SURFACE MODELLING TECHNIQUES

John C. Tipper

SURFACE MODELLING TECHNIQUES

BY

JOHN C. TIPPER*

1979

*Formerly, Research Associate, Geologic Research Section, Kansas Geological Survey; now, Lecturer, Department of Geology, University College Galway, Galway, Ireland.

For certain of the figures in this paper, stereo-pairs have been produced; they are stored in the back pocket. Each stereo-pair is suitable for use with a mirror stereoscope.

Figure No.	Stereo-pair
9(a)	3-left, 3-right
9(b)	4-left, 4-right
9(c)	1-left, 1-right
9(d)	2-left, 2-right
15(a)	7-left, 7-right
15(b)	8-left, 8-right
15(c)	5-left, 5-right
15(d)	6-left, 6-right
21(a)	11-left, 11-right
21(b)	12-left, 12-right
21(c)	9-left, 9-right
21(d)	10-left, 10-right

CONTENTS

ILLUSTRATIONS

INTRODUCTION

Shape Analysis in Geology

In recent years, statistical and mathematical techniques have been increasingly adopted within the geosciences. Geologists have originated few, if any, of the techniques; instead they have borrowed, adapted and used (or misused) them. In earlier papers (Tipper, 1977; Tipper, 1978) I attempted to introduce geologists to a number of additional techniques, chiefly from the field of Computer-Aided Design (CAD), which promise to be valuable in one of geology's most widespread problem areas--the study of complex three-dimensional shapes. This paper provides more comprehensive description and evaluation of these techniques, and attempts to develop a coherent framework to unite them with other techniques of shape analysis with which geologists are already familiar, e.g. trend surface analysis.

Studying the shape of three-dimensional objects and surfaces forms a major part of much geological work, and applications of shape analysis can be found in most branches of geology. Examples include delineation of subsurface mineral deposits, mapping of stratigraphical and structural surfaces, and investigation of the morphology of fossils. Because shape is so general a theme, it is essential that any methods developed for its study should be as efficient as possible. Furthermore, they should be independent of both the origin and geological meaning of the objects concerned.

We may classify methods of shape analysis into two major groups: methods of formal numerical analysis, and methods of mathematical (or statistical) surface representation. In the numerical methods, shapes are considered as points in a multi-dimensional space, the coordinate axes of which represent either variables or combinations of variables measured on the shape. In contrast, surface representation methods define an object by specifying its bounding surfaces, either by surface equations or, less commonly, by using statistical estimation techniques. These surface representation methods do no more than produce mathematical (and necessarily computerized) models of the objects: in consequence I refer to this approach by the general name "Computerized Modelling."

Most shape studies within geology have used the numerical approach. Methods used have included multivariate analysis (Gould, 1967, 1969; Blackith and Reyment, 1971; Brower, 1973; Demirmen, 1973), Fourier analysis (Ehrlich and Weinberg, 1970; Kaesler and Waters, 1972; Christopher and Waters, 1974; Delmet and Anstey, 1974; Gevirtz, 1976; Waters, 1977), and *ad hoc* descriptive methods such as have been developed for terrain analysis (Greysukh, 1966; Speight, 1968; Demirmen, 1973; Prelat, 1974). The great attraction of numerical methods is their operational simplicity, and this advantage is only lost on complex objects. (Here we consider an object to be complex if it has no obvious regularity of shape. Although a simple process may create a complex shape, we are interested more in the final shape than in the process by which it arose.) In practice the surface representation approach is the more efficient for objects with relatively complex curved surfaces, because it is much easier to specify surface equations than to define complex objects by such secondary measures as diameters or point-to-point distances. As the objects with which we shall be concerned in this

paper are quite complex in form, only the surface representation approach is considered further here.

A Geological Problem in Surface Representation

In describing the surface representation techniques, we shall attempt to indicate ways in which they can be compared one with another, both in theory and practice. The theoretical comparisons will be facilitated by maintaining consistent notation (Appendix 1), the practical comparisons by introducing an actual geological data set to which many of the methods will be applied. Before proceeding further we introduce this data set; in the rest of this study it is referred to as the Cherokee data set.

In southeastern Kansas, oil is produced from sandstones of the middle Pennsylvanian Cherokee Group. The reservoir sands form a complex subsurface meshwork; individual sands are impersistent laterally, "migrate" vertically, and frequently merge with one another. The sands in some areas are thought to have been laid down as channel deposits (McQuillan, 1968; Van Dyke, 1975), in other areas as offshore bars (Bass, 1936). Because of the history of its development, the oilfield studied (Fig. 1) provides an unusual volume of high-quality subsurface data. Each well has an associated gamma-ray log and, in most cases, has been cored through the producing interval.

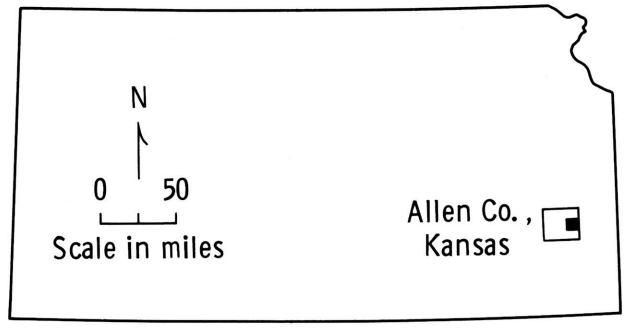


FIGURE 1. Location map showing area from which the Cherokee data set was obtained. Study area in Allen County, Kansas, is shaded.

The actual data were obtained in the following manner. A set of 130 wells was selected, spaced on average at 100-metre intervals, and arranged on a (13-by-10) mesh (Fig. 2). The only restriction placed on the shape of the mesh cells was that they should be convex in plan view (cf. Hessing, *et al.*, 1972). The top and base of each of four sands were picked from the gamma-ray log of each well, verified against the core descriptions where available, and a tentative correlation made for the entire well network. Where individual sand bodies could not be distinguished, as for instance when two superimposed sands had coalesced, arbitrary dividing horizons were chosen. In practice this introduces no problems provided that consistency is maintained. For simplicity in this present work only the two surfaces bounding one sand were retained. Thus the raw data consist of two sets of 130 height values (z) above sea-level datum, each value associated with position coordinates (x, y). These are given in Appendix 2.

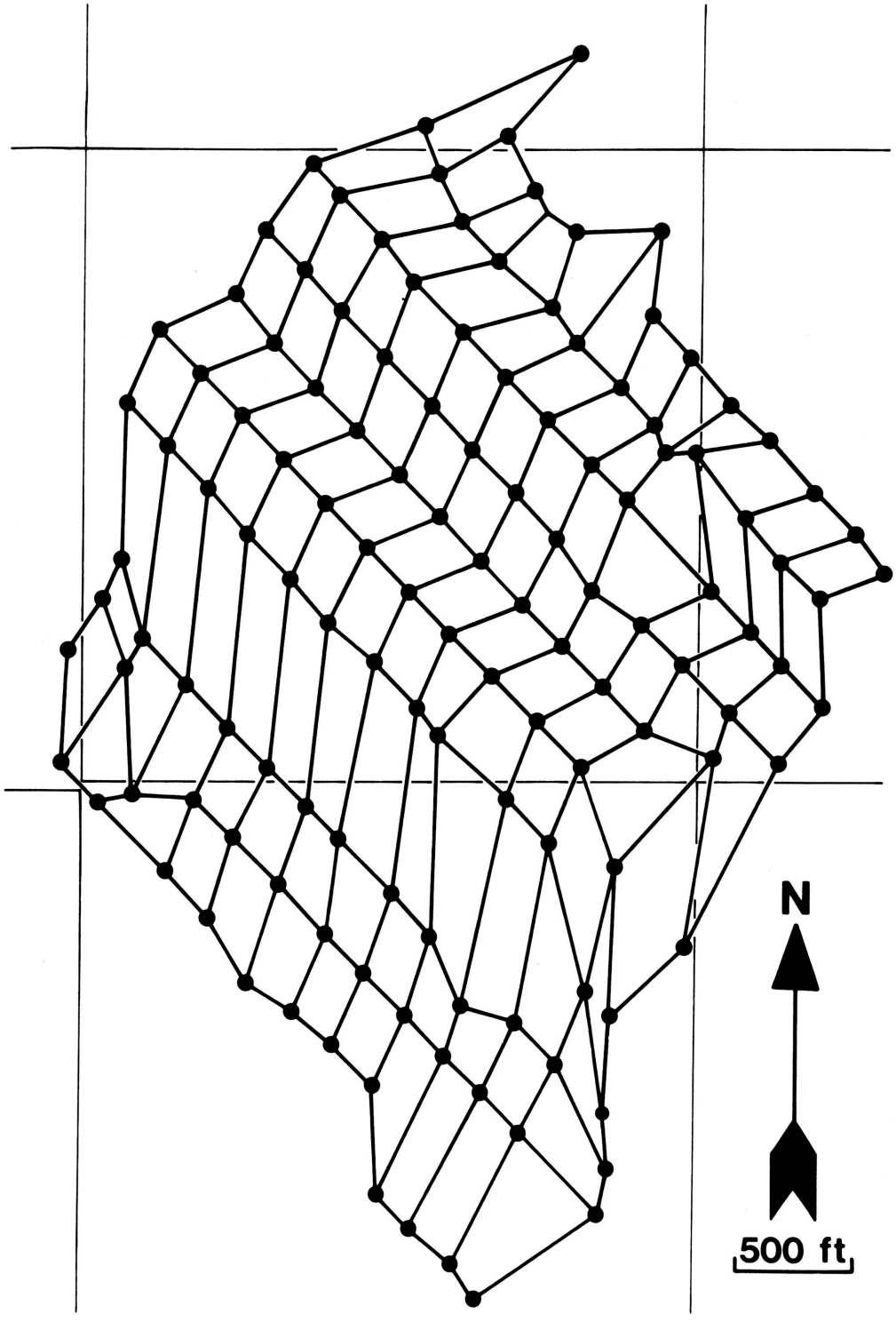


FIGURE 2. Mesh of wells superimposed on 1/2-mile section grid. Each of the 108 cells in the (13-by-10) mesh is convex in plan view.

The reason for selecting this data set to illustrate a paper on shape analysis is that the three-dimensional form of the sands is of importance as a guide to additional drilling for tertiary recovery purposes. In the study area chosen, the subsurface is sufficiently complex that each sand unit must be studied independently. The problem is thus to develop ways to represent mathematically the bounding surfaces of each individual sand, and then to use these representations as bases for subsequent geological work.

Of the two steps involved here, the first is one of modelling, the second one of manipulating the models. In the remaining parts of this paper each step is considered in turn. Neither the treatment of the modelling process nor that of the manipulations which follow it pretends to be exhaustive. The methods described are a broad cross-section of those currently in use. References are given to the original sources (where known), and to important subsequent developments. So fast, however, is the field developing that any paper of this type must inevitably be incomplete. Realizing this, I have tried for balance at the expense of completeness.

PART I -- METHODS OF COMPUTERIZED MODELLING

The General Problem of Surface Representation

The problem with which we are concerned may be generalized in the following way: *Given the rectangular Cartesian coordinates of each of a set of points assumed to lie on one simply connected surface, describe that surface in such a way that it can be manipulated simply and precisely.*

However we choose to describe the surface, we are ultimately involved in defining an interpolating function which will predict the location of points on the surface away from the data points. Argument about how a surface is to be represented thus resolves into argument about the form of this function: this is determined not only by the nature of the surface itself, but also by the purpose for which the description is being made.

The first part of this paper is devoted to systematic description of surface representation techniques (Fig. 3). Consideration is given first to the approach with which geologists will be most familiar, the methods which have together become known as trend surface analysis. Both main techniques are outlined--surface fitting methods (including the method of least-squares), and point-by-point estimation using Regionalized Variable Theory. The idea of using blending functions is then introduced, and a method from geodesy used for illustration (the surface averaging method of Junkins, Miller, and Jancaitis).

By studying in this way the performance of methods already in use in geology, we are able to define more clearly the requirements which a surface representation method must satisfy in order to be suited to our problem. Surface representation methods are shown to be no more

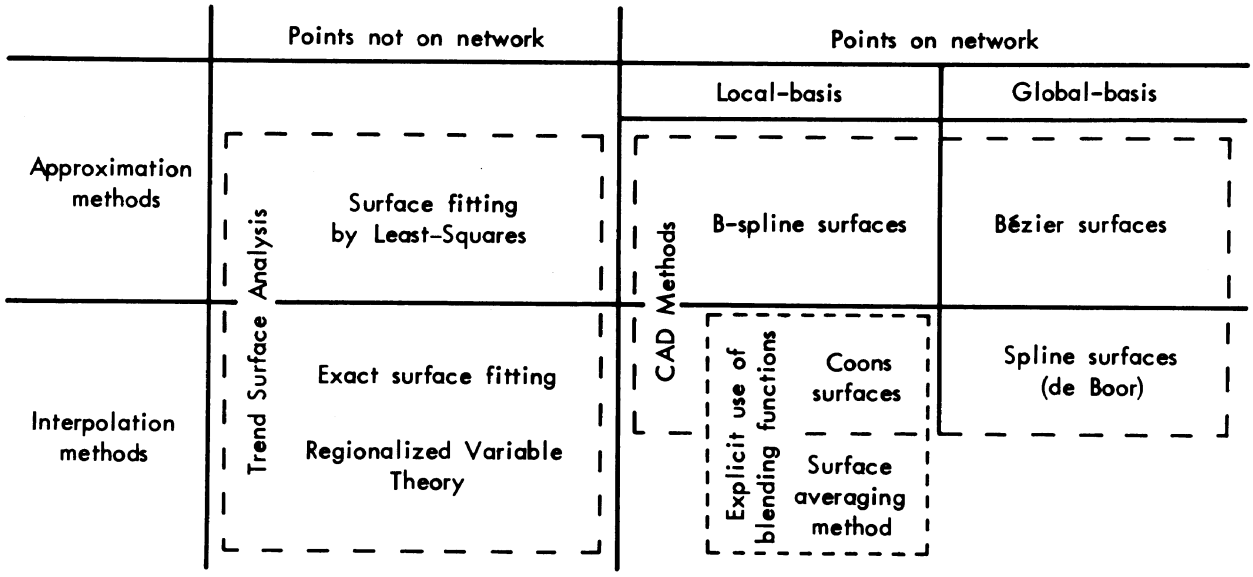


FIGURE 3. Characteristics and interrelationships of the surface modelling techniques described in this paper.

than conventional bivariate interpolation or approximation methods, and the CAD methods are used in illustration. Two approximation methods are described first--Bézier and B-spline curves and surfaces. Interpolation methods are then introduced--bicubic splines and Coons surfaces.

Trend Surface Analysis

In the two decades of its use in geology, trend surface analysis has generated both an extensive literature and considerable controversy. The literature is best approached through reviews such as Harbaugh and Merriam (1968) and Whitten (1975). The controversy has centered on the absolute and relative merits, both theoretical and practical, of the surface fitting and regionalized variable techniques. Heat output from this controversy has generally exceeded that of light [see, for example, Whitten (in Krige, 1966); Matheron, 1967; Watson, 1972; Delfiner and Delhomme, 1975], and it is not the purpose of this paper to stoke the fires further. What is more important is to see how each of the techniques might be applied in the solution of our surface representation problem.

1. Surface fitting methods -- Introduction

Perhaps the most obvious way of describing a surface mathematically is to use some equation of the form

$$z = f(x, y) \qquad .....[1]$$

where z is the height of the surface at a point having base-plane coordinates (x, y). The parameters of the function f are determined by solving the equation for known (x, y, z) triples; clearly if there are k parameters, k of these equations (and hence k distinct data points) are necessary for a unique solution. When all the parameters are known, the value of

z can be obtained for all x and y. The surface, of course, passes through all the data points. The approach is aptly described as one of surface fitting; it is so straightforward that an example of its use is superfluous.

The parameters of the function f are simply combinations of the data point triples from which they were calculated. Each value of z is thus just a weighted average, albeit a complicated one, of the original k data points. This argument can be extended further by noticing that it is hardly necessary to use just weighted averages of points. We can, for instance, construct a system whereby the z-value at a point is a weighted average of k independent functions, each itself of the form [1]. The equation is now

$$z = \sum_{j=1}^{k} g_j \cdot f_j(x, y)$$

where the g_j are weights to be applied to each of the k independent functions, f_j.

An example of this approach is the method of multiquadric surfaces developed by Hardy (1971) for the description of topographic and other irregular surfaces. In essence, the approach he took was to describe a surface as the summation of a set of independent quadrics, most commonly cones or circular hyperboloids. The general multiquadric surface may be written

$$z = \sum_{j=1}^{k} \theta_j[q(x_j, y_j, x, y)]$$

where

$q(x_j, y_j, x, y)$ is a quadric with its vertical axis of symmetry passing through $(x_j, y_j, 0)$,
θ_j is a coefficient describing the sign and "flatness" of the quadric, and
k is the number of quadrics used.

For the case of cones the surface is

$$z = \sum_{j=1}^{k} \theta_j[(x_j - x)^2 + (y_j - y)^2]^{\frac{1}{2}} \qquad .....[2]$$

A unique representation for the surface may be obtained if the coordinates of k points lying on it are known. Let such points be (x_i, y_i, z_i), i = 1, 2, ..., k. Then equation [2] may be rewritten as

$$\sum_{j=1}^{k} \theta_j[(x_j - x_i)^2 + (y_j - y_i)^2]^{\frac{1}{2}} = z_i$$

This is a series of k simultaneous linear equations in k unknowns, the θ_j, and can be easily solved. A matrix formulation for this is

$$\vec{\theta} = \vec{X}^{-1} \vec{Z} \qquad .....[3]$$

where $\vec{\theta}$ is a column vector of order k and general term θ_j,
$\vec{X}$ is a (k × k) coefficient matrix with general term $[(x_j - x_i)^2 + (y_j - y_i)^2]^{\frac{1}{2}}$,
$\vec{Z}$ is a column vector of order k and general term z_i.

Substitution of the coefficients, θ_j, into equation [2] defines the surface for all x and y.

The advantages and disadvantages of the multiquadric surface approach are as follows:

(1) It is computationally simple.

(2) The functions it defines are well-behaved interpolants.

(3) The method is completely empirical, and has no claims to optimality in any sense.

2. Surface fitting methods -- The method of least-squares

The type of approach just described is applicable only when the number of degrees of freedom, k, in the surface equation is exactly equal to the number of data points available. When the number of data points exceeds k, another approach can be used--the method of least-squares. In this method the values of the k unknown parameters are calculated so that the sum of squared deviations between the function and the data points is minimized.

The form of function is not circumscribed by the method; its selection is entirely arbitrary. Linear functions are more usual for reasons which we discuss later, but the methods can also be used for fitting non-linear functions (James, 1968, 1970). Whitten (1975) has tabulated many of the linear functions which have been used to represent surfaces: that most widely used is the simple polynomial series with non-orthogonal coefficients (Krumbein, 1959), although orthogonal polynomial series (Grant, 1957; Whitten, 1970) and double Fourier series (Harbaugh and Preston, 1965; James, 1966) have also found favor for some applications. It is worth remarking that Hardy (1971) has indicated that a least-squares solution can be employed in the method of multiquadric surfaces if the system of equations [3] is overdetermined.

The reason for preferring linear to non-linear functions is that, in some cases at least, the estimates of their parameters given by the least-squares method have certain optimal properties. This is never true for non-linear functions.

The mathematics of the least-squares method can most easily be demonstrated for what is termed the General Linear Model. This has the form:

$$\vec{Z} = \vec{X}\,\vec{\theta} + \vec{\varepsilon}$$

where $\vec{Z}$ is a column vector of order n containing the values of the dependent variable (in this case the height, z, of the surface),

$\vec{X}$ is an (n × k) matrix containing coefficients derived from the independent variables [the (x, y) coordinates of the surface],

$\vec{\theta}$ is a column vector of order k containing the parameters to be estimated,

$\vec{\varepsilon}$ is a column vector of order n containing error terms,

n is the number of data points, and

k is the number of parameters of the model ($n \geqslant k$).

The basis of the method is that the sum of squares of the errors be minimized, i.e. $\vec{\varepsilon}'\,\vec{\varepsilon}$ is to have a minimum value. The necessary condition for minimization is that the derivative with respect to $\vec{\theta}$ is zero, i.e.

$$\frac{d(\vec{\varepsilon}'\,\vec{\varepsilon})}{d\vec{\theta}} = \frac{d([\vec{Z} - \vec{X}\,\vec{\theta}]'\,[\vec{Z} - \vec{X}\,\vec{\theta}])}{d\vec{\theta}} = 2\vec{X}'(\vec{Z} - \vec{X}\,\vec{\theta}) = 0$$

The parameter vector, $\vec{\theta}$, is obtained from this by rearranging the terms: $\vec{\theta} = (\vec{X}'\,\vec{X})^{-1}\,\vec{X}'\,\vec{Z}$.

When the error terms, $\vec{\varepsilon}$, are uncorrelated, have zero mean, and have the same variance, it can be proved (see for example Kendall and Stuart, 1967) that the least-squares estimator, $\vec{\theta}$, gives the best linear unbiased estimates of the parameters. If, in addition, the error terms are normally distributed, then it is also the maximum-likelihood estimator.

The linear model may be summarized by the following equation (Watson, 1972):

Value at any point = Value of the deterministic function + Random error .....[4]

It is important to stress that the least-squares estimators are optimal only when the error terms are truly random. If there is sufficient evidence that they are spatially autocorrelated (and this will generally be true unless the number of degrees of freedom used in the function is almost as great as the number of data points), then the assumptions of the least-squares method are violated, and the estimated parameters are neither unbiased nor of minimum variance. It is thus essential that the error terms from any application of least-squares be tested for spatial autocorrelation, for instance in the manner suggested by Cliff and Ord (1973). Failure to appreciate this point has caused most of the misuse of the least-squares method in surface fitting: sadly this includes much of the earlier geological work using trend surfaces.

If equation [4] is rewritten as

Value at any point = Deterministic function ("regional component") + Local residual

where the residuals represent local phenomena and are spatially autocorrelated, then the estimates of the parameters of the function are totally arbitrary. What are often misleadingly called "trends" may be no more than artifacts (Matheron, 1971).

It is instructive to consider the extent to which surface functions determined by least-squares satisfy the criterion of being good interpolants, as required here. As noted above, a complex function must be used for any but the smallest data set, in order to remove spatial autocorrelation. It has, however, been found in practice, especially for polynomial series, that such functions tend to fluctuate very erratically between data points (Matheron, 1967); even when a perfect fit is obtained at every data point (n = k), the function is rarely acceptable as an interpolant.

In summary, the advantages and disadvantages of the least-squares method are as follows:

(1) It is based on a precise probabilistic model and can, under certain conditions, define surfaces having some optimal properties.
(2) It is computationally straightforward.
(3) Surfaces estimated without regard for spatially autocorrelated error terms are just as arbitrary as surfaces fitted by eye.
(4) Complex surfaces tend to perform poorly as interpolants.
(5) The form of the surfaces is greatly influenced by the spatial distribution of the data points (Doveton and Parsley, 1970).

3. Interpolation using Regionalized Variable Theory

Regionalized Variable Theory was developed by Matheron to describe functions which vary in space with some continuity, yet which are mathematically intractable (Matheron, 1967). Comprehensive descriptions of the theory have been given by Matheron (1965, 1971) and, most recently, by Olea (1975). The theory is extensible to the N-dimensional case; here we consider just surface functions with the form of equation [1].

Matheron's approach is to consider the function at any point as a random variate, the value of which depends on its position. If the statistics of the variate are known, its value at any given location can be estimated. In principle then this approach is ideally suited to the type of surface representation in which we are interested. A more detailed look at the theory, however, results in a more pessimistic judgment.

The random variate, $Z(\vec{x})$, is described by its mean and covariance, and by certain assumptions about its behavior. Under what are termed conditions of weak stationarity, the expected value of the variate, $E[Z(\vec{x})]$, is constant and its covariance is independent of absolute location. In some circumstances these assumptions are too strong and what is termed the intrinsic hypothesis is preferred. Under this, it is not the variate which is weakly stationary, but rather its increments $[Z(\vec{x} + \vec{h}) - Z(\vec{x})]$. The second moment of these increments, expressed as a function of the increment size, is termed the semivariogram, $\gamma(\vec{h})$, where

$\gamma(\vec{h}) = E\{[Z(\vec{x} + \vec{h}) - Z(\vec{x})]^2\}/2$,

$Z(\vec{x})$ is the value of the variate at $\vec{x}$, and

$Z(\vec{x} + \vec{h})$ is the value at a distance $\vec{h}$ from $\vec{x}$.

In some cases even the intrinsic hypothesis is too restrictive, as for example when the expected value of the variate, $E[Z(\vec{x})]$, is not constant. The variate is then a function having a trend in value across the area under study: this trend is called the drift. Under this, the weakest of the three hypotheses, the residuals obtained by removing the drift are assumed to obey the intrinsic hypothesis, i.e. to have a semivariogram of intrinsic form.

For our purposes it is the third of the hypotheses which is the most relevant, as it is the most general: estimation of point values under this hypothesis is termed Universal Kriging. The estimate, $\hat{Z}(\vec{x})$, of the variate $\vec{x}$ is defined as a linear combination of the known values of $Z(\vec{x})$ which are within a neighborhood of radius r around $\vec{x}$. Thus

$$\hat{Z}(\vec{x}) = \sum_{j=1}^{k} \lambda_j \, Z(\vec{x}_j) \qquad \text{.....[5]}$$

where λ_j are unknown weights which can be calculated, and

k is the number of points in the neighborhood.

The estimates produced by this procedure are optimal in the sense of being unbiased and of having minimum estimation variance. It is, however, a prerequisite that either the covariance of the residuals or the semivariogram of the residuals be known (these have been generically termed correlograms). If that is not the case, the drift must first be determined in order that the residuals (and hence their correlograms) can be calculated. The drift, $m(\vec{x})$, at a point $\vec{x}$ within a radius r of a fixed point $\vec{x}_o$ is defined in a manner analogous to the Universal Kriging equation [5]. Thus

$$m(\vec{x}) = \sum_{i=1}^{n} g_i \, f^i(\vec{x})$$

where $f^i(\vec{x})$ are arbitrary functions of $\vec{x}$,

g_i are unknown weights to be calculated, and

n is the number of points within the r-radius neighborhood of $\vec{x}_o$.

It is at this stage that one of the major drawbacks to the regionalized variable approach is found, for it is necessary to know one of the correlograms of the residuals in order to determine the drift. An iterative solution to the problem is usually suggested (Olea, 1975), whereby a theoretical correlogram and expression for the drift are assumed, the coefficients in the drift expression calculated, and an experimental correlogram for the residuals then produced. Unless this is a sufficiently good fit to the theoretical correlogram, it is itself used as an initial approximation in another iteration. There is no guarantee that trial solutions using different initial configurations of correlogram and drift expression will converge, and in consequence the method is, at least in its initial stages, substantially arbitrary.

The advantages and disadvantages of the regionalized variable approach under the third hypothesis are as follows:

(1) The method is optimal in the sense of producing the best linear unbiased estimators, if the following features are known beforehand: a form for the drift, a theoretical correlogram for the residuals, a value for the neighborhood radius.

(2) Because in general these features will not be known, the estimates given by the method will usually be quite arbitrary.

(3) The method is essentially a point-by-point estimation procedure. Thus a surface is never described in a form in which it can easily be manipulated. This above all renders the method unsuitable for our surface representation problem.

(4) The method is computationally expensive and time-consuming.

Blending Function Methods

In each of the methods so far considered, we have come across the idea that a surface can be constructed by some kind of weighted averaging procedure. For some methods this is stated explicitly; for others it is inherent in their structure. A surface is thus a blend of the original data, with the relative importance of each data item varying continuously from place to place.

The idea of a blended surface may be expressed formally in the following way:

$$z(u) = \sum_{j=1}^{k} g_j(u) \cdot f_j(x, y)$$

where $g_j(u)$ are blending functions which vary in value according to the parameter u,

f_j are functions of position (the original data items),

u is a parameter varying continuously across the surface, and

k is the number of data items to be blended.

The purpose of a blending function is simply to ensure that adjacent parts of the final surface merge smoothly into one another. It is the mathematical form of the blending function which determines the continuity of the surface (this is discussed in a later section). As noted earlier, it is not just point coordinates which can be blended together; space curves and surfaces can be treated in the same way. As a result, the blending function approach is widely used in surface representation methods; it is especially important in some of the CAD methods described later. For now, however, a simpler method will be described which illustrates well the blending function approach. This is the surface averaging method, developed by Junkins, *et al.* (1973) for the mathematical modelling of irregular topographic surfaces.

