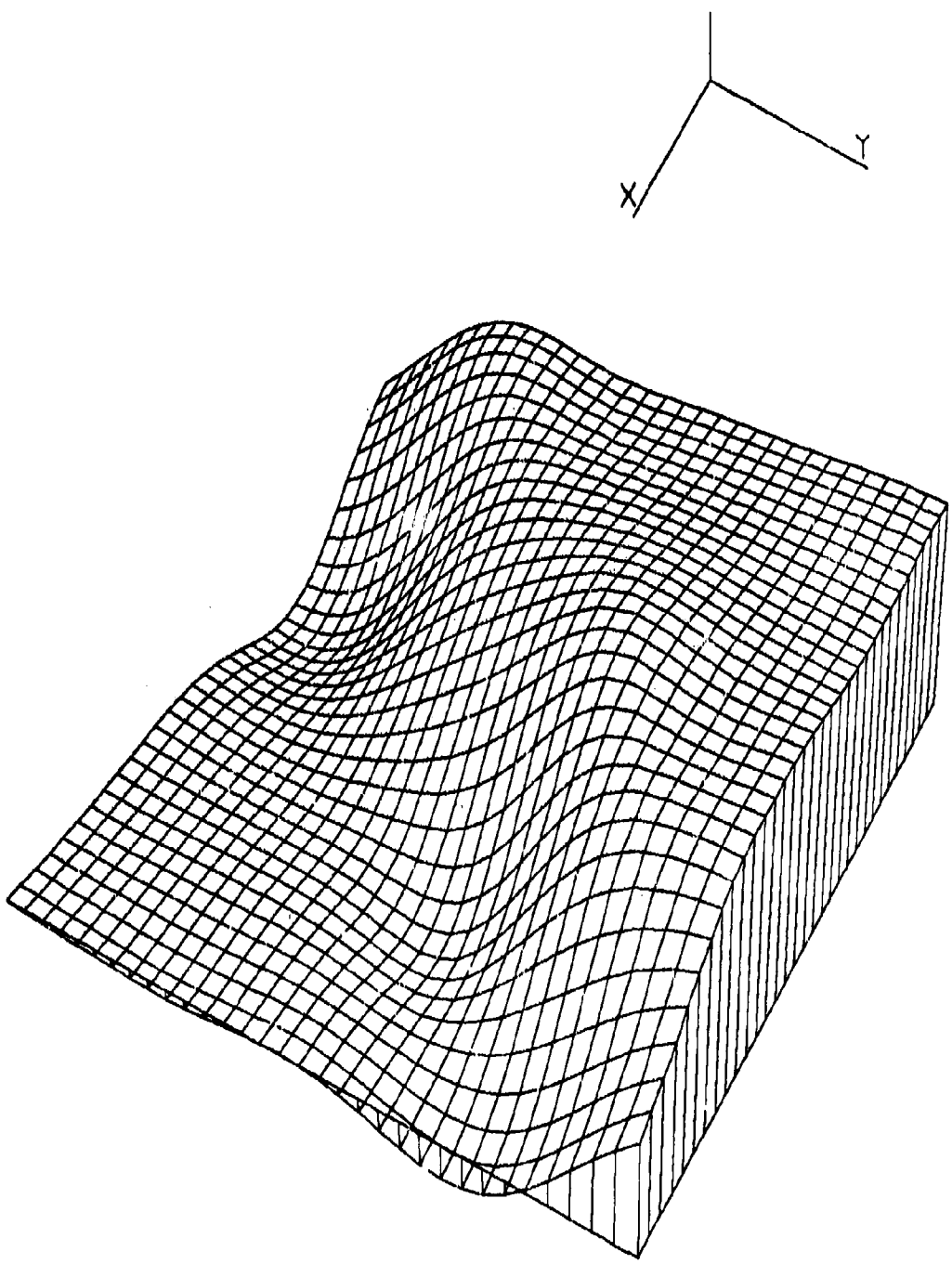
Figure 4.2.2.23



33PT CLIFF FUNCTION ON SPARSE 80879 858

FRANKE W/ THIN PLATE NPPR = 6

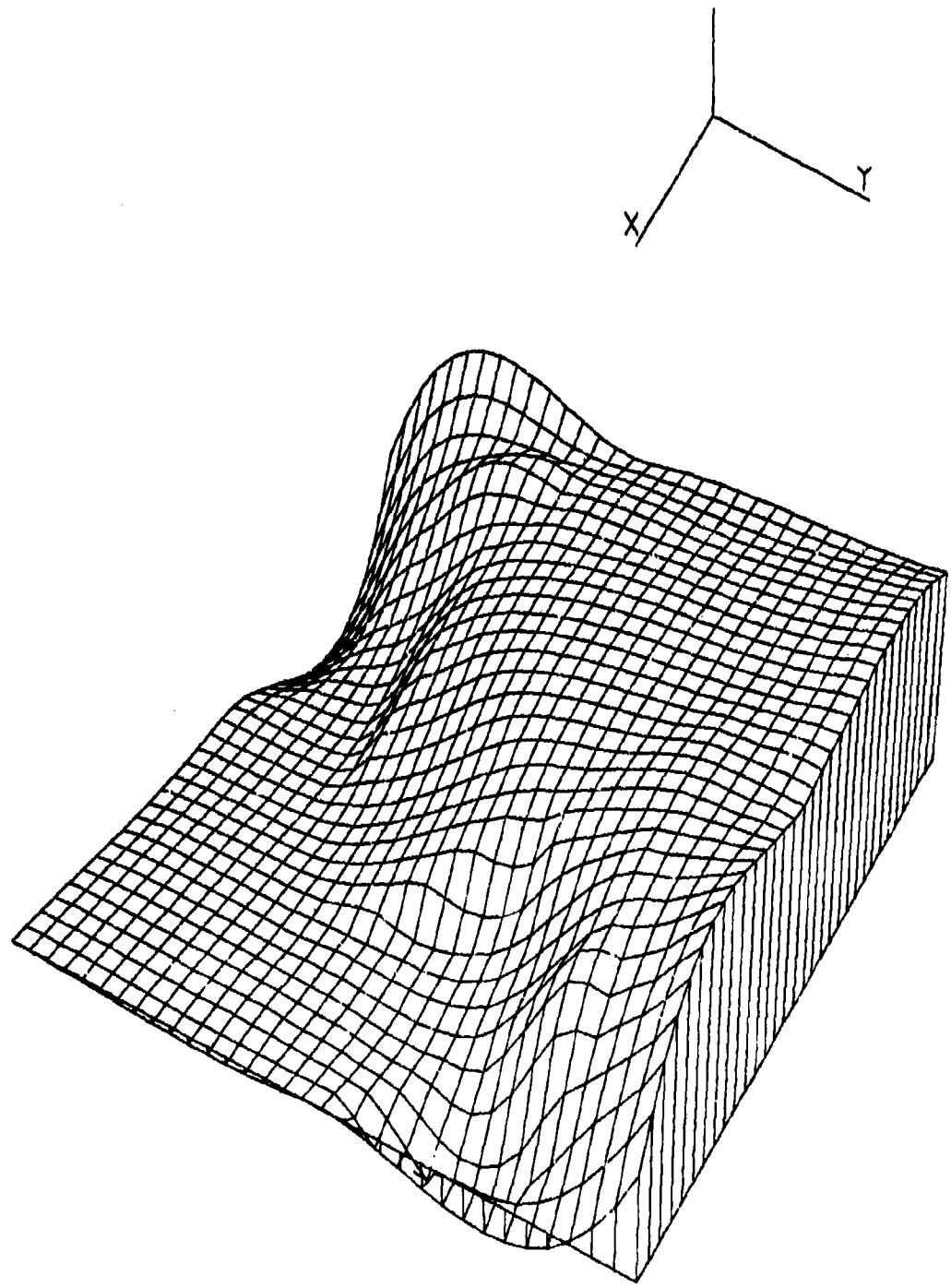
Figure 4.2.2.24



33PT CLIFF FUNCTION ON SPARSE 40679 1528

HARDY RECIP MQ SURFACE NPPR = 25 R = 0.308

Figure 4.2.2.27



33PT CLIFF FUNCTION ON SPARSE 32379 1424

LAWSON TRIANGLE METHOD

Figure 4.2.2.28

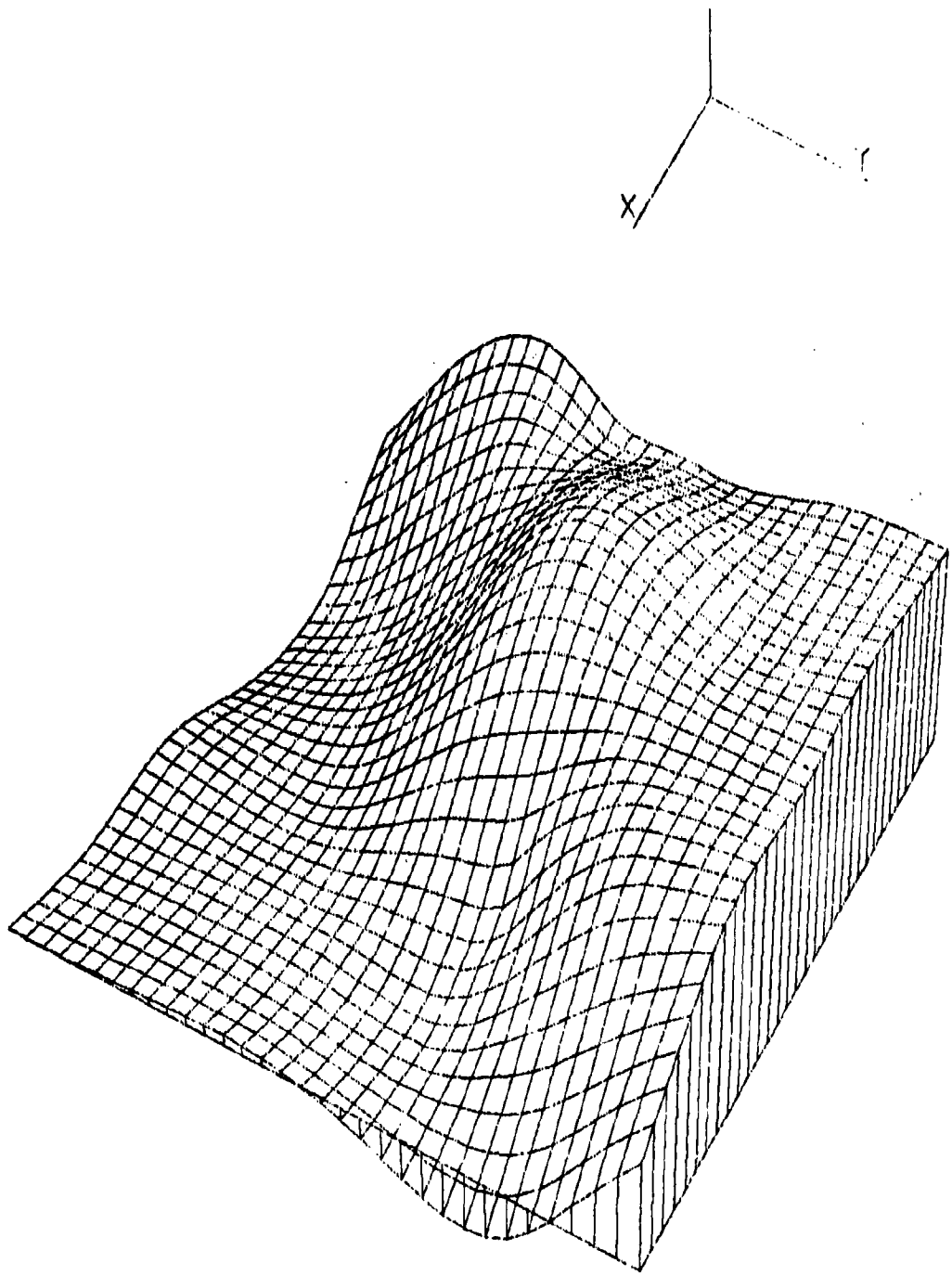
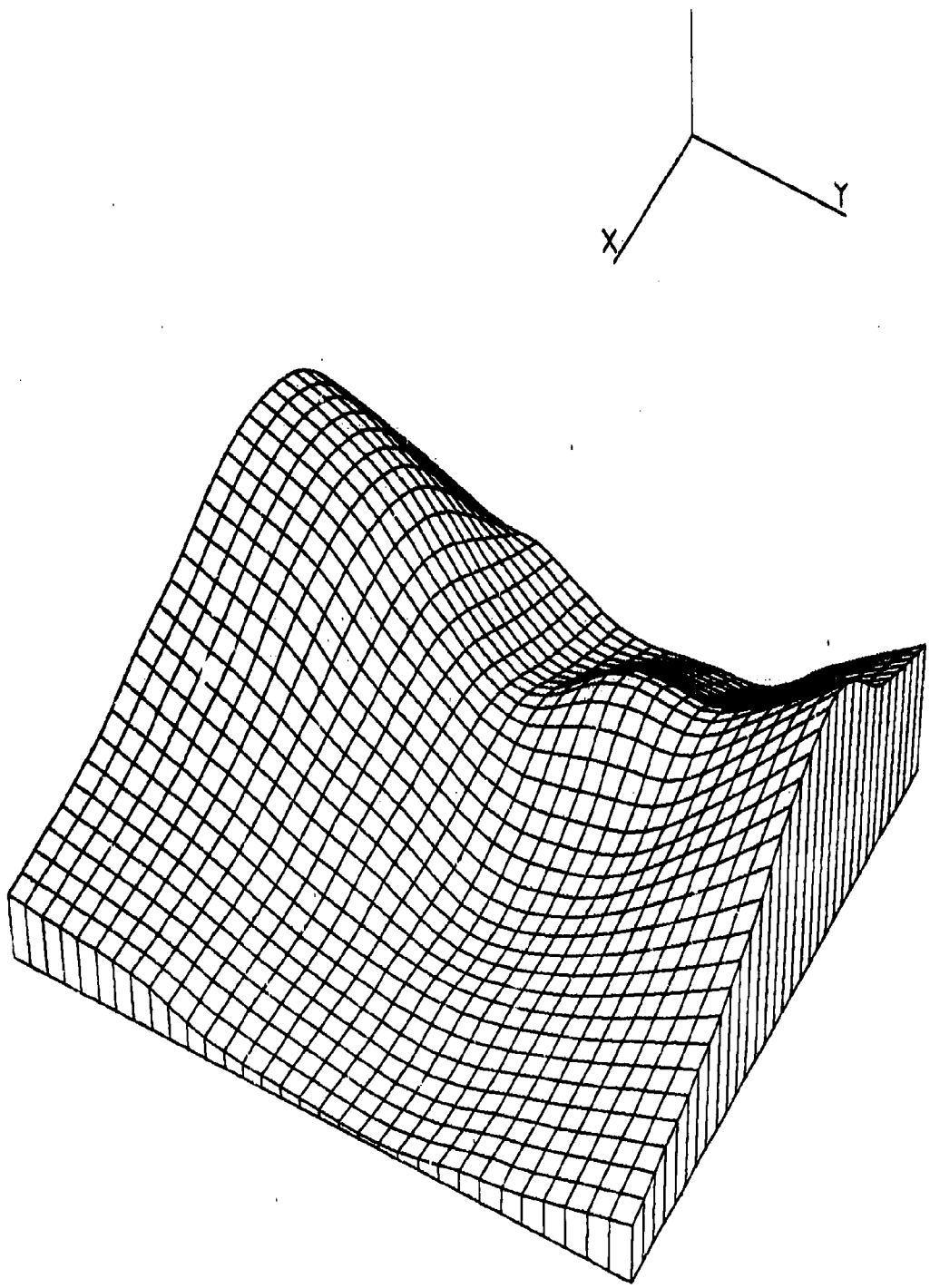


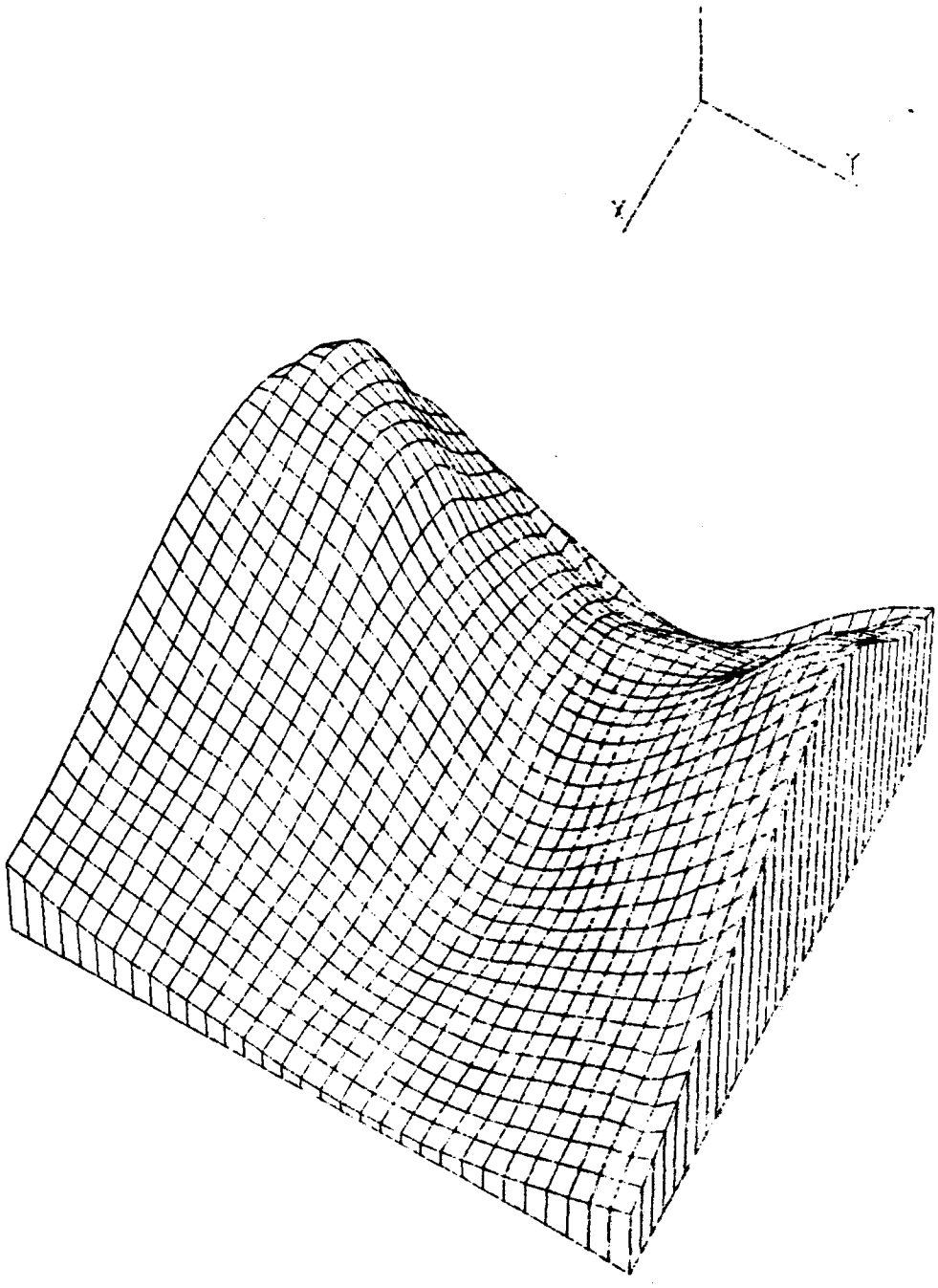
Figure 4.2.2.30



33PT SADDLE ON SPARSE 21679 943

FRANKE - SARD BS PLANE NPPR = 6

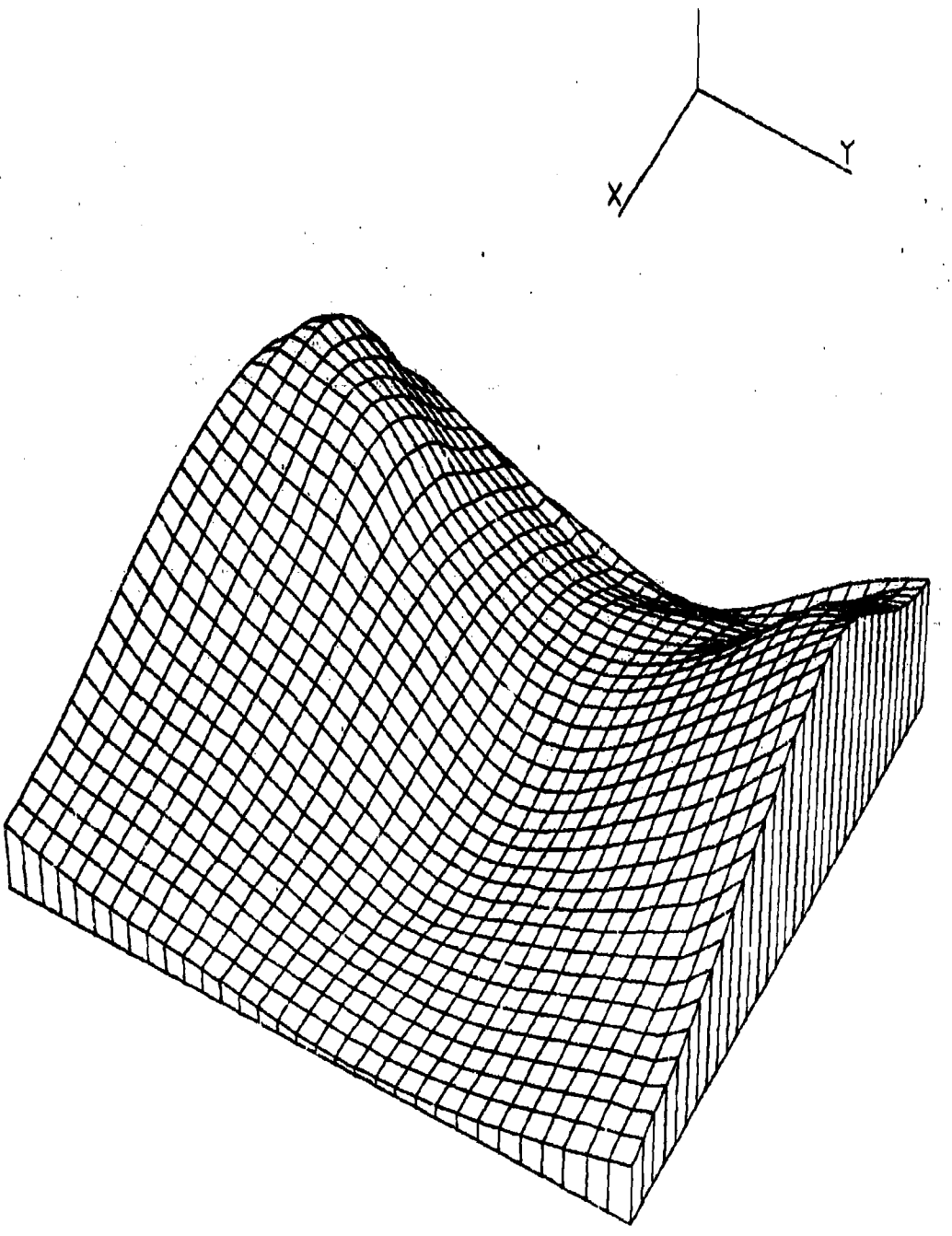
Figure 4.2.3.1



33PT SADDLE ON SPARSE 41879 1021

AKIMA - FEM FIT NPPR = 6

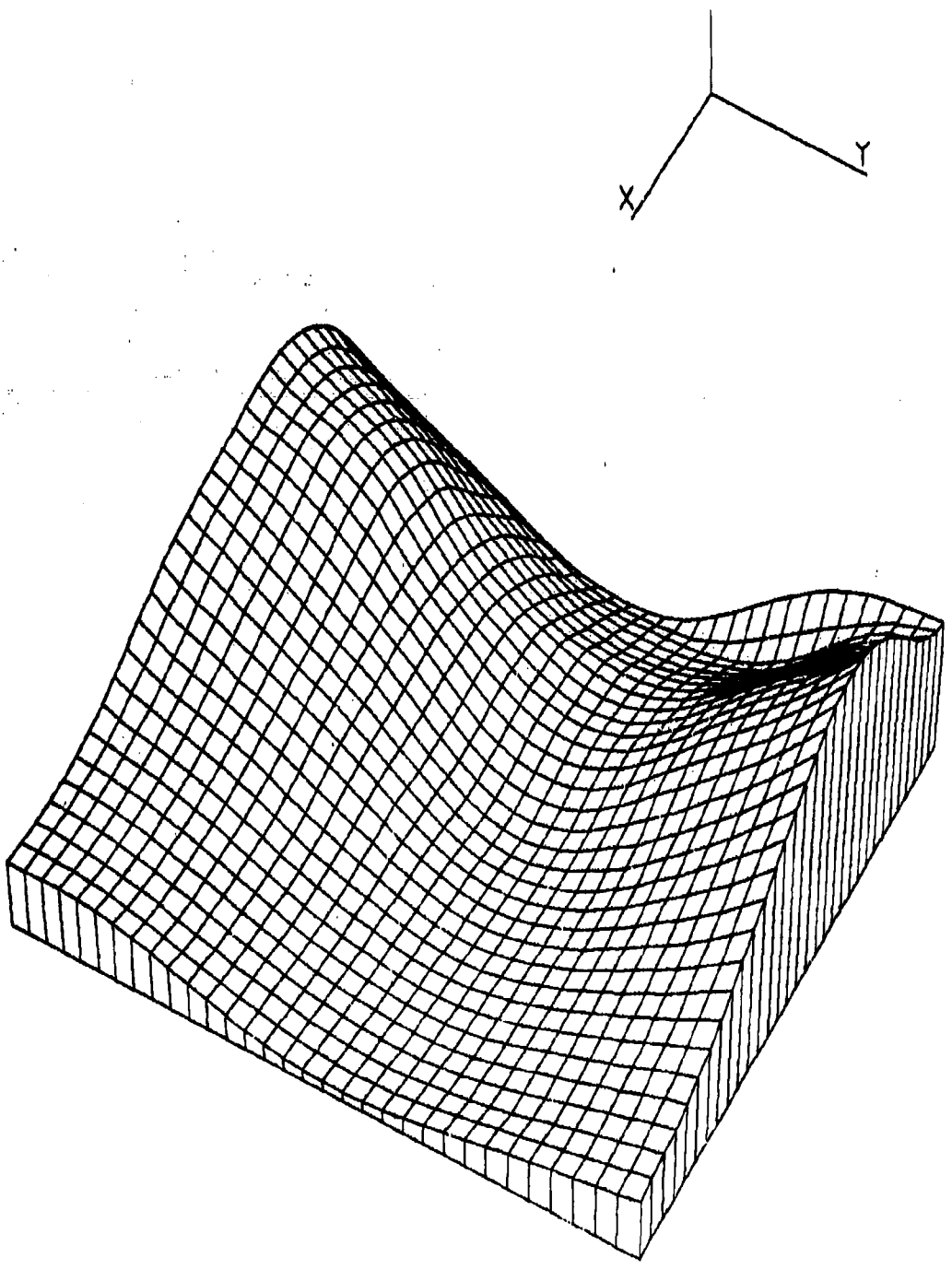
Figure 4.2.3.4



33PT SADDLE ON SPARSE 31079 1933

MODIFIED AKIMA FEM FIT NPPR = 6

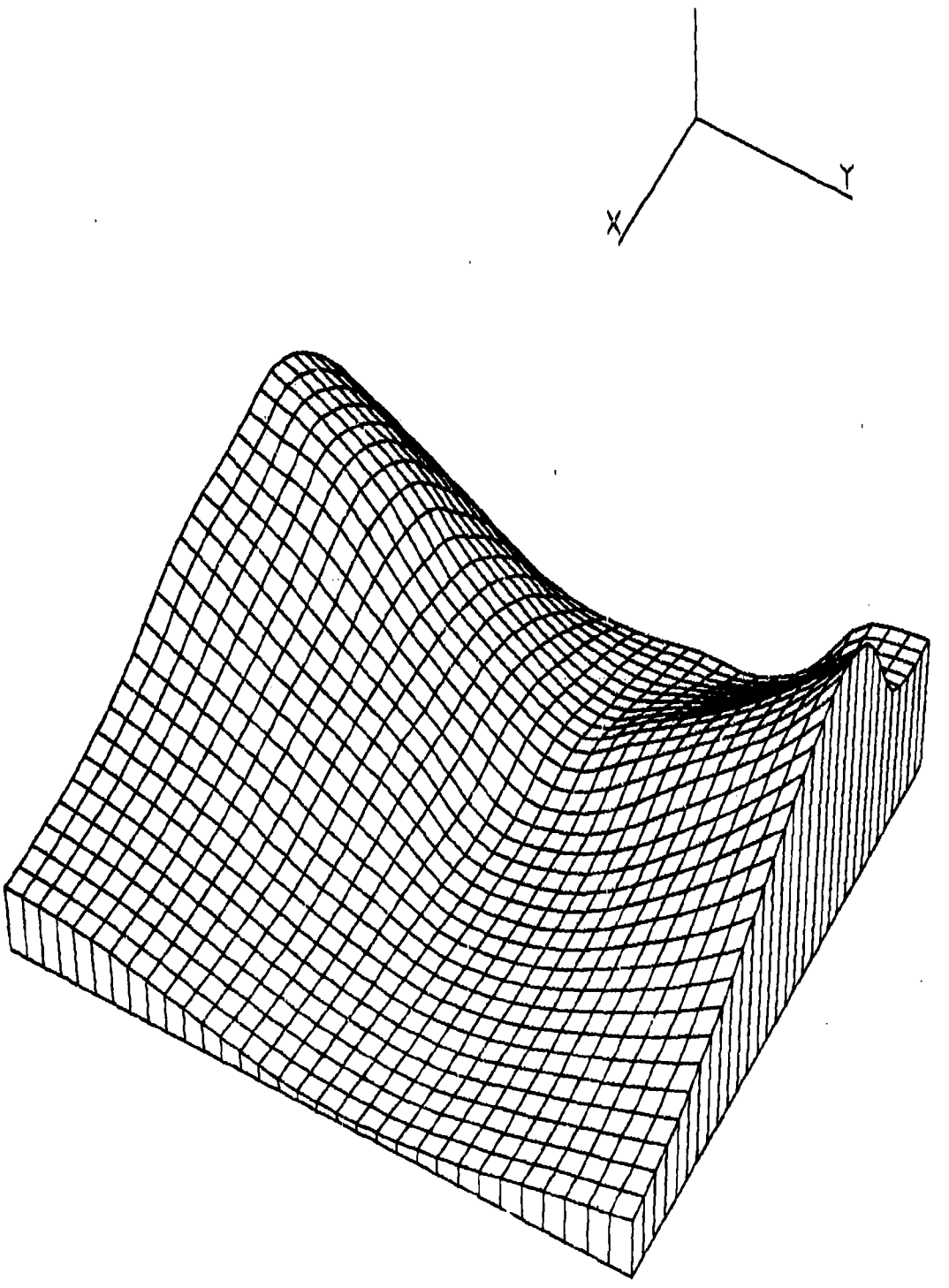
Figure 4.2.3.10



33PT SADDLE ON SPARSE 31079 1909

NIELSON-FRANKE QUAD-TRI NPPR = 18 R = 0.522

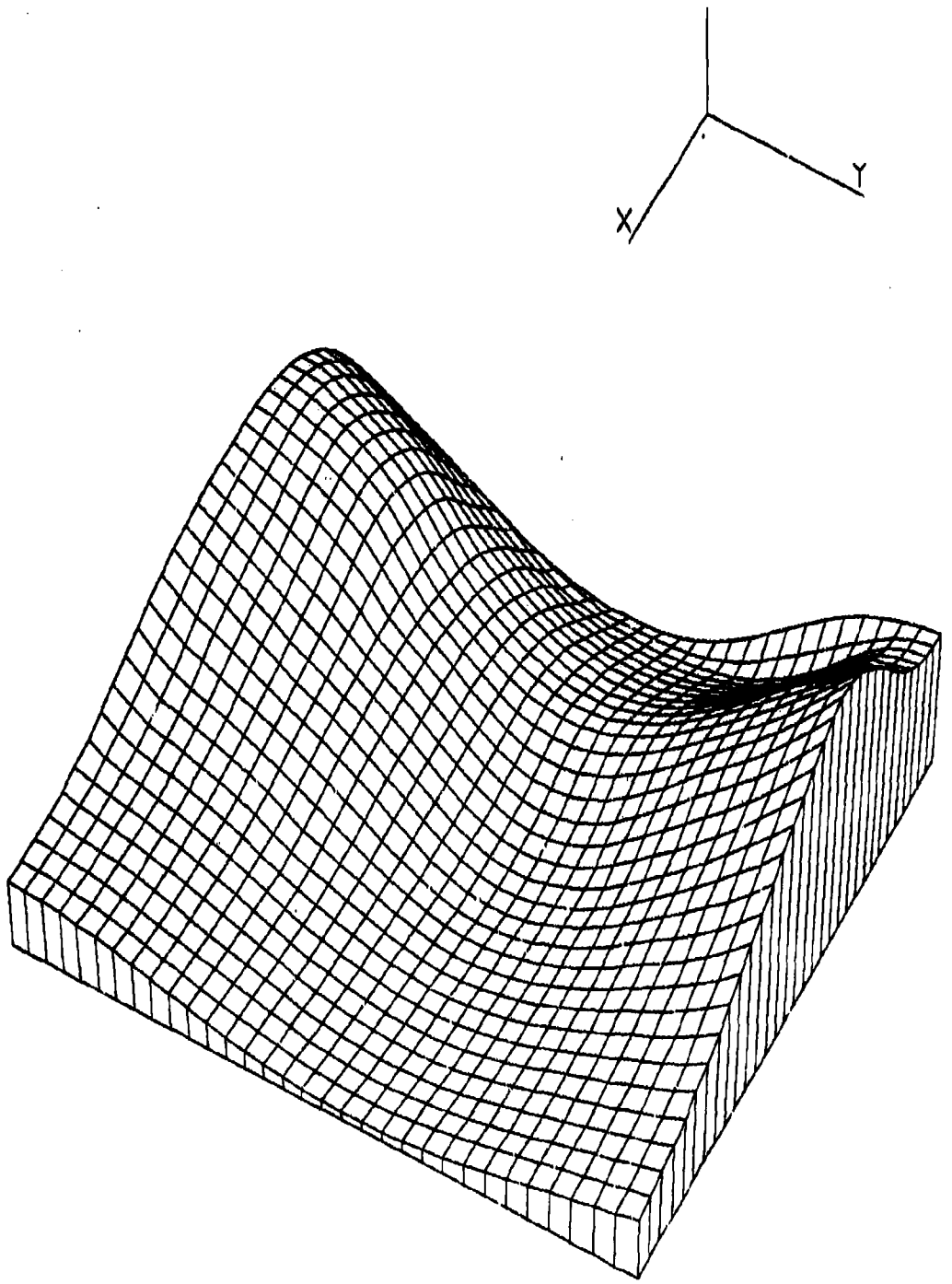
Figure 4.2.3.13



33PT SADDLE ON SPARSE 31179 1441

QUADRATIC SHEPARD METHOD NPPR = 918 R = 0.369

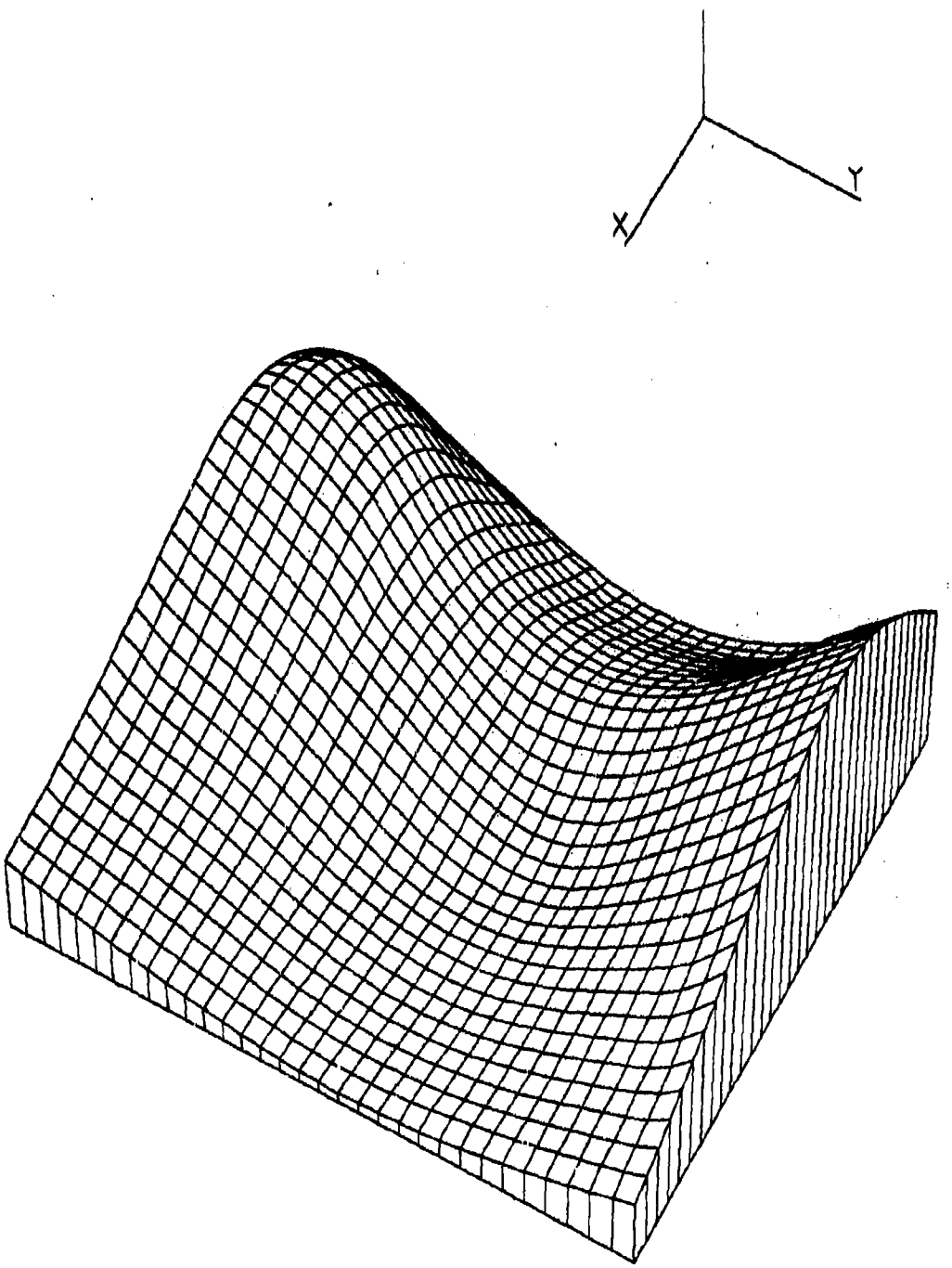
Figure 4.2.3.14



33PT SADDLE ON SPARSE 31079 1932

AKIMA - L2 QUAD DERIVS NPPR = 18 R = 0.522

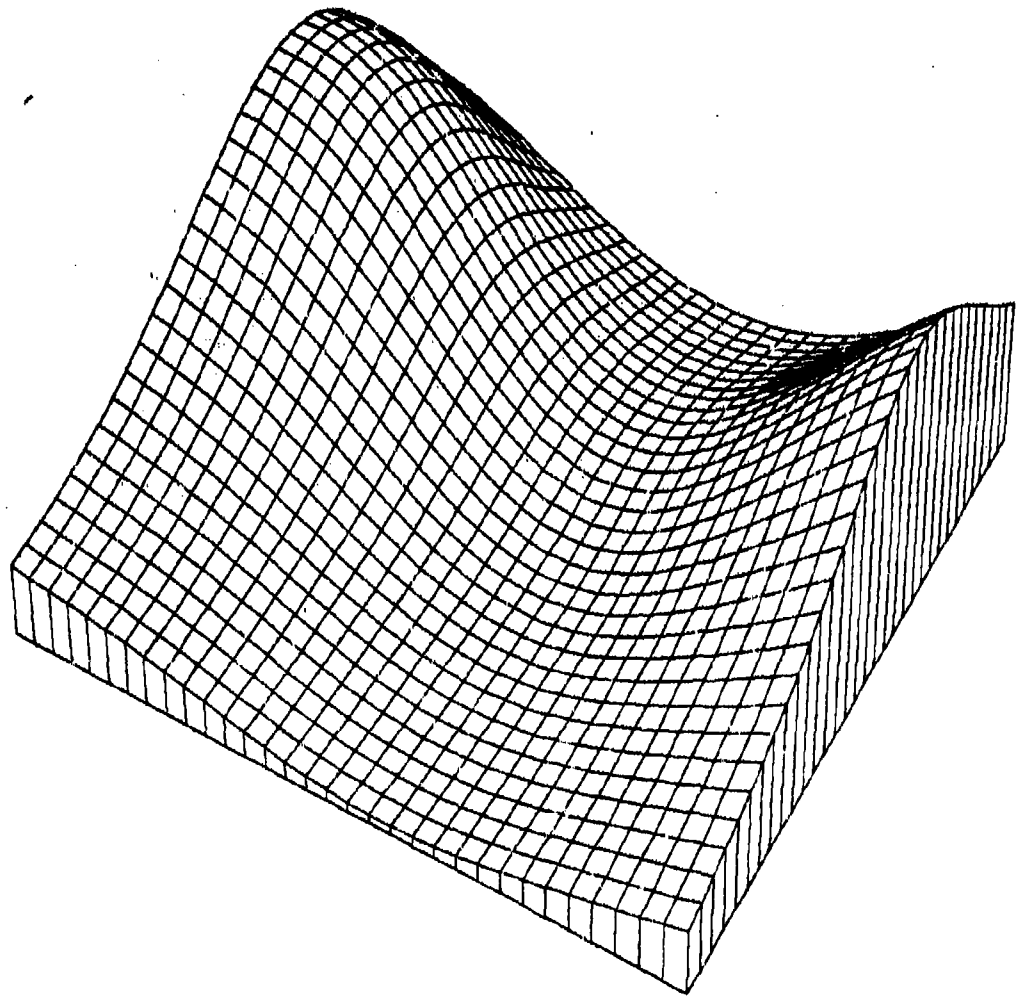
Figure 4.2.3.16



33PT SADDLE ON SPARSE 31079 1910

NIELSON MIN NORM NETWORK

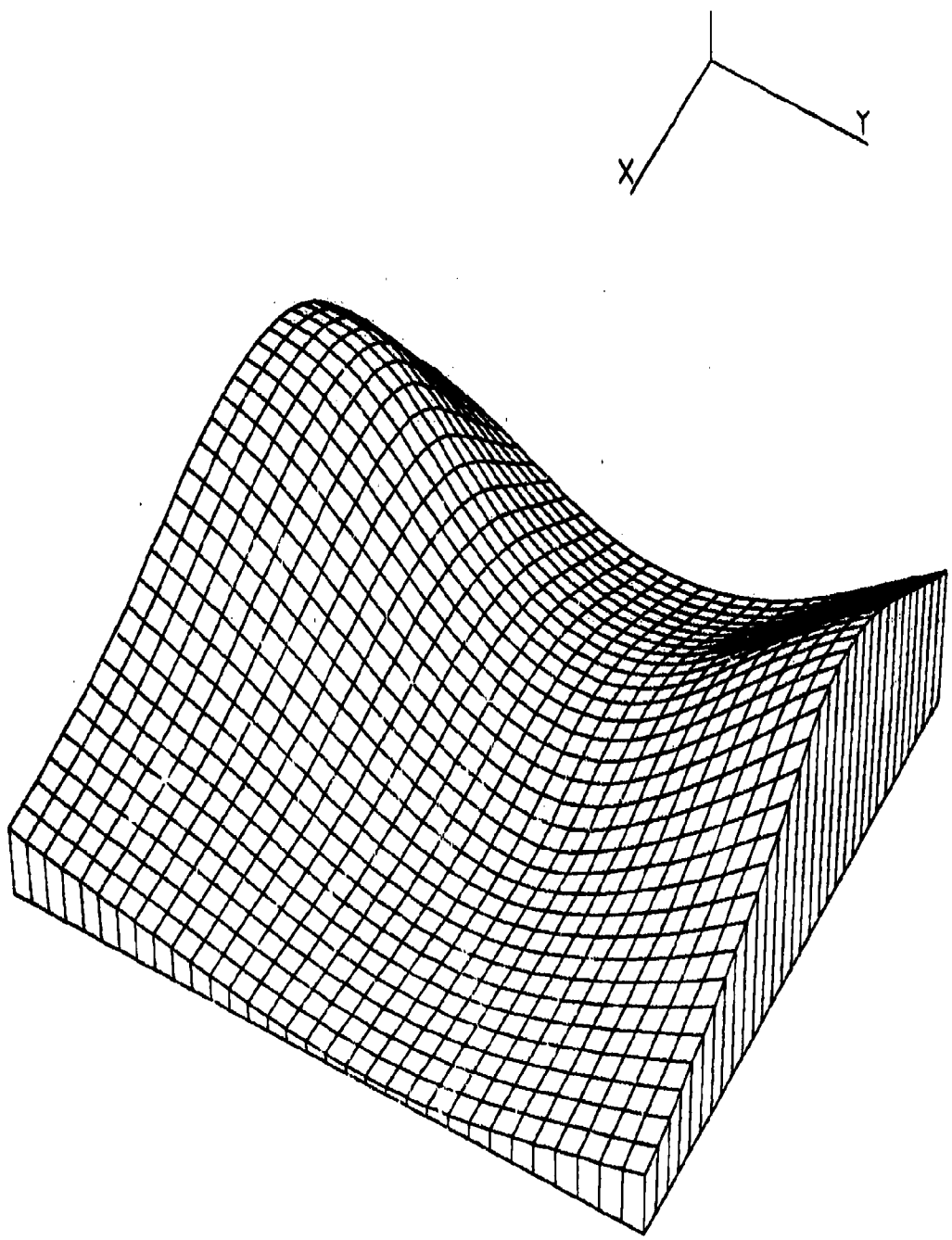
Figure 4.2.3.19



33PT SADDLE ON SPARSE 31579 2016

HARDY MQ SURFACE NPPR = 25 R = 0.308

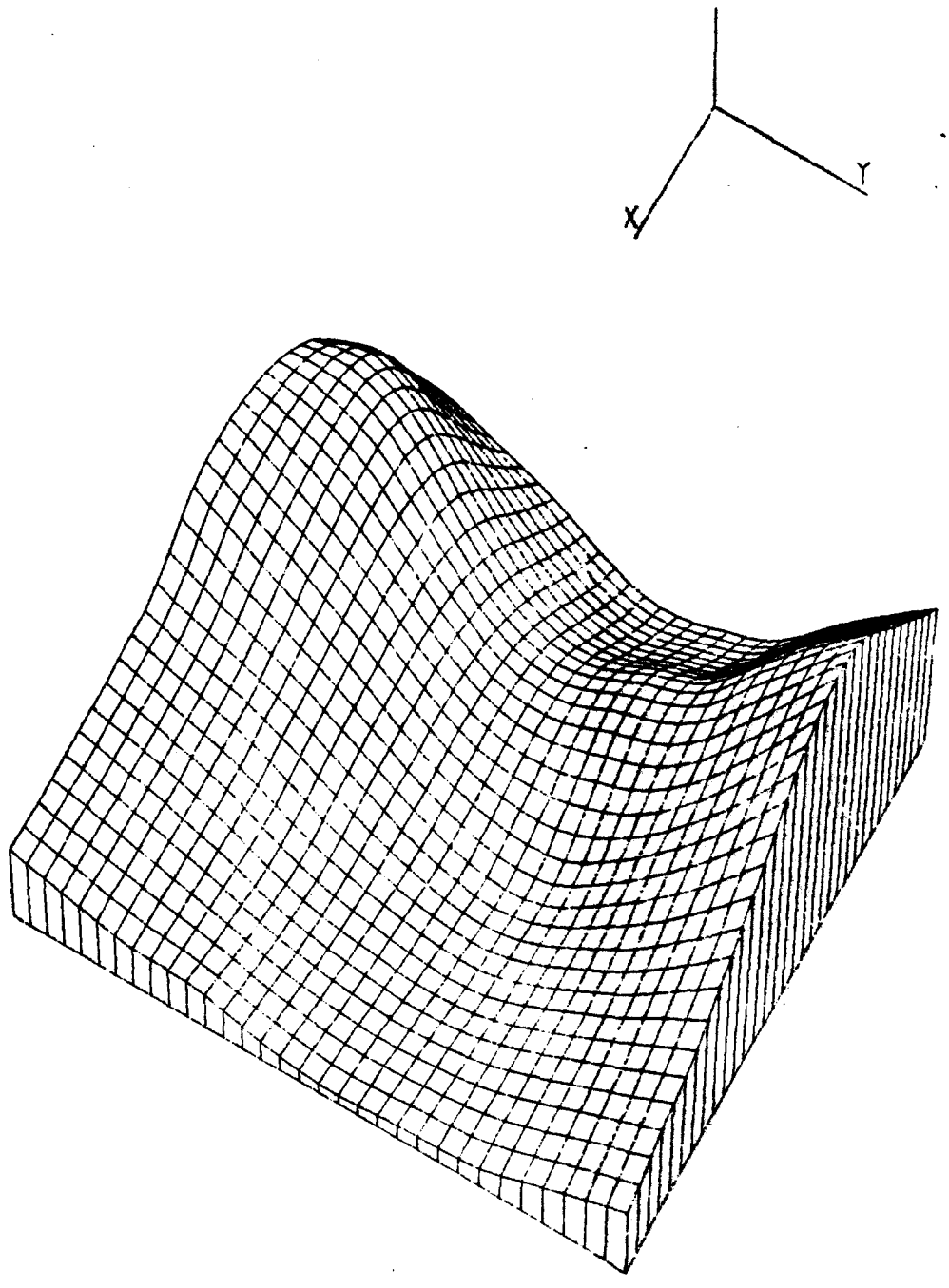
Figure 4.2.3.21



33PT SADDLE ON SPARSE 20779 1038

DUCHON S THIN PLATE

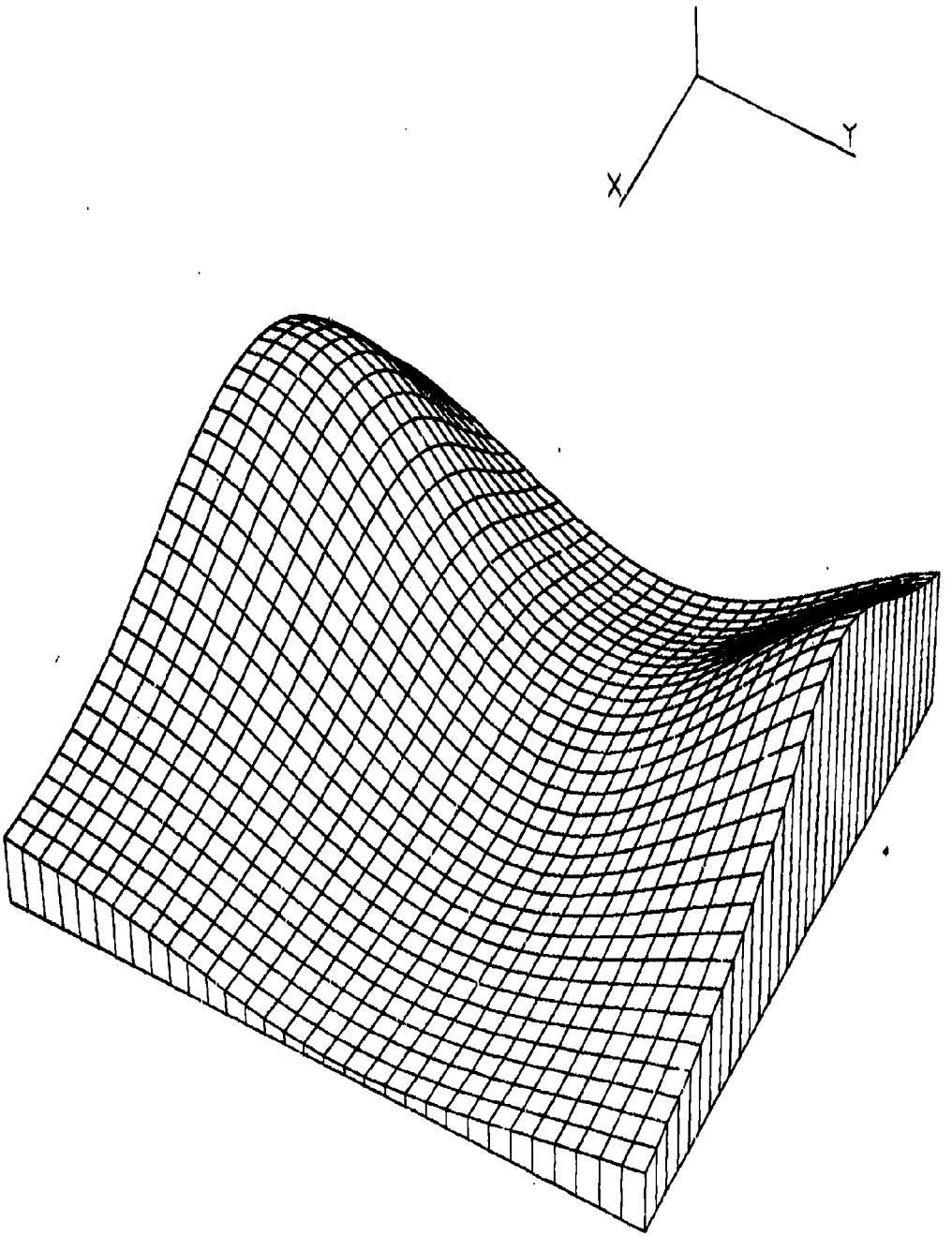
Figure 4.2.3.23



33PT SADDLE ON SPARSE 80873 837

FRANKE W/ THIN PLATE NPPR = 6

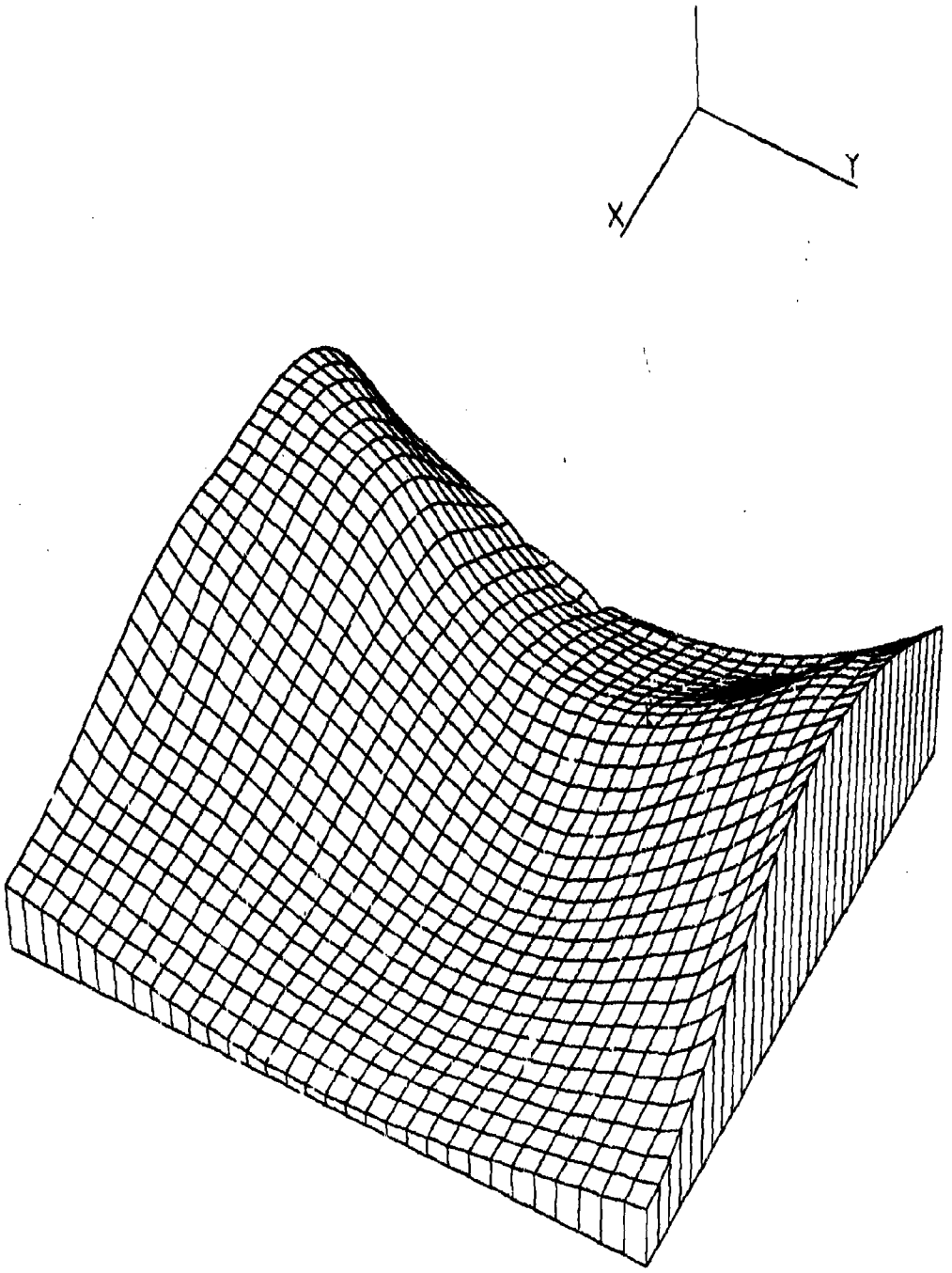
Figure 4.2.3.24



33PT SADDLE ON SPARSE 40679 1527

HARDY RECIP MQ SURFACE NPPR = 25 R = 0.308

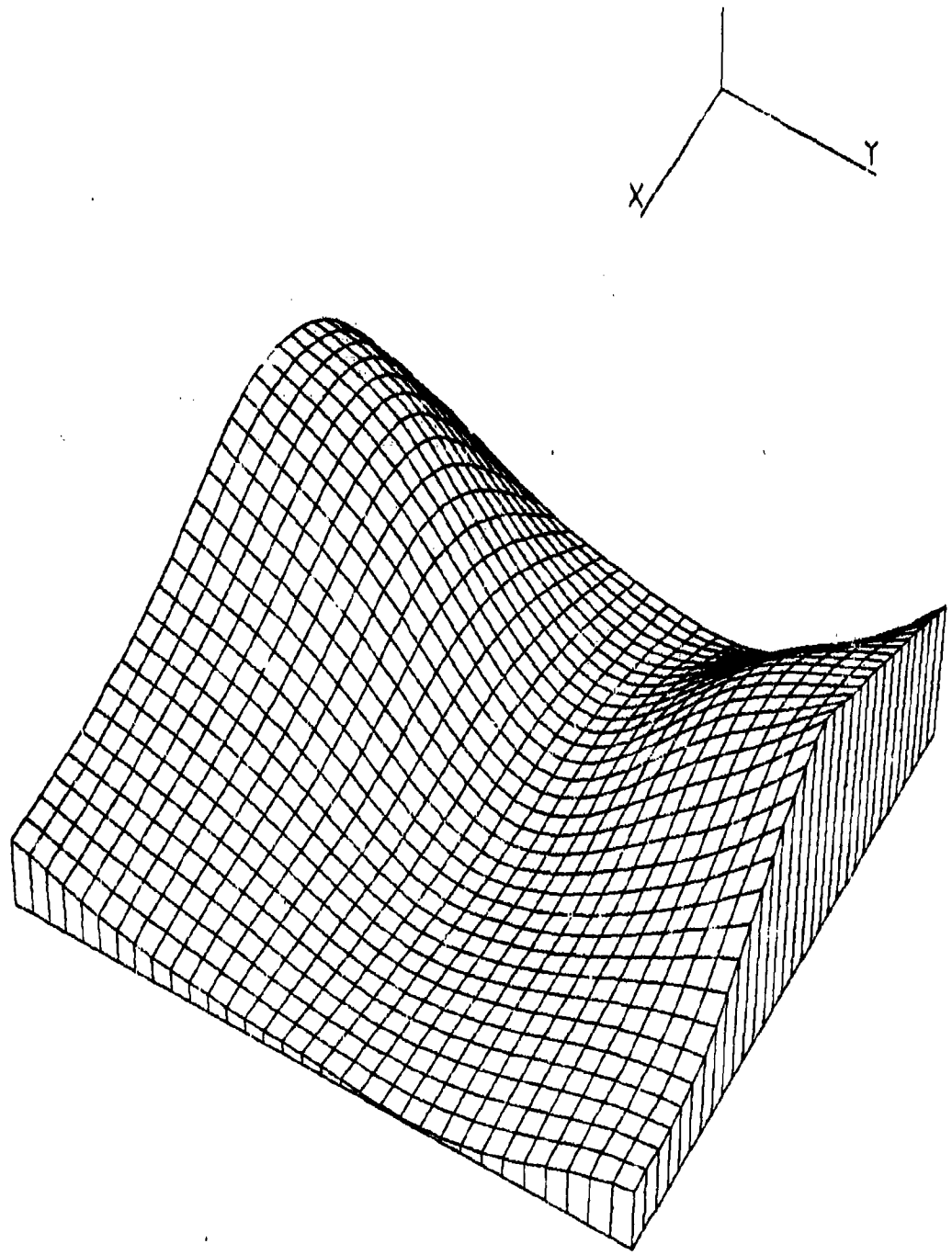
Figure 4.2.3.27



33PT SADDLE ON SPARSE 32379 1422

LAWSON TRIANGLE METHOD

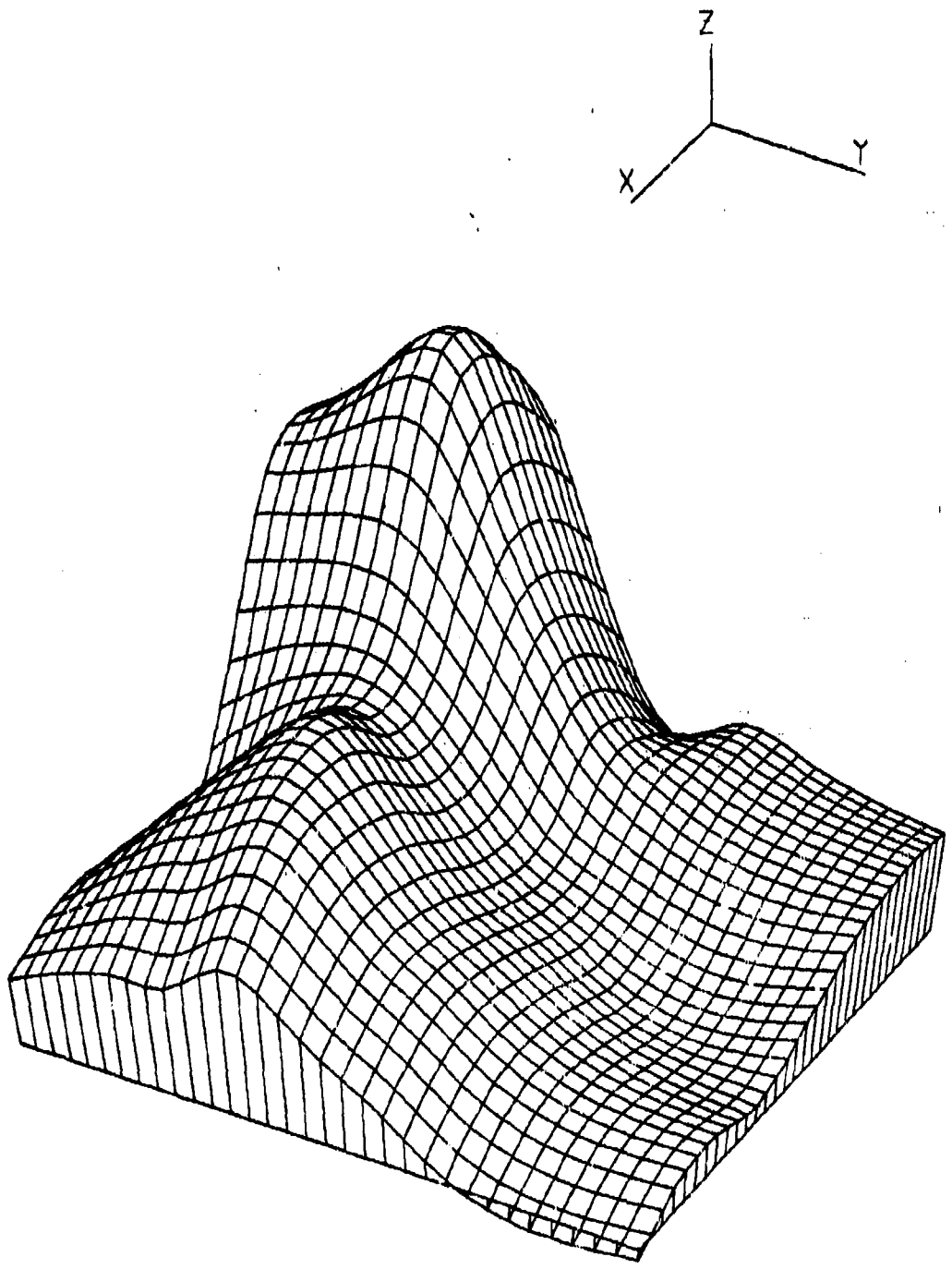
Figure 4.2.3.28



33PT SADDLE ON SPARSE 80279 1103

FOLEY ITERATED BICUBICS NPPR = 3

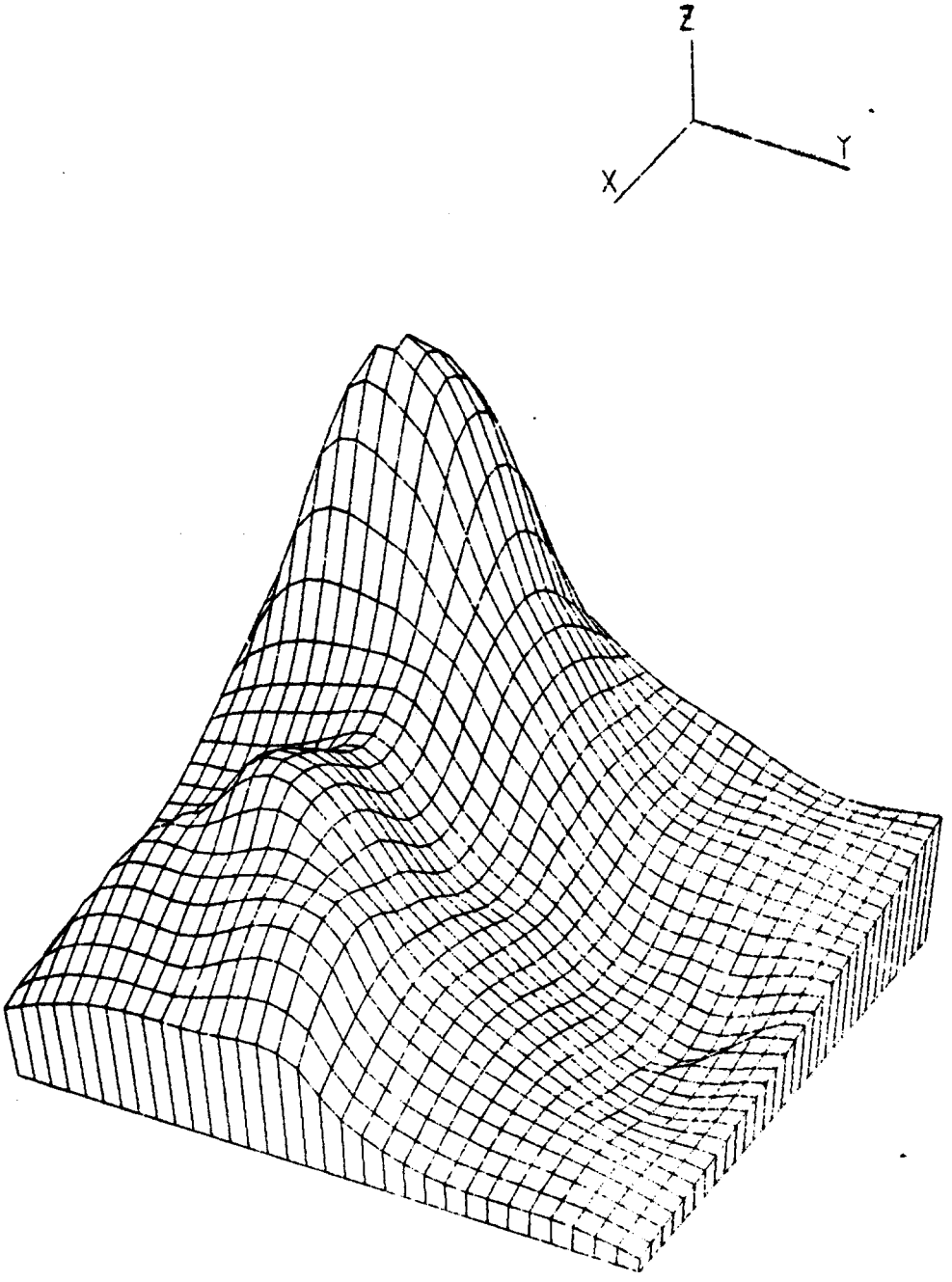
Figure 4.2.3.30



25PT EXP HUMPS, DIP ON GREGS 21679 942

FRANKE - SARD BS PLANE NPPR = 6

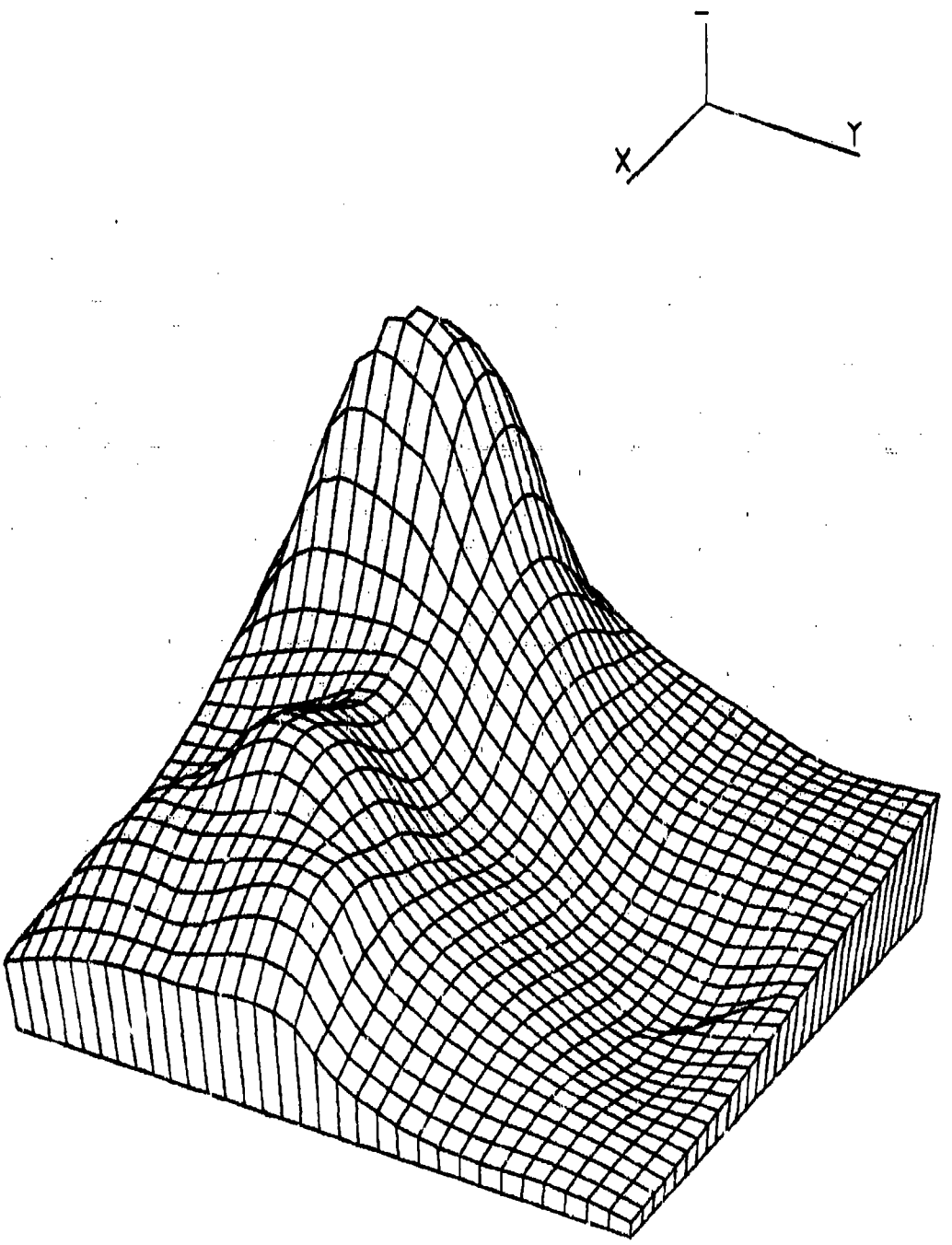
Figure 4.3.1.1



25PT EXP HUMPS, DIP ON GREGS 41679 1021

AKIMA - FEM FIT NPPR = 0

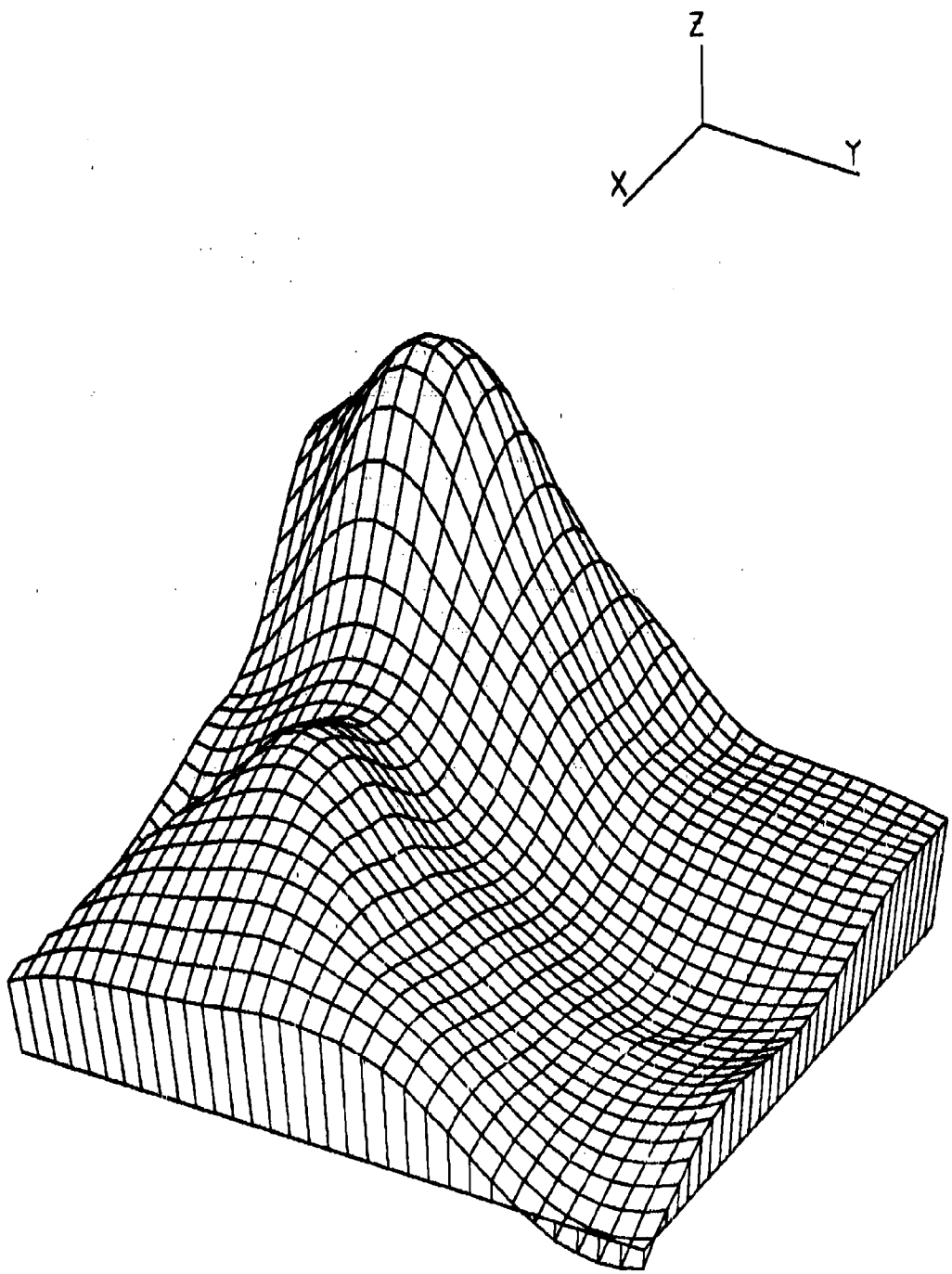
Figure 4.3.1.4



25PT EXP HUMPS, DIP ON GREGS 30779 1334

MODIFIED AKIMA FEM FIT NPPR = 6

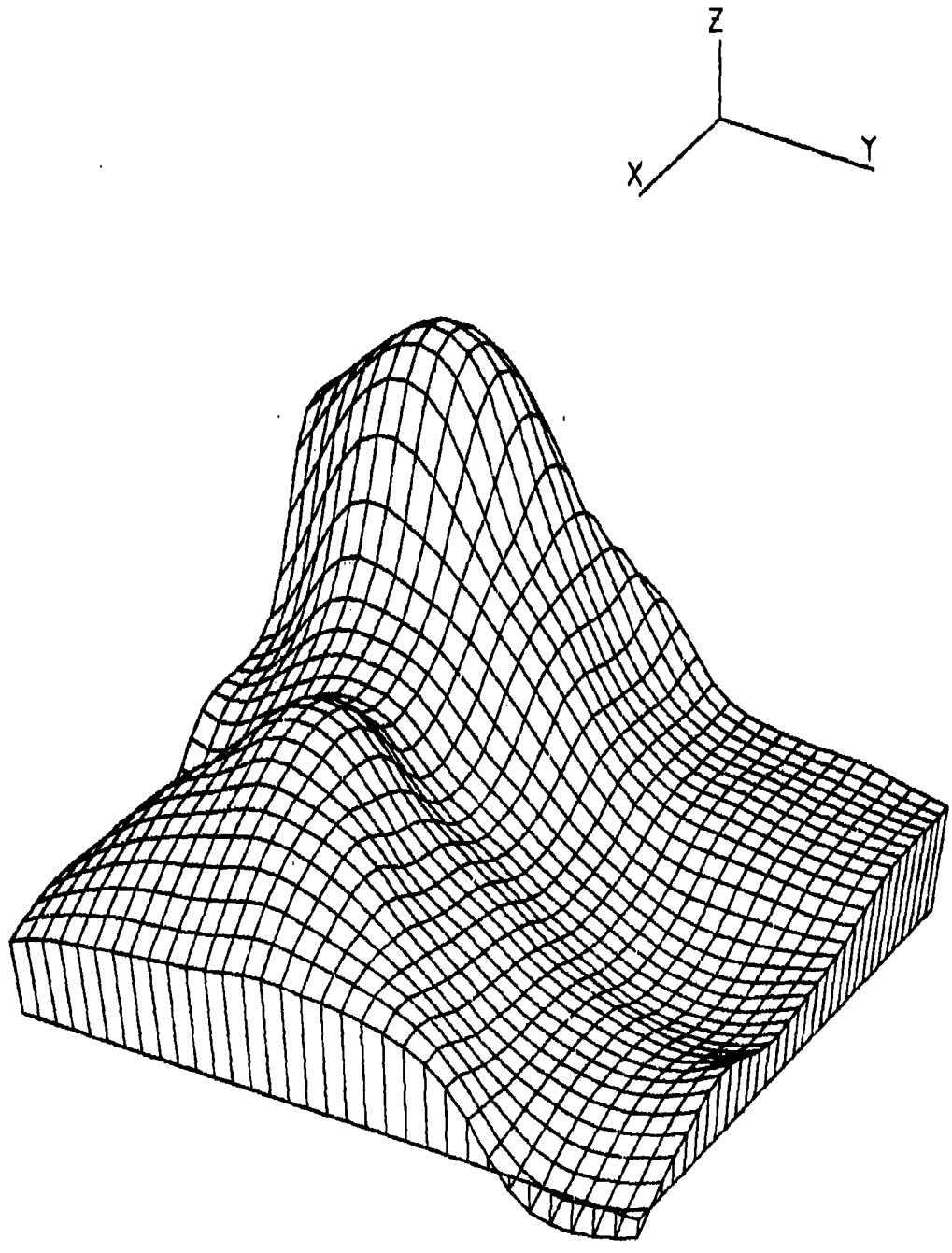
Figure 4.3.1.10



25PT EXPONENTIAL ON GREGS 11579 1158

NIELSON-FRANKE QUAD-TRI NPPR = 18 R = 0.586