The object of the surface averaging method is to fit a surface over a regular grid of data points. It does this by considering the surface as a set of individual sub-surfaces, each valid over a single grid cell, and each defined as a blend of several preliminary surfaces calculated for some of the surrounding cells. Consider an (m by n) network of points, $P_{i,j}$, on the x-y plane, with each cell rectangular in plan view (Fig. 4). As notation, define $c_{i,j}$ as the cell

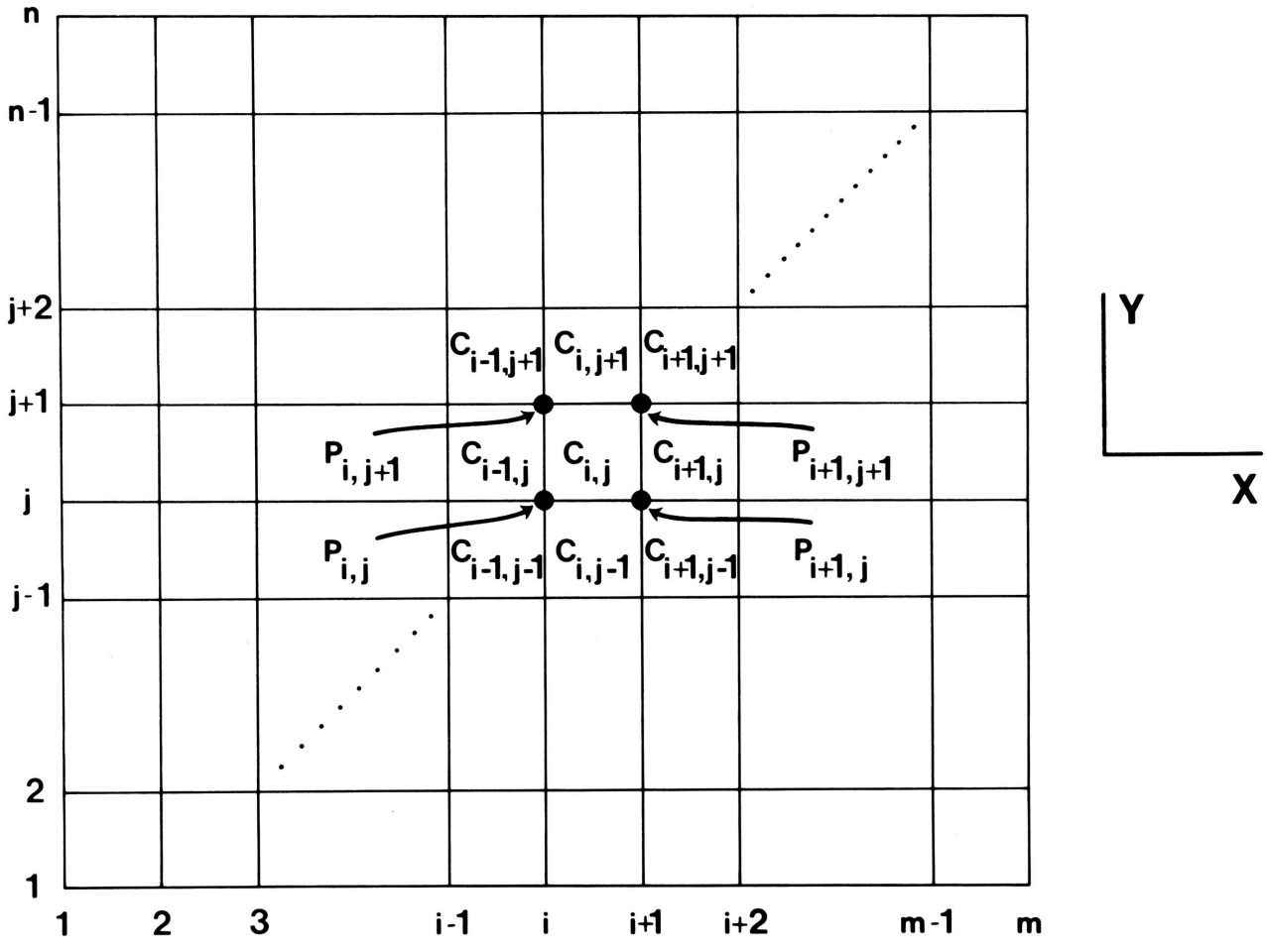


FIGURE 4. Mesh of (m by n) data points, with each cell rectangular in plan view. The notation given for points (P, see abscissa and ordinate indices) and cells (c) is that used in the surface averaging method.

with corner points $P_{i,j}$, $P_{i,j+1}$, $P_{i+1,j+1}$, and $P_{i+1,j}$. The sub-surface, $S^*_{i,j}$, valid over $c_{i,j}$ is then

$$S^*_{i,j}(x, y) = \sum_{k=1}^{4} g_k(x, y) \cdot S'_k(x, y)$$

where g_k are blending functions,

S'_k are preliminary surfaces:

S'_1 is valid over cells $c_{i,j}$, $c_{i,j+1}$, $c_{i+1,j+1}$, $c_{i+1,j}$

S'_2 is valid over cells $c_{i-1,j+1}$, $c_{i,j+1}$, $c_{i,j}$, $c_{i-1,j}$

S'_3 is valid over cells $c_{i-1,j}$, $c_{i,j}$, $c_{i,j-1}$, $c_{i-1,j-1}$

S'_4 is valid over cells $c_{i,j-1}$, $c_{i+1,j-1}$, $c_{i+1,j}$, $c_{i,j}$

$x_{i,j} \leq x \leq x_{i+1,j}$ and $y_{i,j} \leq y \leq y_{i,j+1}$.

Thus each $S^*_{i,j}$ is a weighted average of four preliminary surfaces, each of which is valid over four neighboring cells. In their original description of the method (Junkins, *et al.*, 1973), the authors used least-squares approximation for the preliminary surfaces. Their mathematical form can, however, be any tractable function. It is important to note that the functions must be chosen with care if the final surface, $S^*_{i,j}$, is to interpolate the data points.

The blending functions, $g_k(x, y)$, are determined by the smoothness with which adjacent sub-surfaces are required to blend. For continuity of position and slope along each cell boundary the blending functions are

$$g_1(x, y) = \frac{(x - x_{i,j})^2 (y - y_{i,j})^2}{\Delta x^2 \, \Delta y^2} \left[9 - \frac{6(x - x_{i,j})}{\Delta x} - \frac{6(y - y_{i,j})}{\Delta y} + \frac{4(x - x_{i,j})(y - y_{i,j})}{\Delta x \cdot \Delta y}\right]$$

$$g_2(x, y) = g_1(1 - x, y)$$

$$g_3(x, y) = g_1(1 - x, 1 - y)$$

$$g_4(x, y) = g_1(x, 1 - y)$$

where $\Delta x = x_{i+1,j} - x_{i,j}$

$\Delta y = y_{i,j+1} - y_{i,j}$.

The advantages and disadvantages of the method are as follows:

(1) It is computationally efficient, and applicable to large data sets.

(2) It is flexible; any suitable function can be used for the preliminary surfaces.

(3) Although the authors state that "if one least squares approximation is good, the average of four must be better" (Junkins, *et al.*, 1973, p. 1800), this claim ignores the uniqueness of the optimal least-squares solution. One cannot pretend that the method is other than highly empirical.

(4) When preliminary surfaces are used which have been fitted by the method of least-squares, the final surface does not interpolate the data points.

Requirements for Practical Surface Description

Clearly none of the methods so far described is the ideal for our surface representation problem. Those which claim optimality (the least-squares method; Regionalized Variable Theory) are demonstrably non-optimal except under particular and usually unrealistic assumptions; the others produce results which are acceptable for many applications, but rely entirely on the combination of sets of functions whose form is totally arbitrary. Yet each method has some advantages which should be retained, or drawbacks which should be avoided. With this in mind a set of minimum requirements can be specified which a surface representation method should meet, and against which other methods can be assessed. Because the methods are to be of practical use, utility is emphasized rather than strict statistical or mathematical optimality.

1. Foremost among these requirements is continuity, or local smoothness. Most surfaces in geology are continuous, both in a physical and a mathematical sense. Physical continuity usually results from some areally operative geologic process, as in the formation of geomorphic features by erosion. In cases like this, the surface possesses some degree of smoothness, at least locally, and the mathematical surface representing it can reasonably be required to be continuous both in its value and first derivative. Higher order derivative continuity may of course also be required, and it is common to ask for curvature continuity as well.

2. The requirement of local smoothness is valid for many applications. Other similar requirements include suppression of surface undulations (Akima, 1970, 1974), linear additivity, and invariance under some coordinate transforms (Akima, 1975).

3. More troublesome is the requirement that the surface pass through particular data points, either all the points on which it is based or a pre-defined subset of these. This must depend entirely on the use to which the surface will be put, and on the quality of the data. In some cases it may be sufficient to approximate the data points within pre-specified tolerances; for other work a surface fitting method may have to be chosen whose fidelity depends solely on the accuracy of computation. Sparsely or irregularly distributed data impose only weak constraints on the surface; in such cases only approximations are warranted.

4. Probably the most important requirement is that the surface should, if necessary, be capable of being multiple-valued in any coordinate direction, and that its description should be independent of its orientation. Most methods used for surface representation in geology, such as those described earlier, have used representations with the form of equation [1]. The inadequacy of this representation for other than simple surfaces may be illustrated as follows. For convenience only a two-dimensional example is given--the two-dimensional analogue of equation [1] is

$$z = f(x) \qquad \text{.....[6]}$$

Figure 5 shows a lenticular channel sand in cross-section. Its lower surface is formed by the truncated edges of underlying beds; the upper surface is defined by the bases of overlying deposits, usually with slight angular unconformity. In the original orientation (Fig. 5a) each surface can be represented by equations of the form [6]. Figure 5b shows the sand body after tectonic rotation: use of equations of this form will now result in extreme distortion because of the concentration of values of the independent variable within $x_1 \rightarrow x_2$

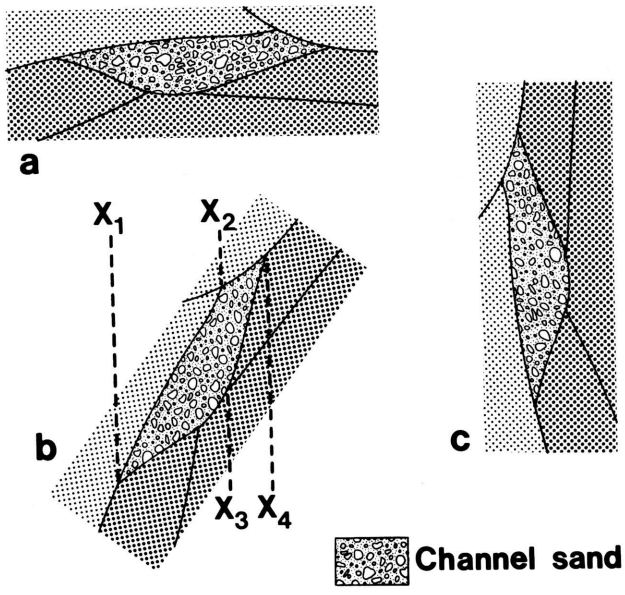


FIGURE 5. Cross - sections through a braided channel sandstone. The underlying beds are shown in the coarser stipple, the overlying beds in the finer stipple. (a) Original orientation. (b) After rotation, equidistant points on the surfaces of the sand are inevitably over - concentrated within $x_1 \rightarrow x_2$ and $x_3 \rightarrow x_4$. (c) After further rotation until the major axis of the sand is almost vertical.

and $x_3 \rightarrow x_4$. After further rotation the major axis of the sand is almost vertical (Fig. 5c) and both surfaces are multiple-valued in z for most values of x. No equation of the form [6] is applicable to either surface.

The restrictions evident in this example may be overcome in two ways: (a) Use of coordinate transforms, such as rotation to restore the sand to approximately its original orientation. Other transforms include those used by Thompson (1961) for comparison of organic shapes. The generalized approach to coordinate transformation is described by Morse and Feshbach (1953). (b) Subdivision of the surface into parts, each simple enough to be represented in the form given by equation [6].

Both approaches have the drawback that the model is dependent less on the actual form of the object than on the position and orientation in which it is presented. Because this is clearly undesirable, neither approach demands more than passing consideration here. A third approach using curvilinear coordinates is more useful, and is outlined in the next section.

Curvilinear Coordinate Systems

In rectangular Cartesian coordinates, a point on the plane z = 0 can be expressed in terms of the axes x and y, which are orthogonal and coplanar. If the latter restriction is removed, the x-y surface can be deformed without breaking to give any simply connected surface. Any point on this surface can still be expressed in terms of the two axes (relabelled u and v), which remain orthogonal (Fig. 6). If a point on the surface has coordinates (x, y, z) with respect to the original Cartesian system, then its curvilinear representation is a vector having components:

$$x = f_1(u, v), \quad y = f_2(u, v), \quad z = f_3(u, v) .$$

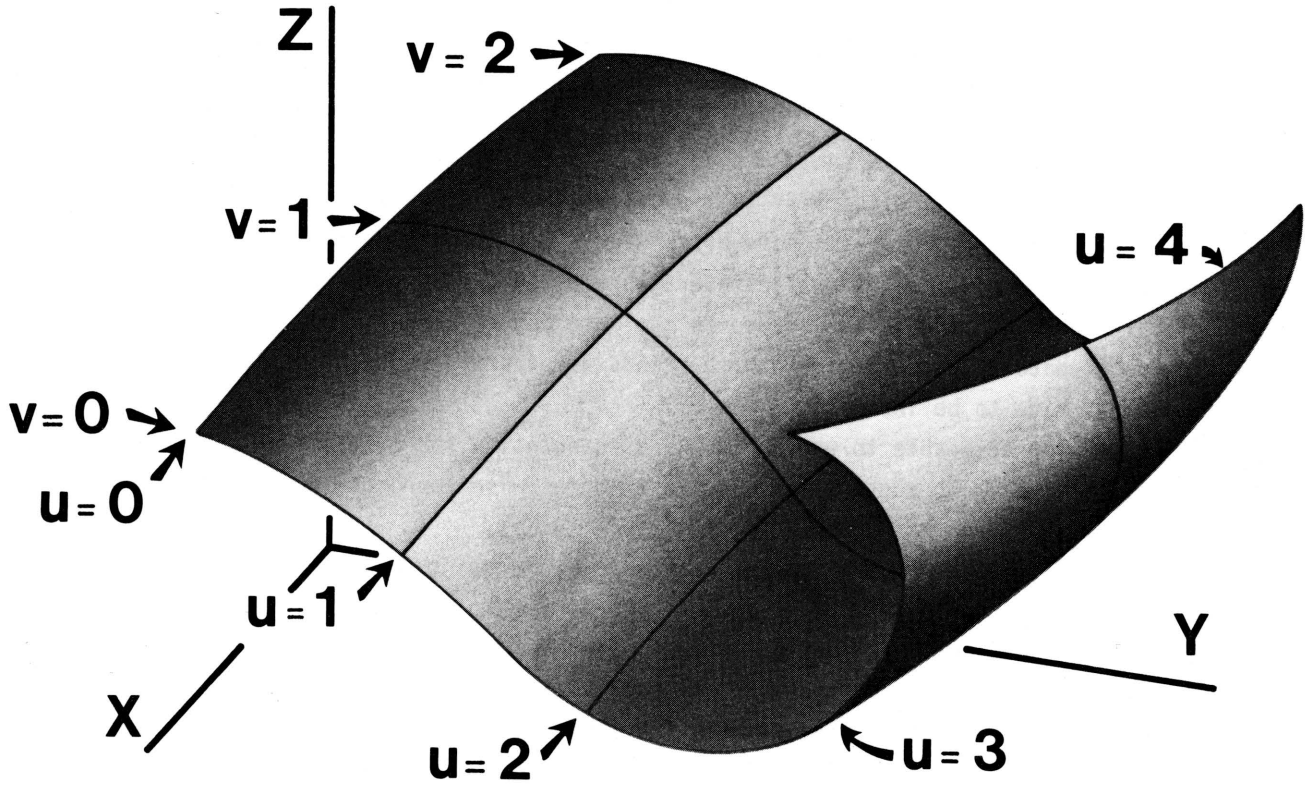


FIGURE 6. A curvilinear coordinate system. Points are described in terms of the parameters u and v, which are defined only on the surface. The surface is completely specified by the function: $f(u, v) = [f_1(u, v), f_2(u, v), f_3(u, v)]$.

The surface is thus completely defined by the functions f_{1-3} and, provided that suitable functions are specified, surfaces of extreme complexity may be represented. In practice the range of available functions is restricted by the necessity of incorporating into them some of the other requirements described earlier, e.g. continuity of a prescribed degree.

Global and Local Representations

Either of two alternative strategies can be applied at this stage. A single triplet of functions valid for all u and v gives what is termed a global representation. This is analogous to the surface fitting approach described earlier, in which a single function was fitted to all the data points. The second strategy is to partition the surface into local u-v domains, and then to define a set of functions for each. These are valid only in that domain. Each of the local sub-surfaces is called a surface patch. This local approach (often termed piecewise) is identical to that used in the surface averaging method of Junkins, *et al.* (1973).

Both global and local strategies have inherent advantages and disadvantages:

(1) A global representation needs only one set of functions for the whole surface, giving savings in computational time and storage.

(2) Global representation of surfaces of even moderate complexity requires extremely complex

functions, even if the tolerance allowed in the model is high.

(3) Any global representation is unreliable at its extremes. Furthermore, if it is at all complex, it will often behave erratically as an interpolant.

(4) Local representation is computationally less efficient in terms of storage required, as one set of functions must be stored for each patch.

(5) Each set of local functions can, however, be relatively simple and yet still give a good representation. The simplicity of the functions and the fact that they are locally based imply that a local representation is often highly efficient for large data sets. This was found to be the case for the surface averaging method (Junkins, *et al.*, 1973).

(6) A global representation is continuous to the same degree everywhere. To achieve this result using local patches, the form of the functions is severely restricted and the number of patches may have to be increased.

(7) It is often impracticable to use a global representation for multiple-valued or concave surfaces. There is no such restriction to the piecewise approach.

Computer-Aided Design Methods

Up to now we have dealt only with methods which have an obvious relevance to our surface representation problem. Each method has been applied in geology or geodesy, and there would be little point in implementing them for the Cherokee data set. There is, however, another class of methods developed in an entirely unrelated field whose relevance to the problem is not nearly as peripheral as one might initially expect; they have in fact been used for contouring subsurface horizons (Hessing, *et al.*, 1972). These methods were mostly devised as computer-aided design tools in the aircraft, motor, and shipbuilding industries. Two excellent reviews of work in this field have been given by Forrest (1972a, 1974). The theoretical basis of many of the individual methods is covered in the next sections of this paper: for now it is sufficient to regard them as methods which generally use sets of local surface patches which, when "sewn" together, create complex surface forms (Fig. 7).

The problem of computer-aided design is not identical to our surface representation problem, although both share many theoretical ideas and practical difficulties (Ahlberg, 1974; Barnhill, 1977; Wu, *et al.*, 1977). The CAD problem is essentially one of *ab initio* shape design. The designer has no model, except for his own conception of how the finished object should appear. His approach is simply to build up the object from scratch, either by amalgamating simple solid forms (see for example Braid, 1975) or by defining the object by its bounding surfaces (Coons, 1967; Bézier, 1972), until it satisfies certain aesthetic or functional constraints.

In contrast, our surface representation problem involves developing a mathematical model which conforms, however approximately, to a specific object. Most, if not all, of the surface design methods are in theory applicable: the main difficulty in using them is the practical one of devising a suitable interface between the model being built and the object which it is to represent. Given particular data on the shape of the object (usually the coordinates of points on its surface, as for the Cherokee data set), how are they to be incorporated into the

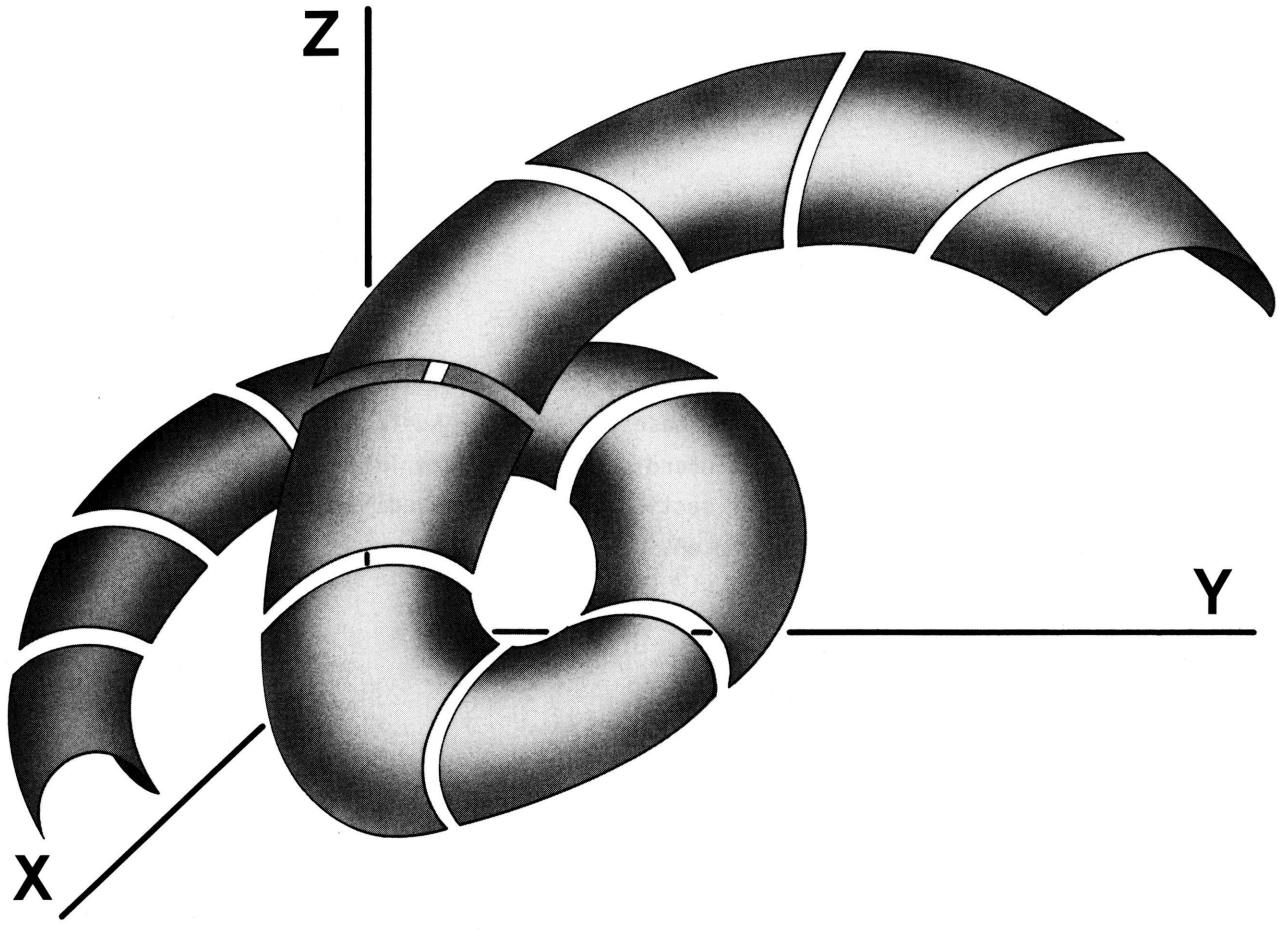


FIGURE 7. The use of simple surface patches to construct a complex three-dimensional figure.

model, and how are discrepancies between model and object to be measured and corrected? One elegant and practical solution to the problem is offered by Wu, *et al.* (1977).

One way to overcome the difficulty is to organize the data points into some form of network, and to use it as an approximation to the surface being modelled. This approach contrasts markedly with the surface representation methods considered earlier which, with the exception of the surface averaging method, put relatively little stress on the topological configuration of points controlling surface shape in any local area.

The most commonly used type of network (Fig. 8) is one where the cells are quadrilateral in plan view: the connections between data points imply that the network is topologically equivalent to a planar rectangular grid. Triangular networks (Goël, 1968; Sabin, 1968b, 1969a, 1971a; Barnhill, 1974, 1977; Birkhoff and Mansfield, 1974; Lawson, 1977) and networks having some pentagonal cells (Sabin, 1968b) can also be used. The networks need not be regular, although this is usually the case. Practical applications of irregular network representation of complex objects have used combinations of twisted quadrilateral and triangular cells (Tipper, 1976, 1977). From now on, however, it will be assumed that the data are available as coordinates of the nodes of a network of quadrilaterals: this conforms to the Cherokee data set (Fig. 9).

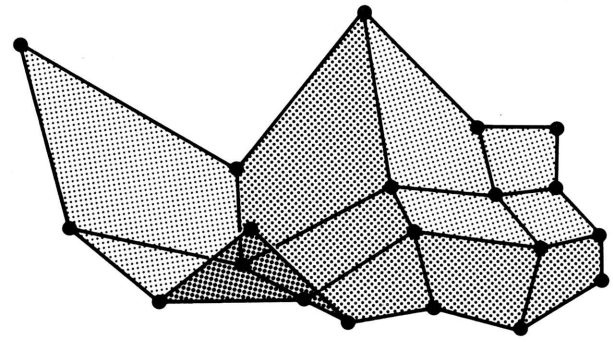
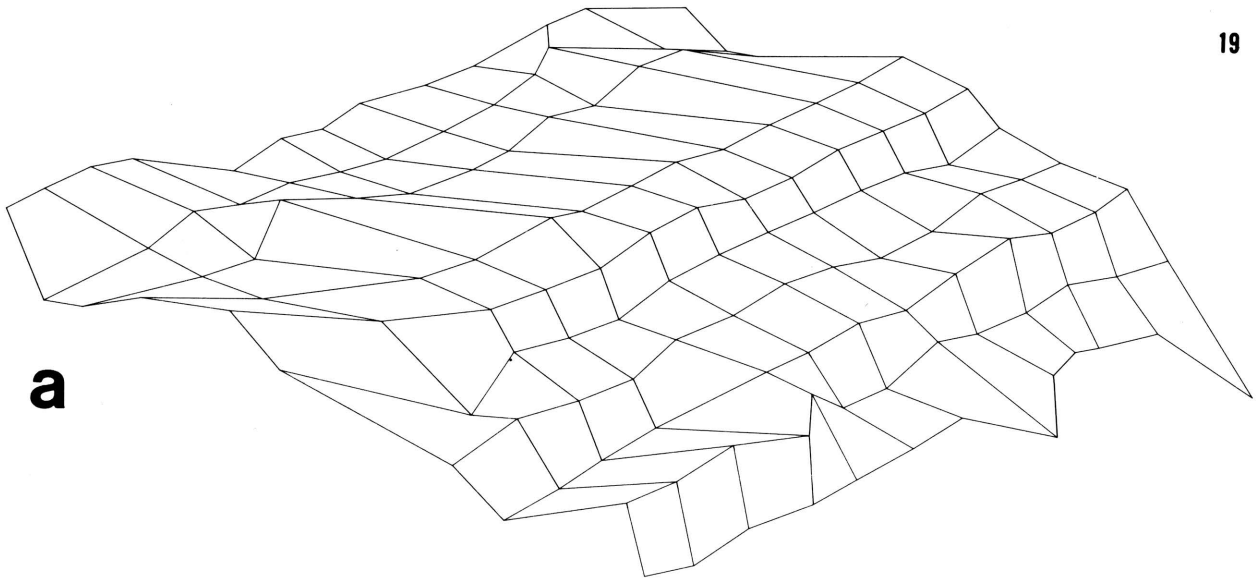


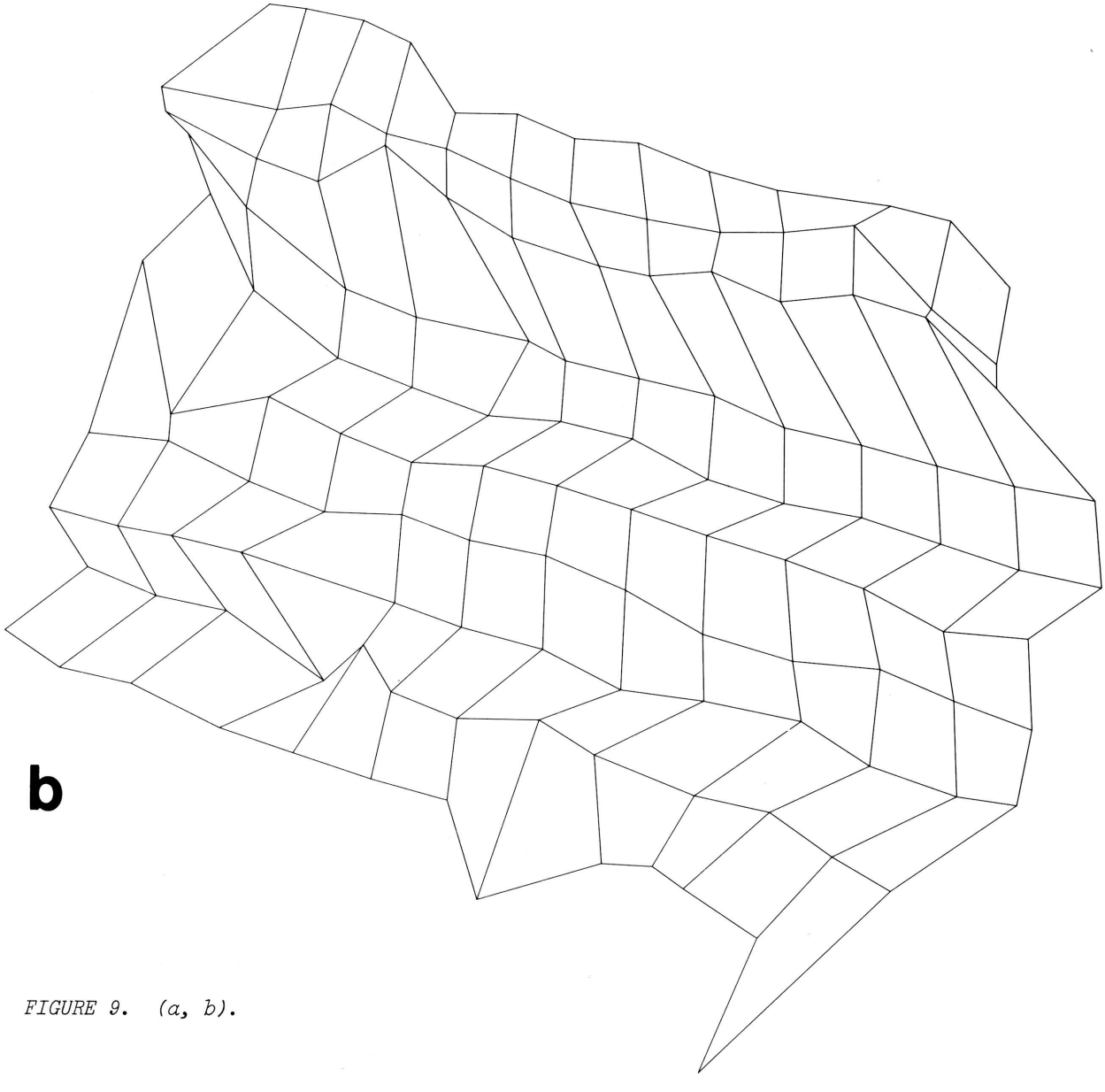
FIGURE 8. A (5 by 4) mesh of points which is topologically equivalent to a planar rectangular grid.

There are two advantages to this network approach: (1) It is compatible with most CAD methods--each cell is used simply as the base for a single surface patch; (2) It reduces the surface representation problem to a standard one of bivariate interpolation or approximation over a quasi-planar grid. In the next section some of the fundamental features of univariate and bivariate interpolation are introduced.

FIGURE 9. Perspective projections of the Cherokee data set: center of projection located at $z = -50$. See Figure 29 for identification of viewing angles ϕ and θ. (a) Top surface, $\phi = -100°$, $\theta = -50°$. (b) Top surface, $\phi = -160°$, $\theta = -50°$. (c) Bottom surface, $\phi = -100°$, $\theta = -50°$. (d) Bottom surface, $\phi = -160°$, $\theta = -50°$.

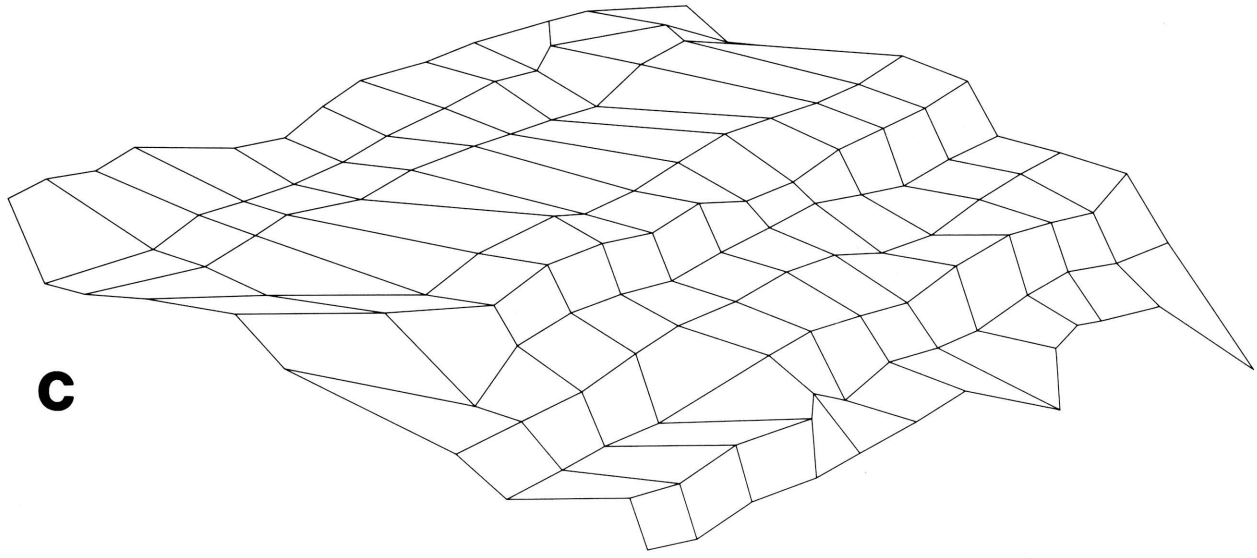


a

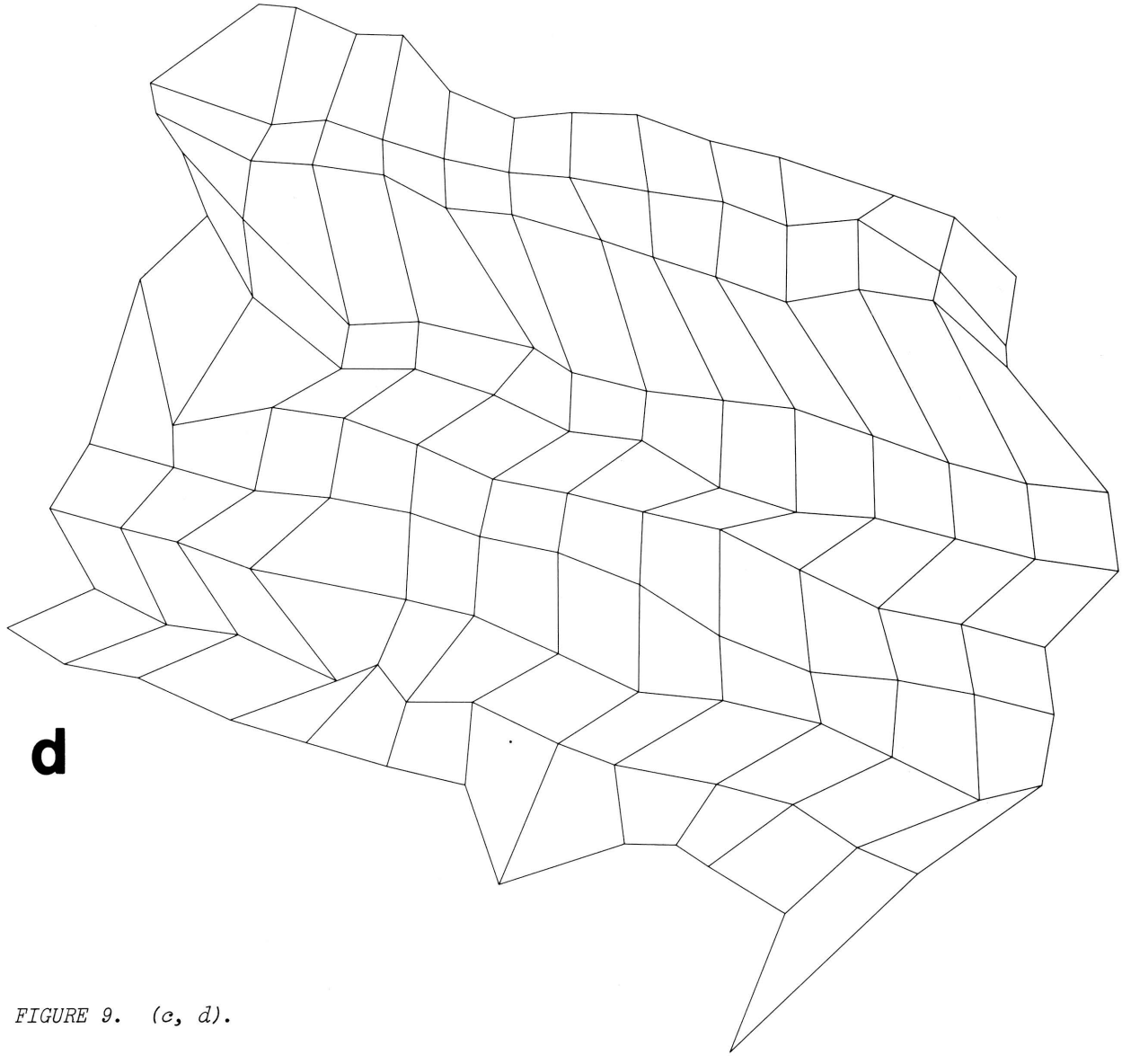


b

FIGURE 9. (a, b).



c



d

FIGURE 9. (c, d).

Univariate and Bivariate Interpolation

The mathematical theory of interpolation and approximation is treated at length by Davis (1963); here we are concerned only with some of its more practical aspects. For simplicity's sake only polynomials of restricted degree are here considered as interpolants. The notation refers to rectangular Cartesian coordinates (x, y, z), but the approach applies also in a curvilinear coordinate system.

Univariate interpolation

In the univariate case, an interpolant is simply a function whose form is constrained by an ordered set of data points. It is usually required to pass through each point, but may in addition have to conform to some specified conditions on its derivatives. For polynomials the following theorem may be proved (Walsh, 1935, p. 49):

Let values $w_i^{(j)}$ *be given at each of* $(m + 1)$ *distinct points,* x_i, *where* $i = 0, 1, \ldots, m$ *and* $j = 0, 1, \ldots, n_i$. *In general* $n_i \neq n_k$ *for* $i \neq k$. *Then there exists a unique polynomial* $P(x)$ *of degree* $N = -1 + \sum_{i=0}^{m} (n_i + 1)$, *whose derivatives satisfy the conditions* $P^{(j)}(x_i) = w_i^{(j)}$ *for* $j = 0, 1, \ldots, n_i$ *and* $i = 0, 1, \ldots, m$.

The general case of this theorem (arbitrary values of n_i; polynomial of degree N) leads to what is known as Full Hermite interpolation (Davis, 1963). A less elaborate system is obtained when $n_i = 1$, for all i. This system, termed Simple Hermite (or osculatory) interpolation, gives only positional and first-degree interpolation at each data point. The interpolant is of degree $(2m + 1)$; its form is given by Hildebrand (1956, p. 316).

The simplest interpolation system is one where positional interpolation alone is required ($n_i = 0$, for all i). This is the commonest situation because in practice it is very unusual to find a data set containing anything save the crudest estimates of derivatives (cf. Birkhoff, 1969). The interpolating polynomial is of degree m [class $C^{(m-1)}$] and is, of course, unique. When m is small, the coefficients of the polynomial can be simply obtained by substitution of the $w_i^{(0)}$ into the polynomial equation, followed by solution of $(m + 1)$ simultaneous linear equations. When m is large, it is more convenient to express the interpolant differently, and numerous such schemes have been developed (see for example the descriptions in Whittaker and Robinson, 1944). One of the most useful of these is Lagrange's formula:

$$P(x) = \sum_{i=0}^{m} w_i^{(0)} \cdot L_i(x)$$

where

$$L_i(x) = \left[\prod_{\substack{j=0 \\ j\neq i}}^{m} (x - x_j)\right] / \left[\prod_{\substack{k=0 \\ k\neq i}}^{m} (x_i - x_k)\right]$$

Each of the alternative schemes can be rearranged algebraically and shown to be equivalent to the Lagrange formula. For this reason the interpolation of point values alone is often termed Lagrangian interpolation.

Bivariate interpolation

By analogy with univariate interpolation, a bivariate interpolant is a function which reproduces positional and derivative information at points in a three-dimensional space. Two other features also carry over from the univariate case; the uniqueness theorem, and the concepts of Lagrangian and Hermitian interpolation. Here the points to be interpolated are restricted to be the nodes of a planar rectangular grid. In general, however, an interpolation polynomial will exist if the points form either a rectangular or triangular mesh (cf. Davis, 1963, p. 27).

The most obvious way of obtaining a bivariate interpolant is to generalize a univariate one. We can develop this idea by considering interpolation over a single cell (Fig. 10), in which B_x and B_y are univariate interpolation operators acting in the x and y directions respectively. Once their mathematical form is specified, these operators become the interpolating functions we discussed earlier: they may be Hermitian or Lagrangian. If approximation is required rather than interpolation, B_x and B_y are simply approximation operators. Whereas in the univariate case one can only interpolate (or approximate) between points on a line, in the bivariate case two alternatives are possible: (1) interpolation to points in space, e.g. to the corner points of the cell [P(0,0), P(0,1), P(1,0), P(1,1)]; (2) interpolation to lines in space (space curves), e.g. to the sides of the cell [P(x,0), P(0,y), P(x,1), P(1,y)]. Note that either alternative can itself be Lagrangian or Hermitian, depending on whether derivative values are specified at the points and on the space curves.

There are four possible ways in which the operators B_x and B_y can be combined (Gordon, 1971). These are:

(1) use of B_x to interpolate between P(0,y) and P(1,y),

(2) use of B_y to interpolate between P(x,0) and P(x,1),

(3) the Cartesian product B_xB_y (= B_yB_x), giving interpolation between P(0,0), P(0,1), P(1,0) and P(1,1),

(4) the Boolean sum $B_x + B_y - B_xB_y$, giving interpolation between P(0,y), P(1,y), P(x,0), P(x,1) and, by implication, also between P(0,0), P(0,1), P(1,0) and P(1,1).

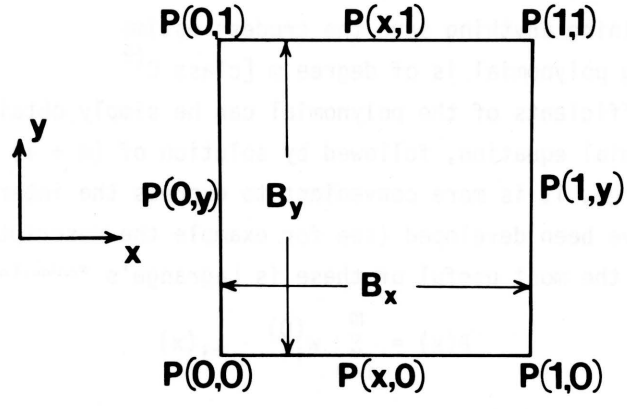


FIGURE 10. Bivariate interpolation over a single rectangular cell. B_x *and* B_y *are generalized univariate interpolation operators acting in the x and y directions respectively.*

The first and second alternatives produce what are termed lofted (ruled) surfaces; each surface can be regarded as a set of isoparametric curves parallel to either the x or y axis. The third alternative is also known as the tensor product (or cross product) representation, and is the most commonly used, both for approximation and interpolation. A bivariate Cartesian product generalization of Hermitian interpolation is given by Ahlin (1964). The fourth alternative is the most general, and of substantial theoretical interest. Despite this, it is in practice unused (Forrest, 1972a, p. 351). It will be discussed further during the description of Coons surfaces.

We are now in a position to consider specific schemes of approximation and interpolation over a network of quadrilaterals. In the succeeding sections individual CAD methods will be described which implement some of these schemes.

Bernstein-Bézier Curves and Surfaces

One of the earliest practical CAD systems was that devised and implemented by Bézier (1972, 1974) at Renault. We select it as the first CAD method to be described because it is a straightforward application of the idea of networks. In Bézier's system, termed UNISURF, surfaces are described in a particularly elegant manner. Subsequently Forrest (1972b) and Gordon and Riesenfeld (1974a) have shown that the method is simply an application of classical Bernstein polynomial approximation (Davis, 1963) in a curvilinear coordinate framework.

Fundamentals of the Bézier method

The mathematics of Bézier's method is quite straightforward; for simplicity it is developed first for curve fitting and then generalized for surfaces. Consider a string of points, $\vec{P}_i$, where $i = 0, 1, \ldots, m$. Vector notation is used in accordance with the curvilinear coordinate system outlined earlier. The Bernstein polynomial approximant to the polygon $[\vec{P}_i]$ is

$$\vec{B}(u) = \sum_{i=0}^{m} \psi_i(u)\, \vec{P}_i \qquad \ldots\ldots[7]$$

where the $\psi_i(u)$ are the Bernstein basis functions, and u is a parameter constrained to lie in the range $0 \rightarrow 1$. The basis functions are in fact the discrete binomial probability density functions, i.e.

$$\psi_i(u) = \binom{m}{i} u^i (1 - u)^{m-i} \qquad \ldots\ldots[8]$$

The functions for $m = 3$ are plotted in Figure 11.

Three properties of the Bernstein approximant are especially important:

(1) It interpolates the first and last points, $\vec{P}_0$ and $\vec{P}_m$, of the polygon;

(2) It is a convex combination of the polygon vertices, and hence $\vec{B}(u)$ lies within the convex hull of the polygon (Fig. 12). The polygon $[\vec{P}_i]$ is known as the control polygon, and each of its individual points is known as a control point.

(3) It provides variation diminishing approximation (Schoenberg, 1967), i.e. the approximant is always less undulatory than the function being approximated. In practice this means that no straight line can cut the approximant more times than it cuts the function itself.

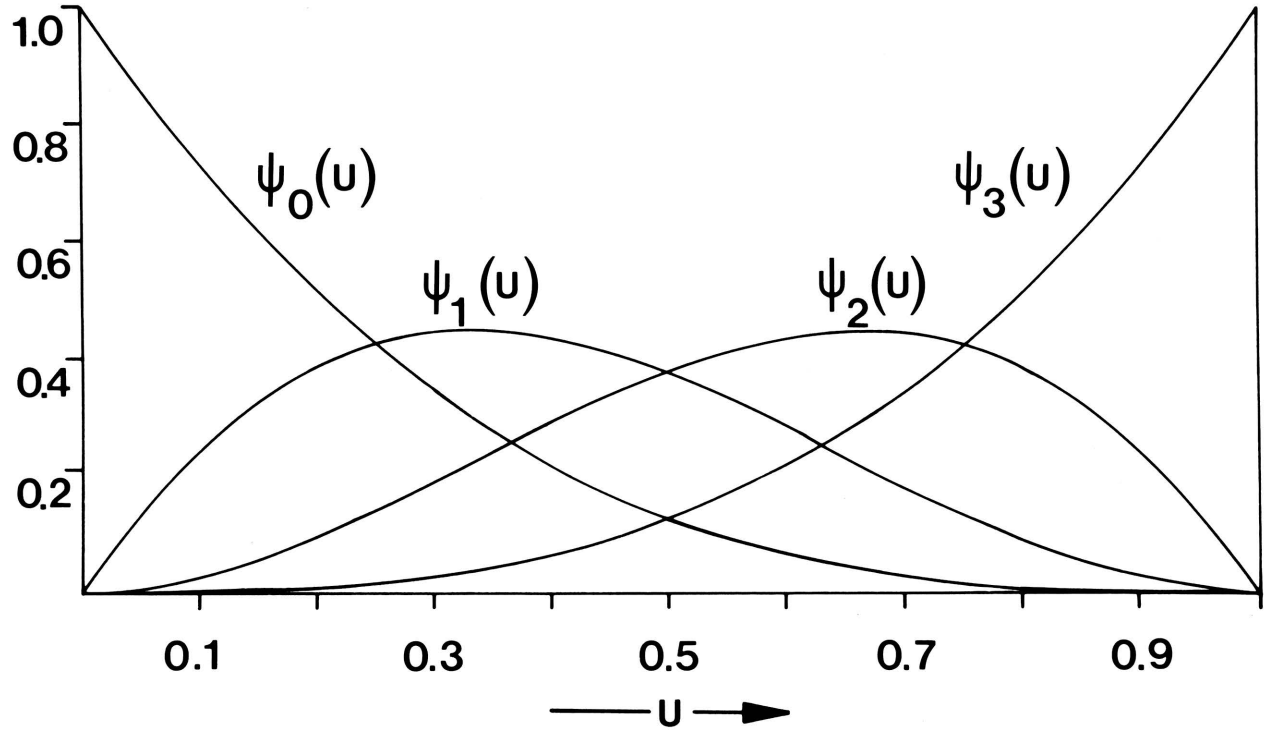


FIGURE 11. Bernstein basis functions for m = 3.

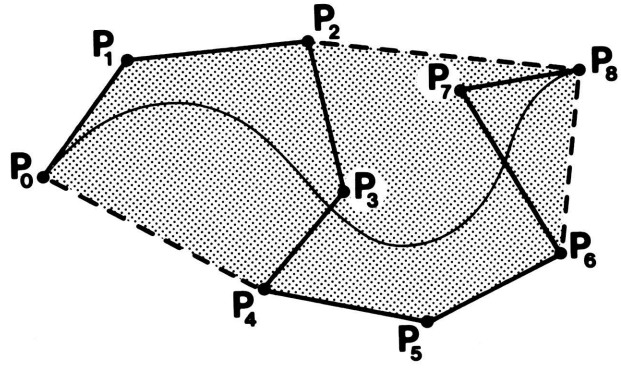


FIGURE 12. Bézier curve, m = 8. Control polygon is indicated by unbroken heavy line, interpolated curve by light line. The convex hull of the control polygon is shaded.

Another interesting property (Bézier, 1972, p. 121-122) is the geometric interpretation of equation [7]. This is illustrated in Figure 13 for a three-sided polygon (m = 3). To determine the value of the approximant at $u = u_o$, find on each side of the polygon the point $\vec{P}'_i$, where $\vec{P}'_i - \vec{P}_i = u_o(\vec{P}_{i+1} - \vec{P}_i)$, $i = 0, 1, \ldots, (m-1)$. These points $\vec{P}'_i$ now define an $(m - 1)$-sided polygon, and can be treated in the same way as the original points $\vec{P}_i$. After m applications of this procedure a single point is left. This is the value of $\vec{B}(u_o)$.

The generalization of the Bernstein-Bézier method to surface approximation is via the Cartesian product of the univariate basis functions. Thus the approximant is

$$\vec{B}(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \psi_i(u)\, \psi_j(v)\, \vec{P}_{i,j} \qquad \ldots\ldots[9]$$

where $\psi_i(u)$ and $\psi_j(v)$ are defined in equation [8].

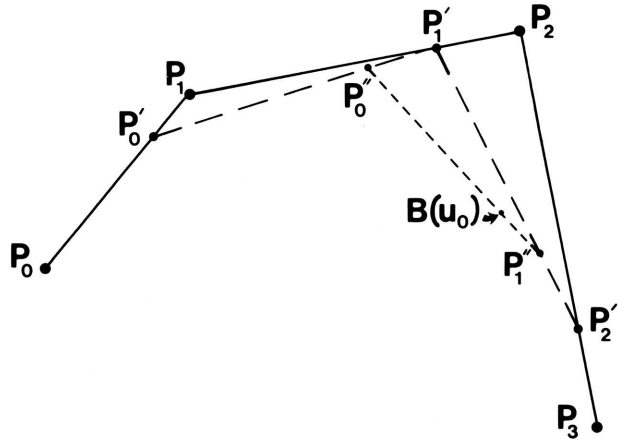


FIGURE 13. The geometric interpretation of a Bézier curve, for m = 3, u_o = 0.75. The original control polygon is $\vec{P}_0 \rightarrow \vec{P}_1 \rightarrow \vec{P}_2 \rightarrow \vec{P}_3$ and the interpolated point is $\vec{B}(u_o)$.

This approximant interpolates the four corner points of the network ($\vec{P}_{0,0}$, $\vec{P}_{0,n}$, $\vec{P}_{m,0}$, $\vec{P}_{m,n}$) (see Fig. 14), and lies within its convex hull. These properties are useful in practice because (1) they enable the surface to be "tied down" at its four corners, and (2) they give a polyhedral box which bounds the surface very closely.

Interpolation and inversion

As an interpolant to the set of points, $\vec{P}_{i,j}$, the Bernstein-Bézier surface is far from satisfactory. This is hardly surprising because Bernstein polynomials are known to have poor convergence properties (Davis, 1963). Where a Bernstein-Bézier surface must interpolate every data point in a network, it is necessary to generate first of all a network of control points which, when used as the $\vec{P}_{i,j}$ in equation [9], will force the surface through the data points.

Determination of this control point network (the control polyhedron) is in principle a straightforward inversion problem. A total of (m × n + m + n - 3) versions of equation [9] are set up, with $\vec{B}(u, v)$ being in each case a node of the data point network and the $\vec{P}_{i,j}$ the control points to be calculated. Arbitrary values are used for the parameters u and v, albeit ones which appear reasonable. Inversion of the square coefficient matrix then gives the values of the control points. It is obviously unnecessary to calculate the corner points of the control polyhedron, because they are identical to the corners of the data point network.

The solution of these equations is clearly not unique--an unlimited number of control polyhedra can be constructed which produce Bézier surfaces satisfying the data point constraints. The dilemma thus exists of finding an adequate parametrization for the surface. There appears to be no obvious solution other than trial-and-error.

Much of the inversion problem can be eased if an interactive computer graphics system is available. A set of control points can then be defined initially, and its shape modified at will until the controlled Bézier surface is adequate. The formal inversion stage is done away with, and consequently the explicit selection of parameter values rendered unnecessary. This is the approach taken at Renault (Forrest, 1972b).

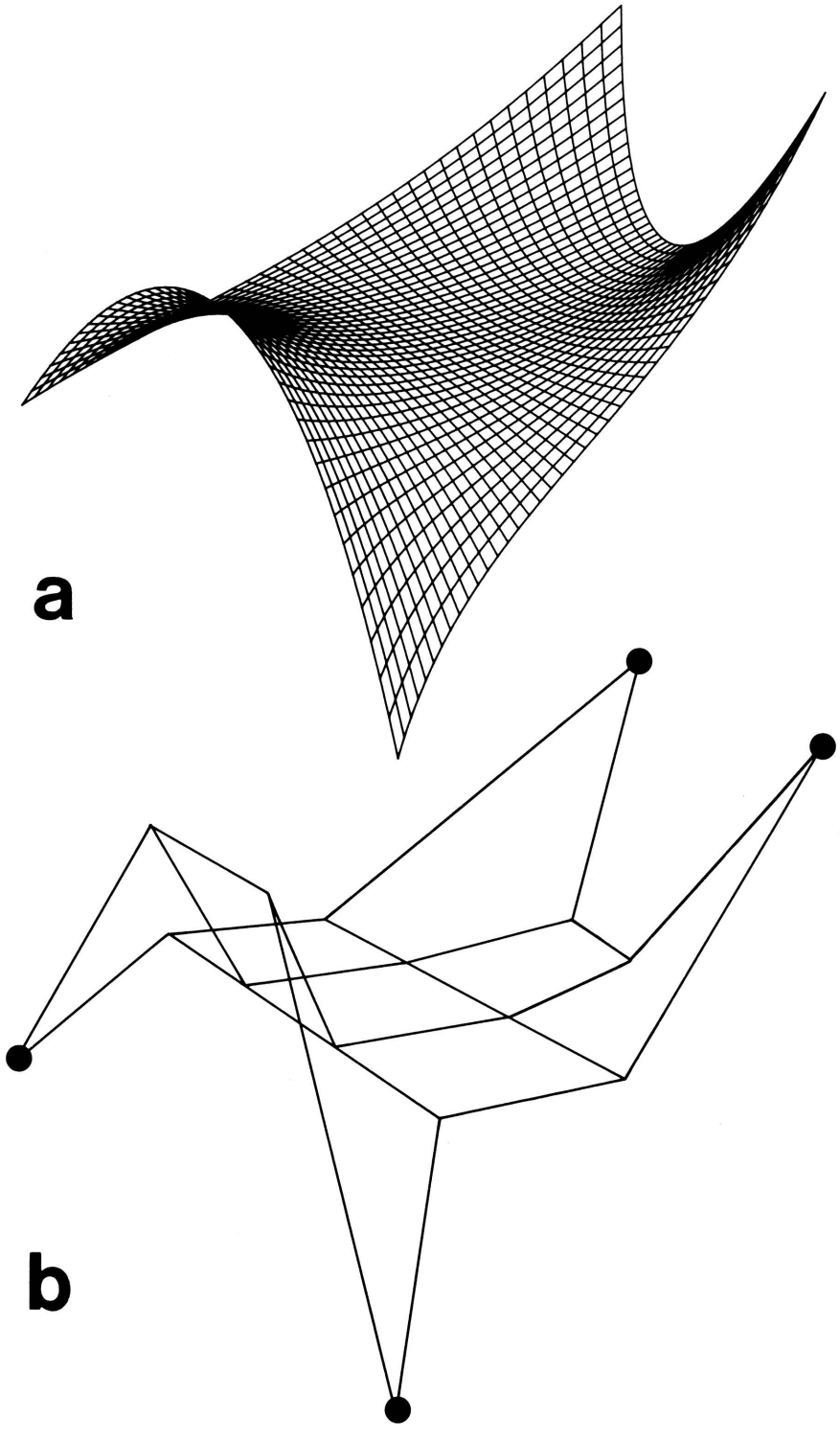


FIGURE 14. Bézier surface (a) generated from (4 by 4) control polyhedron (b). The surface interpolates the four corner points of the polyhedron [shown by solid circles in (b)], but for the sake of clarity has been shown separately.

Alternative formulations

The formulation given here is equivalent (but not identical) to Bézier's original form (Bézier, 1972). This used the relative vectors between adjacent nodes of the network, rather than the absolute point vectors, $\vec{P}_i$ or $\vec{P}_{i,j}$. The reasons for abandoning the original formulation are that it is less elegant, and in practice more susceptible to rounding error problems in computation (Forrest, 1972b). In Bézier's own words (1974, p. 134), it is "complicated and useless."

One special case is of great interest, that for m = n = 3. The basis functions (Fig. 11) are

$$\psi_0(u) = (1 - u)^3$$

$$\psi_1(u) = 3u(1 - u)^2$$

$$\psi_2(u) = 3u^2(1 - u) = \psi_1(1 - u)$$

$$\psi_3(u) = u^3 \qquad = \psi_0(1 - u)$$

The $\psi(v)$ are similarly defined. Equation [9] is thus a bicubic in u and v. In matrix form it is

$$\vec{B}(u, v) = \begin{bmatrix}\psi_0(u) & \psi_1(u) & \psi_2(u) & \psi_3(u)\end{bmatrix}\begin{bmatrix}\vec{P}_{0,0} & \vec{P}_{0,1} & \vec{P}_{0,2} & \vec{P}_{0,3} \\ \vec{P}_{1,0} & \vec{P}_{1,1} & \vec{P}_{1,2} & \vec{P}_{1,3} \\ \vec{P}_{2,0} & \vec{P}_{2,1} & \vec{P}_{2,2} & \vec{P}_{2,3} \\ \vec{P}_{3,0} & \vec{P}_{3,1} & \vec{P}_{3,2} & \vec{P}_{3,3}\end{bmatrix}\begin{bmatrix}\psi_0(v) \\ \psi_1(v) \\ \psi_2(v) \\ \psi_3(v)\end{bmatrix} \qquad .....[10]$$

It is instructive to see how the control points, $\vec{P}_{i,j}$, relate in this case to other features of the surface, particularly to the derivative conditions at the four corners of the network--we know, of course, that each corner is interpolated in position. The square matrix of equation [10] can be re-expressed solely in terms of position vectors and partial derivatives at the corner points (cf. Sabin, 1969b). To ensure homogeneity of notation with later parts of this paper, the corners are referred to by their parametric coordinates for a unit-square cell, i.e., $\vec{P}_{0,0}$ is $\vec{P}(0,0)$, $\vec{P}_{0,3}$ is $\vec{P}(0,1)$, $\vec{P}_{3,0}$ is $\vec{P}(1,0)$, and $\vec{P}_{3,3}$ is $\vec{P}(1,1)$.