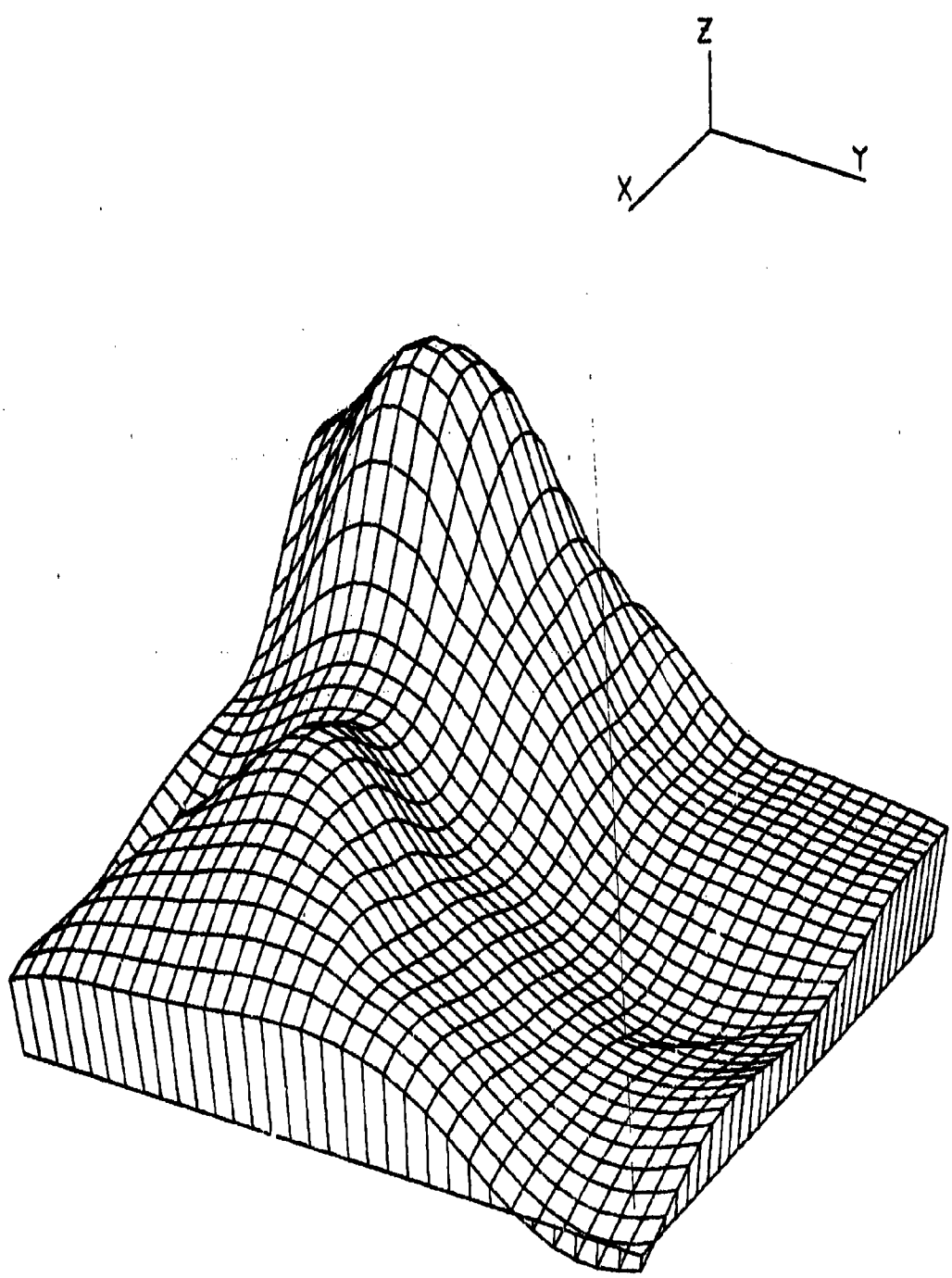
Figure 4.3.1.13



25PT EXPONENTIAL ON GREGS 11579 1209

QUADRATIC SHEPARD METHOD NPPR = 9,18 R = 0.414

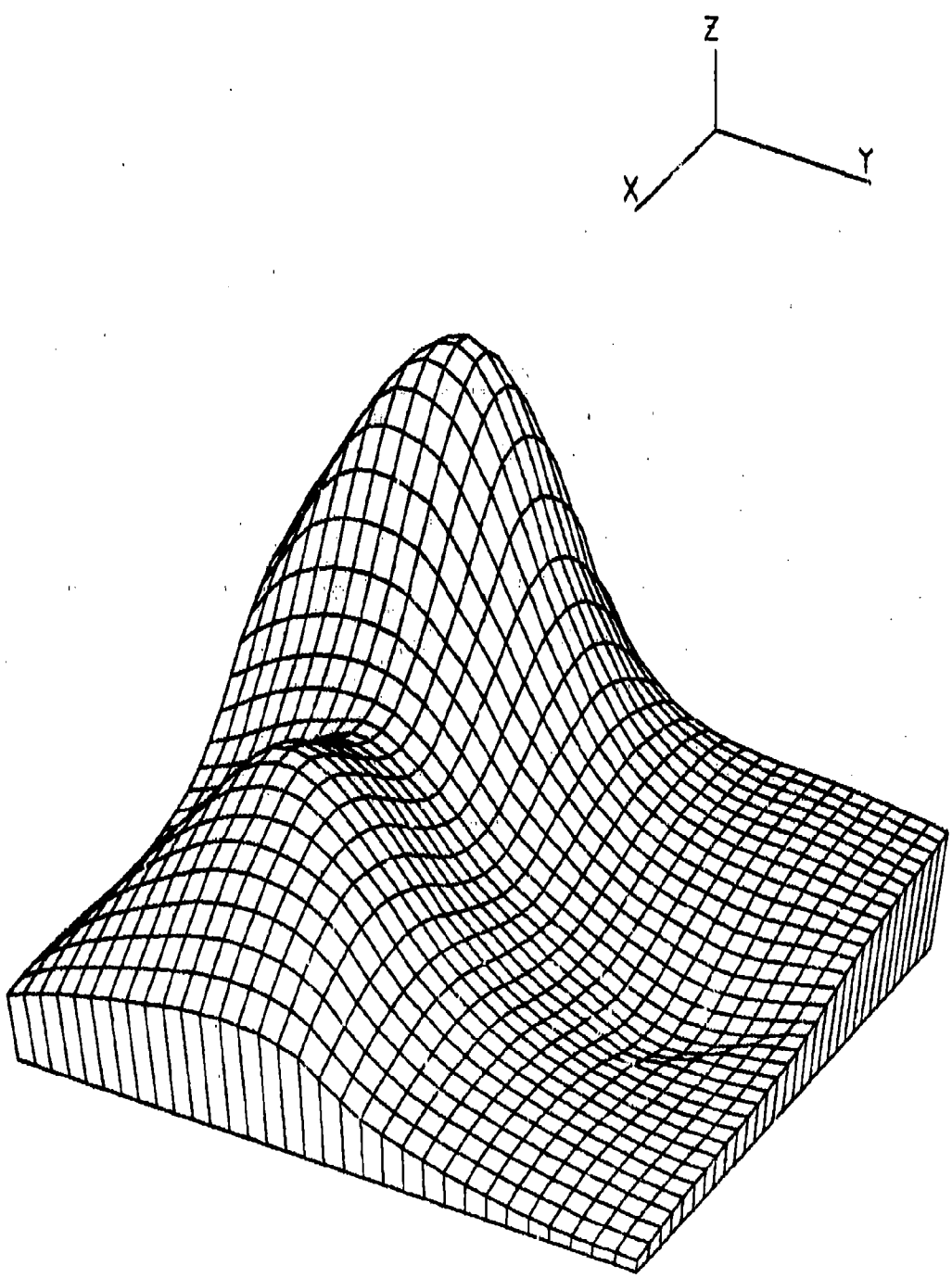
Figure 4.3.1.14



25PT EXP HUMPS, DIP ØN GREGS 30979 1142

AKIMA - L2 QUAD DERIVS NPPR = 18 R = 0.586

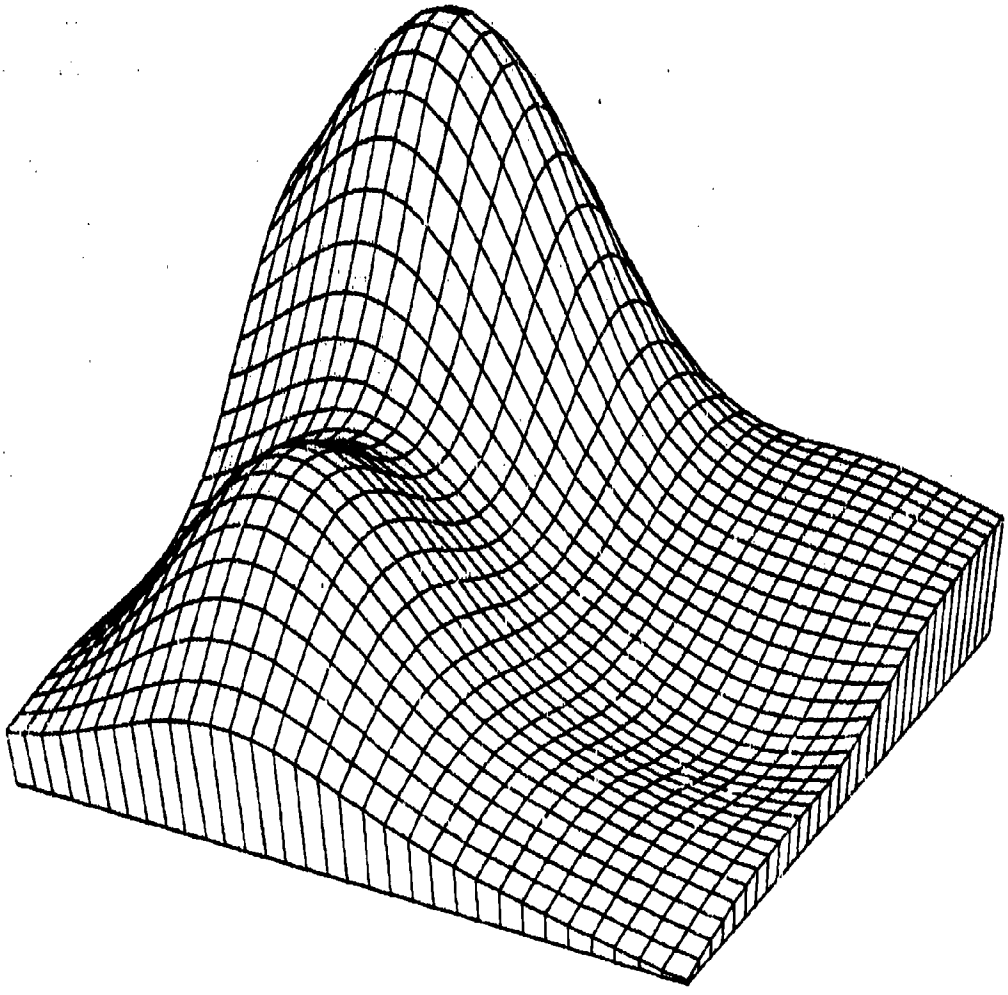
Figure 4.3.1.16



25PT EXPONENTIAL ON GREGS 13179 929

NIELSON MIN NORM NETWORK

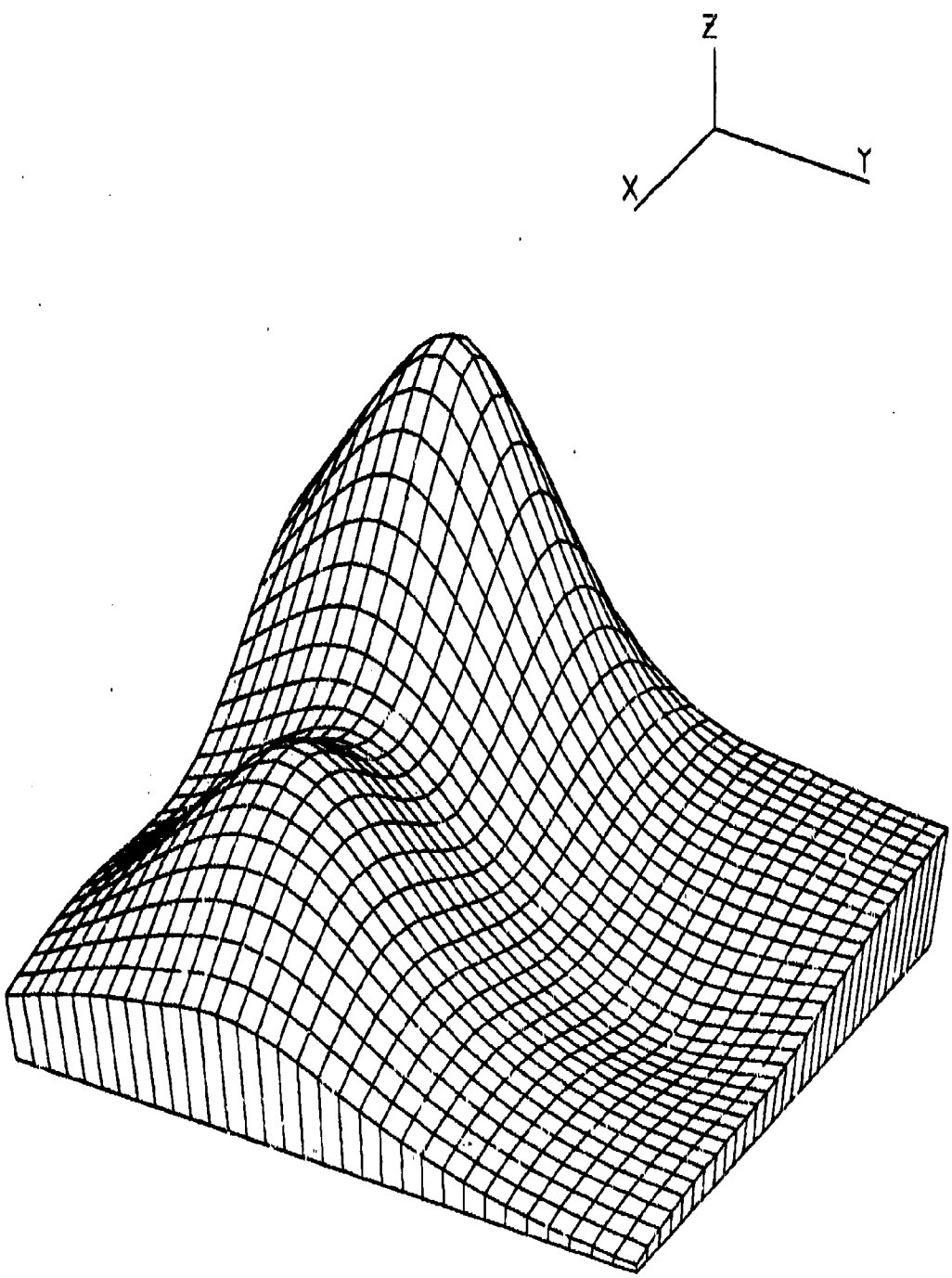
Figure 4.3.1.19



25PT EXP HUMPS, DIP ØN GREGS 32279 1717

HARDY MQ SURFACE NPPR = 25 R = 0.345

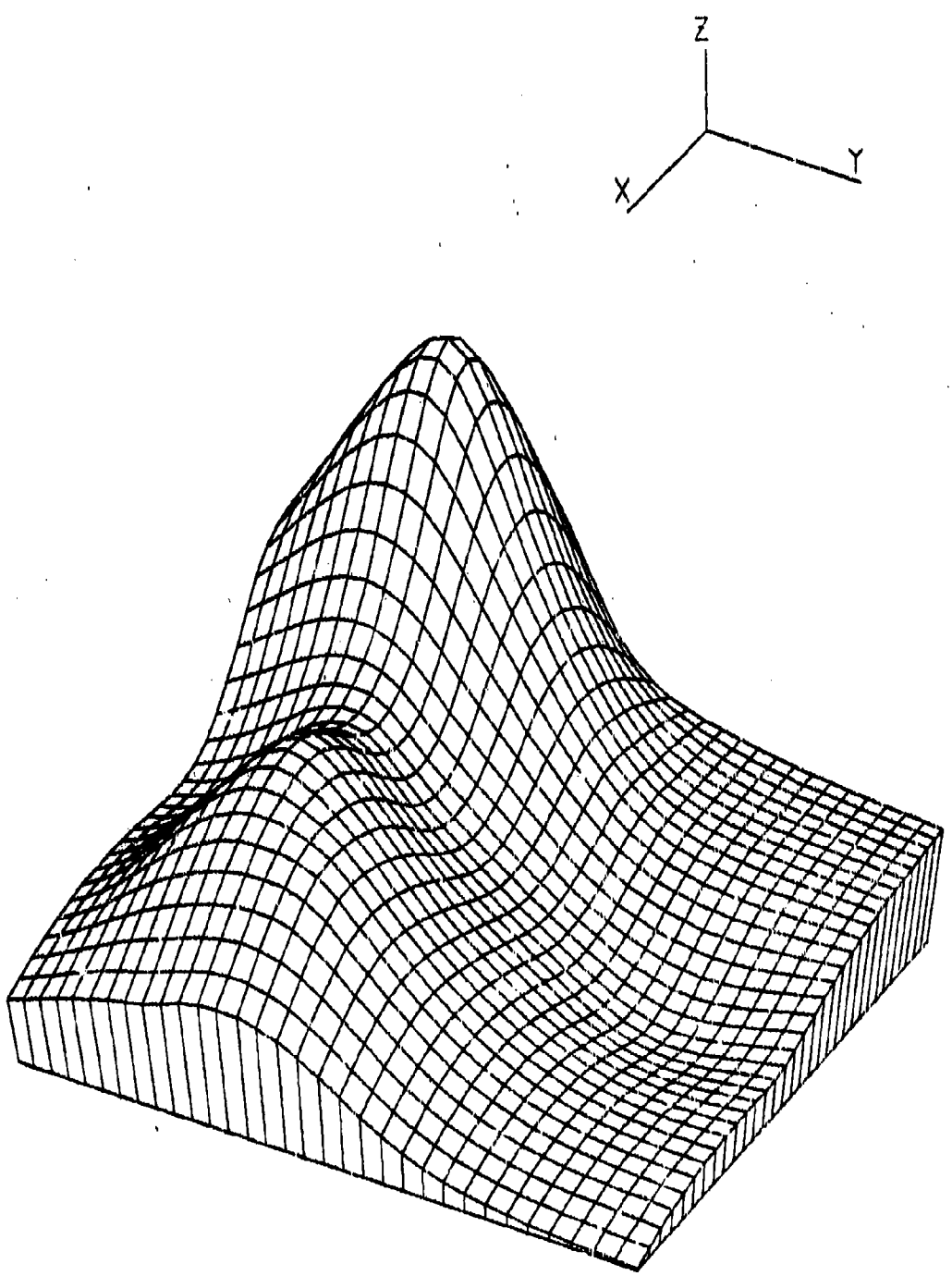
Figure 4.3.1.21



25PT EXP HUMPS, DIP ØN GREGS 20679 1535

DUCHØN S THIN PLATE

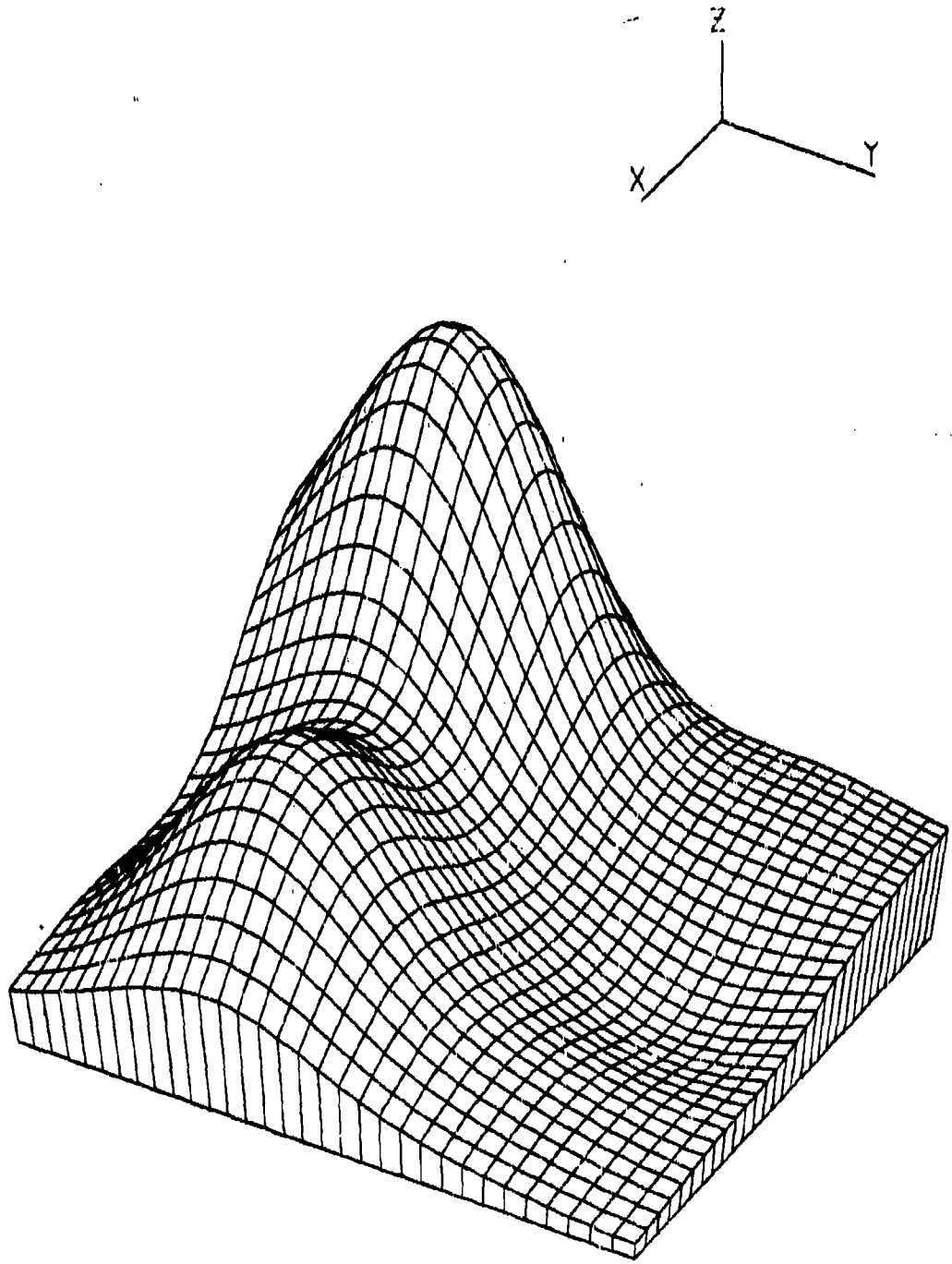
Figure 4.3.1.23



25PT EXP HUMPS, DIP ØN GREGS 80879 909

FRANKE W/ THIN PLATE NPPR = 6

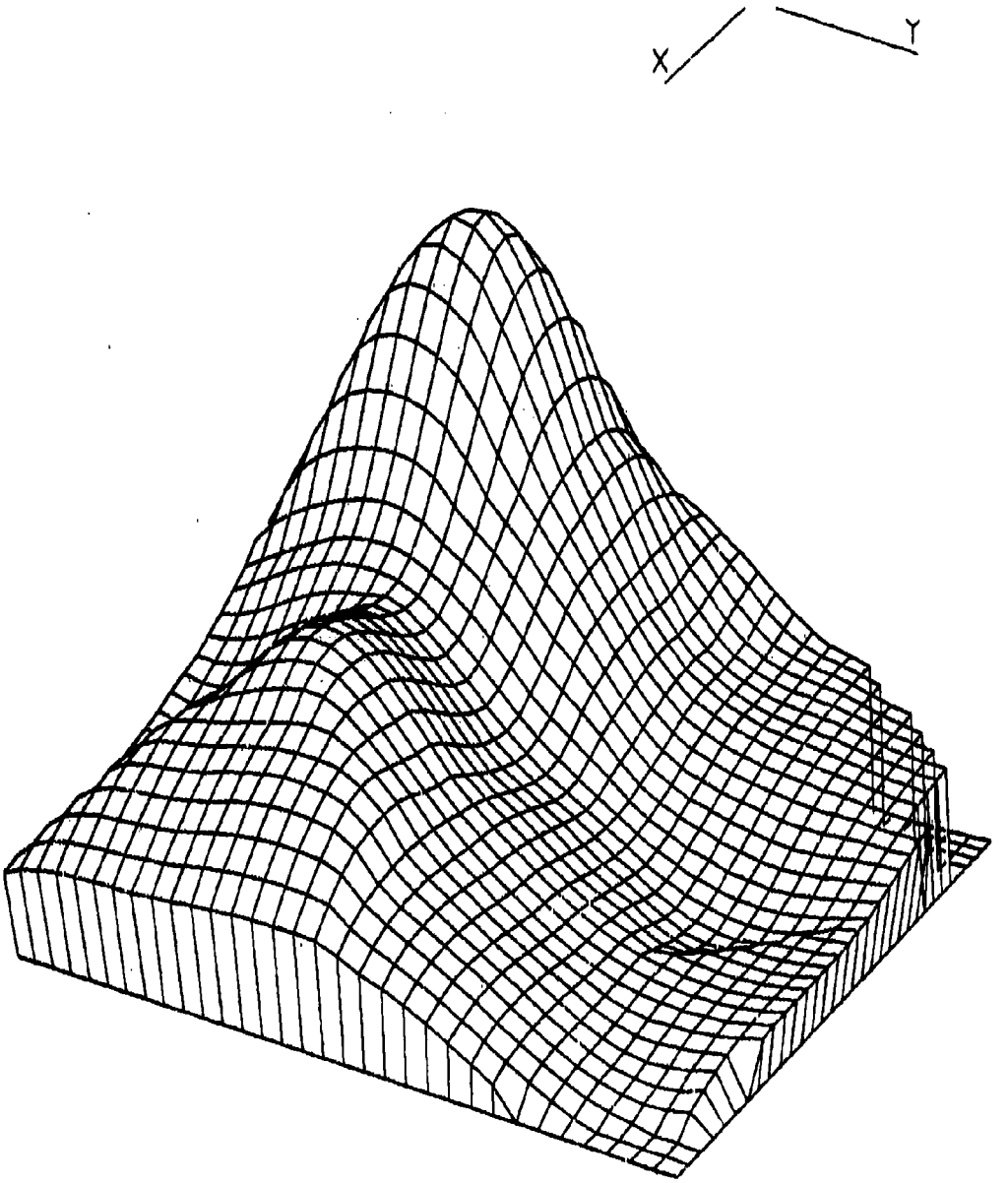
Figure 4.3.1.24



25PT EXP HUMPS, DIP ON GREGS 41079 1824

HARDY RECIP MQ SURFACE NPPR = 25 R = 0.345

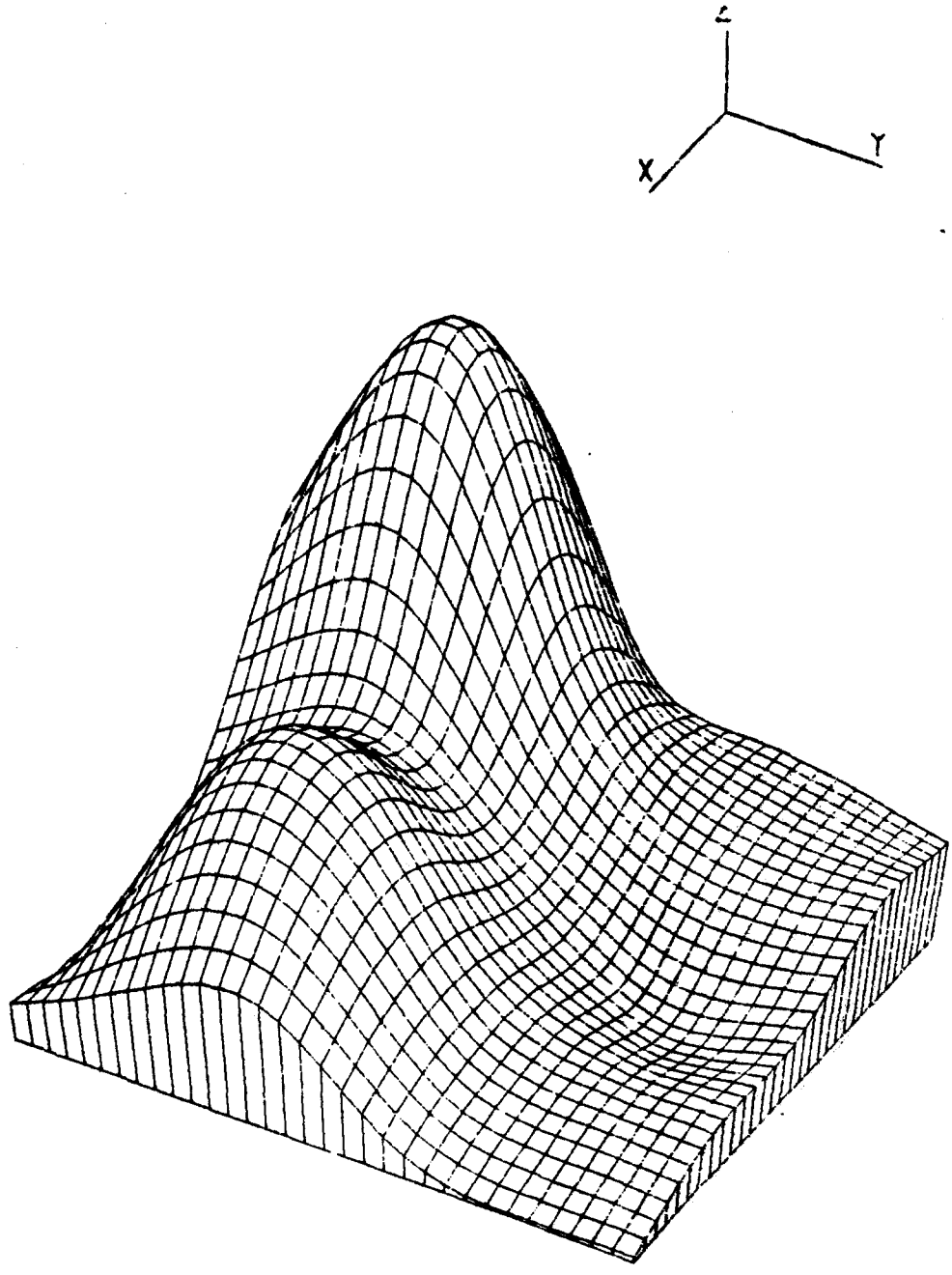
Figure 4.3.1.27



25PT EXP HUMPS, DIP ON GREGS 32379 1846

LAWSON TRIANGLE METHOD

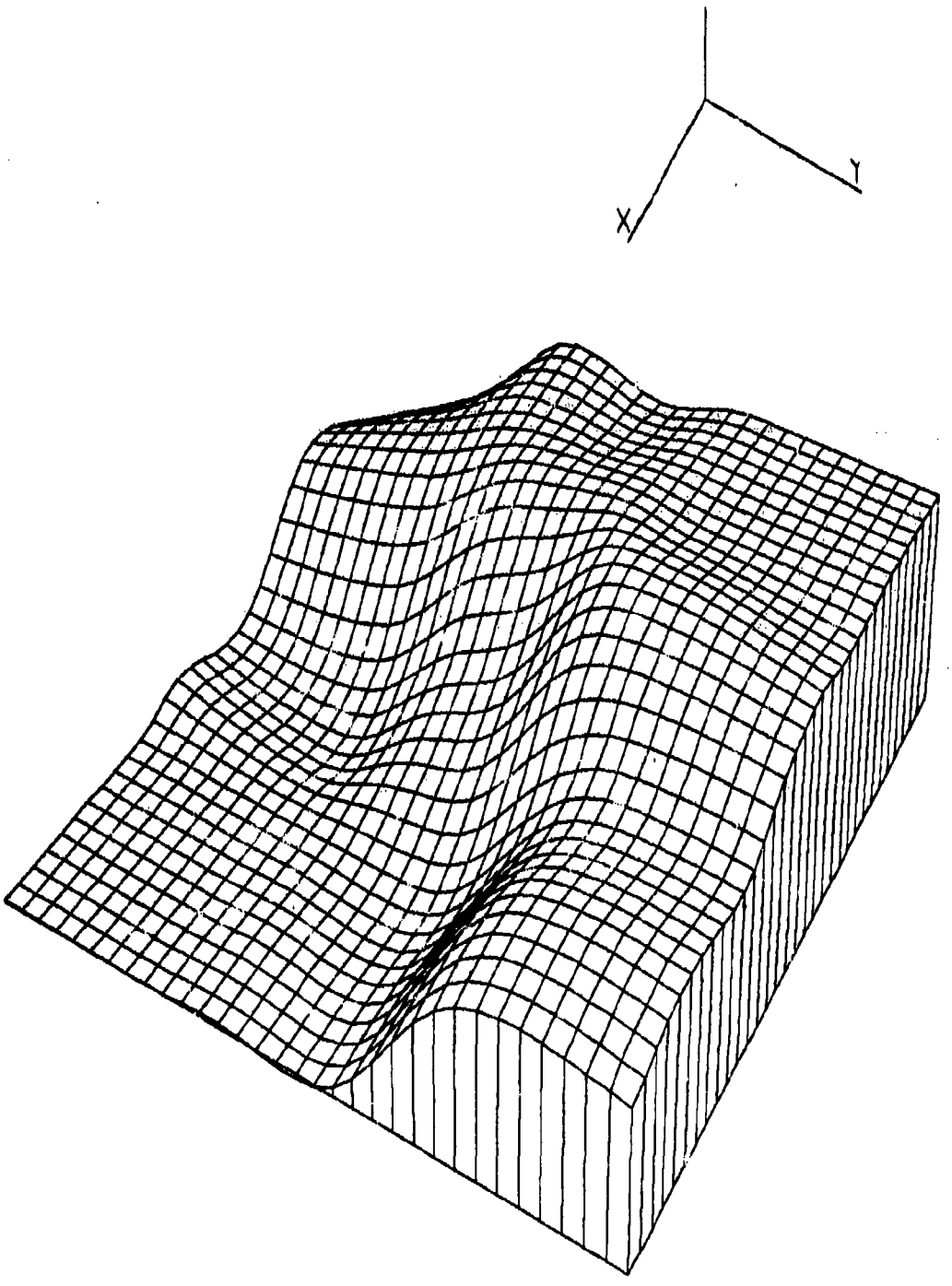
Figure 4.3.1.28



25PT EXP HUMPS, DIP ON GREGS 80279 1049

FOLEY ITERATED BICUBICS NPPR = 3

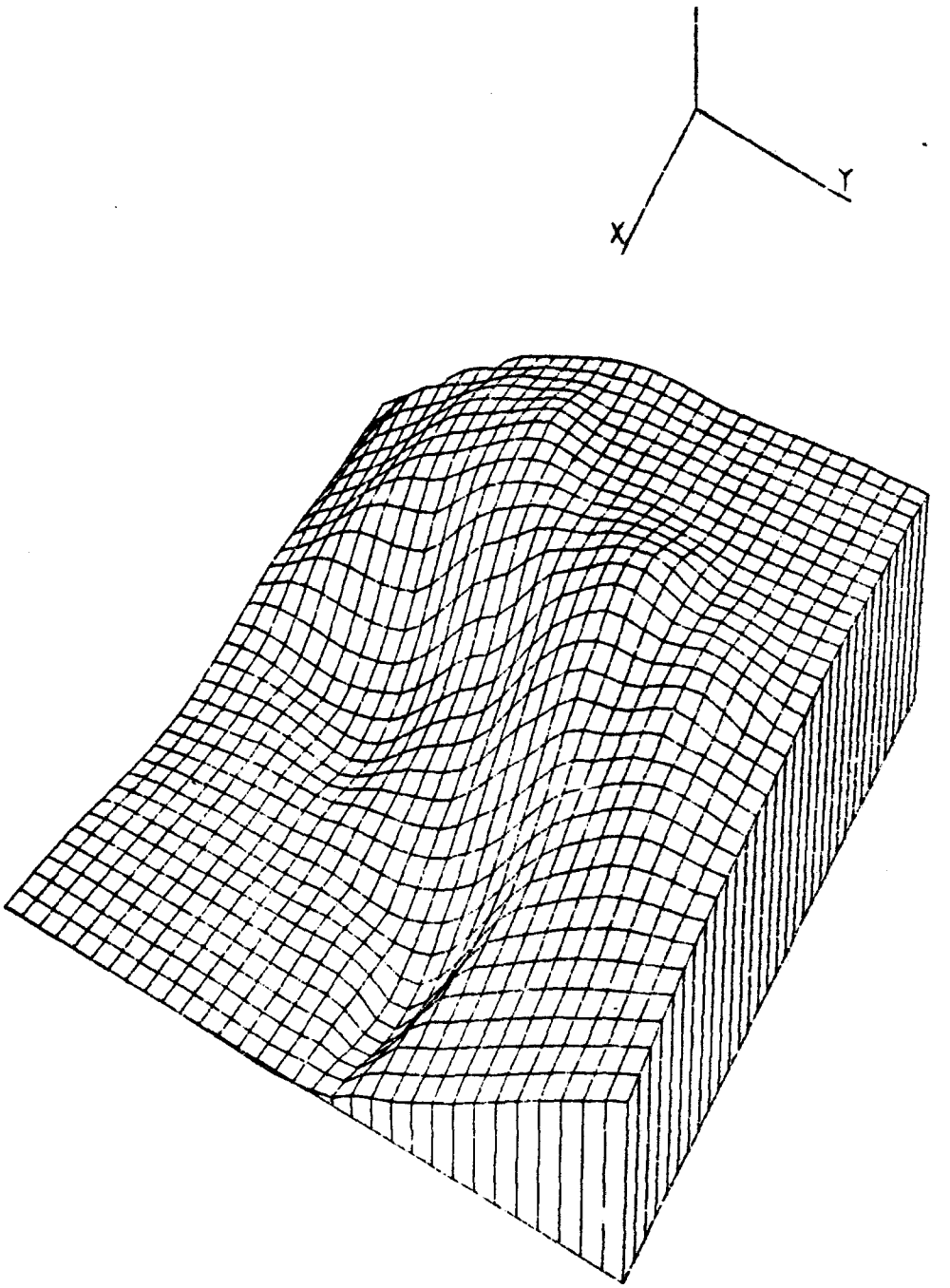
Figure 4.3.1.30



25PT CLIFF ON GREGS 21679 944

FRANKE - SARD BS PLANE NPPR = 3

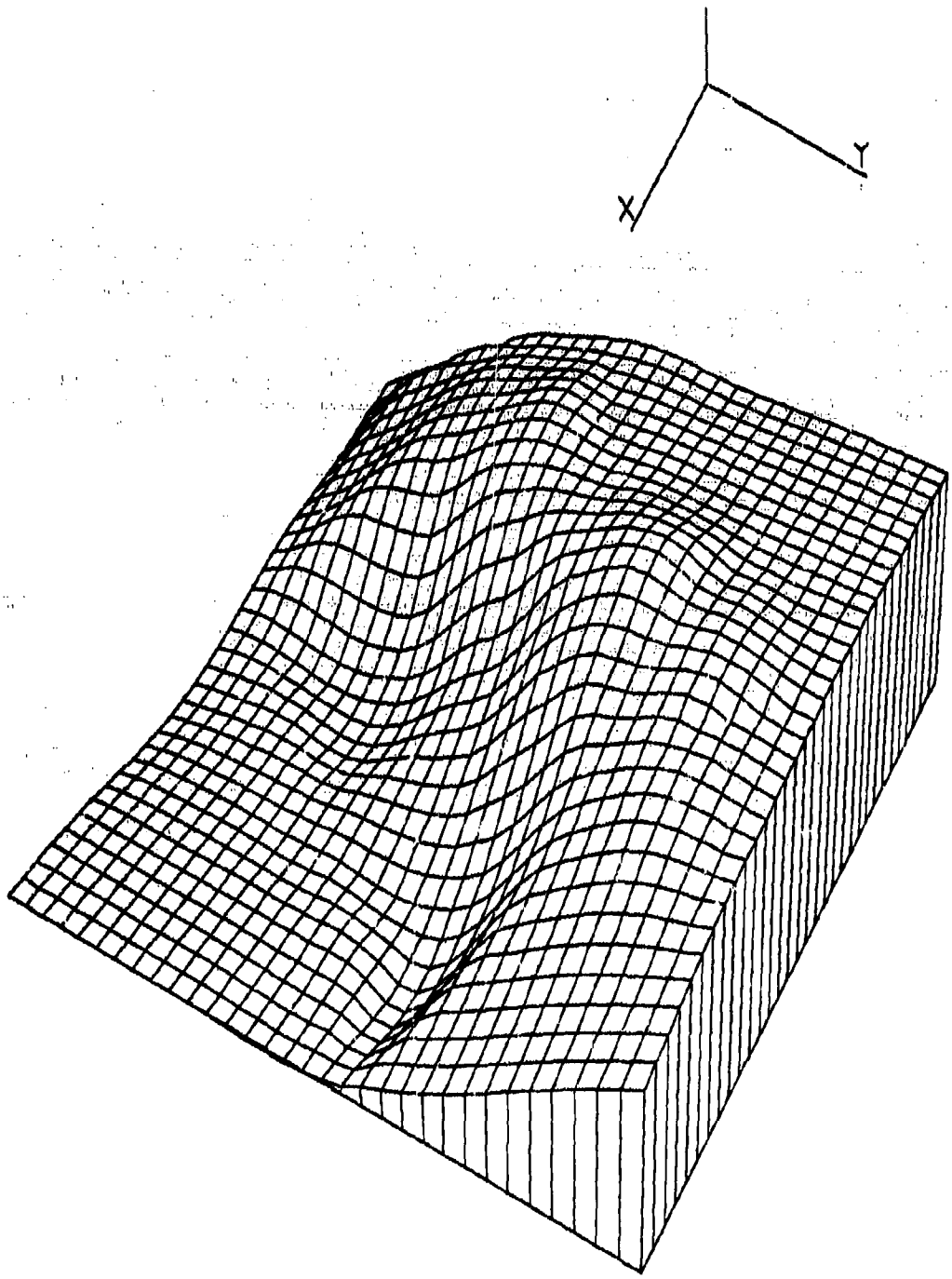
Figure 4.3.2.1



25PT CLIFF ON GREGS 41879 1022

AKIMA - FEM FIT NPPR = 6

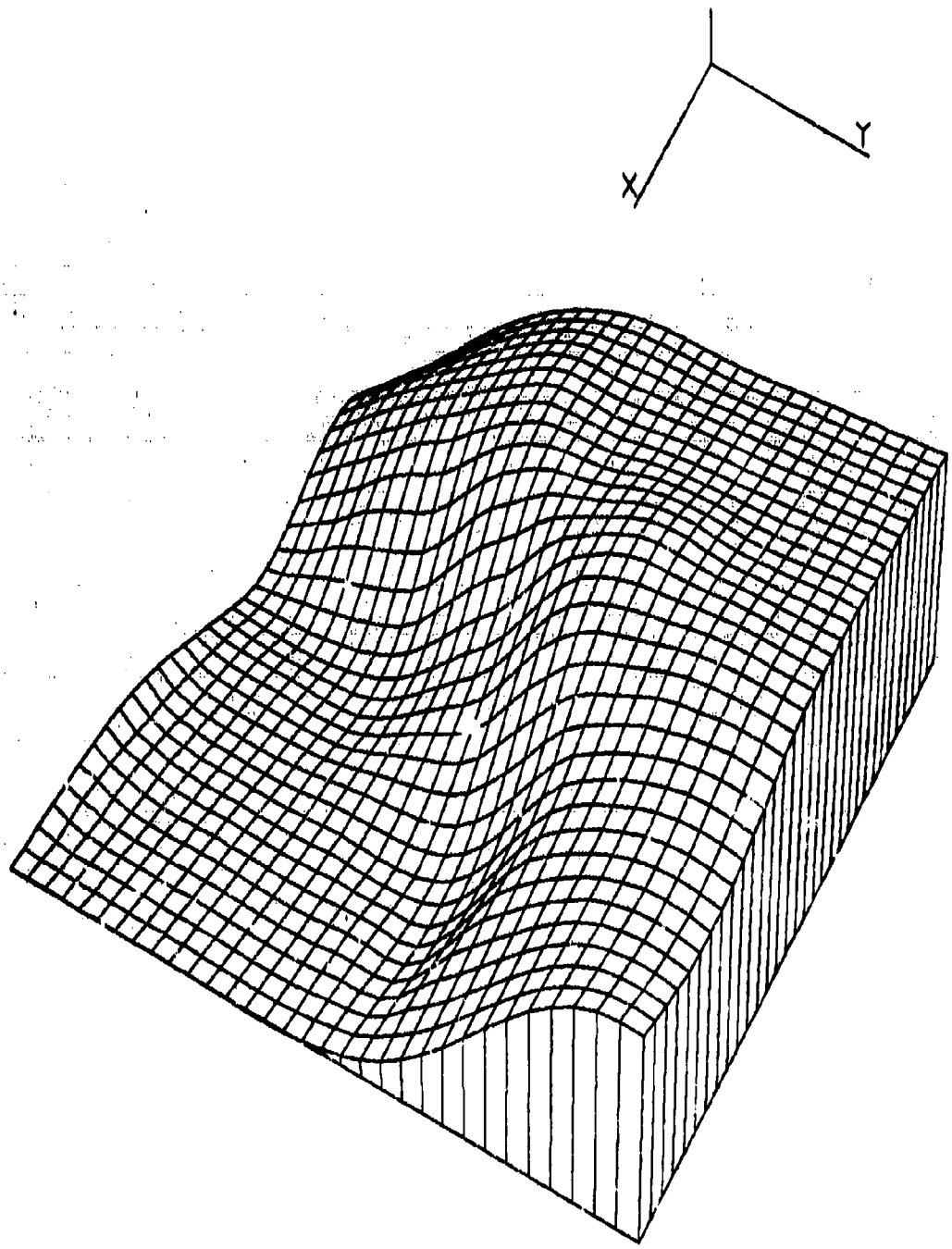
Figure 4.3.2.4



25PT CLIFF ON GREGS 31079 1934

MODIFIED AKIMA FEM FIT NPPR = 6

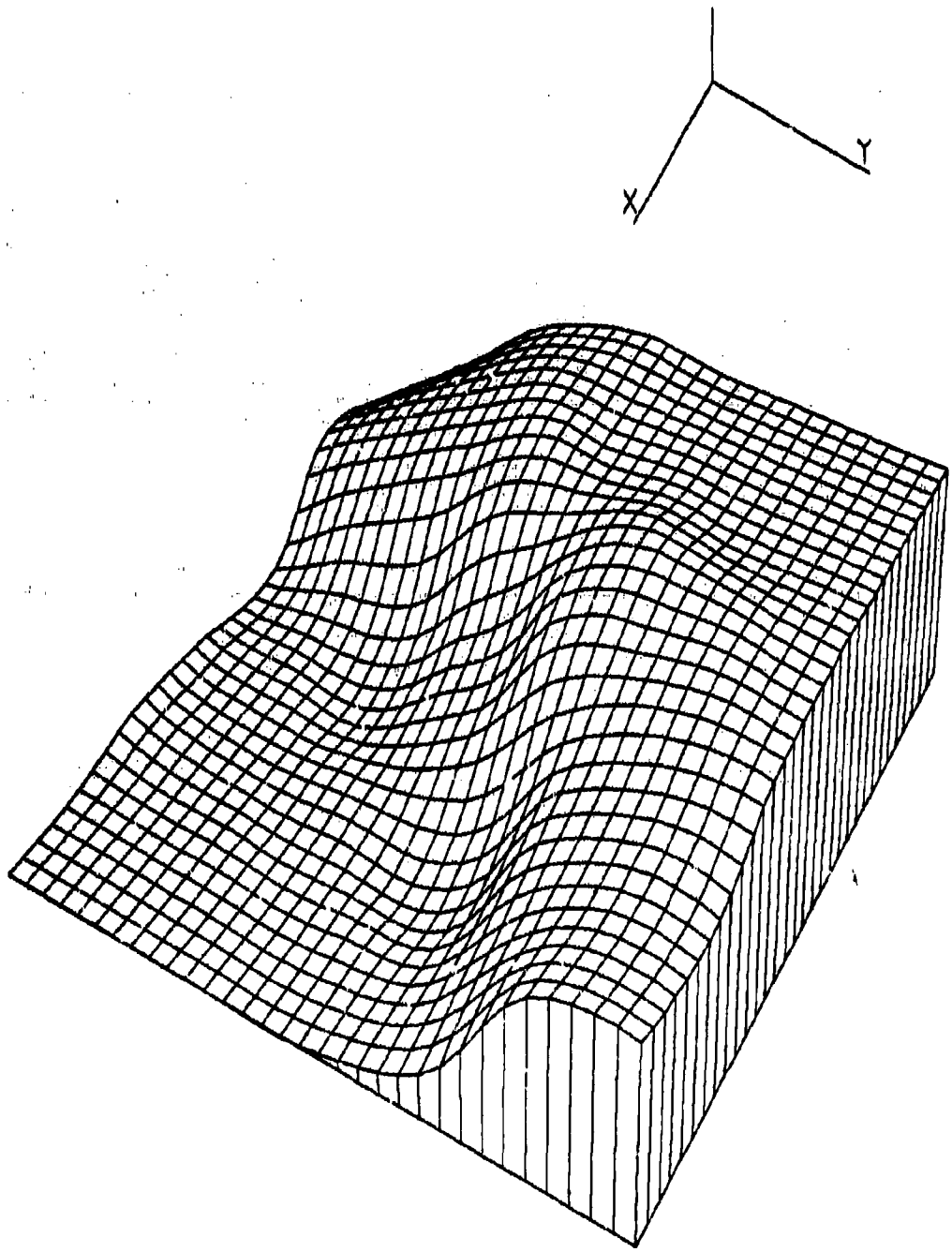
Figure 4.3.2.10



25PT CLIFF ON GREGS 31079 1919

NIELSON-FRANKE QUAD-TRI NPPR = 18 R = 0.586

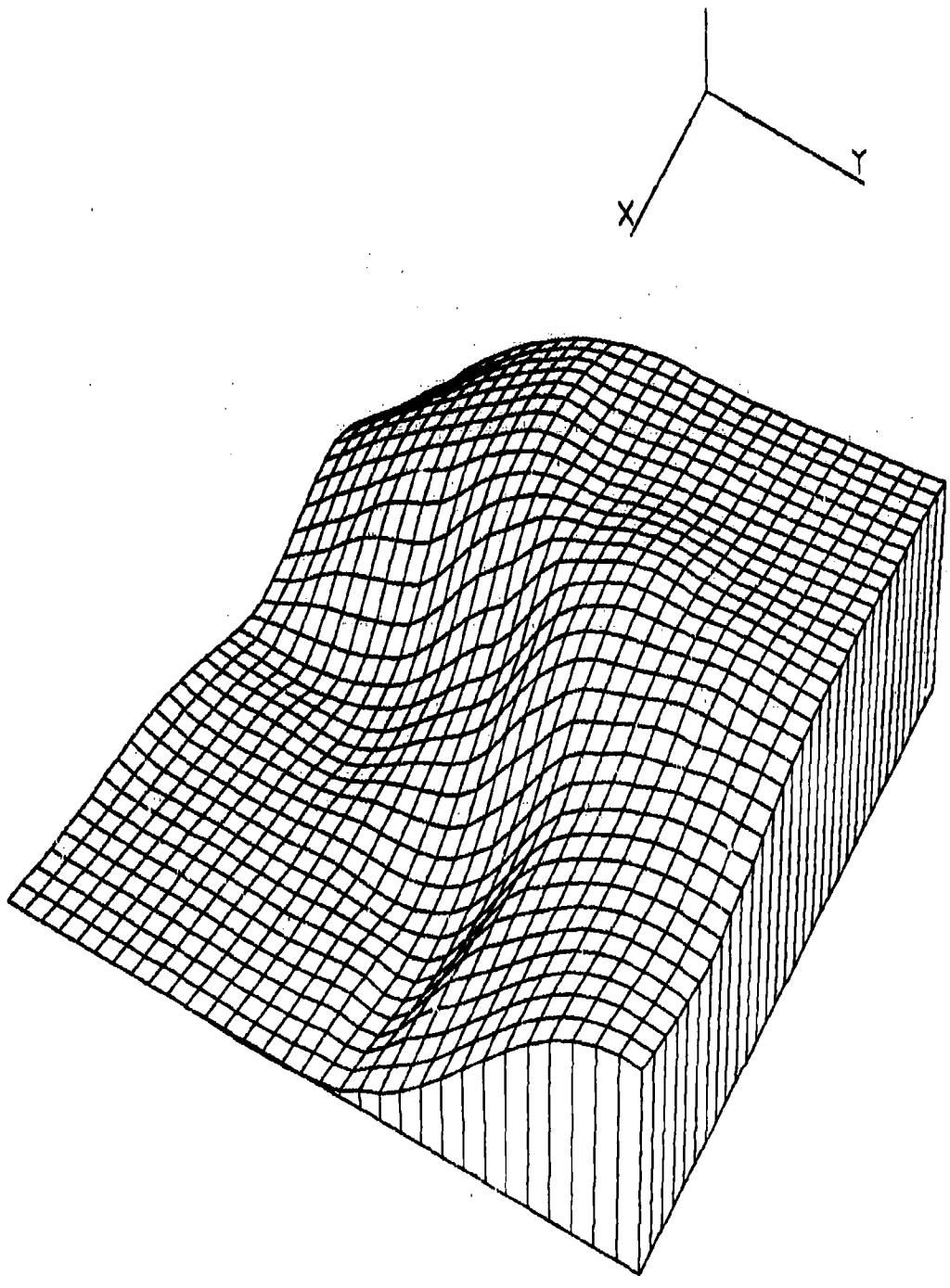
Figure 4.3.2.13



25PT CLIFF ON GREGS 31179 1442

QUADRATIC SHEPARD METHOD NPPR = 918 R = 0.414

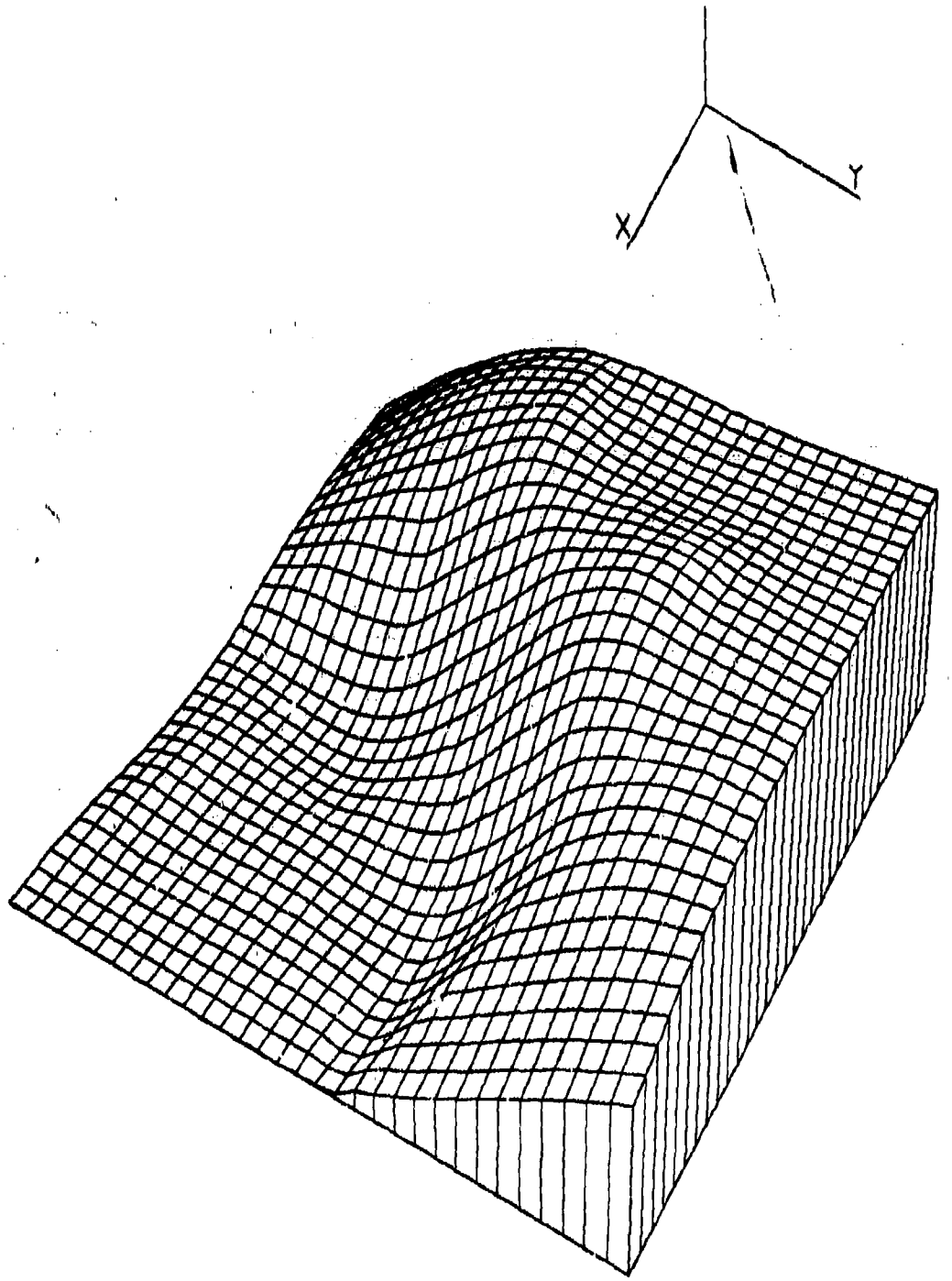
Figure 4.3.2.14



25PT CLIFF ON GREGS 31079 1933

AKIMA - L2 QUAD DERIVS NPPR = 18 R = 0.586

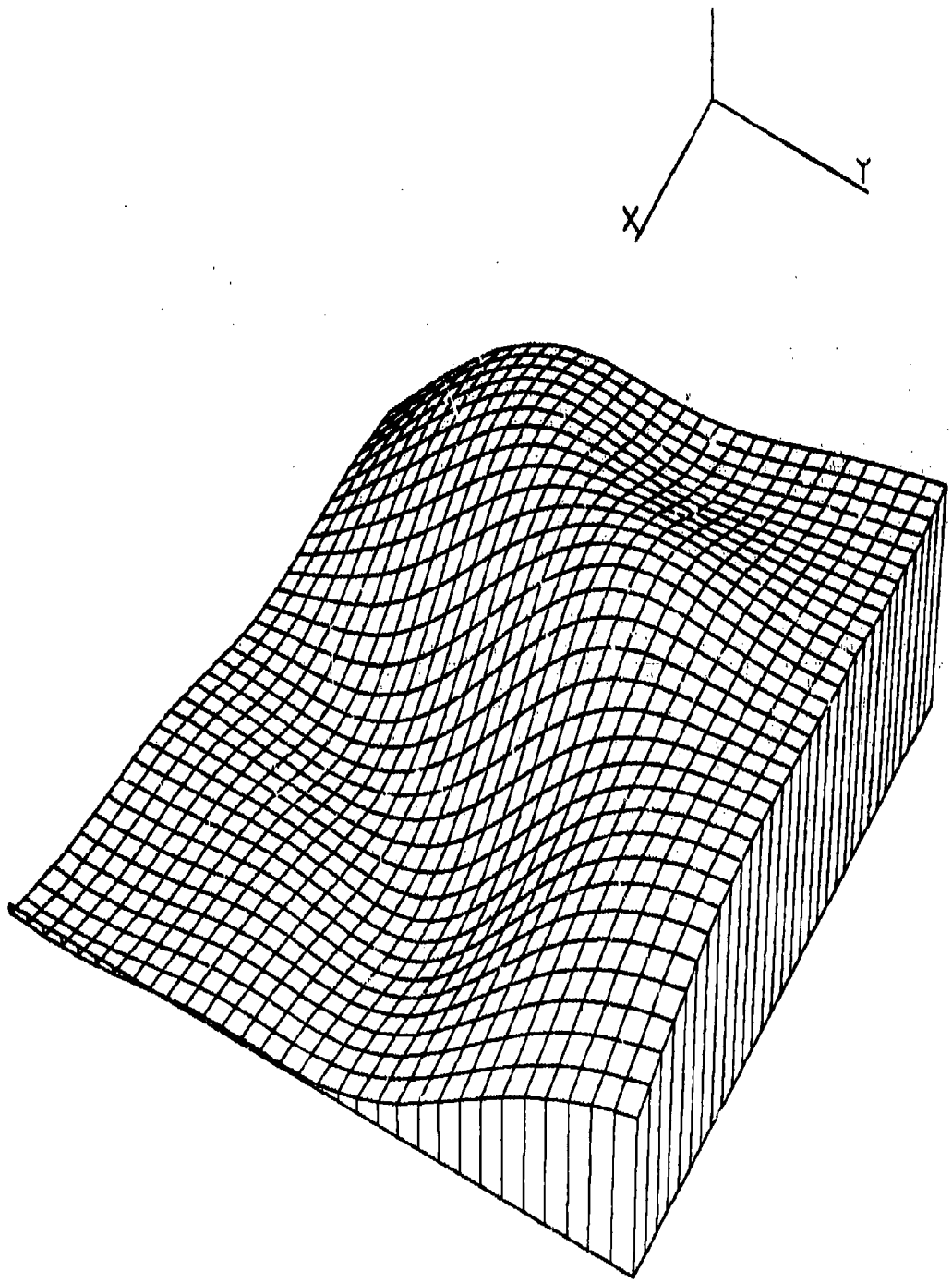
Figure 4.3.2.16



25PT CLIFF ON GREGS 13179 1217

NIELSON MIN NORM NETWORK

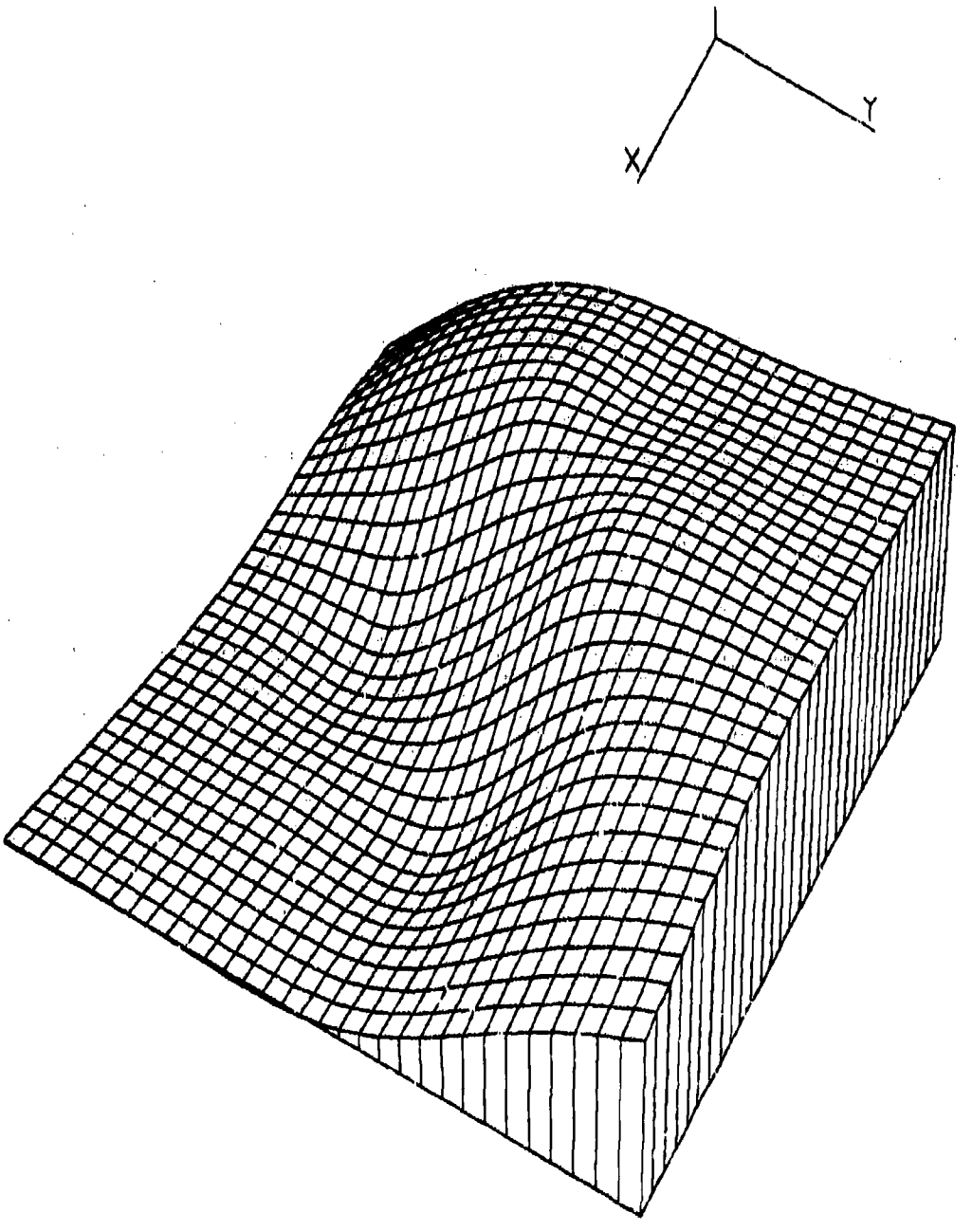
Figure 4.3.2.19



25PT CLIFF ON GREGS 31579 2017