The matrix is:

$$\begin{bmatrix} \vec{P}(0,0) & \frac{\vec{P}_v(0,0)}{3} + \vec{P}(0,0) & \frac{-\vec{P}_v(0,1)}{3} + \vec{P}(0,1) & \vec{P}(0,1) \\ \frac{\vec{P}_u(0,0)}{3} + \vec{P}(0,0) & \frac{\vec{P}_{uv}(0,0)}{9} + \frac{\vec{P}_u(0,0)}{3} + \frac{\vec{P}_v(0,0)}{3} + \vec{P}(0,0) & \frac{-\vec{P}_{uv}(0,1)}{9} + \frac{\vec{P}_u(0,1)}{3} - \frac{\vec{P}_v(0,1)}{3} + \vec{P}(0,1) & \frac{\vec{P}_u(0,1)}{3} + \vec{P}(0,1) \\ \frac{-\vec{P}_u(1,0)}{3} + \vec{P}(1,0) & \frac{-\vec{P}_{uv}(1,0)}{9} - \frac{\vec{P}_u(1,0)}{3} + \frac{\vec{P}_v(1,0)}{3} + \vec{P}(1,0) & \frac{\vec{P}_{uv}(1,1)}{9} - \frac{\vec{P}_u(1,1)}{3} - \frac{\vec{P}_v(1,1)}{3} + \vec{P}(1,1) & \frac{-\vec{P}_u(1,1)}{3} + \vec{P}(1,1) \\ \vec{P}(1,0) & \frac{\vec{P}_v(1,0)}{3} + \vec{P}(1,0) & \frac{-\vec{P}_v(1,1)}{3} + \vec{P}(1,1) & \vec{P}(1,1) \end{bmatrix}$$

where $\vec{P}(h,k)$ is the position vector at $u = h$, $v = k$,

$$\vec{P}_u(h,k) \text{ is } \frac{\partial \vec{P}}{\partial u}\Big|_{h,k} ,$$

$$\vec{P}_v(h,k) \text{ is } \frac{\partial \vec{P}}{\partial v}\Big|_{h,k} ,$$

$$\vec{P}_{uv}(h,k) \text{ is } \frac{\partial^2 \vec{P}}{\partial u \cdot \partial v}\Big|_{h,k} .$$

Example

Control polyhedra were generated for the Cherokee data set (Appendix 3). The values for u and v for each control point were selected by either of two methods:

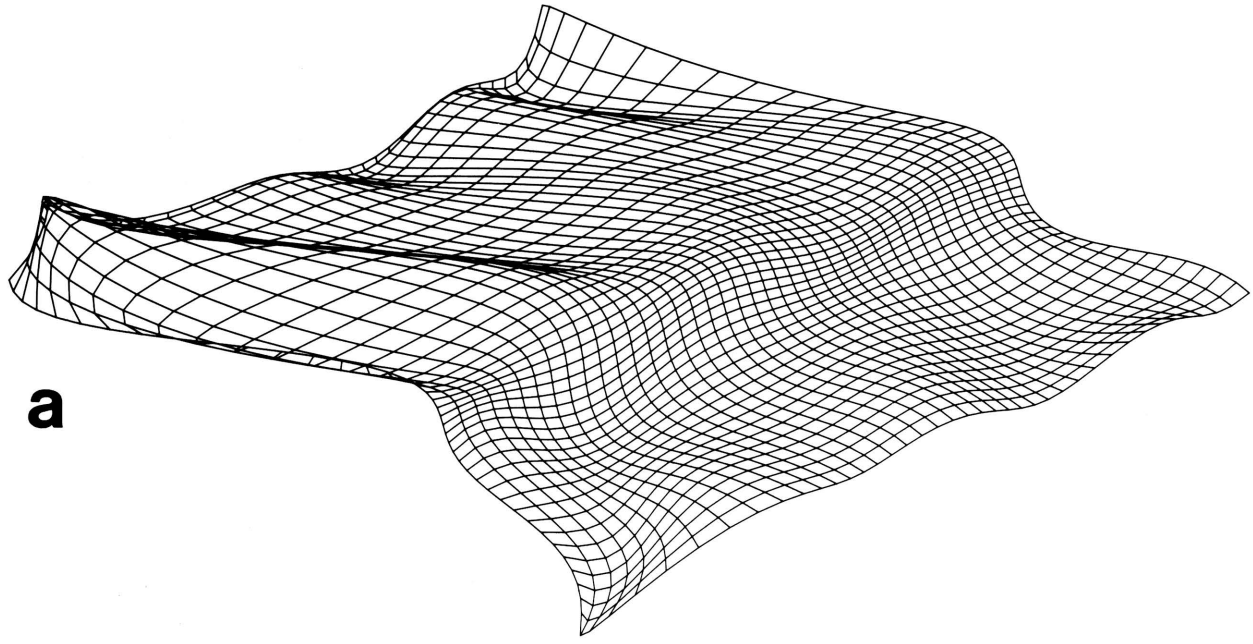
(1) for control point (i, j), $u_{i,j} = i/m$ and $v_{i,j} = j/n$;

(2) for control point (i, j), $u_{i,j} = d^{(j)}_{0\to i} / d^{(j)}_{0\to m}$ and $v_{i,j} = d^{(i)}_{0\to j} / d^{(i)}_{0\to n}$

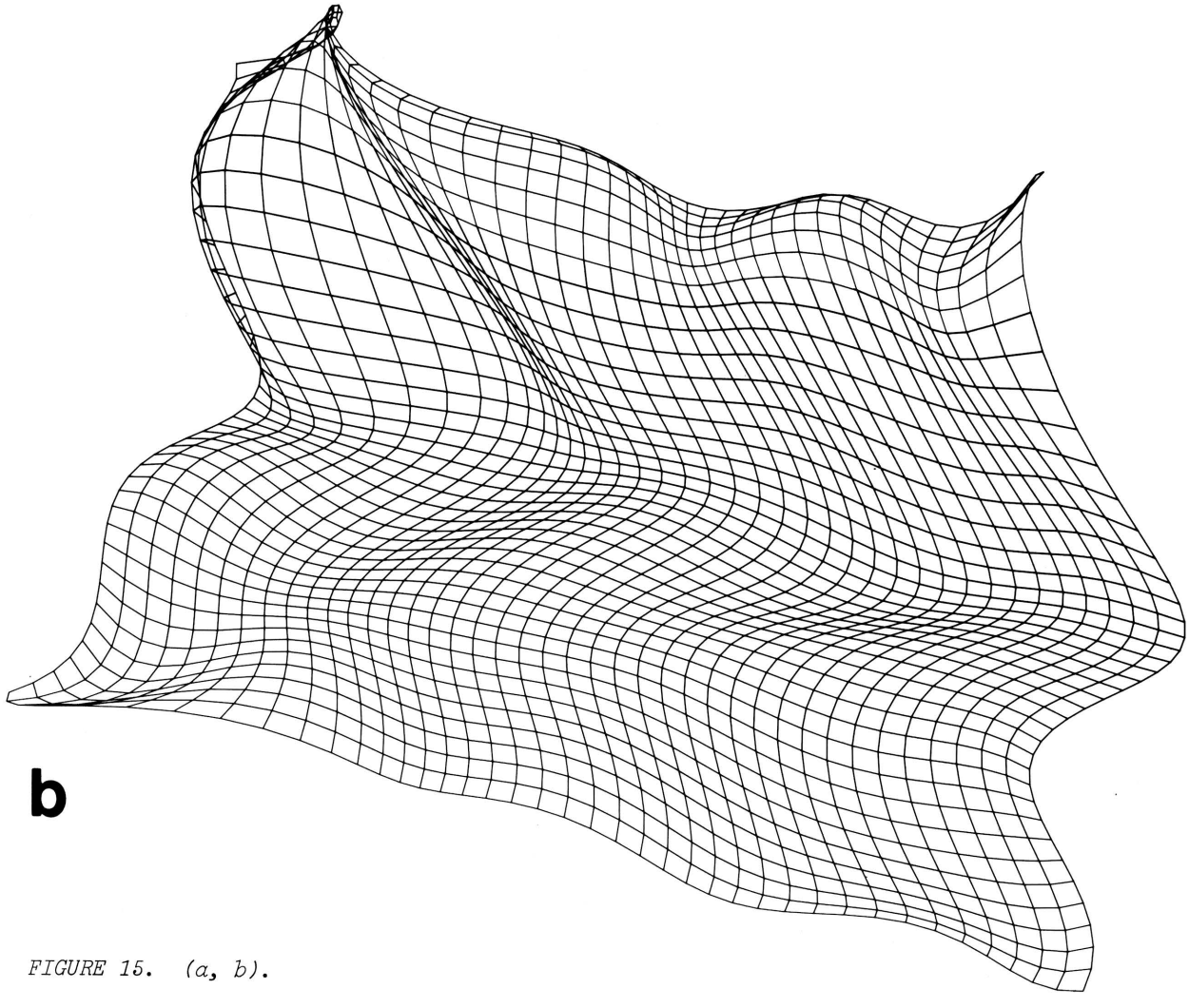
where $d^{(j)}_{h\to k}$ $\left[= \sum_{\ell=h}^{k-1} d^{(j)}_{\ell\to\ell+1}\right]$ is the Euclidean distance from data point $\vec{P}_{h,j}$ to data point $\vec{P}_{k,j}$ measured along the indexing line j [$d^{(i)}$ is similarly defined for the indexing line i].

No attempt was made at any type of optimization; furthermore, the inversion was carried out non-interactively. Neither parametrization is entirely satisfactory, although the former is better, at least for this data set. The Bézier surfaces generated from these polyhedra behave wildly at their edges, although elsewhere they are quite acceptable. The central areas of the former surfaces ($0.17 < u < 0.83$, and $0.22 < v < 0.78$) are shown in Figure 15. It is evident that Bézier surfaces, when implemented in this manner, are not particularly usable.

FIGURE 15. Bézier surface representations of the Cherokee data set, shown in perspective view: center of projection at z = -50. Only the central areas of each surface are shown ($0.17 < u < 0.83$, $0.22 < v < 0.78$). (a) Top surface, $\phi = -100°$, $\theta = -50°$. (b) Top surface, $\phi = -160°$, $\theta = -50°$. (c) Bottom surface, $\phi = -100°$, $\theta = -50°$. (d) Bottom surface, $\phi = -160°$, $\theta = -50°$.



a



b

FIGURE 15. (a, b).

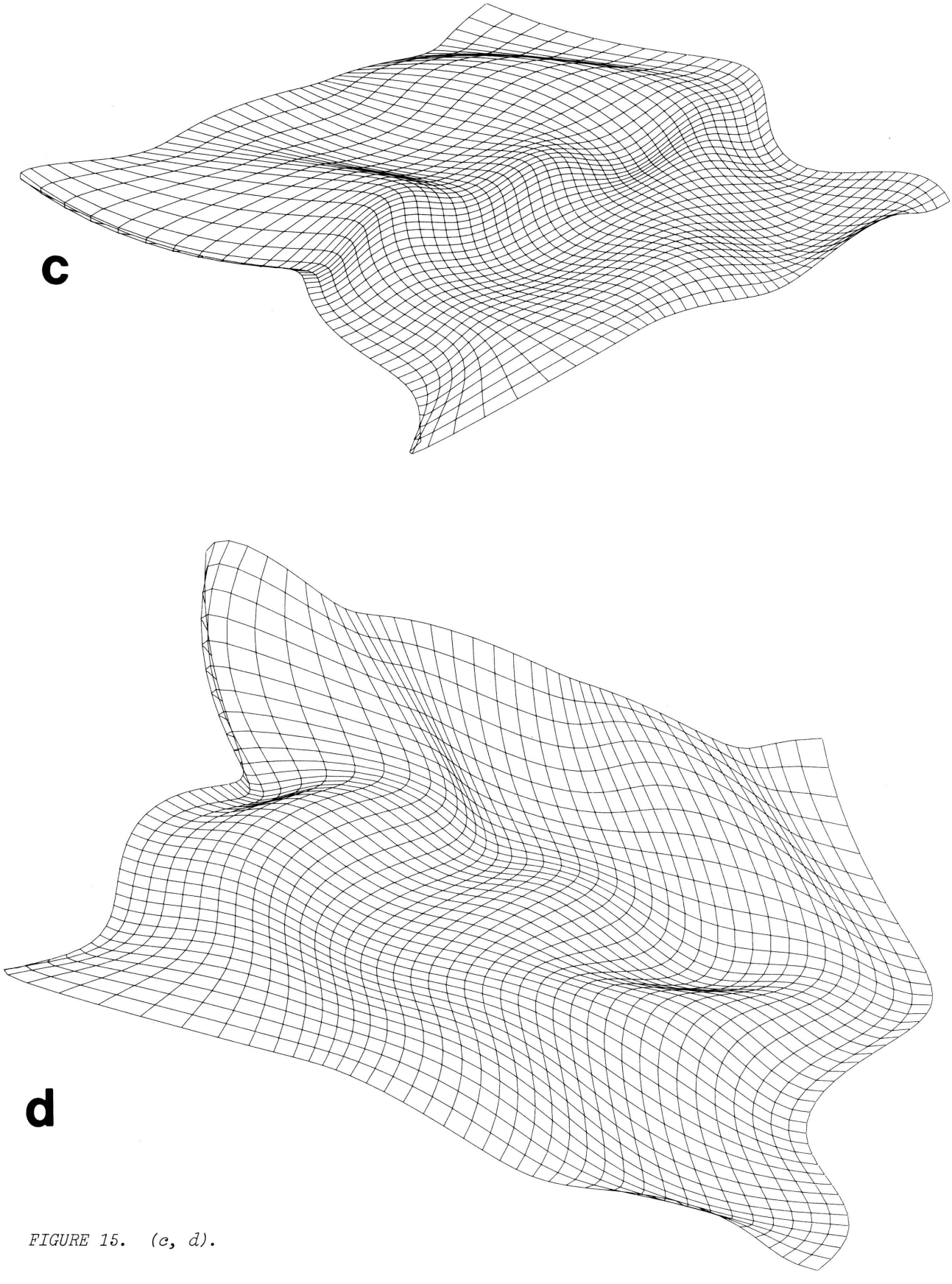


FIGURE 15. (c, d).

Summary

Bézier surfaces have now been used for long enough that most of their advantages and disadvantages are known. In particular, the original Renault system has operated successfully for several years. Features which should be stressed about the method are:

(1) The surfaces are based on a mathematical formulation which is both simple to understand and computationally efficient.

(2) Once the inversion problem is overcome, the surfaces are tractable, well-behaved interpolants. Non-interactive inversion, however, may prove troublesome.

(3) The method is global, i.e. moving one control point alters the whole surface.

(4) The degree of the approximant is controlled by the dimensions of the control polyhedron, and thus the continuity of the surface changes with the number of points defining it, much as in the simpler surface-fitting methods described earlier. This inability to maintain a prescribed continuity is one of the principal drawbacks to Bézier's method.

(5) Although the control polyhedron does bound the surface closely, the surface shape is often only a poor representation of that of the polyhedron (Gordon and Riesenfeld, 1974a). Extensions of the method have been developed to counter this, for example by re-parametrization of the surface, or by use of conditional probability density functions instead of the true Bernstein basis (Gordon and Riesenfeld, 1974a). Even with these added features, however, the Bézier surface is not flexible enough for many purposes. A better surface can be devised, based on the appropriate locally based extension of the Bernstein basis. This is the B-spline basis (Gordon and Riesenfeld, 1974b, p. 103), and underlies the next method to be described.

B-Spline Curves and Surfaces

Mathematical splines are a versatile class of functions originally developed for use in data smoothing (Schoenberg, 1946). Subsequently they have found wide application in interpolation and approximation. For practical purposes we can regard a mathematical spline as a set of functions of constant degree which reproduces certain of the properties of a physical spline--a thin elastic strip used by draftsmen to draw smooth curves through sets of coplanar points. Each function is valid only between one pair of adjacent data points, and neighboring functions are constructed so that they merge together with some specified degree of continuity. (In the spline of degree three, which most closely resembles the physical analogue, the functions are cubic polynomials and give second derivative continuity.) The functions can be constructed using either global or local bases. We return to the global type of spline function later in this paper, and for the present concentrate on how to construct locally based spline approximants which can be used both for curve and surface design. These are termed B-splines.

B-splines were first introduced into the field of computer-aided design by Riesenfeld (1973); see also Gordon and Riesenfeld (1974b). A closely related treatment is due to Clark (1974, 1976a).

Fundamentals of B-spline approximation

It was pointed out earlier that B-spline approximation is the appropriate locally based extension of Bernstein approximation. For simplicity we introduce B-splines not from the strictly mathematical viewpoint (see for example de Boor, 1972; Gordon and Riesenfeld, 1974b), but use instead an approach similar to our treatment of the Bernstein-Bézier method. Again the curve-fitting case is considered first, and later generalized to cover surfaces.

Consider, as before, a string of points, $\vec{P}_i$, where i = 0, 1, ..., m. It is required to approximate this by a piecewise curve having specified continuity between its segments. By analogy with equation [7], each segment is written as

$$\vec{T}_r(u) = \sum_{i=0}^{k} \xi_i(u) \cdot \vec{P}_{r+i-1} \qquad \text{.....[11]}$$

$$\text{for } r = 1, 2, \ldots, (m-k+1)$$

where $\xi_i(u)$ are the B-spline basis functions,

u is a parameter constrained to lie in the range $0 \rightarrow 1$, and

k is the degree of the basis functions.

The curve is composed of (m-k+1) segments, and the points at which adjacent segments join are termed the knots of the spline function. Because the knots have uniform separation in the u-parameter space throughout the whole curve, the approximant is known as a uniform B-spline.

The basis functions are polynomials in u, and their coefficients can be simply determined. As an example, consider the cubic B-spline (k = 3) and let ρ_{1-8} be the required coefficients. Then,

$$\xi_0(u) = \rho_1(1 - u)^3 + \rho_2(1 - u)^2 + \rho_3(1 - u) + \rho_4$$

$$\xi_1(u) = \rho_5(1 - u)^3 + \rho_6(1 - u)^2 + \rho_7(1 - u) + \rho_8$$

By symmetry

$$\xi_2(u) = \rho_5 u^3 + \rho_6 u^2 + \rho_7 u + \rho_8$$

$$\xi_3(u) = \rho_1 u^3 + \rho_2 u^2 + \rho_3 u + \rho_4$$

In this case adjacent segments are required to join with continuity up to the second derivative. Hence for the join between segments r and (r + 1),

$$\vec{T}_{r+1}(0) = \vec{T}_r(1)$$

$$\vec{T}'_{r+1}(0) = \vec{T}'_r(1)$$

$$\vec{T}''_{r+1}(0) = \vec{T}''_r(1)$$

These stipulations require that the basis functions satisfy the following conditions (Clark, 1974):

$$\xi_i(0) = \xi_{i+1}(1)$$

$$\xi'_i(0) = \xi'_{i+1}(1) \qquad \text{for } i = 0, 1, \ldots, (k - 1)$$

$$\xi''_i(0) = \xi''_{i+1}(1)$$

and

$$\xi_0(1) = \xi'_0(1) = \xi''_0(1) = 0$$

$$\xi_k(0) = \xi'_k(0) = \xi''_k(0) = 0$$

Furthermore, the basis functions must be normalized:

$$\sum_{i=0}^{k} \xi_i(u) = 1 \qquad \text{for all } u \ .$$

These conditions lead to the following equations:

$$\begin{aligned}
\rho_1 + \rho_2 + \rho_3 + \rho_4 - \rho_8 &= 0 \\
3\rho_5 + 2\rho_6 + \rho_7 &= 0 \\
3\rho_1 + 2\rho_2 + \rho_3 - \rho_7 &= 0 \\
3\rho_1 + \rho_2 - \rho_6 &= 0 \\
\rho_2 = \rho_3 = \rho_4 &= 0 \\
\rho_1 + \rho_2 + \rho_3 + 2\rho_4 + \rho_5 + \rho_6 + \rho_7 + 2\rho_8 &= 1
\end{aligned}$$

Solution of these equations gives values for the coefficients, and the basis functions are found to be

$$\begin{aligned}
\xi_0(u) &= (1 - u)^3 / 6 \\
\xi_1(u) &= [-3(1 - u)^3 + 3(1 - u)^2 + 3(1 - u) + 1] / 6 \\
\xi_2(u) &= [-3u^3 + 3u^2 + 3u + 1] / 6 \\
\xi_3(u) &= u^3 / 6
\end{aligned} \qquad \text{.....[12]}$$

These are illustrated in Figure 16. If only positional and tangent continuity are required between segments, the cubic basis functions are replaced by quadratics (k = 2). In this case the functions (Fig. 17) are

$$\begin{aligned}
\xi_0(u) &= (1 - u)^2 / 2 \\
\xi_1(u) &= (-2u^2 + 2u + 1) / 2 \\
\xi_2(u) &= u^2 / 2
\end{aligned}$$

Examples of cubic and quadratic B-spline curves are given in Figure 18 for two control polygons: the curves are respectively of class C^2 and C^1 everywhere along their length.

B-spline approximation shares with Bernstein approximation the variation diminishing and convex hull properties mentioned earlier. It is clear though that the B-spline approximant will not, in general, interpolate any of its control points. The only way to achieve interpolation is to use repeated control points. Coons (1974) has described some of the interesting features which can be produced in a B-spline curve by this technique.

As an example of the use of repeated control points, consider a single cubic B-spline segment, with control points $\vec{P}_1$, $\vec{P}_2$, $\vec{P}_3$, $\vec{P}_4$ (Fig. 19). When $\vec{P}_1 \neq \vec{P}_2 = \vec{P}_3 = \vec{P}_4$, i.e. when three adjacent control points are coincident

$$\vec{T}(1) = \vec{P}_2$$

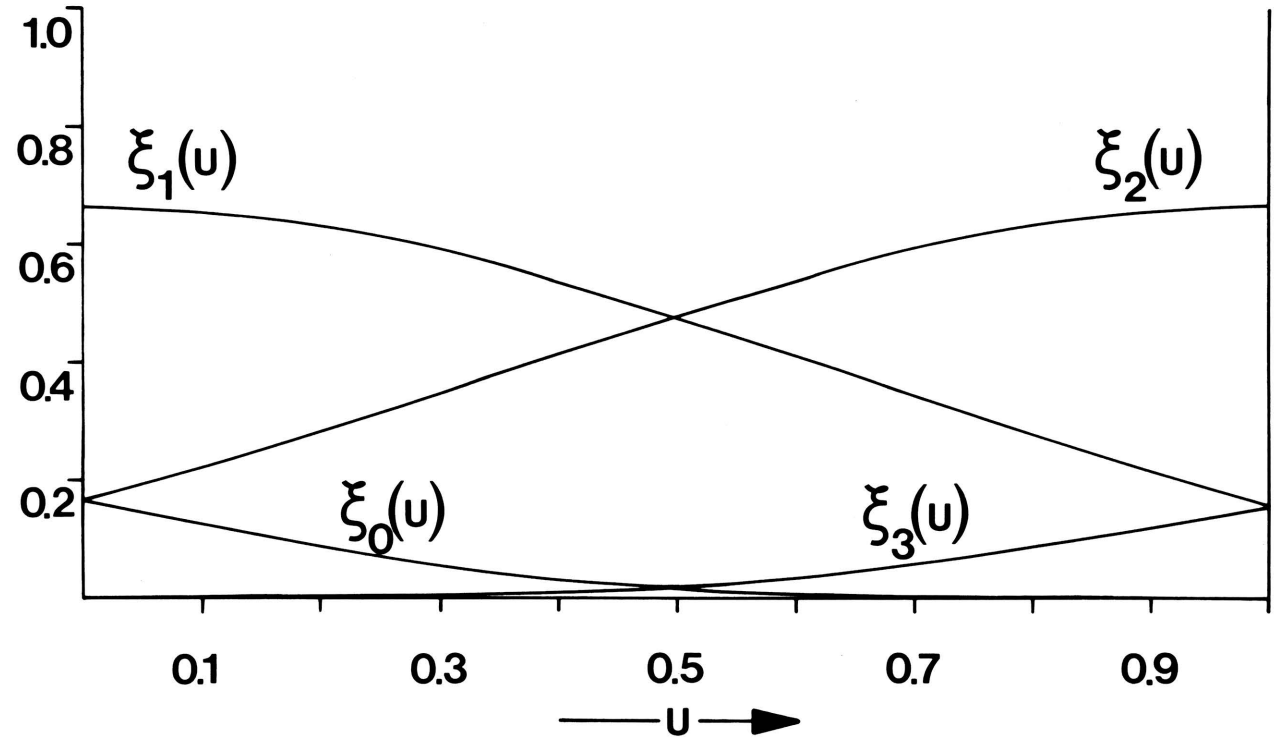


FIGURE 16. Cubic B-spline basis functions.

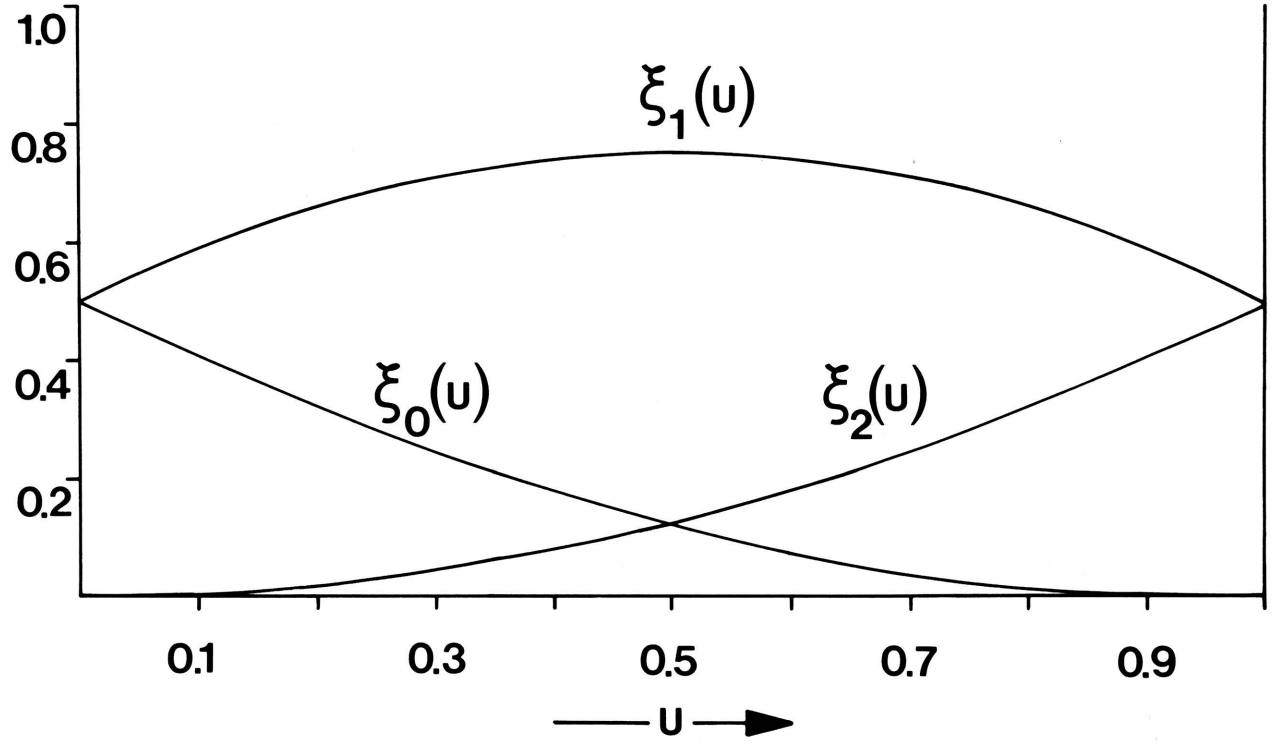


FIGURE 17. Quadratic B-spline basis functions.

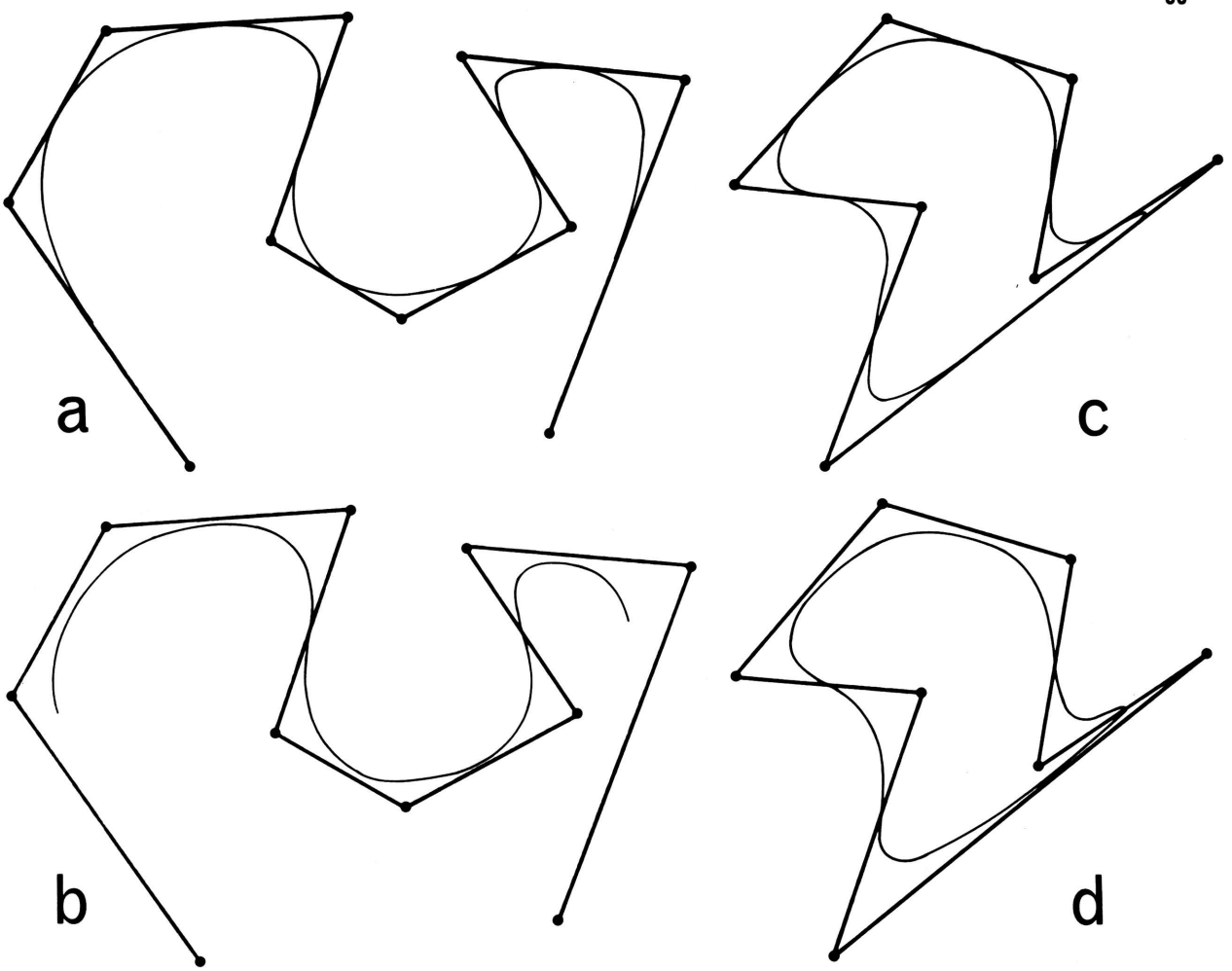


FIGURE 18. Examples of B-spline curves. The control polygons (heavy lines) join the control points (circled). The light lines are the interpolated B-spline curves. (a) Quadratic B-spline, open polygon. (b) Cubic B-spline, open polygon. (c) Quadratic B-spline, closed polygon. (d) Cubic B-spline, closed polygon.

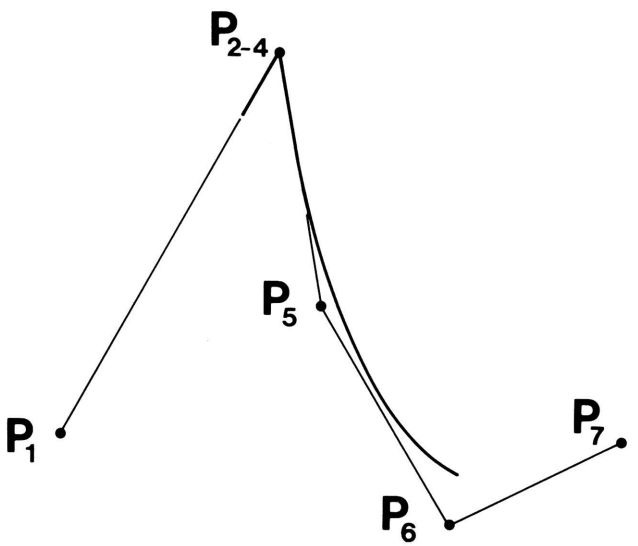


FIGURE 19. Cubic B-spline with repeated control points. Light line is the control polygon, heavy line is the B-spline curve.

Thus the curve passes through the repeated control point, in this case for the parameter u = 1. (The corresponding condition for a quadratic B-spline segment occurs when two adjacent control points are coincident.) If another segment is adjoined at the repeated control point, C^2 continuity is maintained throughout, although it is clear that the curve has a distinct cusp and hence is discontinuous in slope (Fig. 19). This paradox occurs because the tangent vector vanishes at the repeated control point. Interpolation in a B-spline curve is thus obtained at the price of slope discontinuity. It is worth noting that the ability to produce slope discontinuities at specified points is especially important in a practical CAD system, although perhaps less valuable for our purposes.

B-spline surfaces

Generalization from curves to surfaces is via the Cartesian product, as for the Bézier method. Each surface patch is

$$\vec{T}_{r,s}(u, v) = \sum_{i=0}^{k} \sum_{j=0}^{k} \xi_i(u)\, \xi_j(v)\, \vec{P}_{r+i-1,s+j-1}$$

$$\text{for } r = 1, 2, \ldots, (m-k+1) \quad \text{and} \quad s = 1, 2, \ldots, (n-k+1)$$

where $\xi_i(u)$ and $\xi_j(v)$ are the B-spline basis functions (see equation [12] for the bicubic case),
u and v are parameters lying between 0 and 1,
(m + 1 by n + 1) is the size of the control polyhedron, and
k is the degree of the basis functions.

For a bicubic patch (k = 3), a total of 16 points control its shape; for a biquadratic patch (k = 2), only nine (Fig. 20). Surface shape at any point is therefore completely insensitive to movement of control points outside a given range (in the parametric space). The movement of single points never, of course, affects the degree of continuity

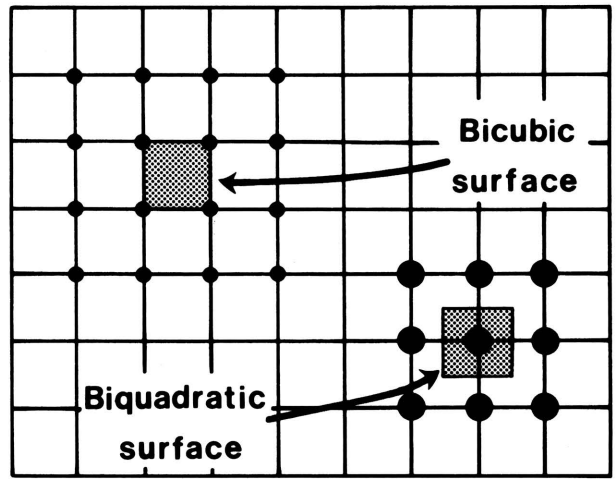


FIGURE 20. Network of control points for B-spline surfaces. The bicubic surface is controlled by 16 points (small circles); the biquadratic surface is controlled by 9 points (large circles).

between adjacent patches. These characteristics make B-spline surfaces very attractive both for surface design and surface representation applications, and numerous systems have been designed which use this approach (see for example Clark, 1974, 1976a).

Perhaps the greatest drawback to the use of B-spline surfaces is that, even more than in the Bézier system, the final surface owes its shape to control points which it does not interpolate. Although the concept of a control polyhedron is simple, and the modification of a surface by movement of individual control points is straightforward, it has been found in practice that interpolation would be a most desirable condition (Clark, 1976a). A second drawback is that, at least for the Cartesian product formulation, B-spline surfaces cannot be blended into arbitrarily shaped boundaries. We shall see in a later section how the use of B-spline space curves in a Coons surface formulation can remove both of these disadvantages.

The inversion problem

Because in general the B-spline approximant is not a satisfactory interpolant, an inversion procedure is required, as was the case for Bézier curves and surfaces. The problem is much simpler for B-spline curves, however, for two reasons. Firstly, they are piecewise curves of relatively low degree and hence the order of equations to be solved is correspondingly lower. Secondly, because only uniform B-splines are being considered, the computation is drastically reduced in complexity.

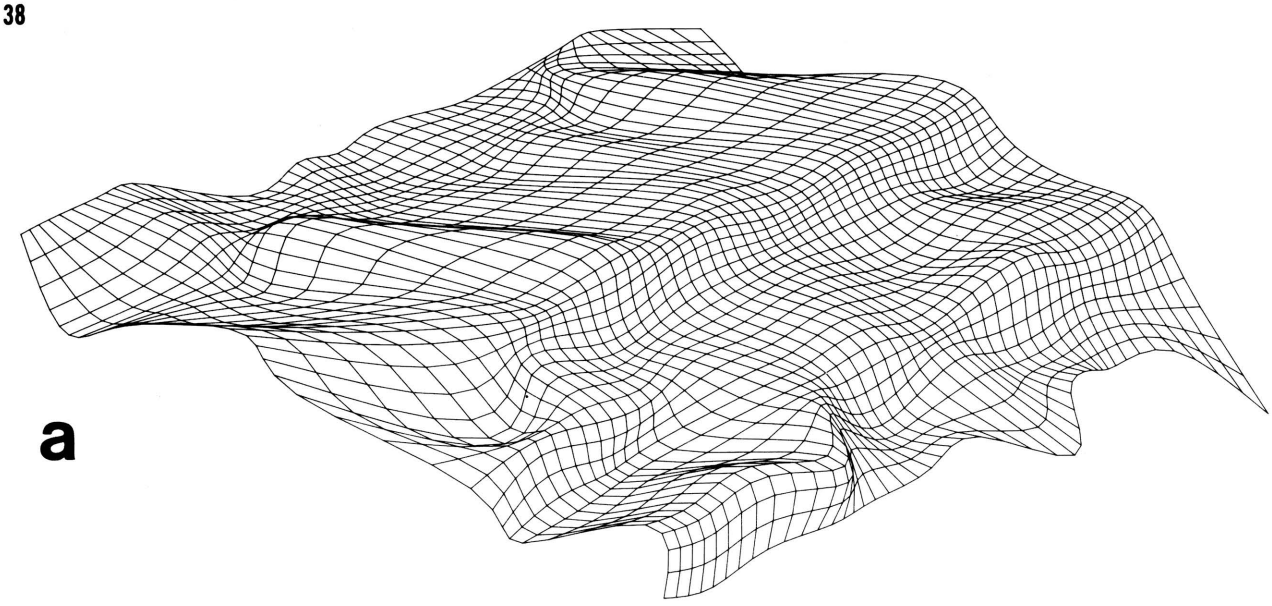
The inversion procedure for B-spline curves is carried out as follows (a more detailed description is furnished by Wu, *et al.*, 1977): A total of (m-k+1) versions of equation [11] are set up, one for each segment of the curve. The $\vec{T}(u)$ are the data points and the $\vec{P}$ are the control points to be calculated. In each case the parameter u is set to zero. An additional version of equation [11] is also set up for the (m-k+1)'th segment, using u = 1. To account for the remaining (k - 1) degrees of freedom, it is necessary to make some stipulations about the form of the two end segments. Several approaches are available, and we shall consider some of them at greater length in the description of global-basis spline interpolation. For now, two approaches are of use. The first is to generate artificial control points, $\vec{P}_{-1}$ and $\vec{P}_{m+1}$, at the ends of the curve in such a way as to ensure that the curvature at points $\vec{T}_1(0)$ and $\vec{T}_{m-k+1}(1)$ is zero (Wu, *et al.*, 1977). The second approach is to use repeated control points at the ends, thus tying the curve down in a similar manner to a Bézier curve.

The inversion procedure for B-spline surfaces can be derived in a similar way. As was the case for Bézier curves, the inversion problem is greatly eased if it can be carried out on an interactive computer graphics system.

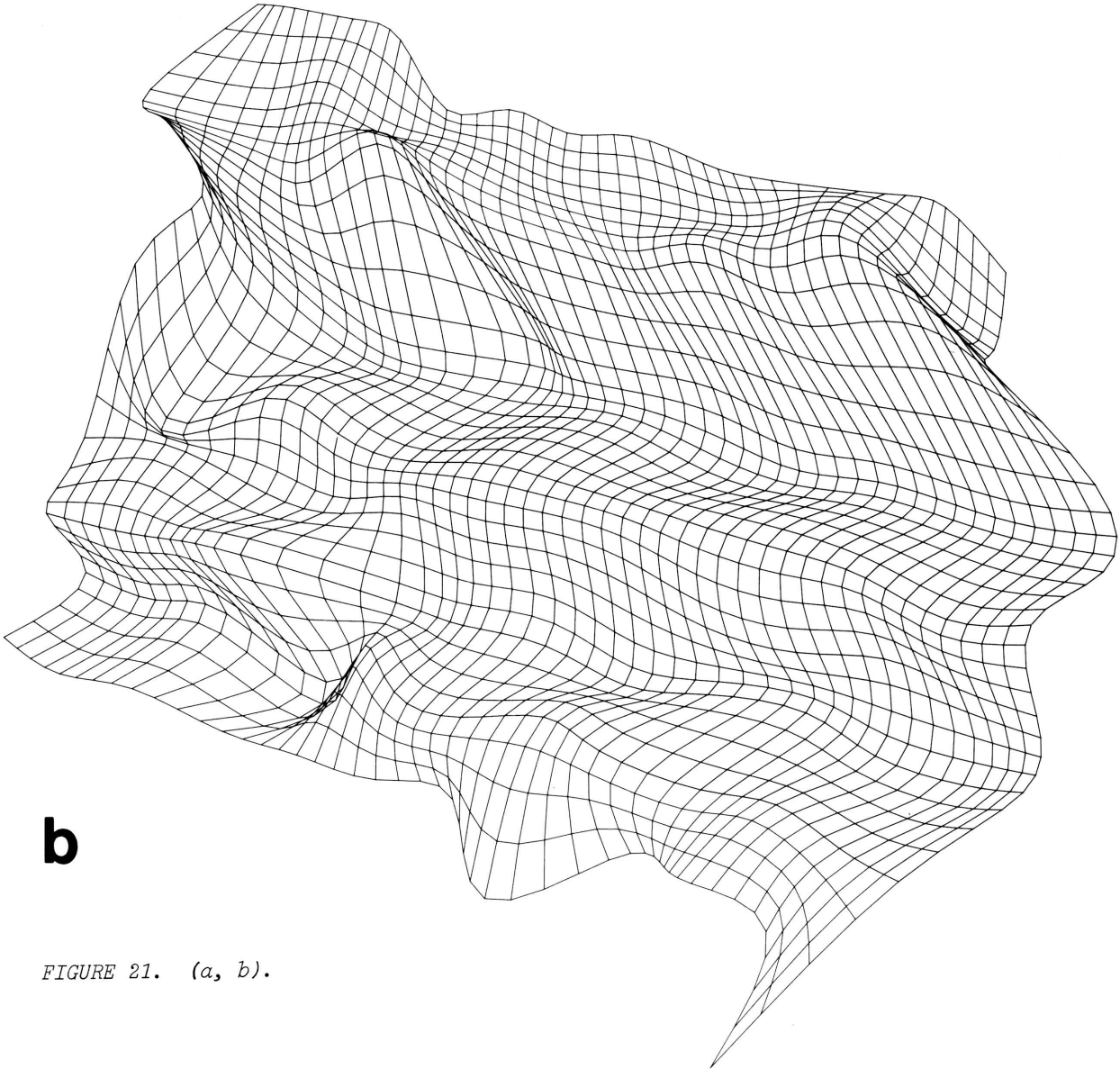
Example

Control polyhedra were generated for bicubic B-spline representations of the Cherokee data set: the control point coordinates are given in Appendix 4. The actual B-spline surfaces are shown in Figure 21, and they are clearly acceptable representations of the data.

FIGURE 21. B-spline surface representations of the Cherokee data set, shown in perspective view: center of projection at z = -50. (a) Top surface, $\phi = -100°$, $\theta = -50°$. (b) Top surface, $\phi = -160°$, $\theta = -50°$. (c) Bottom surface, $\phi = -100°$, $\theta = -50°$. (d) Bottom surface, $\phi = -160°$, $\theta = -50°$.

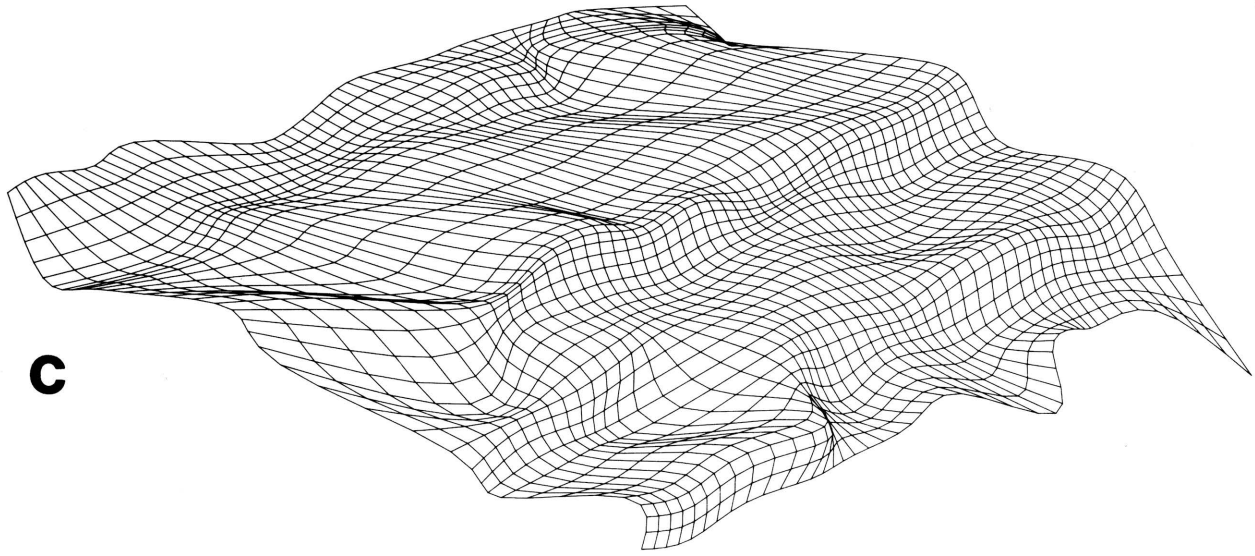


a

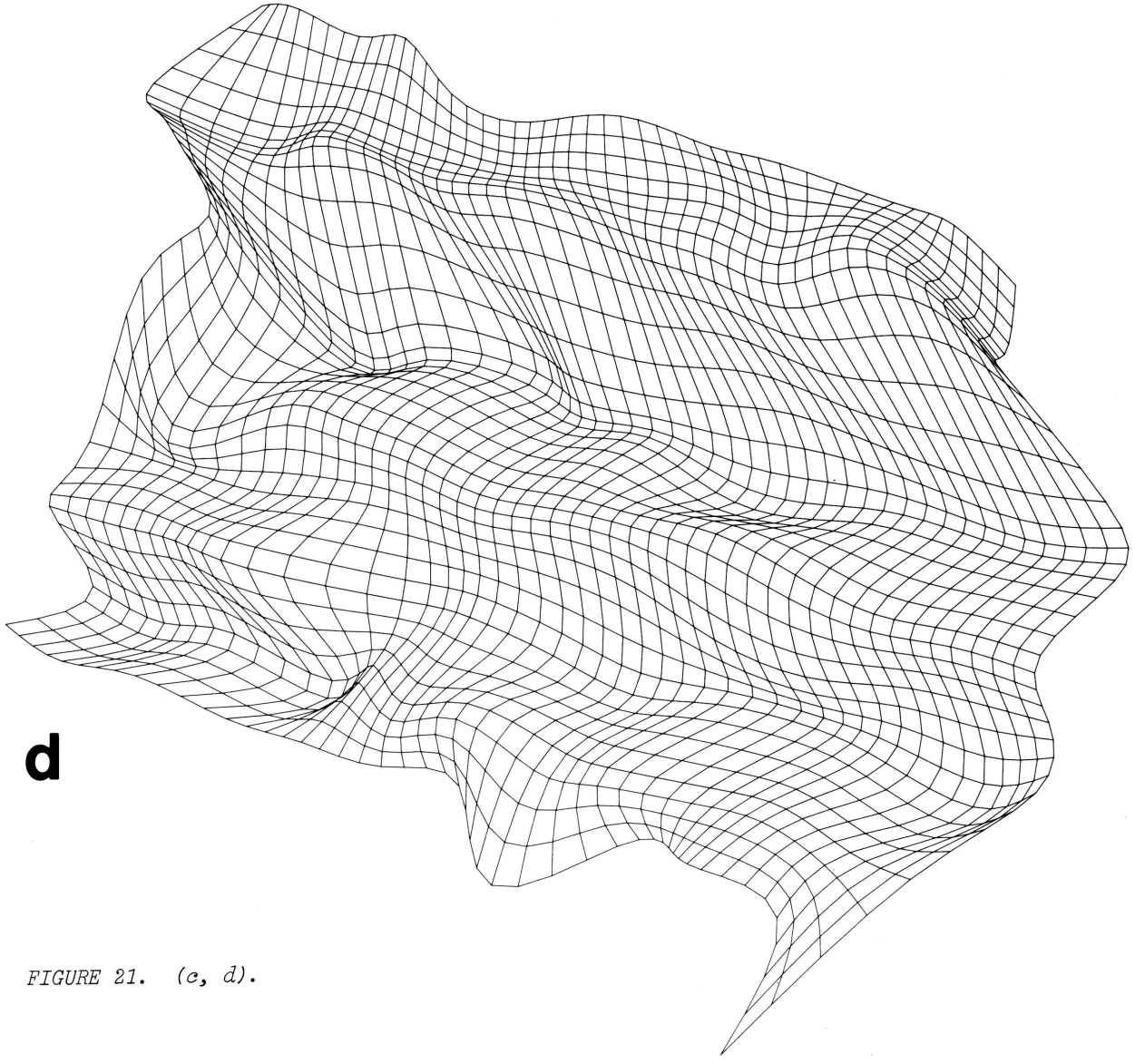


b

FIGURE 21. (a, b).



c



d

FIGURE 21. (c, d).

Summary

The advantages and disadvantages of the B-spline approach to surface representation are as follows:

(1) It uses a local basis formulation.

(2) It is computationally efficient. Clark (1976a) demonstrates how the method can be coded for efficient real-time display.

(3) Pre-specified continuity is always maintained over all the surface. This is an intrinsic feature of the B-spline method.

(4) Control points are not interpolated and hence inversion is necessary. In practice, interpolation would be desirable.

(5) Matching to arbitrarily shaped boundaries is not always possible.

Overall, the B-spline approach has much to recommend it, and it forms the basis of a geological surface modelling system implemented by the author, and used to produce the examples of manipulation given in the second part of this paper. For now, however, another related approach will be examined--the use of globally based spline surfaces.

Global-Basis Spline Interpolation

In the last section we dealt with a class of locally based approximating splines (B-splines). Interpolation by splines is also possible, using either local or global bases. The use of global splines (Schoenberg, 1946) has long been common for surface representation. Some of the earliest work is that of Birkhoff and Garabedian (1960) and de Boor (1962). Spline surfaces have been used for surface fitting in geology (Anderson, 1971; Whitten and Koelling, 1973, 1975), and in geophysics (Bhattacharyya, 1969; Holroyd and Bhattacharyya, 1970).

Many descriptions of global-basis spline interpolation have successfully obscured its essential simplicity, both in theory and in application. There has been, furthermore, an unfortunate tendency to segregate it from many of the other CAD techniques, to which it is closely related: for a noteworthy exception, see Ahlberg (1974).

Some fundamentals of spline interpolation

Much of this treatment follows de Boor (1962), although the notation is in many cases reversed. Once again the curve fitting problem is studied before the generalization to surfaces. Consider the set of points, P_i, where $i = 0, 1, \ldots, m$. It is required to define a piecewise curve which interpolates each of these points (termed knots), and which is of class C^2 (continuity is maintained up to second derivative). For the curve fitting problem we write each point as (u_i, x_i), and stipulate that the values of the u_i be distinct and monotonically increasing with i. Thus the technique is restricted initially to the univariate Hermitian interpolation of a single-valued function.

In order to maintain C^2 continuity, each of the functions must be at least a cubic polynomial. Let the function valid between P_r and P_{r+1} be

$$x(u) = \sum_{i=0}^{3} \alpha_i (u - u_r)^i \qquad \ldots\ldots[13]$$

This may be rewritten in terms of the values of the function and its tangent at the end-points of the interval. Equation [13] is then

$$x(u) = x_r + x'_r(u - u_r) + \left[3 \frac{(x_{r+1} - x_r)}{(u_{r+1} - u_r)^2} - \frac{(x'_{r+1} + 2x'_r)}{(u_{r+1} - u_r)} \right] (u - u_r)^2 +$$

$$\left[-2 \frac{(x_{r+1} - x_r)}{(u_{r+1} - u_r)^3} + \frac{(x'_{r+1} + x'_r)}{(u_{r+1} - u_r)^2} \right] (u - u_r)^3 \qquad .....[14]$$

where x_r and x_{r+1} are the function values at P_r and P_{r+1} respectively, and
x'_r and x'_{r+1} are the tangents.

Now x_r and x_{r+1} are known for all r, and so x(u) can be calculated if the derivatives at each of the (m + 1) points can first of all be found. This is done by equating second derivatives at each of the (m - 1) interior knots for the pairs of functions which join there. For knot P_r, these are the r'th and (r+1)'th functions. A set of (m - 1) simultaneous linear equations is obtained:

$$\Delta u_r \, x'_{r-1} + 2 (\Delta u_r + \Delta u_{r-1}) x'_r + \Delta u_{r-1} \, x'_{r+1} = 3 \left[\frac{\Delta u_{r-1}}{\Delta u_r} (x_{r+1} - x_r) + \frac{\Delta u_r}{\Delta u_{r-1}} (x_r - x_{r-1}) \right]$$

.....[15]

where $\Delta u_r = u_{r+1} - u_r$, and
$\Delta u_{r-1} = u_r - u_{r-1}$.

If the tangents at the two end-points are known (x'_0 and x'_m), then the equations can be simply solved. The coefficient matrix is tridiagonal and strictly diagonally dominant, and consequently non-singular. The equations are therefore independent and the solution for x'_r unique (de Boor, 1962). There is thus but one piecewise cubic of class C^2 which interpolates the data points.

In a substantial number of cases the end-point tangents are not known, and other end-conditions must be used. This is the same problem discussed earlier in the context of B-spline inversion. One common end-condition is to stipulate that the end-point curvatures be zero, i.e.

$$x''_0 = x''_m = 0 \ .$$

The two equations which follow from this are

$$2 \Delta u_0 \, x'_0 + \Delta u_0 \, x'_1 = 3(x_1 - x_0)$$

and

$$\Delta u_{m-1} \, x'_{m-1} + 2 \Delta u_{m-1} \, x'_m = 3(x_m - x_{m-1}) \ .$$

These can be added to equations [15] to give a set of (m + 1) simultaneous linear equations in (m + 1) unknowns, which can then be solved. The solution is unique for the same reasons

as before. Another end-condition is to require that the end functions be parabolic in form, i.e., that the coefficient of the cubic term in equation [14] be identically zero. The equations consequent on this are:

$$\Delta u_0\, x'_0 + \Delta u_0\, x'_1 = 2(x_1 - x_0)$$

and

$$\Delta u_{m-1}\, x'_{m-1} + \Delta u_{m-1}\, x'_m = 2(x_m - x_{m-1}) \ .$$

The solution proceeds as before. The final end-condition to be considered is for a periodic spline of period $(u_m - u_0)$. In this case $x_m = x_0$, $x'_m = x'_0$, and $x''_m = x''_0$, i.e., there are only m distinct points in the system. To use up the remaining degree of freedom, a version of equation [15] is set up for P_0.

Spline surfaces

The extension of spline interpolation from curves to surfaces is particularly simple. We define a rectangular mesh in the space (u,v,x) (Fig. 22), and for each of the (m-by-n) cells construct a bicubic polynomial which satisfies the following conditions: (1) it interpolates each corner point in value, x; (2) its corner tangents, $\frac{\partial x}{\partial u}$ and $\frac{\partial x}{\partial v}$, and cross-derivatives, $\frac{\partial^2 x}{\partial u \cdot \partial v}$, have prescribed values. For notational convenience we write $\frac{\partial x}{\partial u}$ as ${}_u x$, $\frac{\partial x}{\partial v}$ as ${}_v x$, and $\frac{\partial^2 x}{\partial u \cdot \partial v}$ as ${}_{uv} x$.

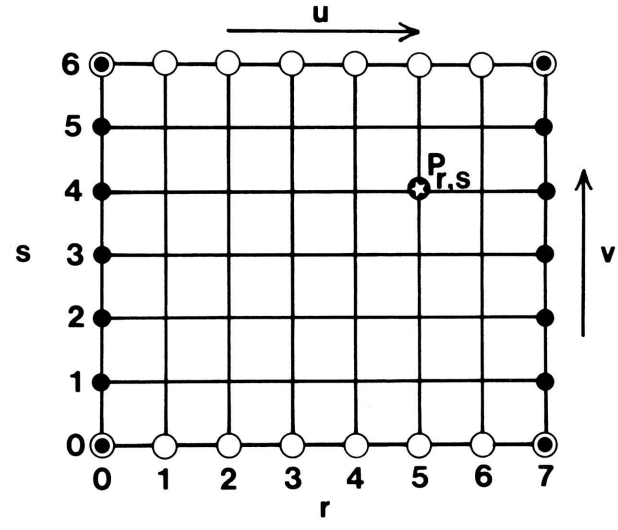


FIGURE 22. Configuration of mesh for fitting spline surfaces; m = 7, n = 6. Coordinate system is (u,v,x), with the x-axis perpendicular to the plane of the paper. The point $P_{r,s}$ shown is for r = 5, s = 4. Points at which the derivative ${}_u x$ is supplied are shown by solid circles; points at which ${}_v x$ is supplied are shown by open circles. The cross-derivative ${}_{uv} x$ is supplied at the four corner points of the mesh.

By analogy with the univariate case, boundary conditions must be prescribed. It is most convenient to use the following set:

(1) ${}_u x$ is known at r = 0, m; s = 0, 1, ..., n

(2) ${}_v x$ is known at r = 0, 1, ..., m; s = 0, n

(3) ${}_{uv} x$ is known at r = 0, m; s = 0, n .

Thus the tangents are supplied on opposite sides of the mesh, and the cross-derivatives at the four corner points. The remaining values of ${}_u x$ are calculated by univariate spline in-

terpolation along each line v = constant, for v = 0, 1, ..., n. In the same way, the remaining values of ${}_{v}x$ are calculated by univariate spline interpolation along each line u = constant, for u = 0, 1, ..., m. The unknown values of the cross-derivative are calculated in a two-step univariate spline interpolation procedure: (1) values for points along v = 0 and v = n are calculated from the four corner values; (2) remaining values are calculated along each line u = constant, for u = 0, 1, ..., m.

It is clear that the bicubic polynomial for each cell is simply a Cartesian product Hermitian interpolant. In a manner similar to the treatment of the univariate case, it can be proved that the piecewise bicubic polynomial comprising all the individual cells is the unique one which gives C^2 continuity (de Boor, 1962).

The bicubic polynomial for the cell with its origin at $P_{r,s}$ is

$$S_{r,s}(u,v) = \sum_{i=0}^{3} \sum_{j=0}^{3} \alpha_{ij} (u - u_{r,s})(v - v_{r,s}) \qquad .....[16]$$

where $P_{r,s}$ is $(u_{r,s}, v_{r,s}, x_{r,s})$, and the α_{ij} are 16 coefficients.