HARDY MQ SURFACE NPPR = 25 R = 0.345

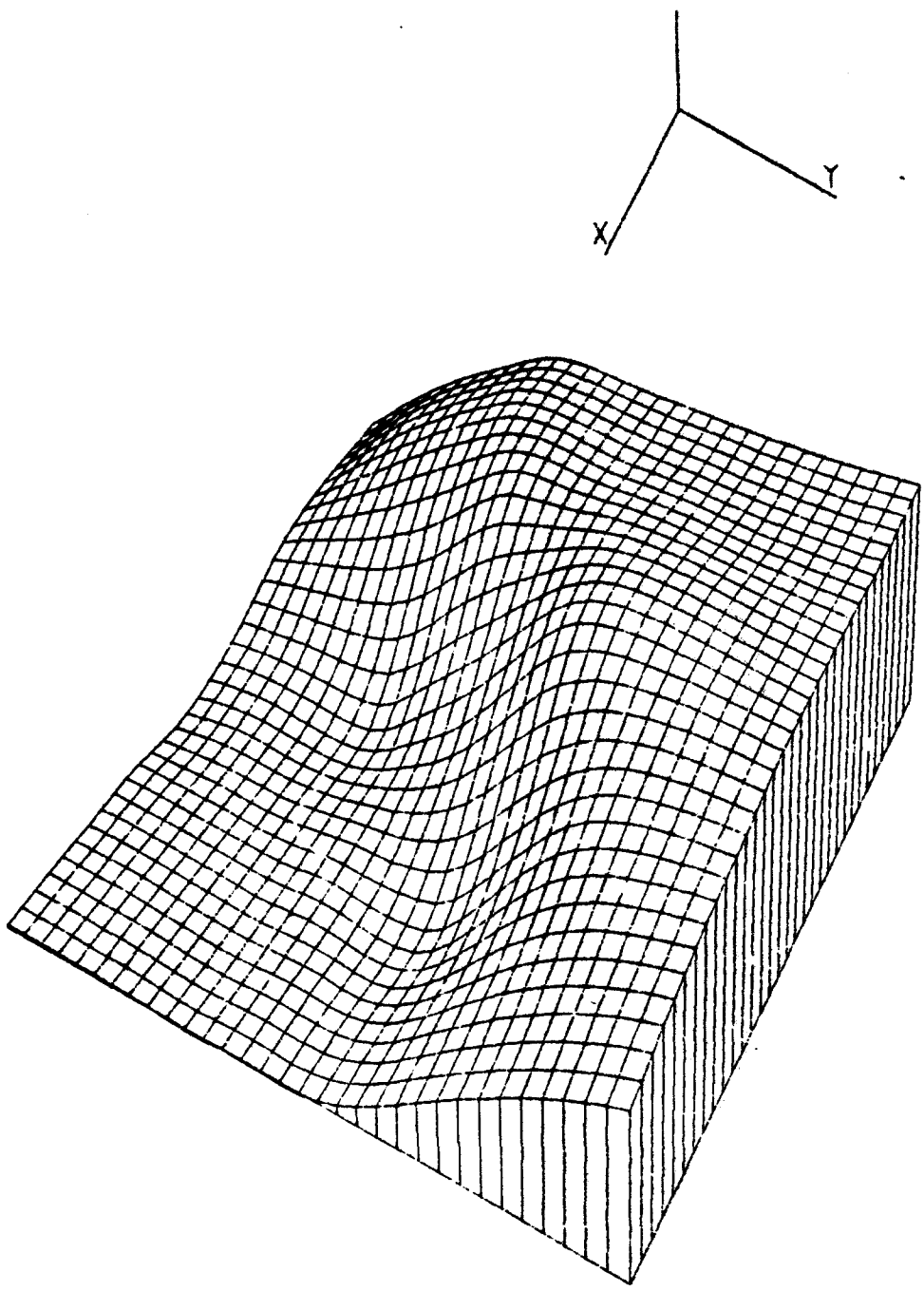
Figure 4.3.2.21



25PT CLIFF ON GREGS 20779 1040

DUCHON S THIN PLATE

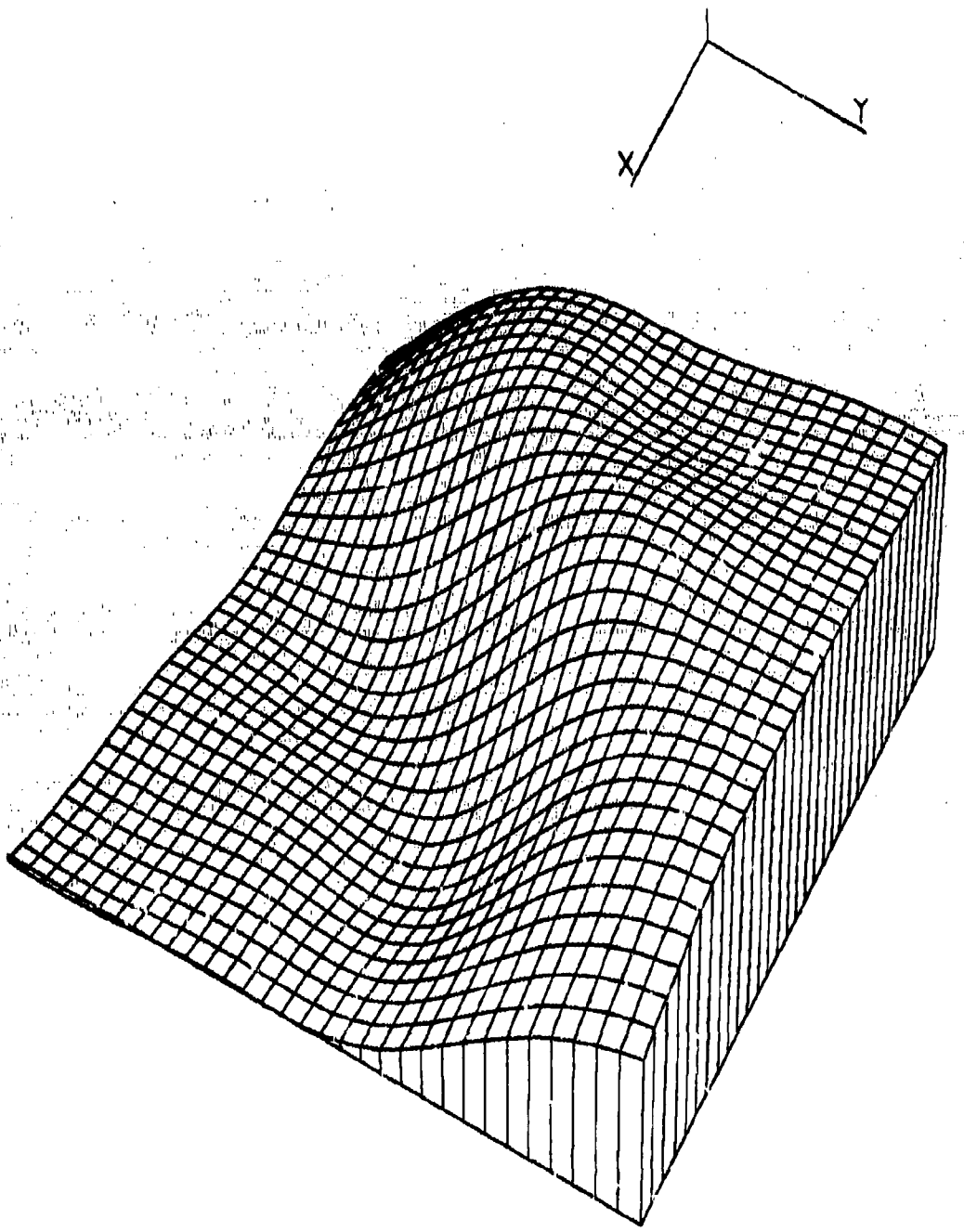
Figure 4.3.2.23



25PT CLIFF ON GREGS 80879 838

FRANKE W/ THIN PLATE NPPR = 6

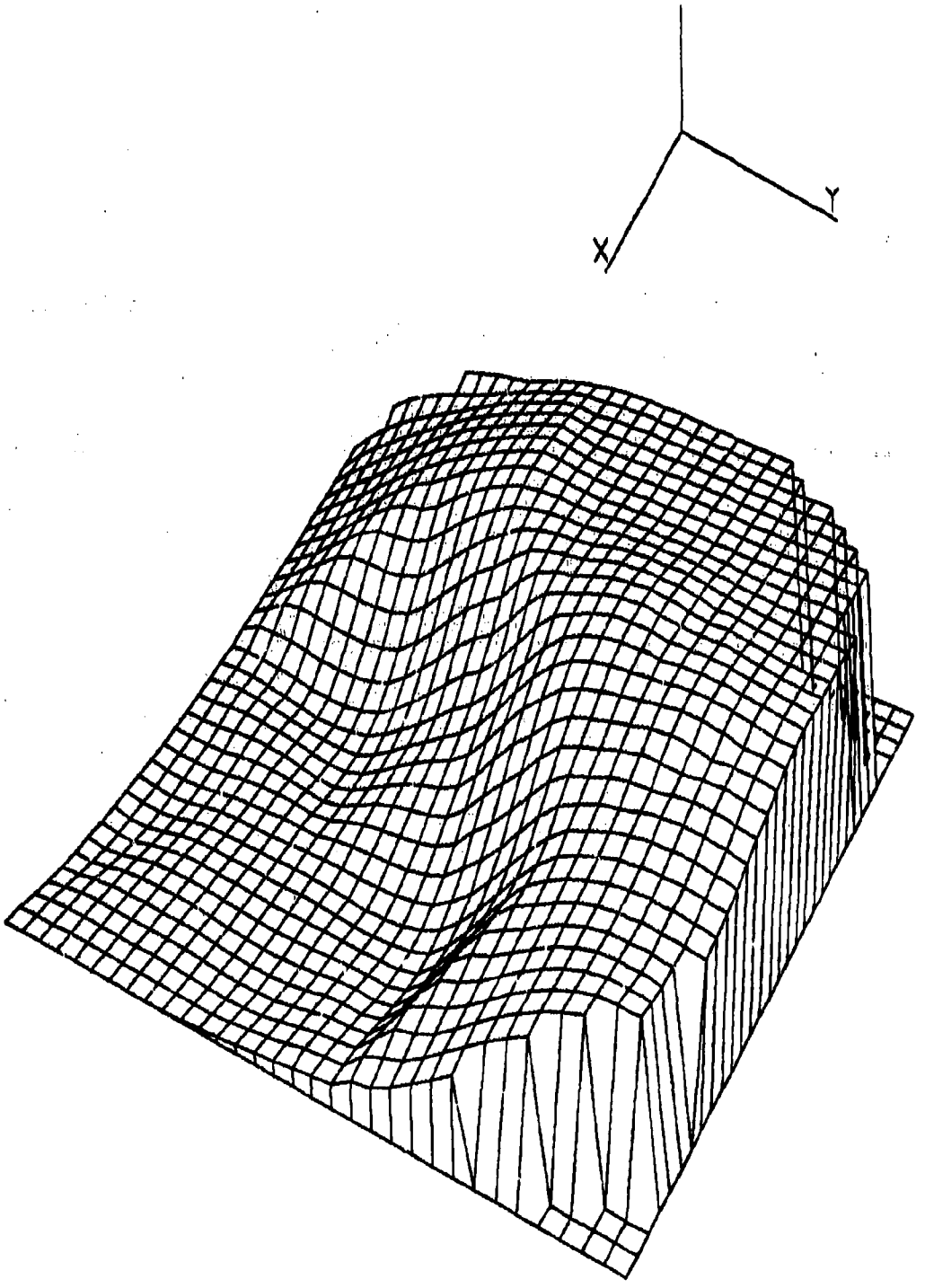
Figure 4.3.2.24



25PT CLIFF ON GREGS 40679 1528

HARDY RECIP MQ SURFACE NPPR = 25 R = 0.345

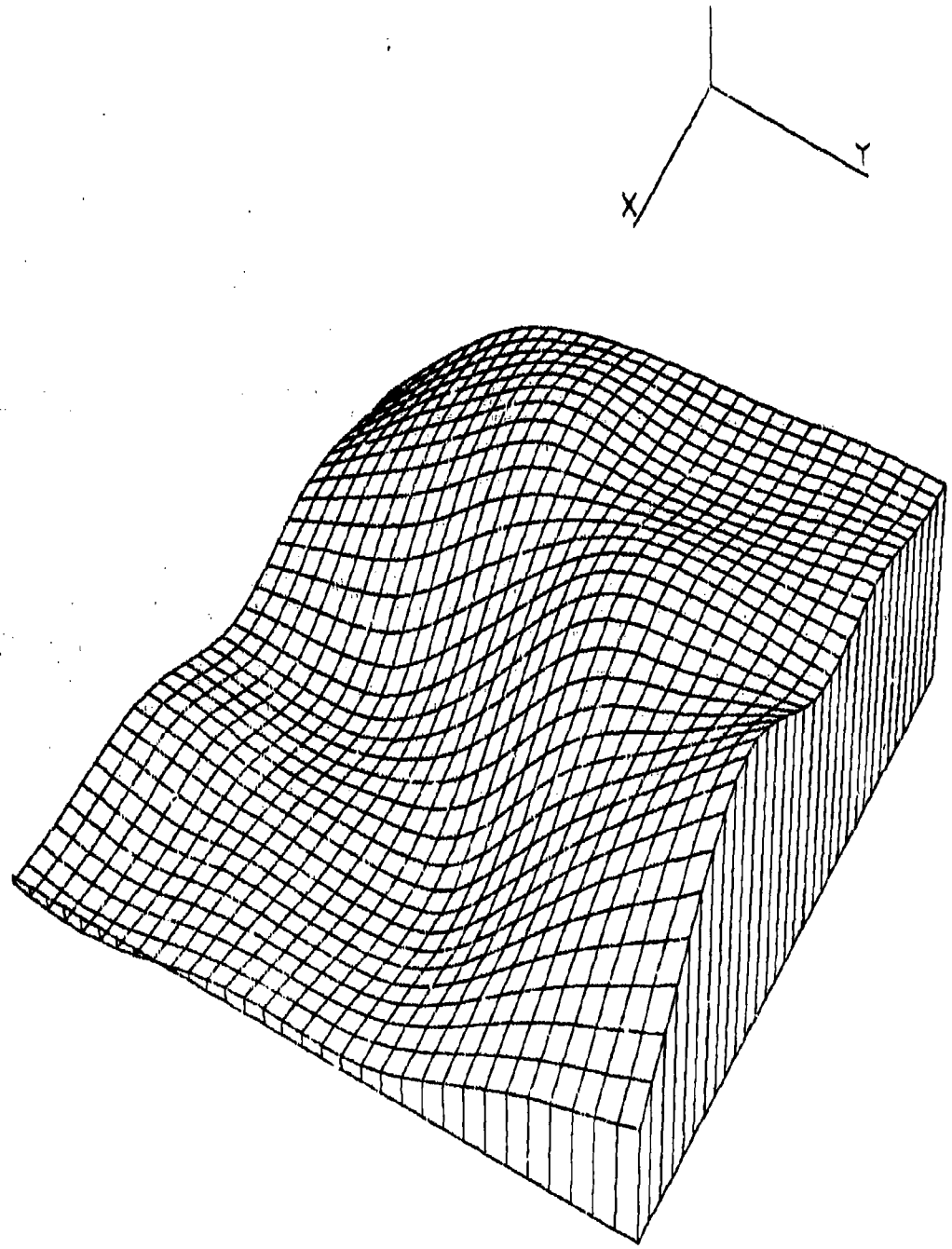
Figure 4.3.2.27



25PT CLIFF ON GREGS 32379 1425

LAWSON TRIANGLE METHOD

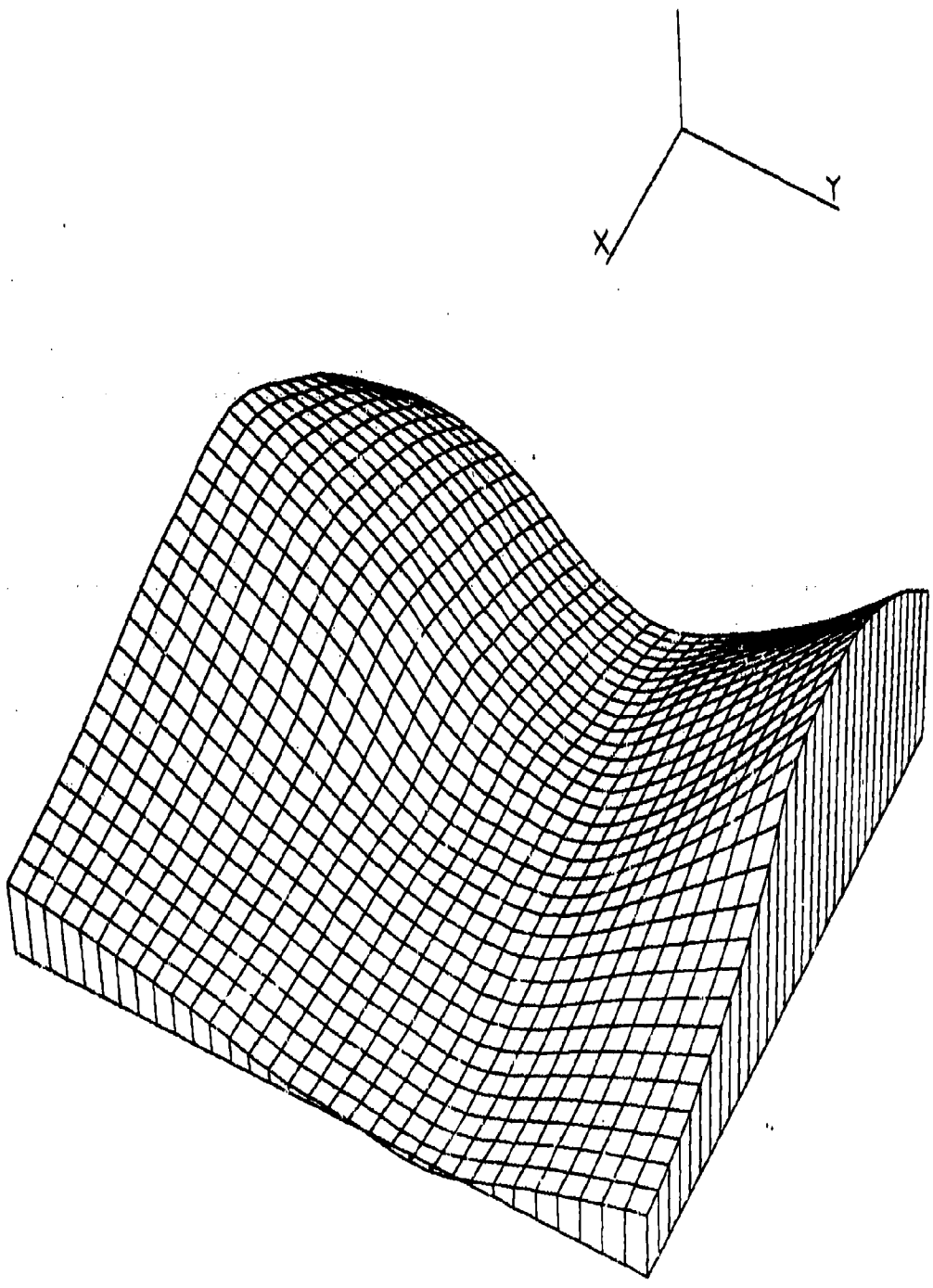
Figure 4.3.2.28



25PT CLIFF ON GREGS 80279 1106

FOLEY ITERATED BICUBICS NPPR = 3

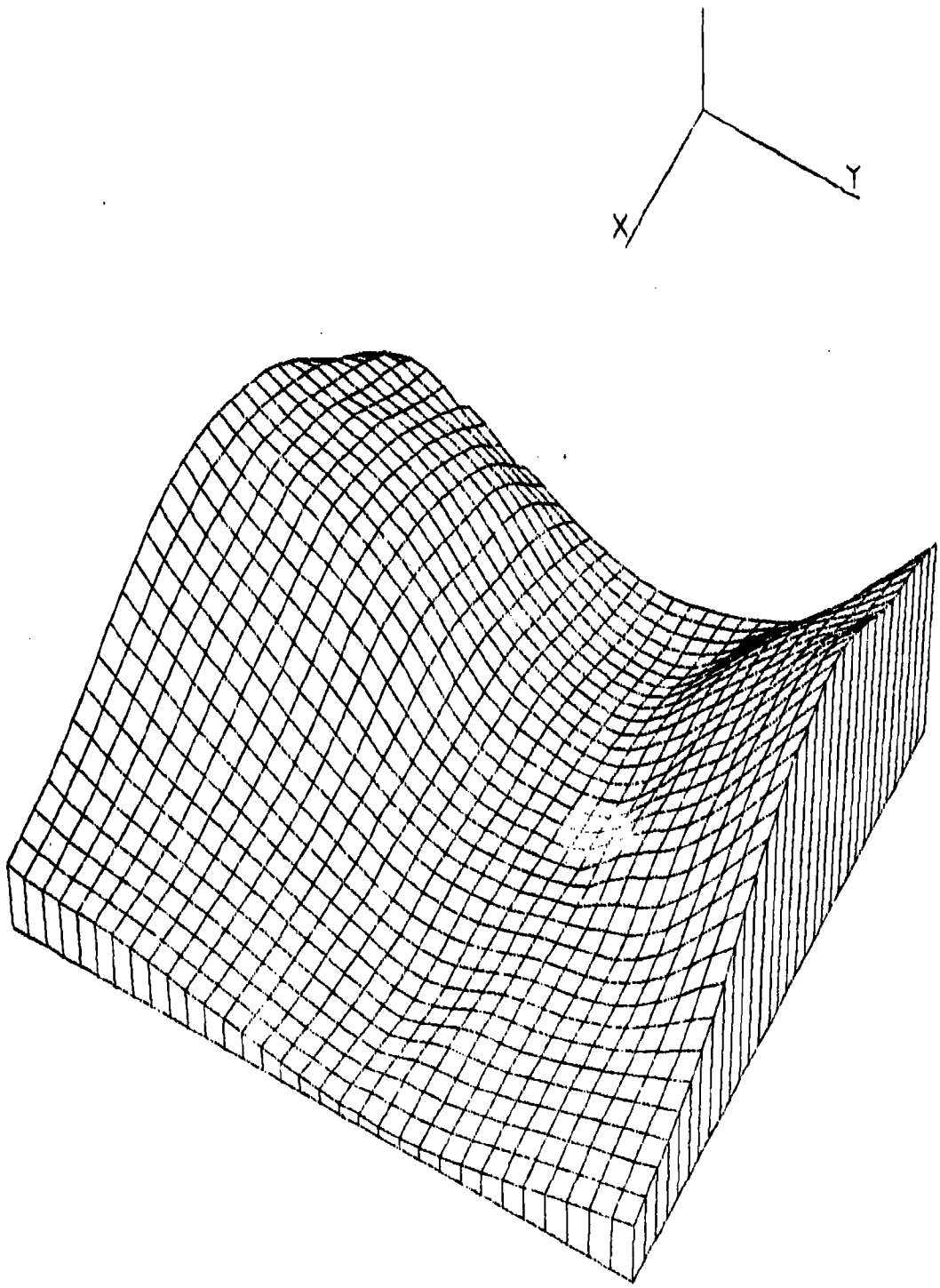
Figure 4.3.2.30



25PT SADDLE ON GREGS 21679 943

FRANKE - SARD BS PLANE NPPR = 6

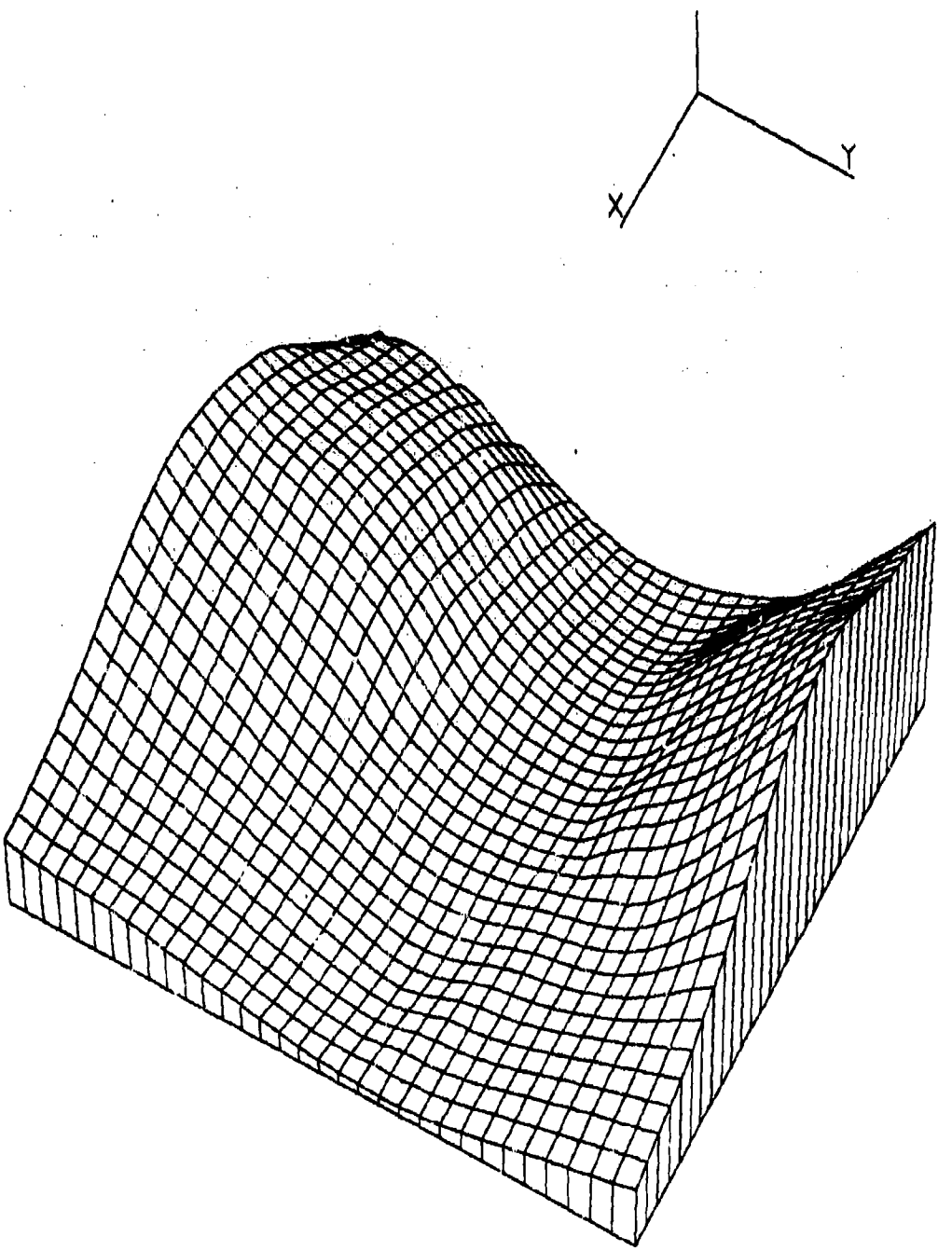
Figure 4.3.3.1



25PT SADDLE ON GREGS 41879 1022

AKIMA - FEM FIT NPPR = 6

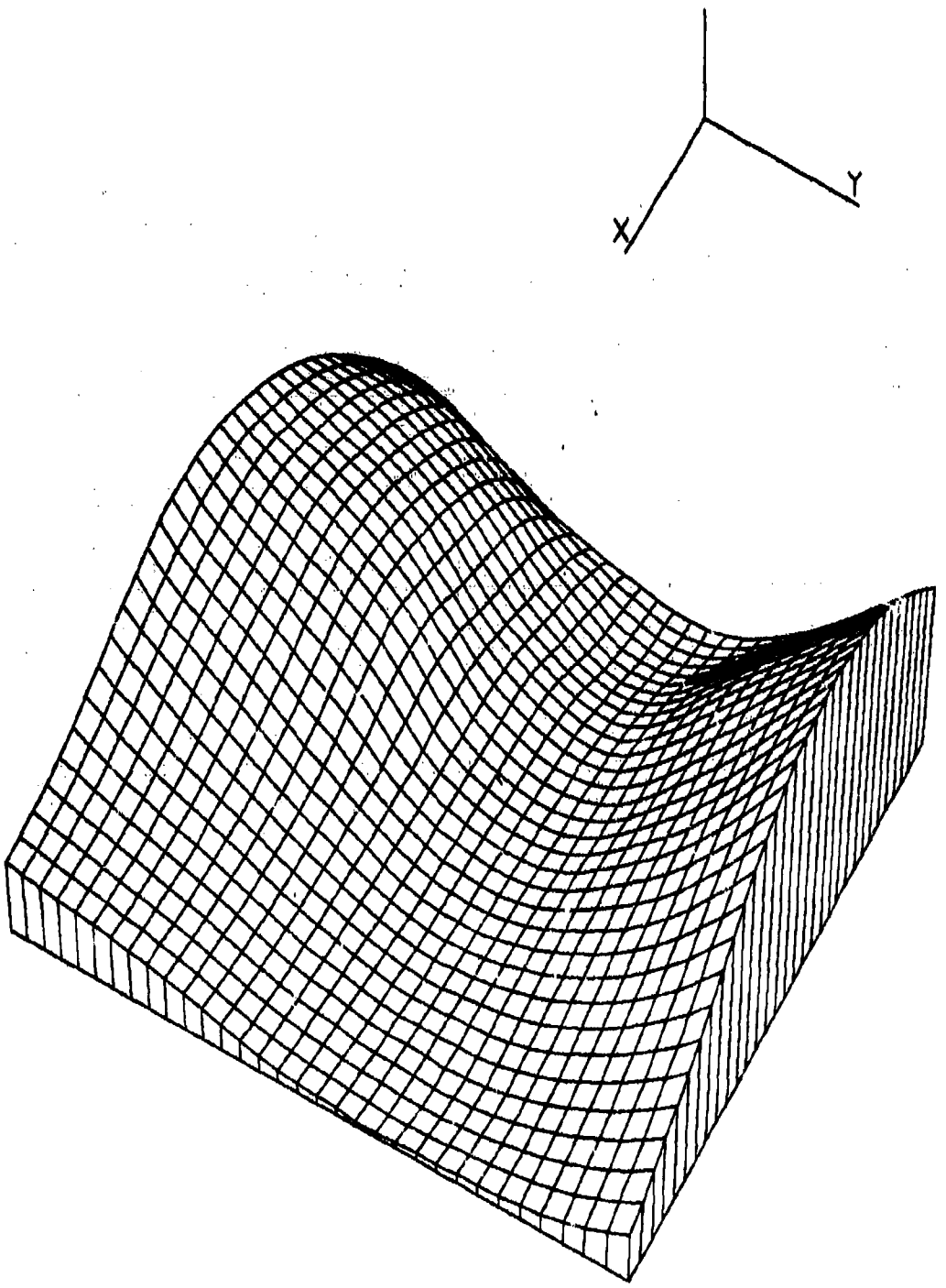
Figure 4.3.3.4



25PT SADDLE ON GREGS 31079 1934

MODIFIED AKIMA FEM FIT NPPR = 6

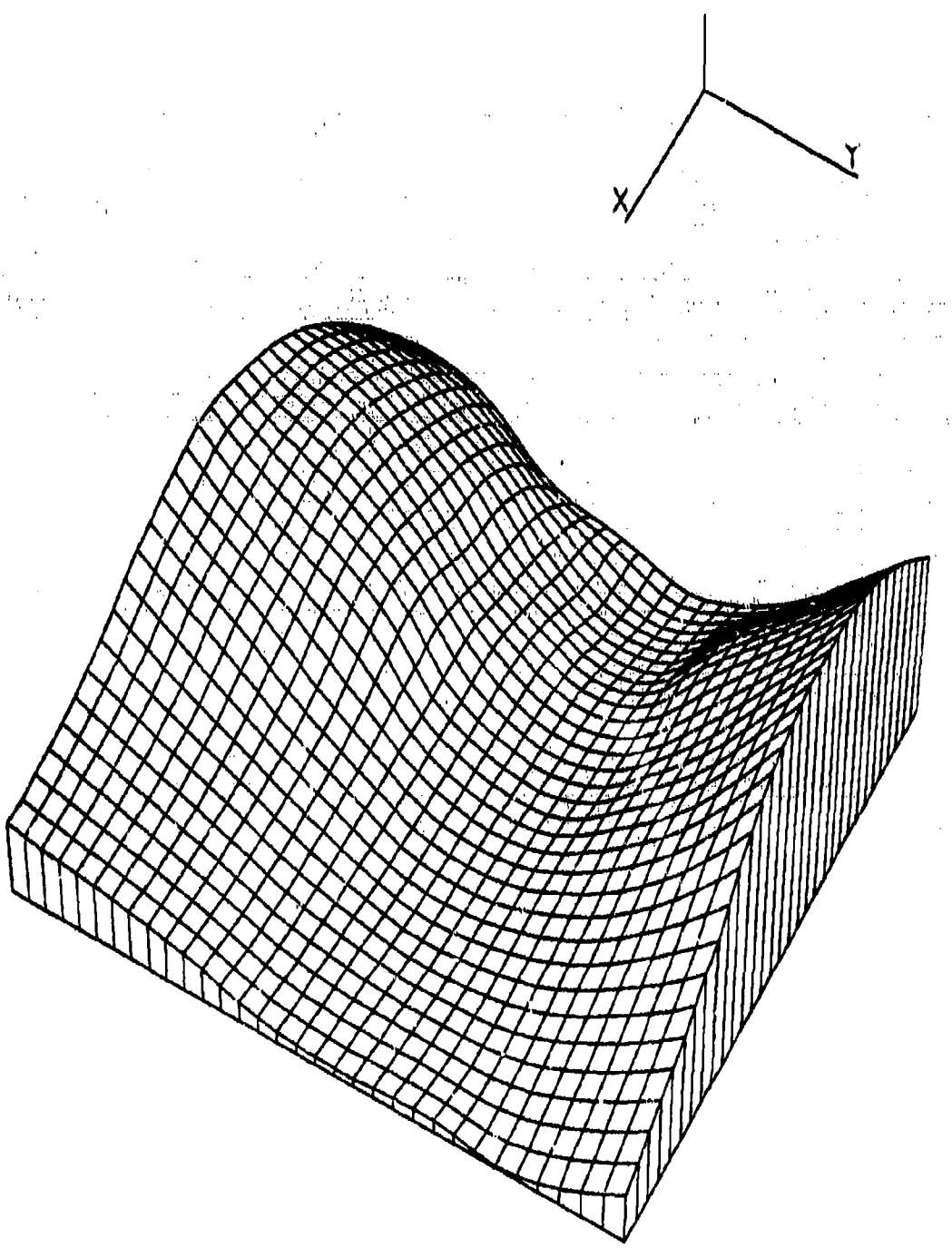
Figure 4.3.3.10



25PT SADDLE ON GREGS 31079 1919

NIELSON-FRANKE QUAD-TRI NPPR = 18 R = 0.586

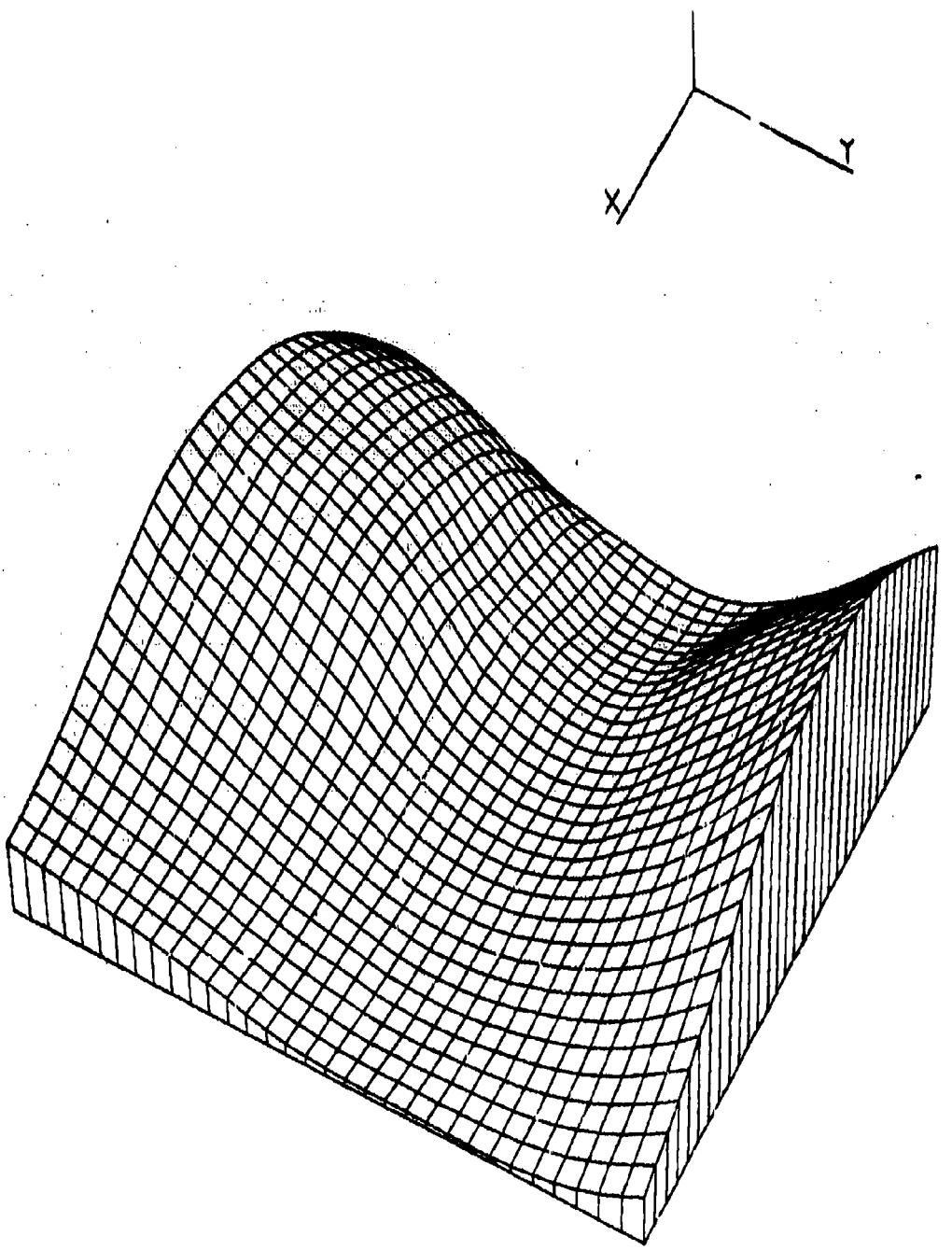
Figure 4.3.3.13



25PT SADDLE ON GREGS 31179 1442

QUADRATIC SHEPARD METHOD NPPR = 918 R = 0.414

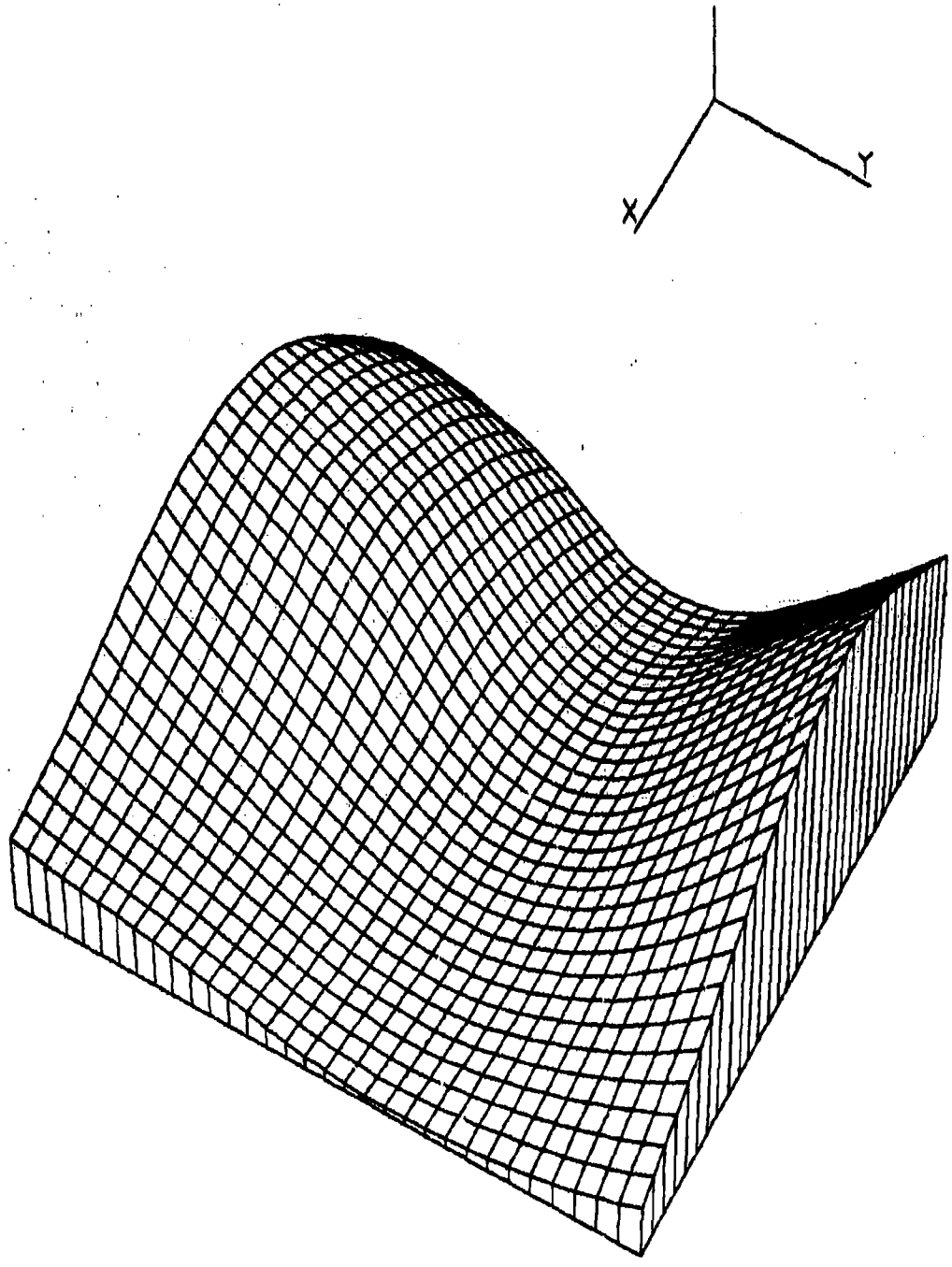
Figure 4.3.3.14



25PT SADDLE ØN GREGS 31079 1933

AKIMA - L2 QUAD DERIVS NPPR = 18 R = 0.586

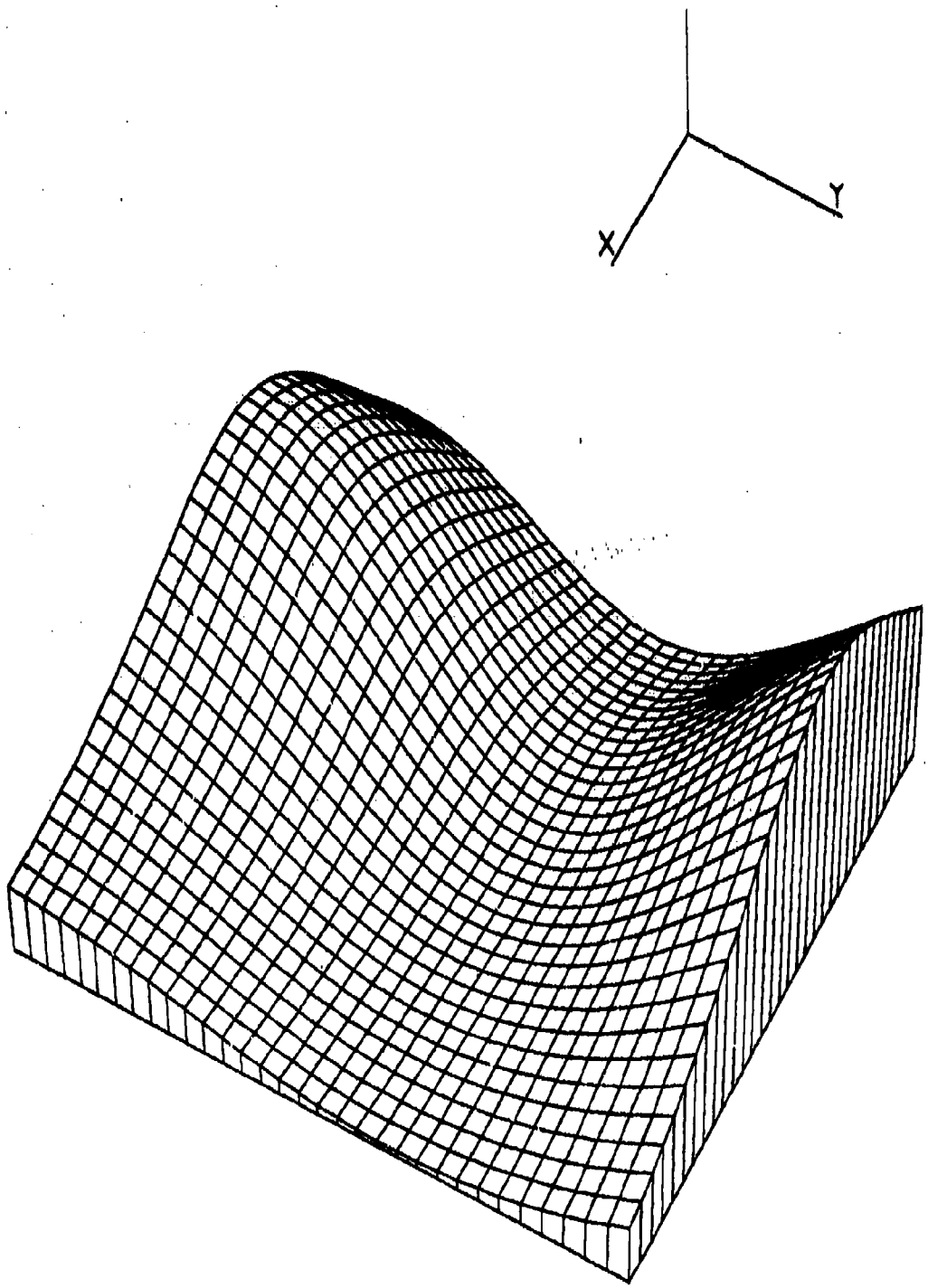
Figure 4.3.3.16



25PT SADDLE ØN GREGS 13179 1217

NIELSØN MIN NØRM NETWØRK

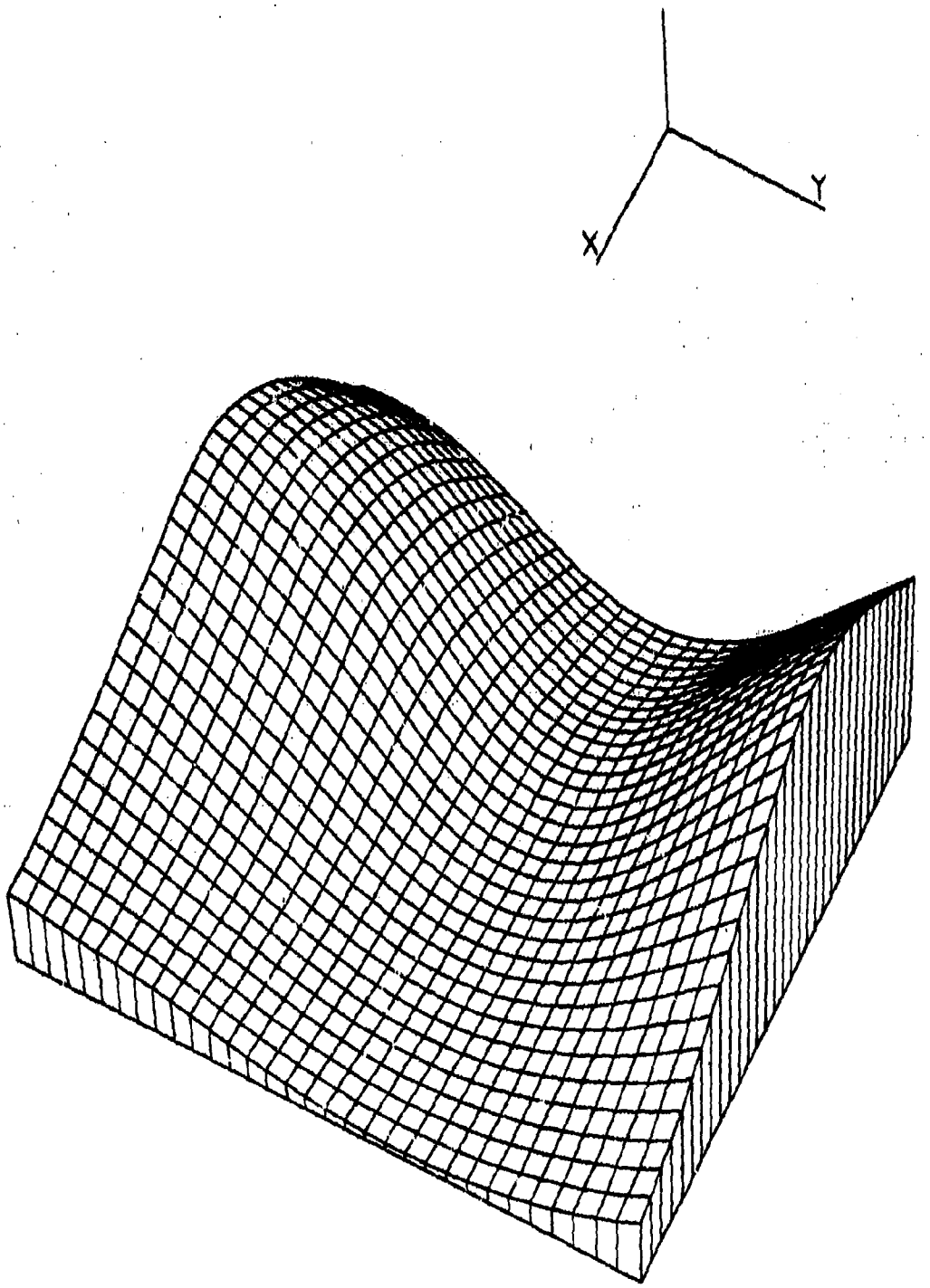
Figure 4.3.3.19



25PT SADDLE ON GREGS 31579 2017

HARDY MQ SURFACE NPPR = 25 R = 0.345

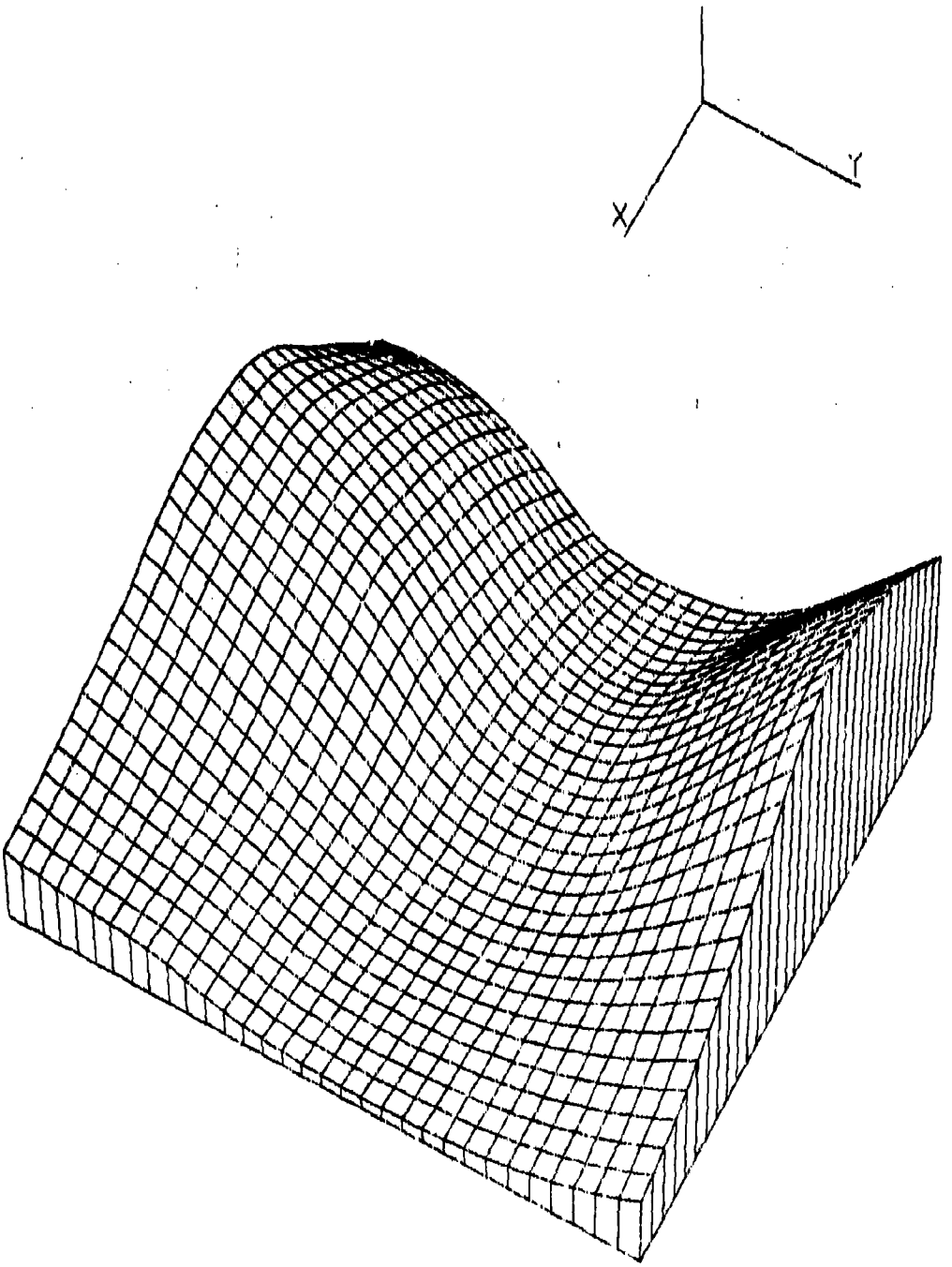
Figure 4.3.3.21



25PT SADDLE ON GREGS 20779 1039

DUCHON S THIN PLATE

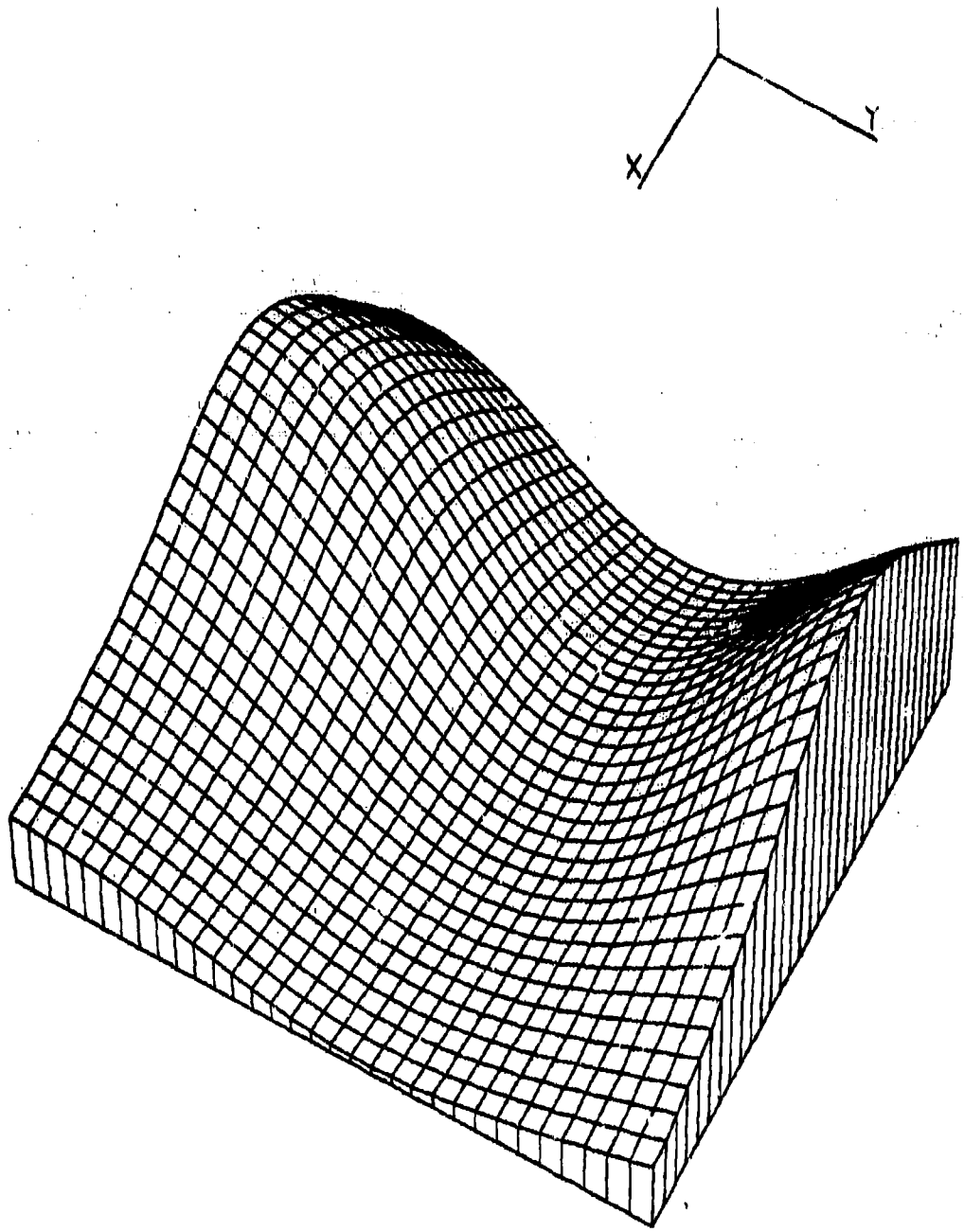
Figure 4.3.3.23



25PT SADDLE ON GREGS 80879 837

FRANKE W/ THIN PLATE NPPR = 6

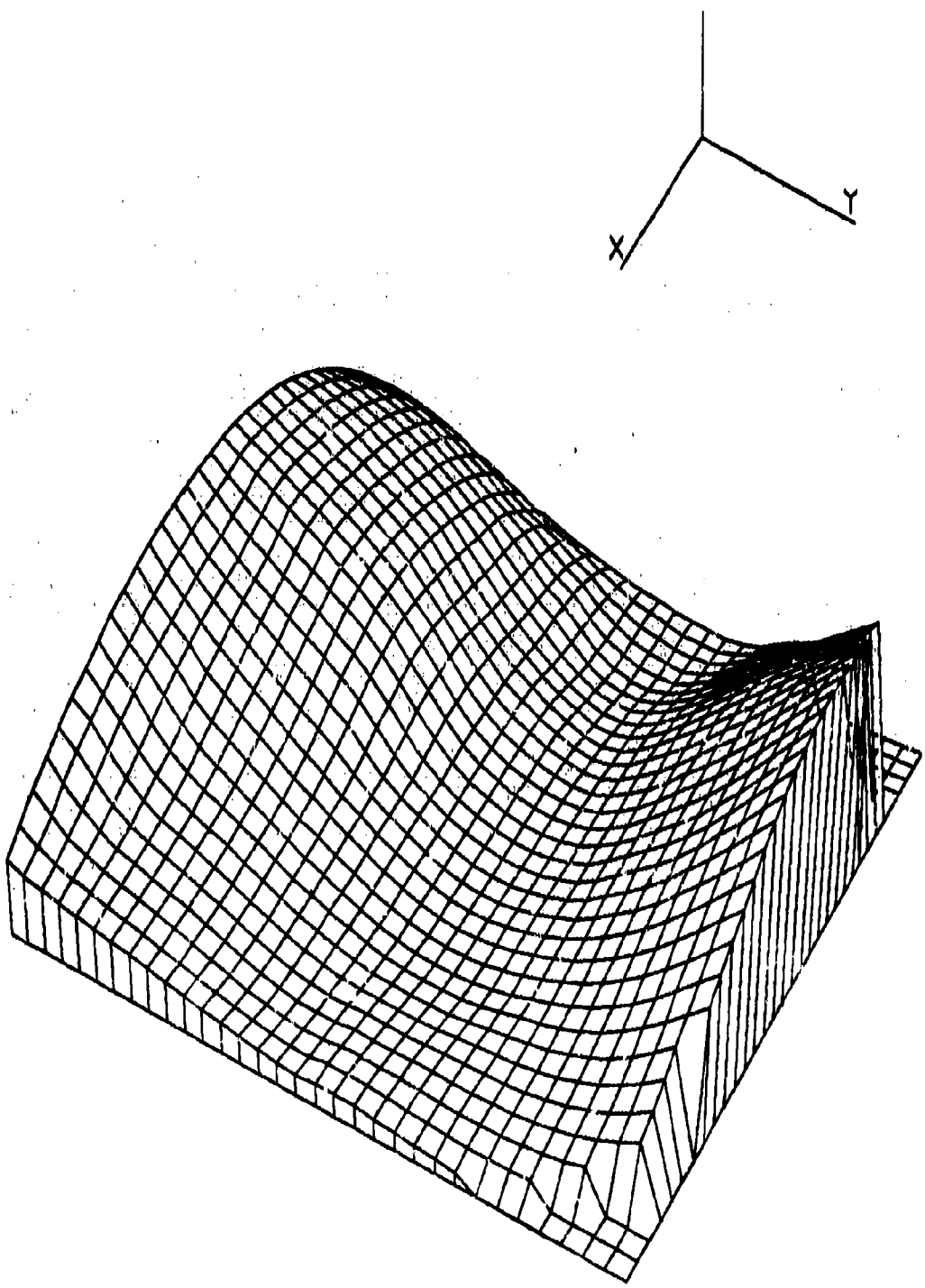
Figure 4.3.3.24



25PT SADDLE ON GREGS 40679 1527

HARDY RECIP MQ SURFACE NPPR = 25 R = 0.345

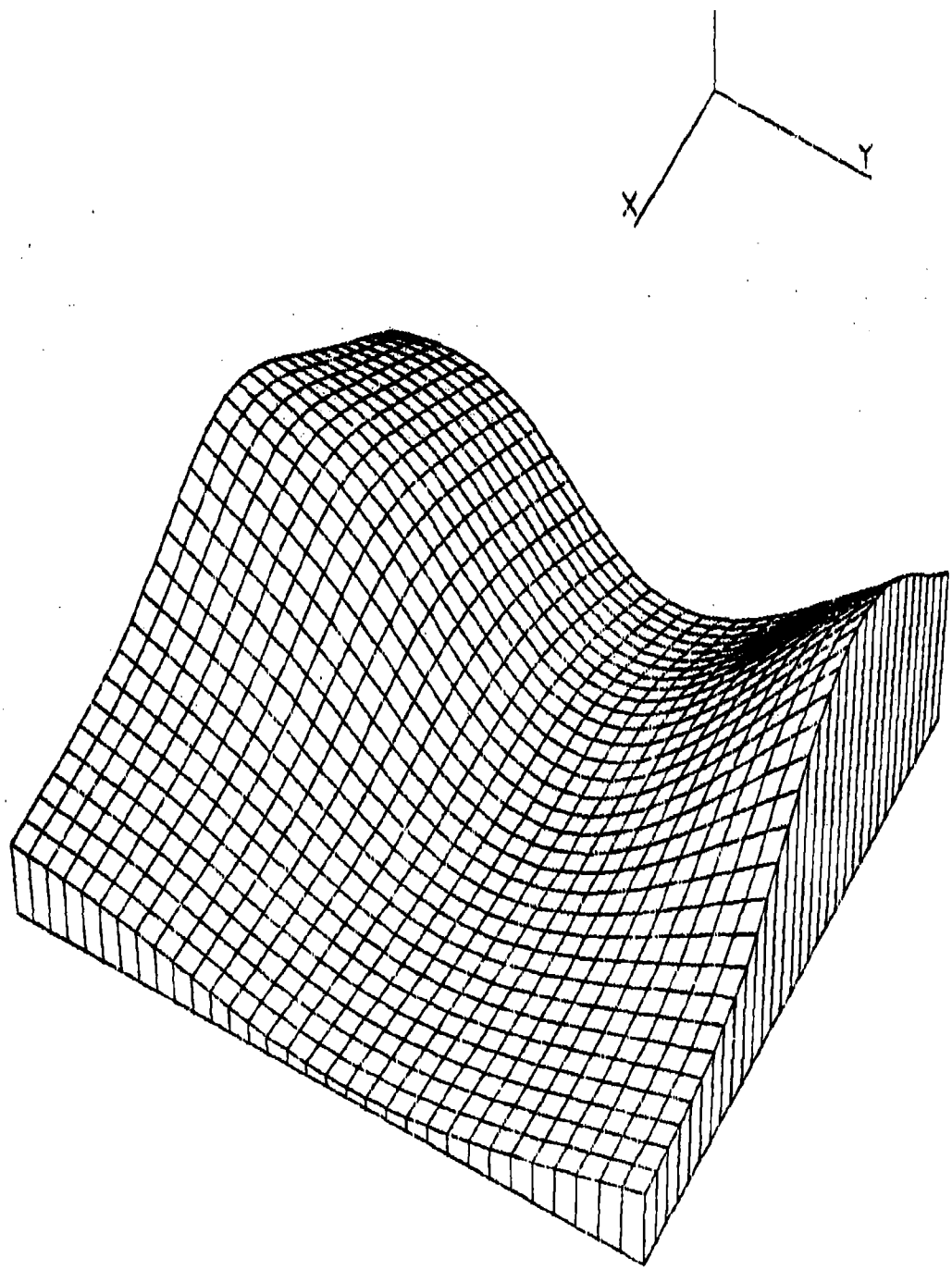
Figure 4.3.3.27



25PT SADDLE ON GREGS 32379 1424

LAWSON TRIANGLE METHOD

Figure 4.3.3.28



25PT SADDLE ØN CREGS 80279 1104