The coefficients, α_{ij}, can be determined from the corner values, tangents and cross-derivatives (these may be termed the corner point conditions) after some straightforward although laborious algebraic manipulation (see for example Whitten and Koelling, 1973). It is more convenient, however, to rewrite equation [16] in matrix form and to arrange the coefficients into a (4 by 4) matrix ∇. Then

$$S_{r,s}(u,v) = \begin{bmatrix} 1 & (u - u_{r,s}) & (u - u_{r,s})^2 & (u - u_{r,s})^3 \end{bmatrix} \cdot \nabla \cdot \begin{bmatrix} 1 \\ (v - v_{r,s}) \\ (v - v_{r,s})^2 \\ (v - v_{r,s})^3 \end{bmatrix} \qquad .....[17]$$

where

$$\nabla = \begin{bmatrix} \alpha_{00} & \alpha_{01} & \alpha_{02} & \alpha_{03} \\ \alpha_{10} & \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{20} & \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{30} & \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix}$$

The matrix ∇ can itself be more conveniently expressed as the product of a matrix, B, containing the cell corner point conditions, and a standard matrix, A(h), which incorporates information on the cell side, h, and which blends together the terms in matrix B. We shall discuss this blending function matrix again when describing Coons surfaces: its form is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3/h^2 & 3/h^2 & -2/h & -1/h \\ 2/h^3 & -2/h^3 & 1/h^2 & 1/h^2 \end{bmatrix}$$

Matrix ∇ in equation [17] is thus

$$\nabla = A(\Delta u_r) \cdot B \cdot A'(\Delta v_s) \qquad .....[18]$$

where $\Delta u_r = u_{r+1,s} - u_{r,s} = u_{r+1,s+1} - u_{r,s+1}$,

$\Delta v_s = v_{r,s+1} - v_{r,s} = v_{r+1,s+1} - v_{r+1,s}$,

$$A(h) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3/h^2 & 3/h^2 & -2/h & -1/h \\ 2/h^3 & -2/h^3 & 1/h^2 & 1/h^2 \end{bmatrix},$$

A' is the transpose of A , and

$$B = \left[\begin{array}{cc:cc} {}^{x}r,s & {}^{x}r,s+1 & {}^{v}x_{r,s} & {}^{v}x_{r,s+1} \\ {}^{x}r+1,s & {}^{x}r+1,s+1 & {}^{v}x_{r+1,s} & {}^{v}x_{r+1,s+1} \\ \hdashline {}^{u}x_{r,s} & {}^{u}x_{r,s+1} & {}^{uv}x_{r,s} & {}^{uv}x_{r,s+1} \\ {}^{u}x_{r+1,s} & {}^{u}x_{r+1,s+1} & {}^{uv}x_{r+1,s} & {}^{uv}x_{r+1,s+1} \end{array}\right] .$$

The boundary matrix B is clearly divisible into four similar partitions. The formulation used here for equation [18] is functionally identical to equation (10) of de Boor (1962), but has been rearranged to conform to standard Coons patch usage.

Equations [17] and [18] can be consolidated into one equation, which can be regarded as the basic equation for a bicubic spline surface. It is

$$S_{r,s}(u,v) = [1 \;\; (u - u_{r,s}) \;\; (u - u_{r,s})^2 \;\; (u - u_{r,s})^3] \cdot A(\Delta u_r) \cdot B \cdot A'(\Delta v_s) \cdot \begin{bmatrix} 1 \\ (v - v_{r,s}) \\ (v - v_{r,s})^2 \\ (v - v_{r,s})^3 \end{bmatrix} \quad .....[19]$$

Special cases

Two versions of equation [19] are of special interest: the case where the cell is square, and the extension to a curvilinear coordinate system. In the first case

$$\Delta u_r = \Delta v_s = 1 ,$$

and hence

$$A(\Delta u_r) = A(\Delta v_s) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} = A .$$

In addition, we can simplify matters further by setting $u_{r,s}$ to zero. This simply makes $P_{r,s}$ the origin of the coordinate system for the cell. Equation [19] then reduces to

$$S_{r,s}(u,v) = [1 \;\; u \;\; u^2 \;\; u^3] \cdot A \cdot B \cdot A' \cdot \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \quad .....[20]$$

The row and column vectors are both simply the cubic polynomial basis functions.

The extension to a curvilinear coordinate system (see Ahlberg, 1974, for a discussion of some problems) just involves making the $S_{r,s}$ into a three-component vector, $\vec{S}_{r,s}$, and using

a boundary matrix $\vec{B}$ in which each entry is also vector-valued. Matrix $\vec{B}$ thus contains ($4 \times 4 \times 3 = 48$) boundary conditions. The basis functions and the matrix A are the same for each component. Equation [20] becomes

$$\vec{S}_{r,s}(u,v) = [1 \quad u \quad u^2 \quad u^3] \cdot A \cdot \vec{B} \cdot A' \cdot \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \qquad .....[21]$$

Example

Global-basis splines cannot be applied directly to the Kansas data set because its cells are not rectangular when projected onto the base-plane of a rectangular Cartesian coordinate system. The vector-valued extension (equation [21]) can, however, be used provided that suitable boundary conditions are specified. As neither the tangent vectors at the edge-points nor the cross-derivatives at the corner points have been supplied, these must first be estimated. Suitable procedures for doing this are discussed in the following section on Coons surfaces, and so illustration of spline surfaces will be deferred until then.

Summary

Global-basis splines have been used successfully for surface representation for many years. Their advantages and disadvantages are as follows:

(1) The representation which they give is global, because it is necessary to specify tangents and cross-derivatives at the mesh boundary.
(2) It is usually highly inconvenient to have to specify derivatives of any kind anywhere in a network. This is especially true for cross-derivatives. Hence surface representation using global-basis splines may well be a good solution to an unrealistic problem.
(3) The surface produced is a Hermitian interpolant at every data point.
(4) The method is naturally related to both of the most important CAD methods--B-splines and Coons patches.

Coons Surfaces

We now proceed to the class of CAD surfaces about which more has probably been written than any other. These are known as Coons surfaces, and were developed in the earliest days of computer-aided design. The original work was carried out by Ferguson (1964) and Coons (1967), but perhaps the clearest and most comprehensive description of the method is that given in Forrest's (1972a) survey. This also provides a substantial bibliography of both published and unpublished material.

In previous sections we discussed surfaces which are either local or global, and which either interpolate data or approximate it. Thus Bézier surfaces were found to be global approximants, B-spline surfaces to be local approximants, and de Boor's spline surfaces to be global interpolants. At the end of the last section, however, it was shown that each of the surface patches given by the bicubic spline representation could be defined explicitly in terms of its boundary conditions (equation [19]). The representation can thus be a local

Hermitian interpolant provided that the boundary conditions for each cell are known independently. Because the representation is in Cartesian product form, the boundary conditions refer only to the corner points of each cell, i.e. to the nodal points of the network. Coons surfaces are simply a generalization of this approach to include positional and derivative information from the cell edges in addition to that from its corners. The whole theory is developed for a curvilinear coordinate system.

Coons' original description of his method (1967) uses a somewhat confusing mathematical representation. Here we follow substantially his later work (Coons, 1974), and that of Forrest (1972a). Forrest has also pointed out (1972a, p. 351) that there is now considerable laxness in the usage of the term "Coons surface." In its strict sense it should be reserved for surfaces which are blends of space curves. In its widest sense it includes as well the Cartesian product spline surfaces of equation [19].

Fundamentals of Coons surfaces

To obtain the most general interpolant, a Boolean sum representation is used instead of the familiar Cartesian product. Consider initially a surface which is to be defined over the domain $0 \leqslant u \leqslant 1$ and $0 \leqslant v \leqslant 1$ (Fig. 23). Use of a unit square domain is simply a mathematical convenience; the results can be generalized to any domain which maps one-to-one onto it. The surface, $\vec{Q}(u,v)$, is to be constructed such that it is a linear blend of the corner points $[\vec{P}(0,0), \vec{P}(1,0), \vec{P}(0,1), \vec{P}(1,1)]$ and of the edge curves $[\vec{P}(0,v), \vec{P}(1,v), \vec{P}(u,0), \vec{P}(u,1)]$. The surface will be a Lagrangian interpolant: it will match in position with each of its four neighbors.

The Boolean sum surface is $\vec{B}_u + \vec{B}_v - \vec{B}_u\vec{B}_v$, where $\vec{B}_u$ and $\vec{B}_v$ are the linear operators which interpolate in the u and v directions respectively. Each can be defined as a linear blend of pairs of corner points on opposite edges of the cell. Thus

$$\vec{B}_u = [f_0(u) \quad f_1(u)] \begin{bmatrix} \vec{P}(0,v) \\ \vec{P}(1,v) \end{bmatrix} \quad \text{and} \quad \vec{B}_v = [\vec{P}(u,0) \quad \vec{P}(u,1)] \begin{bmatrix} f_0(v) \\ f_1(v) \end{bmatrix}$$

where f_0 and f_1 are blending functions.

The Cartesian product is

$$\vec{B}_u\vec{B}_v = [f_0(u) \quad f_1(u)] \begin{bmatrix} \vec{P}(0,0) & \vec{P}(0,1) \\ \vec{P}(1,0) & \vec{P}(1,1) \end{bmatrix} \begin{bmatrix} f_0(v) \\ f_1(v) \end{bmatrix}$$

Hence the bilinear Boolean sum surface (termed the bilinear Coons surface) is

$$\vec{Q}(u,v) = [f_0(u) \quad f_1(u)] \begin{bmatrix} \vec{P}(0,v) \\ \vec{P}(1,v) \end{bmatrix} + [\vec{P}(u,0) \quad \vec{P}(u,1)] \begin{bmatrix} f_0(v) \\ f_1(v) \end{bmatrix} \qquad \text{.....[22]}$$

$$- [f_0(u) \quad f_1(u)] \begin{bmatrix} \vec{P}(0,0) & \vec{P}(0,1) \\ \vec{P}(1,0) & \vec{P}(1,1) \end{bmatrix} \begin{bmatrix} f_0(v) \\ f_1(v) \end{bmatrix}$$

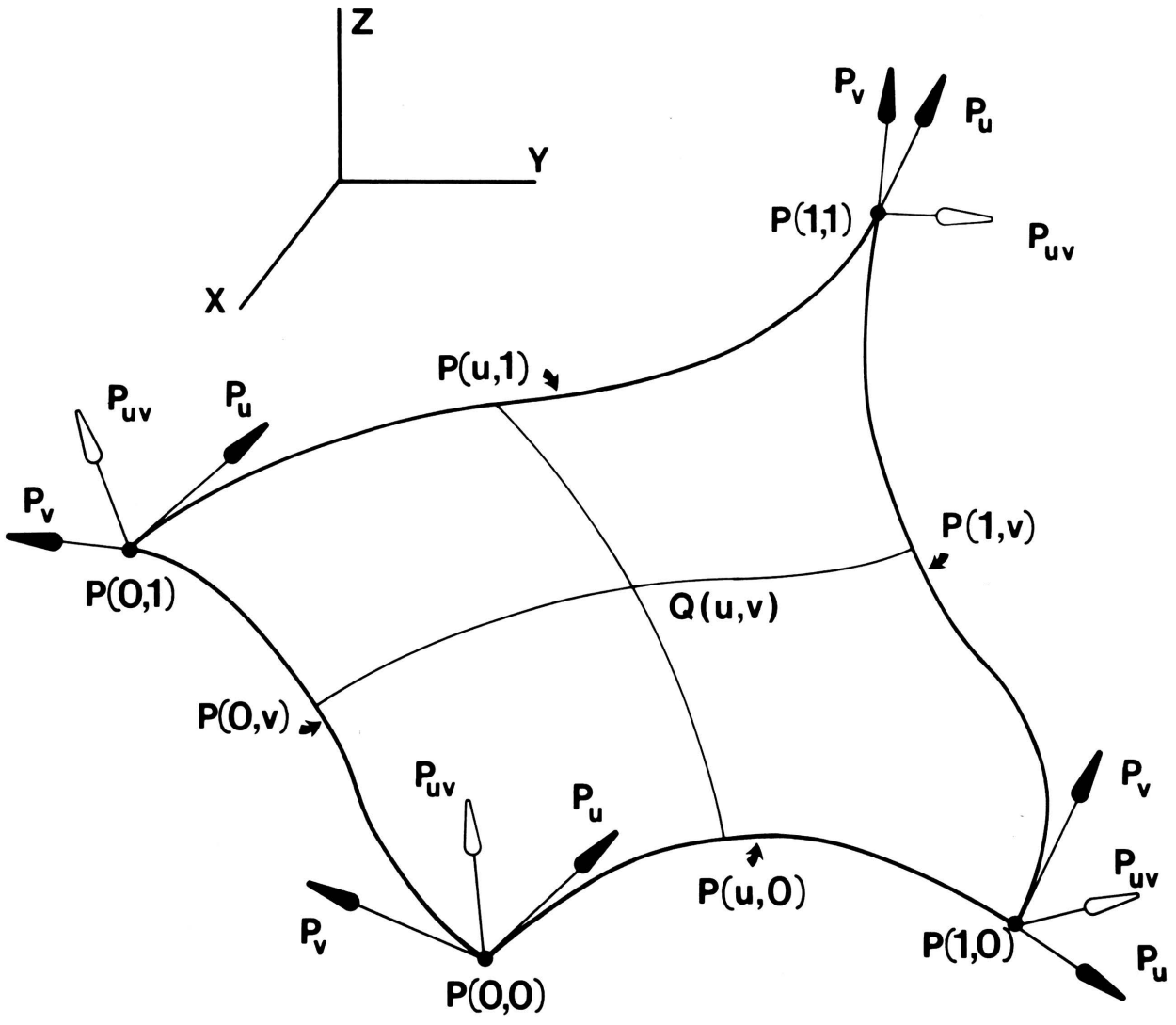


FIGURE 23. Coons surface, $\vec{Q}(u,v)$, defined over a unit-square u-v domain. Partial derivatives are indicated only at the corner points: the single partial derivatives $\vec{P}_u$ and $\vec{P}_v$ are shown by solid arrows, the cross-derivatives $\vec{P}_{uv}$ by open arrows.

The selection of blending functions is controlled by three factors:

(1) for interpolation it is necessary that

$$f_0(0) = 1, \quad f_0(1) = 0, \quad f_1(0) = 0, \quad f_1(1) = 1,$$

(2) for axis-independence the u and v functions must be normalized, i.e.

$$f_0(u) + f_1(u) = 1 \quad \text{and} \quad f_0(v) + f_1(v) = 1,$$

(3) the functions are usually chosen to be continuous.

The simplest functions satisfying these conditions and giving a linear blend are

$$f_0(u) = (1 - u) \qquad f_1(u) = u$$

$$f_0(v) = (1 - v) \qquad f_1(v) = v$$

It is instructive now to consider some specializations of the bilinear Coons surface. Consider first the case when the space curve $\vec{P}(u,0)$ is a linear blend of the points $\vec{P}(0,0)$ and $\vec{P}(1,0)$, and $\vec{P}(u,1)$ is a linear blend of $\vec{P}(0,1)$ and $\vec{P}(1,1)$, i.e.

$$\vec{P}(u,0) = [f_0(u) \quad f_1(u)] \begin{bmatrix} \vec{P}(0,0) \\ \vec{P}(1,0) \end{bmatrix}$$

and

$$\vec{P}(u,1) = [f_0(u) \quad f_1(u)] \begin{bmatrix} \vec{P}(0,1) \\ \vec{P}(1,1) \end{bmatrix}$$

Equation [22] then reduces to

$$\vec{Q}(u,v) = [f_0(u) \quad f_1(u)] \begin{bmatrix} \vec{P}(0,v) \\ \vec{P}(1,v) \end{bmatrix} .$$

This is simply the linear lofted surface obtained by using the operator $\vec{B}_u$. We can obtain the other linear lofted surface by making analogous restrictions on $\vec{P}(0,v)$ and $\vec{P}(1,v)$. If both sets of restrictions are made simultaneously, equation [22] reduces to

$$\vec{Q}(u,v) = [f_0(u) \quad f_1(u)] \begin{bmatrix} \vec{P}(0,0) & \vec{P}(0,1) \\ \vec{P}(1,0) & \vec{P}(1,1) \end{bmatrix} \begin{bmatrix} f_0(v) \\ f_1(v) \end{bmatrix} .$$

This is just the Cartesian product surface, giving bilinear interpolation between the four corners of the patch. Thus the Boolean sum (Coons) representation can be regarded as more general than either the lofted or Cartesian product surfaces. Intuitively what we are doing when constructing a Coons surface is to add together the u and v lofted surfaces, and then to subtract a Cartesian product surface to nullify the effect of having considered each corner point twice.

The Coons representation can now be developed even further to give simple Hermitian rather than Lagrangian interpolation.

The bicubic Coons surface

It was noted earlier in this paper that most practical surface representation methods must give surfaces having at least class C^1 continuity. To achieve this with a Coons surface, tangent and cross-derivative vectors must be specified at each corner point, and also the components of the tangent vectors *across* each of the edges (these components are referred to here as the "normal vectors"). These normal vectors are univariate functions of u or v, just as the edges themselves, and so each must be known for all values of its parameter. For convenience, the normal vectors are written as follows: that across $\vec{P}(u,0)$ is $\vec{P}^*_v(u,0)$, that across $\vec{P}(u,1)$ is $\vec{P}^*_v(u,1)$, that across $\vec{P}(0,v)$ is $\vec{P}^*_u(0,v)$, and that across $\vec{P}(1,v)$ is $\vec{P}^*_u(1,v)$. Interpolation of the tangent vector components *along* each edge is, of course, implicit in the interpolation of the edge itself.

The bicubic Coons surface is

$$\vec{Q}(u,v) = [f_0(u) \quad f_1(u) \quad g_0(u) \quad g_1(u)] \begin{bmatrix} \vec{P}(0,v) \\ \vec{P}(1,v) \\ \vec{P}^*_u(0,v) \\ \vec{P}^*_u(1,v) \end{bmatrix}$$

$$+ [\vec{P}(u,0) \quad \vec{P}(u,1) \quad \vec{P}^*_v(u,0) \quad \vec{P}^*_v(u,1)] \begin{bmatrix} f_0(v) \\ f_1(v) \\ g_0(v) \\ g_1(v) \end{bmatrix} \qquad .....[23]$$

$$- [f_0(u) \quad f_1(u) \quad g_0(u) \quad g_1(u)] \begin{bmatrix} \vec{P}(0,0) & \vec{P}(0,1) & \vec{P}_v(0,0) & \vec{P}_v(0,1) \\ \vec{P}(1,0) & \vec{P}(1,1) & \vec{P}_v(1,0) & \vec{P}_v(1,1) \\ \vec{P}_u(0,0) & \vec{P}_u(0,1) & \vec{P}_{uv}(0,0) & \vec{P}_{uv}(0,1) \\ \vec{P}_u(1,0) & \vec{P}_u(1,1) & \vec{P}_{uv}(1,0) & \vec{P}_{uv}(1,1) \end{bmatrix} \begin{bmatrix} f_0(v) \\ f_1(v) \\ g_0(v) \\ g_1(v) \end{bmatrix},$$

where f_0, f_1, g_0, g_1 are blending functions.

The blending functions are in this case the product of the matrix A in equation [20] and the cubic basis functions, i.e.

$$[f_0(u) \quad f_1(u) \quad g_0(u) \quad g_1(u)] = [1 \; u \; u^2 \; u^3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix}$$

$$= [1 - 3u^2 + 2u^3 \;\vdots\; 3u^2 - 2u^3 \;\vdots\; u - 2u^2 + u^3 \;\vdots\; -u^2 + u^3]$$

The method for calculating higher order basis functions is given by Forrest (1972a).

It is possible to obtain the lofted and Cartesian product surfaces from equation [23] in the same way as for the bilinear Coons surface. The Cartesian product surface is identical to equation [21], and is generally known as the Coons bicubic patch (Peters, 1974). It is the surface most used in practical CAD systems (see for example Craidon, 1975).

In a typical CAD system, the surface to be modelled is formed into a network of quadrilateral cells, and a bicubic patch defined for each cell. In few cases are the tangent and cross-derivative vectors known at each node of the network--usually they must be estimated. It is in this estimation that the main weakness of the Coons bicubic patch lies.

To estimate the tangent vectors one can use two approaches, one global and one local. The global approach is that outlined earlier in the description of globally based interpolating splines. Data is provided at the edge of the network, and a univariate spline interpolation technique used to calculate the tangent vectors at each interior node. The resulting

surface is termed "spline-blended." An approach of this type was used by Ferguson (1964) in one of the earliest bicubic patch systems.

The alternative approach is to use some local estimation procedure. We may illustrate this by the estimation of the u-partial derivative ($\vec{P}_u$) at the point (j,k) in Figure 24. This corresponds to $\vec{P}_u(0,0)$ for patch (j,k), $\vec{P}_u(0,1)$ for patch (j,k-1), $\vec{P}_u(1,0)$ for patch (j-1,k) and $\vec{P}_u(1,1)$ for patch (j-1,k-1). The simplest estimator is the divided-difference

$$\vec{P}_u = [\vec{P}_{j+1,k} - \vec{P}_{j-1,k}] \; / \; (u_{j+1} - u_{j-1}) \qquad .....[24]$$

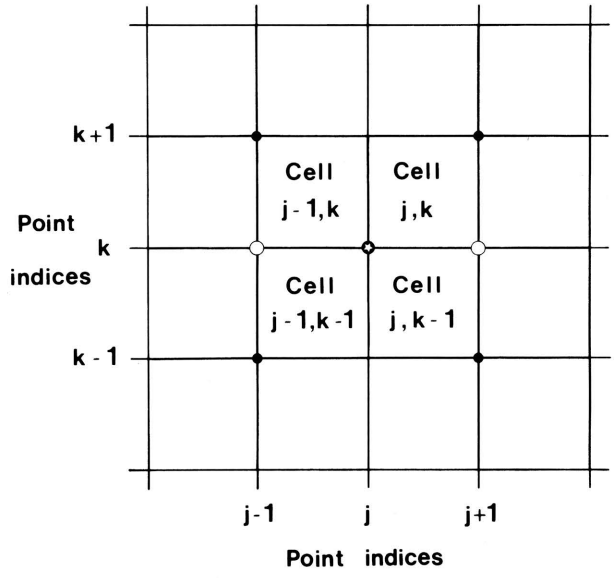


FIGURE 24. Local estimation of derivatives. The points used to estimate the u-partial at point (j,k) are shown by open circles, those used to estimate the cross-derivative by solid circles.

For unit square u and v domains the divisor of equation [24] reduces to 2, and the estimator is simply the straight-line slope between mesh points on either side of the one being estimated. Similar approaches have been used by Hessing *et al.* (1972) and Akima (1974). The advantages of this estimation method are that it is local, and computationally simple. It may, however, produce results which are definitely sub-optimal (see for instance Tipper, 1977, p. 606). It is well suited to be a "first attempt" strategy in modelling a surface.

Whereas it is relatively straightforward to estimate the tangent vectors, the estimation of cross-derivatives is considerably trickier. It is quite appropriate to consider these vectors as describing the twist of the surface at a point (Coons, 1974), and for this reason they are often termed the "twist vectors" of the point. Yet it is not readily apparent (even to the mathematically sophisticated user) how to obtain a required twist by specifying a vector of particular magnitude and direction (Forrest, 1972a, 1972b). There appear to be three solutions to the problem: (1) Values can be assigned by trial-and-error when operating in an interactive computational environment. Adjustments are then made until a surface is obtained which is visually acceptable to the user. (2) A local estimation procedure can be used

(Birkhoff and de Boor, 1965). In this case the cross-derivative, $\vec{P}_{uv}$, at point (j,k) (Fig. 24) is defined by

$$\vec{P}_{uv} = \frac{[\vec{P}_{j+1,k+1} - \vec{P}_{j+1,k-1} + \vec{P}_{j-1,k-1} - \vec{P}_{j-1,k+1}]}{(u_{j+1} - u_{j-1})(v_{k+1} - v_{k-1})} \quad .$$

(3) All cross-derivatives can be set to zero, a practice apparently originated by Ferguson (1964). This results in the generation of what are termed "pseudoflats" at the corners of each patch. These features may in many cases be neither aesthetically pleasing nor mathematically acceptable, but it is this solution which has been used by most authors.

An alternative solution to the twist vector problem has been devised by Inaba (see Bézier, 1972), and used as the basis for a working CAD system (termed F-mesh). Clearly the necessity of specifying the four cross-derivatives for each patch arises because the Cartesian product bicubic patch (equation [21]) has 16 degrees of freedom, of which only twelve are used up by positional and tangent vector information (see also Ahlin, 1964). Inaba's approach is to devise alternative ways of using up the four remaining degrees of freedom, thus by-passing twist vectors entirely. Some of his methods are described by Bézier (1972, p. 152-161).

Further developments of the Coons method

One apparent weakness of the pure Coons formulation (the Boolean sum surface) is that both the space curves defining the cell edges and the normal vectors must be supplied as univariate functions. In most practical applications this is not possible, but in cases where it can be done the Coons method is probably the ideal.

Consider the problem of blending a surface into a boundary curve of known form. In a geological context this might, for instance, be the trace of a subsurface horizon as it intersects the face of a quarry. Only interpolation of position along the curve is required, and so a bilinear surface is appropriate. Without exception, the Cartesian product methods described earlier are unable to give satisfactory results--indeed this was one of the criticisms levelled at (Cartesian product) B-spline surfaces by Clark (1976a). The bilinear Coons surface (equation [22]) is the ideal solution here, because nowhere does it circumscribe the form of the boundary curves $[\vec{P}(0,v), \vec{P}(1,v), \vec{P}(u,0), \vec{P}(u,1)]$.

It is profitable then to examine how Coons' method can be used to "represent with curves" rather than just to "represent with points." The approach is to approximate an object's shape by a network of curves, and then to blend them together by Coons' method. For convenience the form of network used up to now is retained, although Coons (1974) has indicated that this approach can use cells with more than four edges. The boundary curves can be defined in three ways: (1) B-splines, (2) Overhauser curves, (3) Catmull-Rom splines. The definition of B-spline curves in a network is a straightforward application of the theory presented in an earlier section of this paper. The reader should refer to Coons (1974) for more detailed discussion of this approach. Here we consider in greater detail the use of Overhauser curves and Catmull-Rom splines.

1. Overhauser curves

In 1968 Overhauser developed a parabolic blending method for surface design of car bodies. The technique has been much neglected, although interest has recently been revived by Rogers

and Adams (1976) and Brewer and Anderson (1977). In describing Overhauser's method we follow to some extent Brewer and Anderson (1977), but attempt to develop it in a manner not dissimilar to that used earlier for Bézier and B-spline curves.

Consider a set of points, $\vec{P}_i$, i = 0, 1, 2, 3 (Fig. 25). The curve required, $\vec{R}(u)$, is a blend of these points and passes through $\vec{P}_1$ and $\vec{P}_2$. It is a cubic in the parameter u, and has the form

$$\vec{R}(u) = \sum_{i=0}^{3} \beta_i(u)\, \vec{P}_i \qquad .....[25]$$

where the β_i are basis functions; each is a cubic in u.

It is clearly possible to adjoin curves of this form at their ends, thus creating a piecewise Overhauser curve. Because it is known from Walsh's theorem that only global-basis interpolating splines give C^2 continuity between adjacent segments, the piecewise Overhauser curve is restricted to class C^1.

The Overhauser curve is formed as a linear blend of two parametric quadratics. One, parametrized by r, passes through $\vec{P}_0$, $\vec{P}_1$ and $\vec{P}_2$; the other, parametrized by s, passes through $\vec{P}_1$, $\vec{P}_2$ and $\vec{P}_3$. By analogy with equation [25] these curves are defined as

$$\vec{p}(r) = \sum_{i=0}^{2} \mu_i(r)\, \vec{P}_i \qquad \text{and} \qquad \vec{q}(s) = \sum_{i=0}^{2} \eta_i(s)\, \vec{P}_{i+1}$$

where the μ_i and η_i are basis functions, quadratic functions of r and s respectively. The

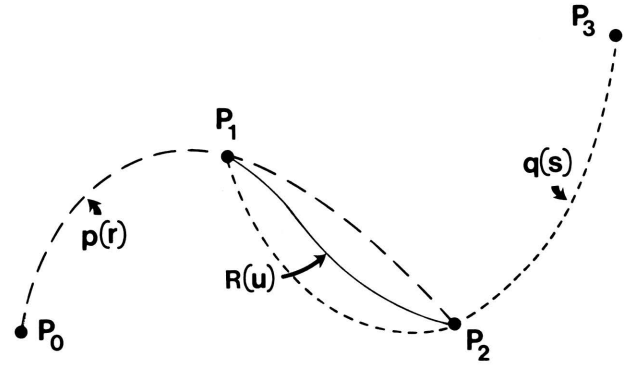


FIGURE 25. Set of points used for constructing the Overhauser curve, $\vec{R}(u)$, shown by the solid line. The blended parabolas are $\vec{p}(r)$ and $\vec{q}(s)$. The points $\vec{P}_1$ and $\vec{P}_2$ correspond to values 0 and 1 respectively of the parameter u.

Overhauser curve itself is parametrized by u. It is written as

$$\vec{R}(u) = (1-u)\, \vec{p}(r) + u\vec{q}(s) \qquad .....[26]$$

The basis functions for the quadratics can be determined provided that values for r and s are specified for each of the points $\vec{P}_i$. By convention the following parametrization is used:

$\vec{P}_0$: r = 0, s is undefined,
$\vec{P}_1$: r = ½, s = 0,
$\vec{P}_2$: r = 1, s = ½,
$\vec{P}_3$: r is undefined, s = 1.