FØLEY ITERATED BICUBICS NPPR = 3

Figure 4.3.3.30

DISTRIBUTION LIST

Defense Documentation Center Cameron Station Alexandria, VA 22314	2
Dudley Knox Library Naval Postgraduate School Monterey, CA 93940	2
Dean of Research Naval Postgraduate School Monterey, CA 93940	2
Department of Mathematics C. O. Wilde, Chairman F. D. Faulkner, Acting Chairman A. L. Schoenstadt R. Franke Naval Postgraduate School Monterey, CA 93940	 1 1 1 15
Dr. Richard Lau Office of Naval Research 1030 East Green St. Pasadena, CA 91106	1
Professor R. E. Barnhill Department of Mathematics University of Utah Salt Lake City, UT 84112	1
Professor G. M. Nielson Department of Mathematics Arizona State University Tempe, AZ 85281	1
Chief of Naval Research ATT: Mathematics Program Arlington, VA 22217	2
Rosemary E. Chang Sandia Laboratories Applied Mathematics, Division 2-8235 Livermore, CA 94550	1
Hiroshi Akima Office of Telecommunications Department of Commerce Boulder, CO 80302	1

PAGE 364

IS

MISSING

IN

ORIGINAL

DOCUMENT

Eric A. Lundstrom
Code 3275
Naval Weapons Center
China Lake, CA 93555 1

John R. Rice
Math Science 428
Purdue University
W. Lafayette, IN 47907 1

Dr. Robert Riffenburgh
Code 18
Naval Ocean System Center
San Diego, CA 92152 1

John K. Munro, Jr.
Bldg. 9104-2
Oak Ridge National Laboratory
P. O. Box Y
Oak Ridge, TN 37830 1

Paul Peterson
Woodward Governor Co.
5001 N. 2nd St.
Rockford, IL 61101 1

John Bobbitt
Continental Oil Co.
P. O. Box 1267
Ponca City, OK 74601 1

Eric Grosse
Computer Science Department
Stanford University
Stanford, CA 94305 1

LCDR Alan J. Pickrell
3233 NE 105th Street
Seattle, WA 98125 1

Dr. Farhad Rajabi
SMC #1717
Naval Postgraduate School
Monterey, CA 93940 1

Visiting Professor O. C. Zienkiewicz
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, CA 93940 1

Arthur R. Paradis
President, Dynamic Graphics, Inc.
2150 Shattuck Avenue
Berkeley, CA 94704 1

Dr. Paul T. Boggs
Army Research Office
P. O. Box 1211
Triangle Park, NC 27709 1

Stavros Busenberg
Department of Mathematics
Harvey Mudd College
Claremont, CA 91711 1

Dr. Theodore Rubin
IBM Watson Research Center
P. O. Box 218
Yorktown Heights, NY 10598 1

Professor J. Ben Rosen
Computer Science Department