The basis functions may then be shown to be

$$[\mu_0\ \mu_1\ \mu_2] = [r^2\ r\ 1]\begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

and

$$[\eta_0\ \eta_1\ \eta_2] = [s^2\ s\ 1]\begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

The relationship between the parameters r, s, and u is linear. In fact r = (u+1)/2 and s = u/2. Equation [26] can thus be written as

$$\vec{R}(u) = (1-u)[r^2\ r\ 1]\begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} \vec{P}_0 \\ \vec{P}_1 \\ \vec{P}_2 \end{bmatrix} + u[s^2\ s\ 1]\begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} \vec{P}_1 \\ \vec{P}_2 \\ \vec{P}_3 \end{bmatrix}$$

$$= (1-u)\left[\frac{(u+1)^2}{4}\ \frac{(u+1)}{2}\ 1\right]\begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} \vec{P}_0 \\ \vec{P}_1 \\ \vec{P}_2 \end{bmatrix} + u\left[\frac{u^2}{4}\ \frac{u}{2}\ 1\right]\begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} \vec{P}_1 \\ \vec{P}_2 \\ \vec{P}_3 \end{bmatrix}$$

On consolidation of this equation, the basis functions, β_i, are determined to be

$$[\beta_0\ \beta_1\ \beta_2\ \beta_3] = [u^3\ u^2\ u\ 1]\begin{bmatrix} -1/2 & 3/2 & -3/2 & 1/2 \\ 1 & -5/2 & 2 & -1/2 \\ -1/2 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The combination of Overhauser curves with a Coons surface formulation (equation [22]) provides a powerful and flexible method of surface representation. Consider the definition of a surface patch over the mesh of points shown in Figure 26. The patch is to interpolate the points $\vec{P}(0,0)$, $\vec{P}(0,1)$, $\vec{P}(1,0)$ and $\vec{P}(1,1)$, and the edges $\vec{P}(u,0)$, $\vec{P}(u,1)$, $\vec{P}(0,v)$ and $\vec{P}(1,v)$. The Overhauser method is first used to define the edge curves--$\vec{P}(u,0)$, for instance, is defined as the Overhauser curve $\vec{R}(u,0)$ by using $\vec{P}_{-1,0}$, $\vec{P}_{0,0}$, $\vec{P}_{1,0}$, and $\vec{P}_{2,0}$ for $\vec{P}_0 \rightarrow \vec{P}_3$ in equation [25]. The Overhauser-Coons patch (Brewer and Anderson, 1977) is then

$$\vec{R}(u,v) = [f_0(u)\quad f_1(u)]\begin{bmatrix} \vec{R}(0,v) \\ \vec{R}(1,v) \end{bmatrix}$$

$$+ [\vec{R}(u,0)\quad \vec{R}(u,1)]\begin{bmatrix} f_0(v) \\ f_1(v) \end{bmatrix} - [f_0(u)\quad f_1(u)]\begin{bmatrix} \vec{P}(0,0) & \vec{P}(0,1) \\ \vec{P}(1,0) & \vec{P}(1,1) \end{bmatrix}\begin{bmatrix} f_0(v) \\ f_1(v) \end{bmatrix} \quad .....[27]$$

2. Catmull-Rom splines

Catmull and Rom (1974) have described a class of local-basis spline functions which can be used both for curve and surface representation. Consider the points, $\vec{P}_i$, in Figure 27. The interpolant, $\vec{D}(u)$, is to be a cubic in the parameter u ($0 \leqslant u \leqslant 1$), and is to interpolate

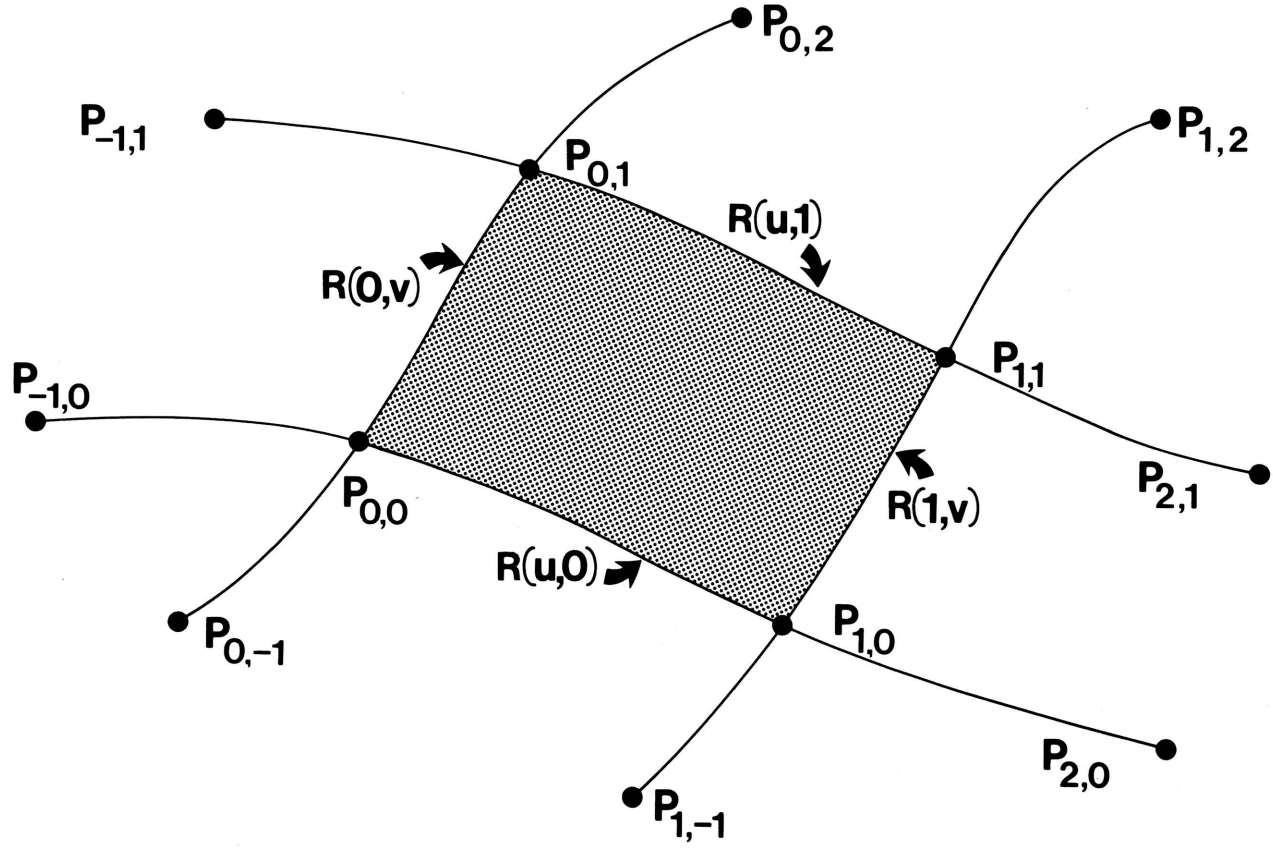


FIGURE 26. An Overhauser-Coons patch (shaded) defined by 12 data points. In terms of the parameters u and v, the points indexed as $\vec{P}_{0,0}$, $\vec{P}_{0,1}$, $\vec{P}_{1,0}$ *and* $\vec{P}_{1,1}$ *are* $\vec{P}(0,0)$, $\vec{P}(0,1)$, $\vec{P}(1,0)$ *and* $\vec{P}(1,1)$ *respectively.*

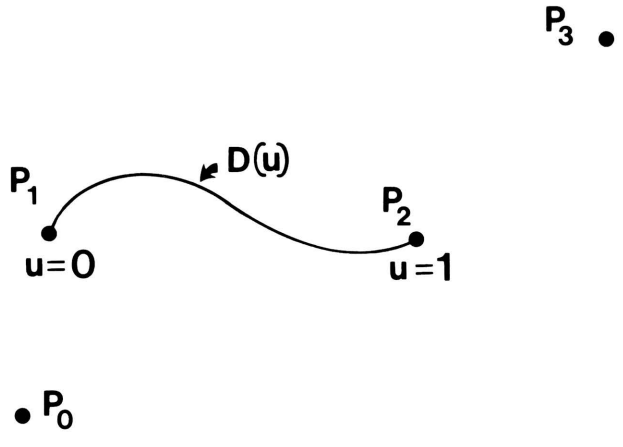


FIGURE 27. Univariate Catmull-Rom spline, $\vec{D}(u)$, *interpolates points* $\vec{P}_1$ *and* $\vec{P}_2$. *Additionally, points* $\vec{P}_0$ *and* $\vec{P}_3$ *are used to estimate the slopes at* $\vec{P}_1$ *and* $\vec{P}_2$ *respectively.*

$\vec{P}_1$ and $\vec{P}_2$. The slopes at $\vec{P}_1$ and $\vec{P}_2$ are to be estimated respectively by

$$\vec{D}'(0) = \omega(\vec{P}_2 - \vec{P}_0) \qquad \text{and} \qquad \vec{D}'(1) = \omega(\vec{P}_3 - \vec{P}_1)$$

The parameter ω controls the slope magnitude. When $\omega = \frac{1}{2}$, the estimates are simply the divided-differences. The Catmull-Rom spline is then

$$\vec{D}(u) = \sum_{i=0}^{3} \zeta_i(u)\, \vec{P}_i$$

where $\zeta_i(u)$ are basis functions. These may be determined to be

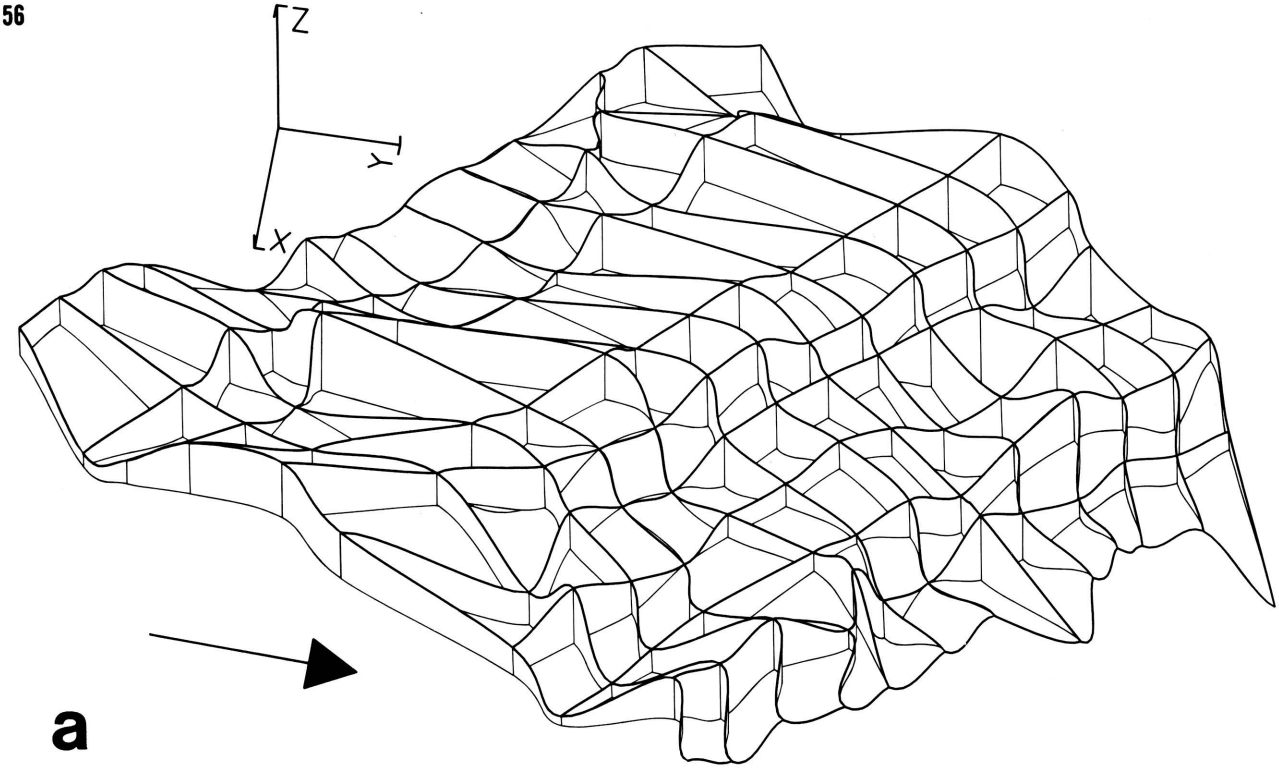
$$[\zeta_0(u) \quad \zeta_1(u) \quad \zeta_2(u) \quad \zeta_3(u)] = [u^3 \; u^2 \; u \; 1] \begin{bmatrix} -\omega & (2-\omega) & (\omega-2) & \omega \\ 2\omega & (\omega-3) & (3-2\omega) & -\omega \\ -\omega & 0 & \omega & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Incorporation of Catmull-Rom splines into a Coons surface formulation proceeds in exactly the same way as for the Overhauser-Coons patch (see the previous section, equation [27]). Catmull-Rom splines are of class C^1, and it has been suggested that they might be used as the univariate basis for a Cartesian product surface, much in the way of B-splines. Clark (1974) has, however, pointed out that, despite having the desirable property of interpolation, such a surface would not possess the convex hull property which renders B-spline surfaces so usable. It is probably best to reserve Catmull-Rom splines for use in the Coons formulation.

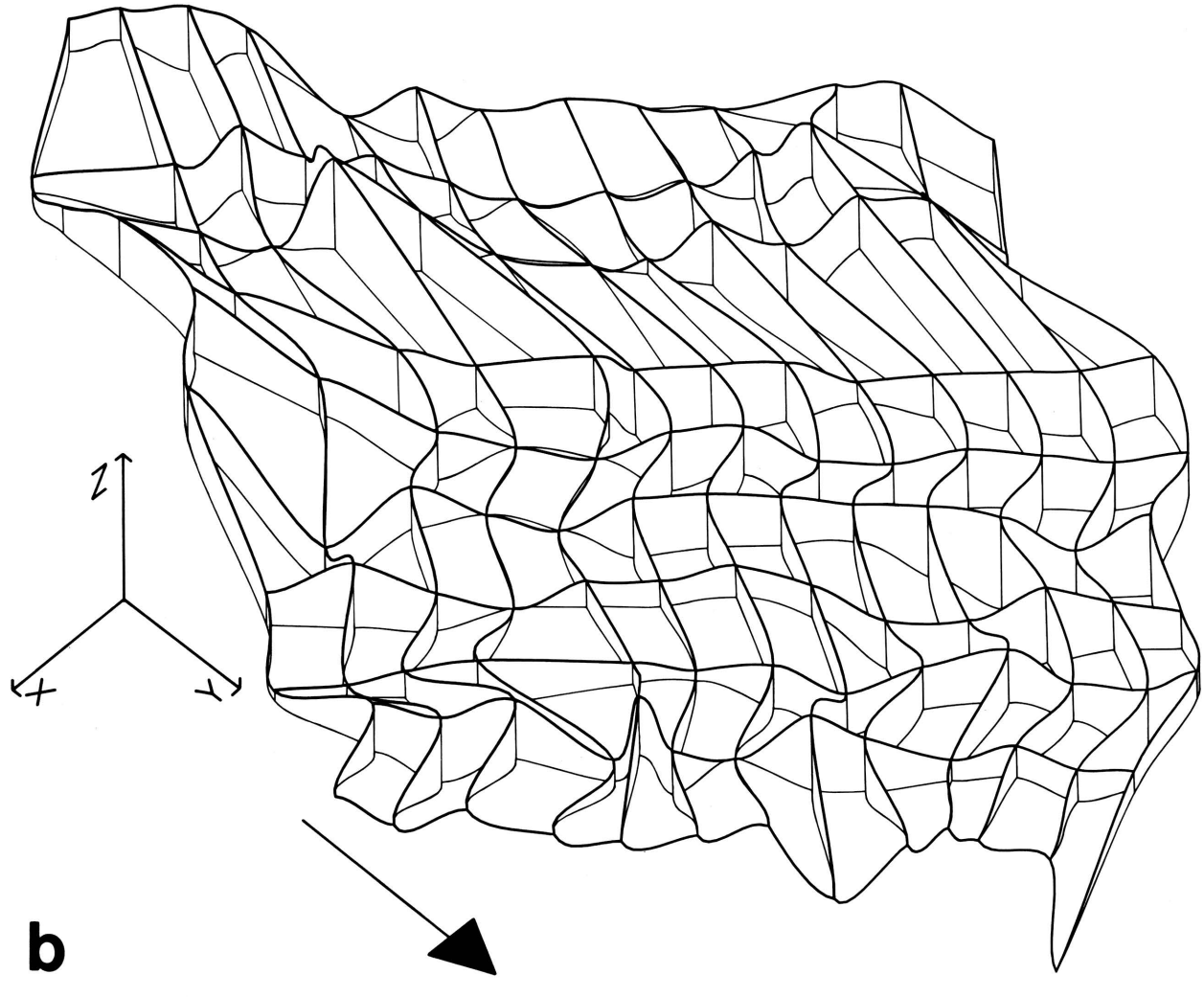
Example

Bicubic Cartesian product surfaces (equation [21]) were fitted to each of the cells in the Cherokee data set. The single partial derivatives at each data point were estimated by the divided-difference method (equation [24]); for data points on the edge of the network they were estimated by the corresponding values from the nearest interior mesh point. The cross-derivatives were all set to zero. It is interesting to note here that had dip-meter readings been available at each well, the surface slopes could have been obtained directly, rendering estimation of the partial derivatives unnecessary. This is one of the rare instances where a data set might comprise both values and partial derivatives at each mesh point.

Figure 28 shows the sand body of the Cherokee data set as it would appear when viewed from three different directions. Hidden lines have been removed to produce a conventional fence diagram. The rippled appearance of parts of the upper and lower surfaces is the pseudo-flat effect which was referred to earlier, and results from having set the cross-derivatives to zero. There are also some places where the lower surface of the sand apparently overlies the upper one. This results from using divided-differences to estimate the partial derivatives; it would probably not have occurred had spline-blending been used. In extreme cases, none of which occur in the examples shown, individual patches may bend back under themselves to create artificial multiple z-valued surfaces. This also results from estimation of derivatives, and will not happen in areas with good data control.



a



b

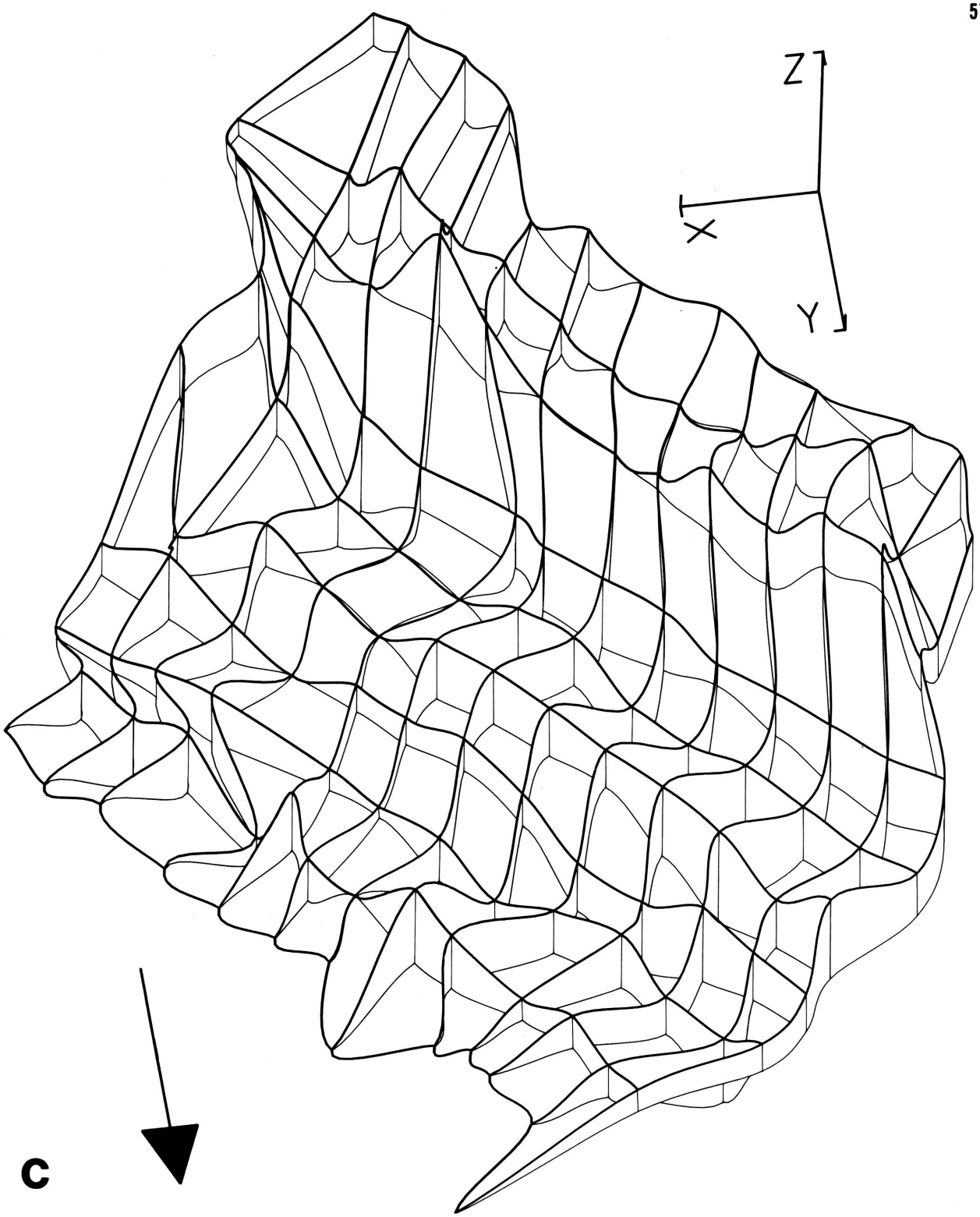


FIGURE 28. Fence diagram of the Cherokee data set; heavy lines define the upper surface, light lines the lower. Length of X and Y axes is 500 ft.; length of Z axis is 25 ft. (a) Trimetric projection, $\phi = -100°$, $\theta = -30°$. (b) Trimetric projection, $\phi = -130°$, $\theta = -40°$. (c) Trimetric projection, $\phi = -160°$, $\theta = -50°$.

Summary

Coons surfaces have formed the basis of many practical CAD systems. Their advantages and disadvantages are as follows:

(1) A Coons surface is a local interpolant, either Lagrangian or Hermitian. The generalization to class C^n is given by Forrest (1972a).

(2) In its strict sense (the Boolean sum surface), the Coons surface is a blend of space curves. Because in this case the edge curves of each cell and their slopes must be specified analytically, the pure Coons surface is of limited use for many applications.

(3) The pure Coons surface may be decomposed into lofted and Cartesian product surfaces. The commonest Cartesian product surface is the bicubic patch, which can be constructed provided that values and derivatives are supplied at its corner points.

(4) For the bicubic patch, two types of derivative must be estimated for each corner point. The tangent vectors can be estimated locally from divided-differences, or globally by spline interpolation. The cross-derivatives (twist vectors) are troublesome to estimate. They may be estimated locally, adjusted interactively, or arbitrarily set to zero.

(5) Coons' approach may be used to combine space curves defined in a variety of ways, to give surfaces which interpolate edges and corner points. Methods used to define the curves include B-splines, Overhauser curves, and Catmull-Rom interpolating splines.

PART II -- MANIPULATIONS OF THE COMPUTERIZED MODEL

Introduction -- Manipulation Methods

Part I of this paper attempted to show how computerized models can be constructed, and examples were given of the use of several modelling methods. Creation of a model cannot, however, be an end in itself. If computerized models are to be of practical use to the geologist, it is essential that he be able to retrieve from them answers to questions of interest. In Sabin's words (1971b), it is essential that the geologist can "interrogate" the model. In this part of the paper we focus on the interrogation problem--the methods used to solve it are termed "manipulation methods."

Surprisingly, this aspect of computerized modelling has received scant attention, and there is little published literature on the subject. Undoubtedly, the most important contribution has been made by Sabin, in a series of technical memoranda issued from the British Aircraft Corporation (Sabin, 1968a, 1968c-e; see also Sabin, 1971b). In his 1971 paper, Sabin listed more than two dozen interrogation facilities used in the BAC Numerical Master Geometry system. These included sectioning (both by planar and by non-planar surfaces), offsetting, intersection by lines, measurement of areas and volumes, and generation of perspective views. Naturally the interrogations were biased heavily towards those most valuable in an aeronautical CAD system. Here we are interested in different and more general facilities, ones which the geologist can use. It is to indicate this greater generality that the methods are termed "manipulations" rather than "interrogations": Sabin's interrogations, however, form an important subset of the manipulation methods.

Categories of Manipulation

The variety of manipulations possible on an object modelled in computerized form is limited only by the imagination of the user and by the availability of suitable computer programs. For our purposes five categories may be identified: transformation, visualization, reduction, simulation, and mensuration. Inevitably in such an arbitrary classification there is overlap between categories.

In the sections which follow, each category will be considered in turn, emphasizing manipulations which are important from the geologist's standpoint. In some cases examples will be given which illustrate the practical use of particular manipulation methods. The data used is the Cherokee data set from southeastern Kansas.

TRANSFORMATION

Inherent in much of geology is the idea of continuous shape change, whether it be tectonic deformation through time, modification of fossil shape during ontogeny, or land form alteration by surface processes. When such changes are considered in a descriptive sense alone, divorced from their causal processes, each can be described simply as an example of coordinate transformation. Because the computerized model is a precise mathematical shape specification, application of this type of transform presents no difficulty.

In any reference coordinate system the transformation is

$$\vec{X}' = \vec{X} \cdot \vec{T} \quad , \qquad \text{.....[28]}$$

where $\vec{X}'$ and $\vec{X}$ are homologous point vectors, post- and pre-transformation respectively, and $\vec{T}$ is some transformation operator. Rigid-body rotation, scaling and shearing are examples of transformations which can be produced.

This simple general formulation can only be used when the point vectors are expressed in homogeneous coordinates. The many advantages of this form of representation are outlined by Ahuja and Coons (1968): basically, a point (x_o, y_o, z_o) is expressed as a vector (x, y, z, h), where $x_o = x/h$, $y_o = y/h$, $z_o = z/h$. Equation [28] is then written

$$[x'\ y'\ z'\ h'] = [x\ y\ z\ h] \left[\begin{array}{ccc:c} \alpha_{11} & \alpha_{12} & \alpha_{13} & \beta_1 \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \beta_2 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \beta_3 \\ \hdashline \gamma_1 & \gamma_2 & \gamma_3 & \delta \end{array}\right] \qquad \text{.....[29]}$$

The square matrix, T, can be regarded as made up of four partitions. The α-partition produces relative scaling of the x-, y-, and z-components, reflection, shearing, and rigid-body rotation; the β-partition produces perspective transformations; the γ-partition produces translation; the δ-partition produces absolute scaling. Each transformation is discussed in detail by Rogers and Adams (1976), who also provide algorithms. An important feature is that individual transformations can be concatenated. Rotation about an arbitrary axis, for instance, is produced by

(1) translation of the object so that the rotation axis passes through the origin of the coordinate system, followed by
(2) rotation through the required angle, followed by
(3) translation of the object back to its initial position [the reverse of (1)]; symbolically this is

$$\vec{X}' = \vec{X} \cdot [\text{Trans}] \cdot [\text{Rotat}] \cdot [\text{-Trans}] \quad .$$

Equation [28] represents the simplest type of coordinate transformation, and it is immediately relevant in fields such as structural geology (Ramsay, 1967). More sophisticated transforms can be developed, however, such as those involving mappings into non-Euclidean coordinate

systems (Morse and Feshbach, 1953). In particular, it is possible to devise purpose-oriented transforms in which the type and degree of transformation of any part of an object are made a function of its position within the object; in structural geology this would correspond to the case of inhomogeneous strain. By combining general- and special-purpose transforms, complex manipulations can be achieved. Processes such as the classic deformation grids of Thompson (1961; see also Bookstein, 1977) can be given a precise mathematical formulation.

VISUALIZATION

In a paper which has so strongly advocated modern mathematical techniques, it may seem strange to read that one of the most important manipulations is the production of pictures of an object. Yet the importance of the geologist being able to visualize effectively an object he has modelled can hardly be overstated. The apparent paradox is rooted in the way in which geology has developed.

From its earliest days geology has been a visually-oriented science (Rudwick, 1976). Reliance on the eye of an experienced geologist is the traditional geological approach. This paragon solves three-dimensional problems by looking at as much as is exposed, and then by following his practiced judgment in filling in the gaps. There is considerable justification for this, because the human visual perception system is probably the best pattern discriminator for all but the simplest of data sets. Yet this most efficient of systems is extremely prone to illusion and, in certain circumstances, may be unable to break away from preconceptions generated when objects are seen initially from unrepresentative viewpoints. [The definitive work on visual perception of form, its strengths and its weaknesses, is by Zusne (1970).] Elias (1972) (see also Tipper, 1978) has quoted a case of visual deception in stereology; Chadwick (1975, 1976) has demonstrated convincingly that geologists, despite their claims to the contrary, are as prone to visual illusion (and delusion) as anyone.

Realization that the purely visual approach to geology has these substantial drawbacks produced the inevitable backlash. In recent years for example, especially in the many uncritical applications of multivariate analysis to morphometric work, it has become commonplace to relegate visual judgment of shape to a relatively lowly role. Fortunately it is possible to steer a middle course between the extreme views. By using computerized modelling techniques to represent three-dimensional objects in an easily manipulated form, we provide ourselves with a most effective aid. For the first time, the geologist has an efficient way of constructing trial configurations of objects, of viewing the results, and hence of removing sources of potential deception. The computerized modelling system becomes part of an experimental laboratory.

The general problem of visualization is to produce pictures of an object as it would be seen from different viewpoints. The pictures may be either line-drawings, or half- or full-tone shaded representations. Three factors are involved in their creation: the projection of the three-dimensional object onto the appropriate two-dimensional viewing plane, the elimination of hidden lines and surfaces to enhance the realism of the image, and the provision of shading to simulate natural lighting effects.