University of Minnesota
Minneapolis, MN 55455 1

Andy Schoene
Computer Science Department
General Motors Research Laboratory
Warren, MI 48090 1

Professor L. L. Schumaker
Department of Mathematics
University of Texas at Austin
Austin, TX 78712 1

L. F. Shampine
Numerical Mathematics Div. 5122
Sandia Laboratories
Albuquerque, NM 87185 1

Patrick W. Gaffney
Union Carbide/Nuclear Division
Computer Sciences Division
Box Y
Oak Ridge, TN 37830 1

Dr. Carl de Boor
Mathematics Research Center
University of Wisconsin
Madison, WI 53706 1

Grant Burgess
Mobil Oil Co.
1001 Howard Avenue, Room 937
New Orleans, LA 70113 1

Jean Duchon 1
University of Grenoble
F-38041
Grenoble, FRANCE

J. K. Reid 1
AERE
Building 8-9
Harwell, Didcot, Birkshire, ENGLAND

Professor John A. Roulier 1
Department of Mathematics
North Carolina State University
Raleigh, NC 27607

G. Bolondi 1
AGIP
S. Donato Milanese
Milan, ITALY

D. H. McLain 1
Computing Centre
University of Sheffield
Sheffield, United Kingdom S10 2TN

Neil Stahl 1
U. W. - Fox Valley
Midway Rd.
Menuyha, WI 54952

C. A. Steele, Jr. 1
P. O. Box 45
Magnolia, MA 01930

Kunio Tanabe 1
Applied Math Department
Brookhaven National Laboratory
Upton, NY 11973

Mr. B. Clifford 1
Department of Computer Services
University of Calgary
Calgary, Alberta, CANADA T2N 1N4

James R. Jancaitus 1
USA Topographic Labs
Ft. Belvoir, VA 22060

John L. Junkins 1
Department of Engineering Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

Harry Joseph Feeney III
Naval Ocean Systems Center
San Diego, CA 92152 1

Dave Cooper
AFHRL/ASM
Wright-Patterson AFB, OH 45433 1

Stanley C. Eisenstat
Department of Computer Science
Yale University
10 Hillhouse Avenue
New Haven, CT 06520 1

Professor Grace Wahba
Department of Statistics
University of Wisconsin
Madison, WI 53706 1

Jesse Y. Wang
Applied Mathematics Division
Argonne National Laboratory
9700 So. Cuss Avenue
Argonne, IL 60439 1

Shraya Yosefa
Department of Applied Mathematics and Statistics
State University of New York at Stony Brook
Stony Brook, NY 11794 1

R. E. Funderlic
P. O. Box X, Bldg. 4500N, D224
Oak Ridge, TN 37830 1

A. J. Goldman
Chief, Operations Research Div.
U. S. Department of Commerce
National Bureau of Standards
Washington, D. C. 20234 1

Dr. Gary Herron
Boeing Computer Services
P. O. Box 24346
Seattle, WA 98124 1

Dr. R. Peter Dube
Boeing Computer Services
P. O. Box 24346
Seattle, WA 98124 1

Professor Gerald D. Taylor
Department of Mathematics
Colorado State University
Ft. Collins, CO 80523 1

Prcfessor Tom Foley
Department of Computer Science
California Polytechnic State University
San Luis Obispo, CA 93407 1

William L. Vittitow
Technical Services
Lockheed - California
P. O. Box 551
Burbank, CA 91503 1

M. J. D. Powell
Department of Applied Mathematics
University of Cambridge
Silver Street
Cambridge, ENGLAND 1

Jean Meinguet
Istitute de Mathematique P. et A
Universita de Louvain
Chemin du Cyclatron 2
B-1348 Louvain-la-Nerive
BELGIUM 1

A. R. Forrest
School of Computing Studies
University of East Anglia
Norwich, ENGLAND NR4 7TJ 1

Professor Gordon E. Latta
Department of Mathematics
Naval Postgraduate School
Monterey, CA 93940 1

Professor Garrett Birkhoff
Mathematics Department
One Oxford Street
Cambridge, MA 02138 1

Commanding Officer
Fleet Numerical Oceanographic Center
Naval Postgraduate School
Monterey, CA 93940 1

Officer in Charge
Naval Environmental Research Prediction Facility
Naval Postgraduate School
Monterey, CA 93940 1

J. G. Hayes
Division of Numerical Analysis and Computing
National Physical Laboratory
Teddington, ENGLAND TW11 OLW 1

Dr. David Kahaner
U. S. Department of Commerce
National Bureau of Standards
Scientific Computing Division
Washington, D. C. 20234 1

Donald Beardsley
Petro Lewis Corporation
One Energy Center
P. O. Box 2250
Denver, CO 80201 1

Dr. James G. Smith
Office of Naval Research
Arlington, VA 22217 1

Frank Hagin
Computing Services
University of Denver
Denver, CO 80208 1

Professor Rolland L. Hardy
Department of Civil Engineering
Iowa State University
Ames, IA 50011 1

John M. Karon
Department of Biostatistics
School of Public Health
Trailer #32 306H
University of North Carolina
Chapel Hill, NC 27514 1

R. W. Klopfenstein
RCA Laboratories
Princeton, NJ 08540 1

Gene Golub
Computer Science Department
Stanford University
Standord, CA 94305 1

Professor William J. Gordon
Department of Mathematics
Drexel University
Philadelphia, PA 19104 1

Charles Micchelli
Thomas J. Watson Research Center
Post Office Box 218
Yorktown Heights, NY 10598 1

Michael Minkoff
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439 1

R. K. Rew
NCAR
P. O. Box 3000
Boulder, CO 80307 1

W. Fichtner
Bell Labs
Murray Hill, NJ 07974 1

Peter R. Eiseman
ICASE
MS 132C
NASA Langley Research Center
Hampton, VA 1

Stephen C. Banks
Sun Production Co.
503 N. Central Expressway
Richardson, TX 75080 1

Pablo Barrera
Facultao de Ciencias
Department de Matematicas
U. N. A. M.
Mexico 20, D. F. MEXICO 1

Richard A. Hansen
308 TMCB
Brigham Young University
Provo, UT 1

P. H. Merz
Chevron Research Co.
P. O. Box 1627
Richmond, CA 94802 1

C. Bunch
P. O. Box 1267
Conoco R & D
Ponca City, OK 74601 1

James Lyness
AMD-ANL
Argonne, Il 60439 1

Fred N. Fritsch
Lawrence Livermore Laboratory
P. O. Box 808 (L-300)
Livermore, CA 94500 1

W. J. Schaffers
Dupont Co., Engineering Department
Exp. Station, 304
Wilmington, DE 19898 1

U. Ascher
Department of Computer Science
University of British Columbia
Vancouver, BC CANADA 1

Joe McGrath
KMS Fusion, Inc.
3941 Research Park Dr.
Ann Arbor, MI 48104 1

P. S. Jensen
Lockheed Research 5233/205
3251 Hanover St.
Palo Alto, CA 94304 1

Myron Ginsberg
Computer Science Department
General Motors Research Laboratory
Warren, MI 48090 1

L. Kratz
Mathematics Department
Idaho State University
Pocatello, ID 83209 1

Alan Pierce
Amoco Production Co.
P. O. Box 591
Tulsa, OK 74102 1

A. K. Cline
Department of Computer Science
University of Texas at Austin
Austin, TX 78712 1

W. Roy Wessel
CDC
7995 E. Prentice Ave.
Englewood, CO 80111 1

J. W. Chalmers
HAO/NCAR
Box 3000
Boulder, CO 80307 1

Richard B. Evans
Ocean Data Systems, Inc.
6000 Executive Blvd.
Rockville, MD 20852 1

L. H. Seitelman
Pratt and Whitney Aircraft
400 Main St.
East Hartford, CT 06108 1

John A. Carpenter
Oak Ridge National Laboratories
Bldg. 4500-N
Room E 208
P. O. Box X
Oak Ridge, TN 37830 1

James C. Ferguson
Los Alamos Scientific Laboratory
Los Alamos, NM 87544 1

Professor David Salinas
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, CA 93940 1

Dr. Nira Richter-Dyn
Department of Mathematical Sciences
Tel-Aviv University
Tel-Aviv, ISRAEL 1