Projection algorithms

Projection of an object onto a viewing plane is a relatively simple example of the transformations discussed in the previous section, and is considered in detail by Newman and Sproull (1973). Probably the most useful simple transformation for our purposes is illustrated in Figure 29. The line of sight along which the object is to be viewed (this is the normal to the viewing plane) is defined by two rotations about the origin of the coordinate system. The first is a rotation through ϕ-degrees about the z-axis; the second is a rotation through θ-degrees about the rotated x-axis. This transformation ensures that lines which are originally vertical appear vertical after projection. The transformed coordinates of a point on the object are

$$[x'\ y'\ z'\ h'] = [x\ y\ z\ 1] \begin{bmatrix} \cos\phi & \sin\phi & 0 & 0 \\ -\sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The actual projection onto the viewing plane may be either axonometric or perspective: Rogers and Adams (1976) discuss these at length. The simplest axonometric projection (the

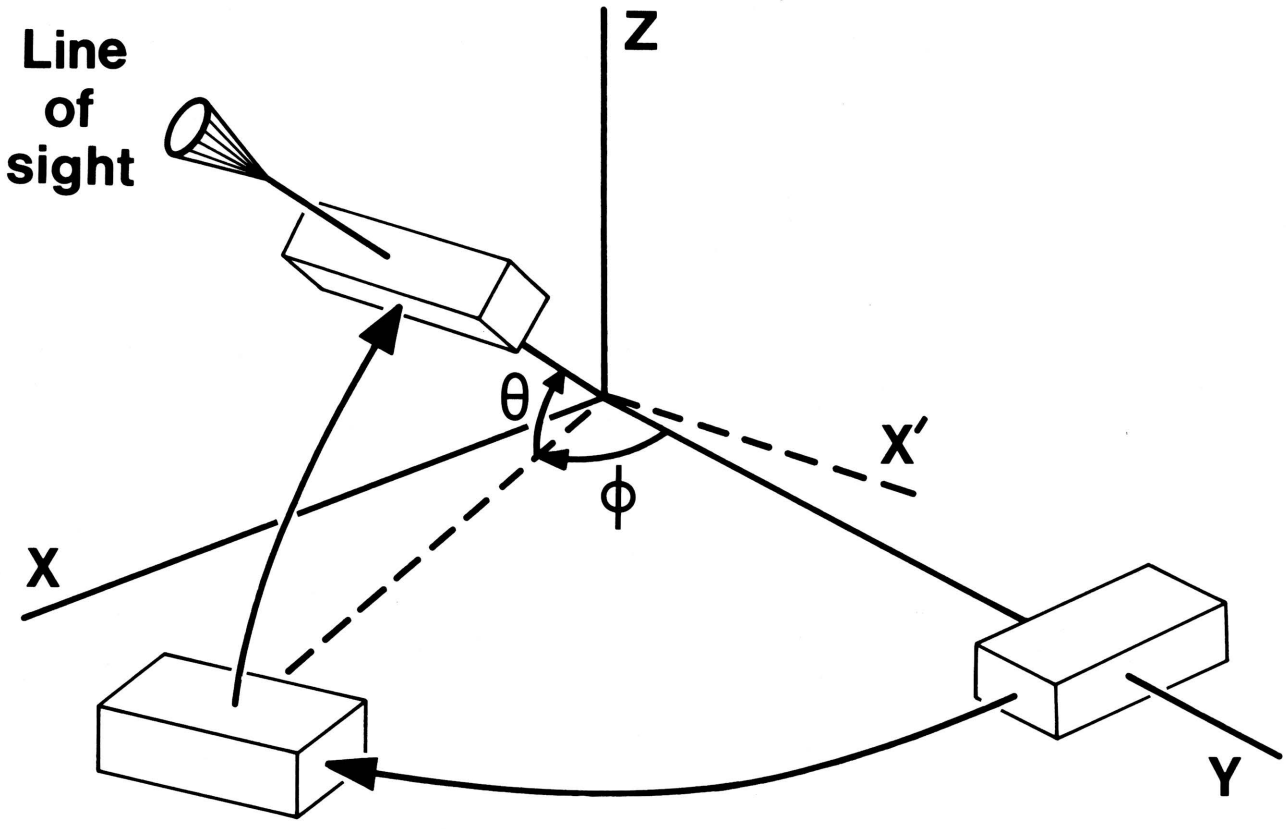


FIGURE 29. Simple viewing transformation. Object is rotated through ϕ degrees about the Z axis, and then through θ degrees about the rotated X axis (X').

trimetric projection along the z-axis) is

$$[x'' \; y'' \; z'' \; h''] = [x' \; y' \; z' \; h'] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The perspective projection is obtained by using non-zero elements in the β-partition of equation [29]. Perspective projection along the z-axis is given by

$$[x'' \; y'' \; z'' \; h''] = [x' \; y' \; z' \; h'] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the center of projection is located at z = -k.

A comparison of the trimetric and perspective projections is given in Figure 30 for the surface shown initially in Figure 15.

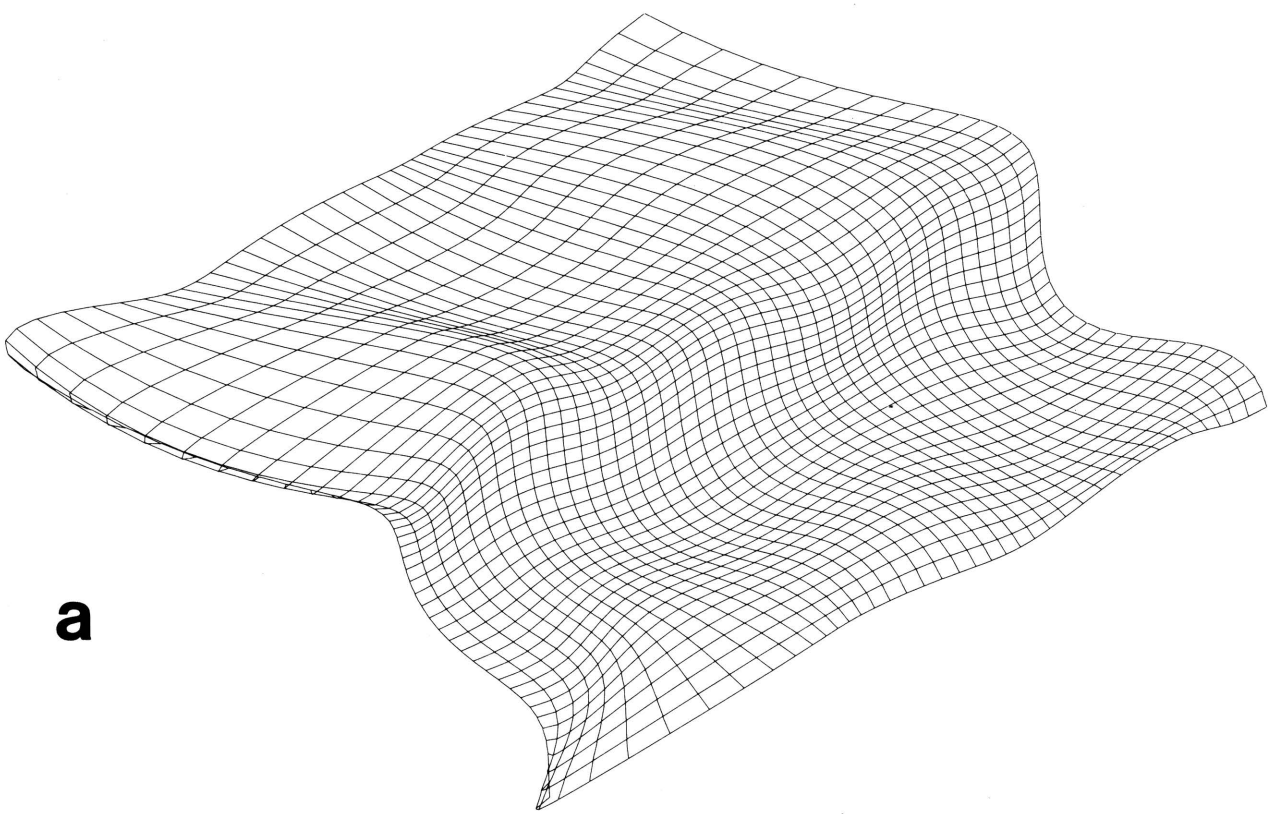


FIGURE 30. Comparison of (a) trimetric and (b) perspective projections of the same surface. The perspective projection is considerably more natural.

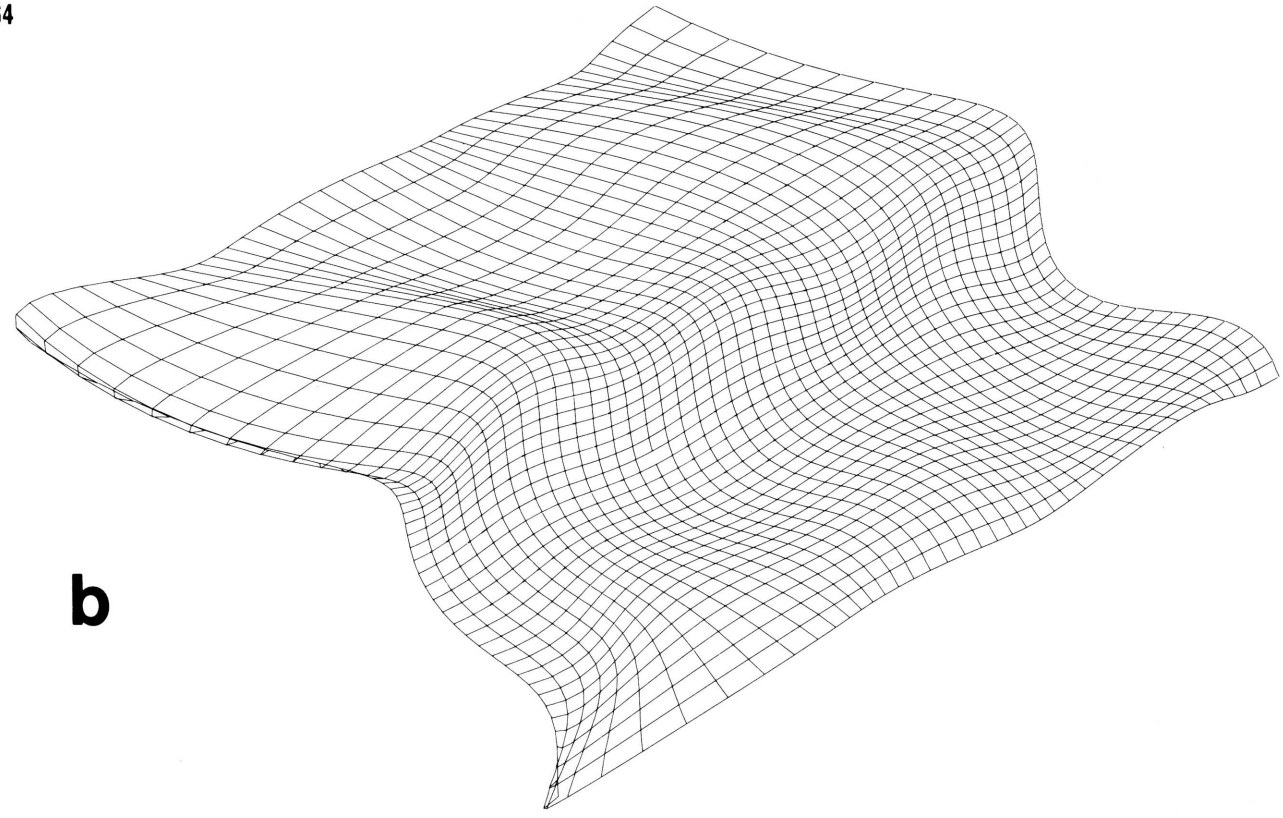


FIGURE 30. (b).

Elimination of hidden lines and surfaces

In contrast to projection, the hidden-line and hidden-surface problem is distinctly non-trivial, and still forms one of the major research areas in computer graphics. In hidden-line elimination, the outlines of objects are only drawn when not obscured by objects closer to the viewer; the final picture is composed of sets of abutting line segments. In hidden-surface elimination the object outlines are ignored, and each object is considered instead to be formed of opaque surfaces. Areas of the final picture are shaded (or colored) according to the surface to which they are closest.

Although conceptually straightforward, the problem is difficult to formulate for efficient computer solution, especially for complex multivalued surfaces and for real-time operation. Sutherland *et al.* (1974) have provided a valuable analysis of a representative selection of the available algorithms, and Clark (1976b) has considered some of the theoretical questions which may lead to more efficient solutions. It is worth pointing out here that for many simple one-off applications it may be most efficient to "paint-out" hidden lines at the production stage (Sprunt, 1975).

In two cases the hidden-line problem may be bypassed entirely. The first of these is when stereo-pairs of an object are used to give an illusion of depth (Rogers and Adams, 1976). The second case is when pictures are viewed in rapid succession. Then any slight rotation of the object will provide the eye with a depth-illusion. It is, of course, immaterial whether the pictures were actually produced in real-time; animated sequences of "still" pictures produce the same effect.

Shading algorithms

Shading of computer-produced pictures has often been attempted, using both single and multiple point illumination schemes. Some of the most successful early results were those of Gouraud (1971). Recently, Phong (1975) and Blinn (1977) have proposed more realistic lighting models, and the latter author has produced some very impressive shaded full-color pictures. Catmull (1975) and Blinn and Newell (1976) have shown how surfaces of computerized models may be given textures. The problem of introducing shadows into computer-synthesized images is analyzed by Crow (1977).

Examples

The bicubic patch model of the Kansas data set (Fig. 28) can also serve to illustrate projection and hidden-line removal. The views of the model were produced by trimetric projection onto specified viewing planes, followed by manual removal of hidden lines to give the illusion of opaque fences. A set of stereo-pairs for the bicubic B-spline model of the Kansas data set is shown in Figure 31.

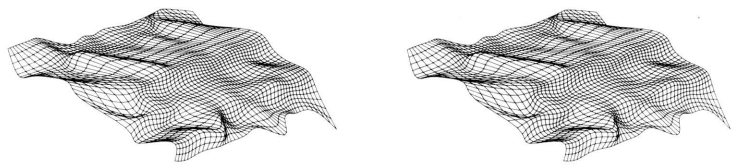


FIGURE 31. Stereo-pair of bicubic B-spline surface, the top surface of the Cherokee data set (Fig. 21a): $\phi = -100°$, $\theta = -50°$. This figure is for use with a standard pocket stereo-viewer.

REDUCTION

Up to now we have been concerned solely with manipulations of an entire model. There are, however, occasions when just part of a model must be studied. The modelled object must literally be taken apart. I have termed this type of manipulation "reduction" (Tipper, 1978). Included within it are processes such as cutting, exfoliation, and dissection. The implementation of each of these is as convenient on a computerized modelling system as it is inconvenient by any other approach.

The cutting process forms the basis of any use of sectioning. As examples may be quoted the synthesis of serial sections of a modelled fossil (Tipper, 1977), or the generation of artificial outcrop patterns by intersection of a modelled landform with a modelled body of rock. Exfoliation is the process of stripping material layer-by-layer from all or part of a model. Geological examples of this process are landform erosion, and the phenomenon of spheroidal weathering. Cutting processes break a model into parts, once and for all, whereas exfoliation processes involve removal of material in piecemeal fashion.

Dissection refers to the breaking up of a model into its constituent parts, much as in the anatomical usage of the term. Cutting and exfoliation processes take no account of the

internal structure of a model--sections, for instance, will commonly cut completely across patch boundaries. In contrast, the dissection process is controlled totally by the model's structure; a model constructed of surface patches can only be dissected into sub-assemblies of those patches.

The cutting process

Assume that the model to be cut apart is described by a mesh of surface patches (Fig. 32). The mathematical form of each patch, e.g. Coons or B-spline, is at this stage immaterial. Initially we consider just the problem of plane sectioning, because it is computationally less taxing than the general case of curved surface cutting curved surface. For the simplest of computerized models, those constructed only of bilinear patches, an efficient algorithm for plane sectioning has been developed by Cottafava and Le Moli (1969). This has been used for geological work (Tipper, 1977) but will not be considered further here.

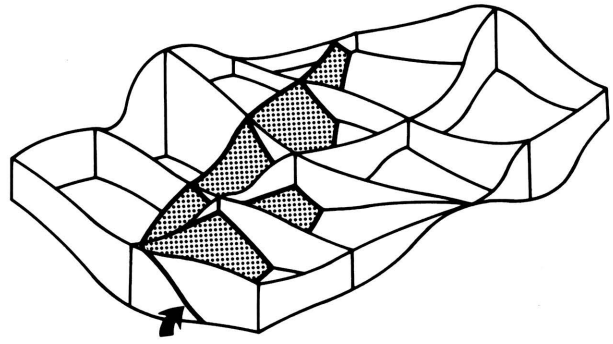


FIGURE 32. Plane sectioning through an object modelled as a mesh of bicubic patches. The sectioning plane is shown shaded, and its traces with the patches and with their associated vertical fences are shown by the heavy line.

The object of any sectioning algorithm is to calculate the space curve defined by the intersection of the surface of the object and the sectioning plane. In practice this curve can be approximated by a sequence of points ordered in space. Sabin (1968c, 1971b) has noted that this sequence can be generated in either of two ways. The first is to calculate the intersection points of the plane with selected isoparametric lines on the object's surface (these are the lines of constant u or v). These intersection points are sorted into order once all have been calculated. The advantages of this method are that it is computationally quite simple, and that it will work for every attitude of plane relative to surface. The disadvantages are that it is difficult to control the spacing between adjacent points on the resultant curve, and that in extreme cases it may be all but impossible to determine the correct way to order the points.

The second approach to sectioning is to determine initially the intersections of the plane with the boundaries of each patch, and then to use these as starting points for a "marching" process across the patch interior (Sabin, 1971b). Given a starting point, an estimate is made of the position of a point on the plane a given distance away within the patch.

This estimate is continually refined by an iterative minimization technique until some pre-specified conditions are met. The estimated point is then used as another starting point. The disadvantages of this method are that it is computationally more complex (with a certain number of special cases which must be given careful treatment), and that it is difficult to ensure that intersection curves will be calculated when the sectioning plane does not cut a patch boundary, even though it does cut the patch interior. Sabin (1968c) reported that this was not found to be a difficulty in practice. The main advantage of the "marching" approach is that the spacing of points can be easily controlled. This is especially important in areas of high curvature, and for applications where high precision is important.

Sabin (1971b) has pointed out that most working systems have used the former alternative (for example Payne, 1968). Here, however, we follow him in regarding point spacing on the intersection curve as of paramount importance, and so describe the "marching" algorithm in greater detail.

The "marching" method

The section plane is defined as the locus of a point $\vec{P}$, where

$$f = [\vec{P} - \vec{Q}] \cdot \vec{N} = 0$$

$\vec{Q}$ is a point known to lie on the plane, and $\vec{N}$ is the plane normal. For points not on the plane, f is their perpendicular distance to the plane. Because the sign of f is positive or negative depending on the side of the plane on which $\vec{P}$ lies, the intersection of a space curve, $\vec{P}(u)$, with the plane can be identified simply by looking for changes in the sign of f as u is varied.

To determine the intersection of the plane with the boundaries of a patch, each of the patch edges is treated in turn as the space curve, $\vec{P}(u)$. As an example, consider $\vec{P}(u,0)$. By testing successive points along the edge from u = 0 in steps of Δu, points $\vec{P}_1$ and $\vec{P}_2$ are determined which span f = 0. Let these have parametric coordinates $(u_1,0)$ and $(u_2,0)$ respectively: $u_2 - u_1 = \Delta u$. Then

$$f_1 = [\vec{P}_1 - \vec{Q}] \cdot \vec{N} \quad \text{and} \quad f_2 = [\vec{P}_2 - \vec{Q}] \cdot \vec{N}$$

$$f_1 \times f_2 < 0 .$$

The divided-difference method is now used to define a point $\vec{P}_0$ between $\vec{P}_1$ and $\vec{P}_2$ which is a close approximation to f = 0. Let $\vec{P}_0$ have parametric coordinates $(u_0,0)$. By linear interpolation in the u-parameter space

$$u_0 = (u_1 \times f_2 - u_2 \times f_1) / (f_2 - f_1)$$

The distance f_0 $(= [\vec{P}_0 - \vec{Q}] \cdot \vec{N})$ is then calculated. If it is sufficiently small, P_0 is taken as a starting point; otherwise u_0 is used to replace either u_1 or u_2 (depending on the sign of f_0) and the process repeated. When the starting point has been found, the actual "marching" process begins.

It is first necessary to calculate the surface normal vector, $\vec{N}_0$, at $\vec{P}_0$, and the unit vector which is tangential to the intersection, $\langle\vec{T}_0\rangle$. These are

$$\vec{N}_0 = [\vec{P}_u \times \vec{P}_v] \quad \text{and} \quad \langle\vec{T}_0\rangle = [\vec{N}_0 \times \vec{N}] \;/\; |[\vec{N}_0 \times \vec{N}]| \qquad .....[30]$$

where $\vec{P}_u$ and $\vec{P}_v$ are the partial derivatives of $\vec{P}$ with respect to the parameters u and v respectively.

We assume now that the distance between points (the step length) along $\vec{T}_0$ is required to be ℓ. Let a first approximation to the required point ℓ units away from $\vec{P}_0$ be $\vec{P}_1$. This has the coordinates $(u_1, v_1) = (u_0 + \delta u, v_0 + \delta v)$, where the increments δu and δv are

$$\delta u = \ell([\langle\vec{T}_0\rangle \times \vec{P}_v] \cdot \vec{N}_0) \;/\; (\vec{N}_0 \cdot \vec{N}_0) \quad \text{and} \quad \delta v = \ell([\langle\vec{T}_0\rangle \times \vec{P}_u] \cdot \vec{N}_0) \;/\; (\vec{N}_0 \cdot \vec{N}_0) \; .$$

A second approximation ($\vec{P}_2$) to the required point is now made. Its coordinates are $(u_1 + \delta u, v_1 + \delta v)$, where the increments δu and δv are determined by the two-variable form of Newton's method (Whittaker and Robinson, 1944, p. 90). First the actual step length at $\vec{P}_1$, ℓ_1, and the distance, f_1, from the sectioning plane are calculated. These are

$$\ell_1 = \sqrt{([\vec{P}_1 - \vec{P}_0] \cdot [\vec{P}_1 - \vec{P}_0])} \quad \text{and} \quad f_1 = [\vec{P}_1 - \vec{Q}] \cdot \vec{N} \qquad .....[31]$$

Differentiating,

$$\ell_u = \vec{P}_u \cdot \langle\vec{P}_1 - \vec{P}_0\rangle \; , \qquad \ell_v = \vec{P}_v \cdot \langle\vec{P}_1 - \vec{P}_0\rangle \; ,$$

and

$$f_u = \vec{P}_u \cdot \vec{N} \; , \qquad f_v = \vec{P}_v \cdot \vec{N} \; ,$$

where ℓ_u, ℓ_v, f_u, and f_v are the components of ℓ and f in the u and v directions respectively.

The first-order terms of a two-variable Taylor series about $\vec{P}_1$, for ℓ and f, are then:

$$\ell = \ell_1 + \ell_u \, \delta u + \ell_v \, \delta v$$

and

$$f = f_1 + f_u \, \delta u + f_v \, \delta v = 0 \; .$$

Thus,

$$\delta u = [-f_1 \, \ell_v - f_v(\ell - \ell_1)] \;/\; (f_u \, \ell_v - f_v \, \ell_u) \quad \text{and} \quad \delta v = [f_1 \, \ell_u + f_u(\ell - \ell_1)] \;/\; (f_u \, \ell_v - f_v \, \ell_u) \; .$$

The point $\vec{P}_2$ and the values of the distance, f_2, and the actual step length, ℓ_2, are now calculated. The lengths $|f_2|$ and $|\ell - \ell_2|$ are tested against zero. If they are both less than a prescribed tolerance, the iteration ends and $\vec{P}_2$ is output as an intersection point of plane and surface. $\vec{P}_2$ is then used to replace $\vec{P}_0$, and the "marching" process begins again (equation [30]). If not, $\vec{P}_2$ replaces $\vec{P}_1$ as the starting point for the Newton's method (equation [31]).

The "marching" process is repeated until a boundary of the patch is encountered. At that stage the complete intersection of patch with plane will have been determined. For a multi-patch surface, the method is simply applied to each patch in turn.

Non-planar cutting

The general case of cutting occurs when two curved surfaces are to intersect. The problem of determining the space curve of the intersection is considerably more complex than for plane sectioning. Unless the surfaces are modelled in a very simple way, a direct analytic solution to the problem is impossible because of the order of equations involved. Recourse to iterative

techniques is generally inevitable. One such approach has been suggested by Sabin (1968e) as a generalization of his plane sectioning algorithm. Others are no doubt possible.

One of the chief difficulties in devising an algorithm which will work in the general case is to ensure that all branches of the intersection curve are obtained. Figure 33 illustrates this problem: two surfaces, each modelled as a set of bicubic patches, cut at very low angles, and the intersection curve has three branches.

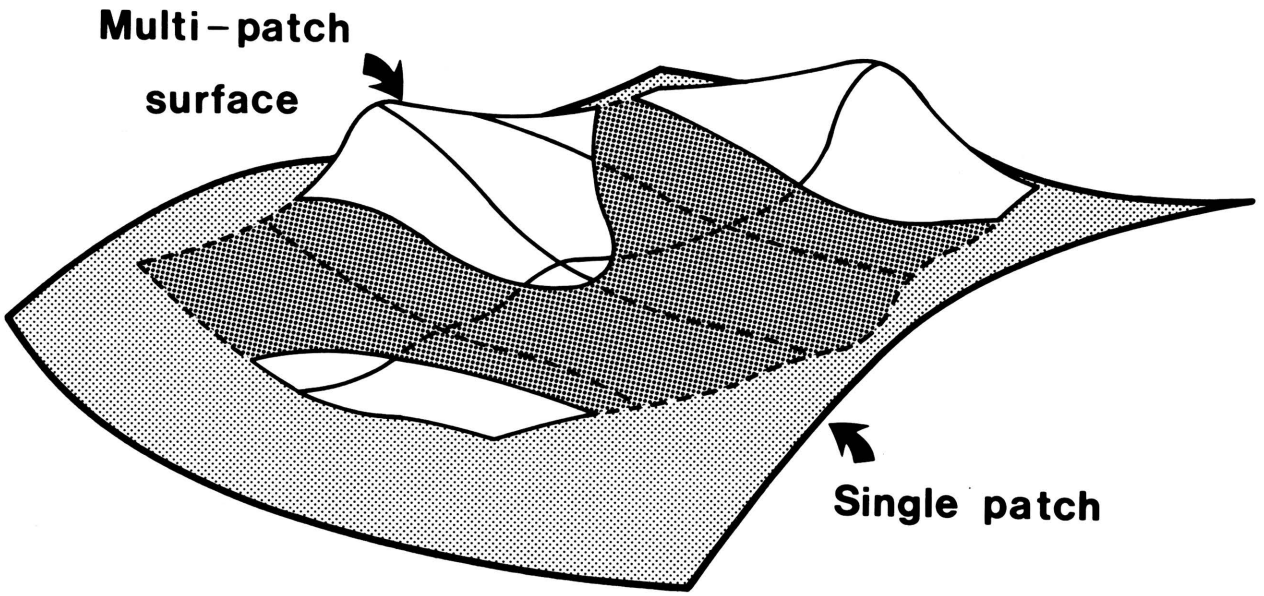


FIGURE 33. Non-planar intersection. One surface, the single bicubic patch defined by the heavier lines, intersects the multi-patch surface shown by the lighter lines. The angles of intersection are low, and the intersection space curve has three branches.

The exfoliation process

In the exfoliation process a surface must be defined which is offset to some extent from an initial surface. The amount of offset is usually measured perpendicular to the initial surface. The offset may either be constant over the whole surface (Fig. 34a), or it may vary as a function of position (Fig. 34b). Offsetting is routinely used in engineering design,

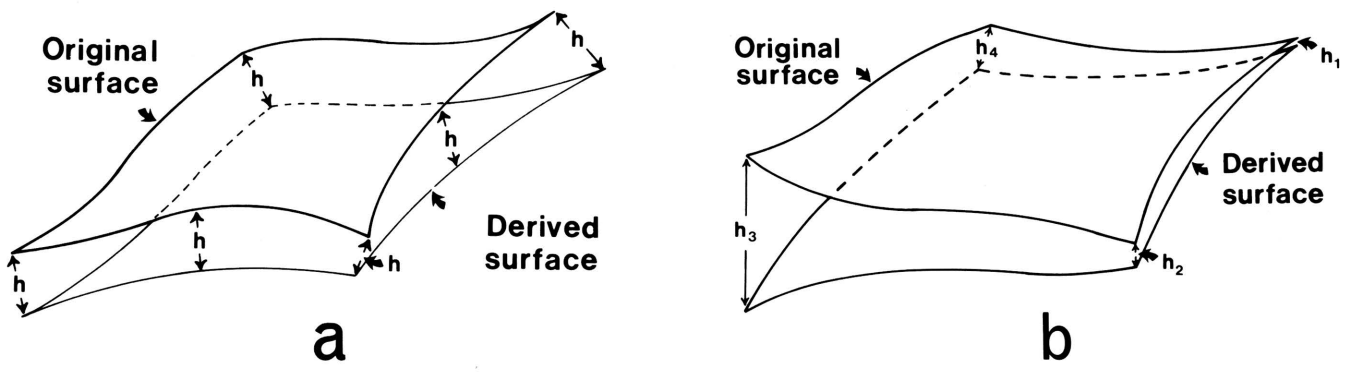


FIGURE 34. Exfoliation. (a) The offset, h, between the original and derived surfaces is constant over the whole surface. (b) The offset is dependent on position: $h_1 \neq h_2 \neq h_3 \neq h_4$.

and Sabin (1968d) has outlined an efficient algorithm. In a geological context, exfoliation is best viewed as a process for removal of a layer of material from an object's surface, e.g. land-form erosion. There is, however, no reason to restrict it just to removal of material; the method is entirely reversible and may, for instance, be simply applied to the modelling of sediment depositional systems (Tipper, 1978).

Sabin's algorithm for the exfoliation process is as follows. Assume that the initial surface is $\vec{P}(u,v)$. The surface which is offset from it by a distance h is $\vec{P}^*(u,v)$, where

$$\vec{P}^* = \vec{P} + h\,\langle\vec{N}\rangle \qquad .....[32]$$

$\vec{N}$ $(= \vec{P}_u \times \vec{P}_v)$ is the surface normal at $\vec{P}$. $(\vec{P}_u, \vec{P}_v, \vec{P}_{uv})$ and $(\vec{N}_u, \vec{N}_v, \vec{N}_{uv})$ are the partial derivatives of $\vec{P}$ and $\vec{N}$ respectively.

In order that the offset surface can be utilized in further manipulations, it is necessary to develop expressions for its derivatives; these are denoted by $\vec{P}^*_u$, $\vec{P}^*_v$, and $\vec{P}^*_{uv}$, and are calculated by differentiation of equation [32]. Sabin (1968d) gives the complete derivation of $\vec{P}^*_u$ and $\vec{P}^*_v$. We find that

$$\vec{P}^*_u = \vec{P}_u + \langle\vec{N}\rangle\{h_u - h(\vec{N}\cdot\vec{N}_u)/(\vec{N}\cdot\vec{N})\} + h\vec{N}_u/|\vec{N}|$$

$$\vec{P}^*_v = \vec{P}_v + \langle\vec{N}\rangle\{h_v - h(\vec{N}\cdot\vec{N}_v)/(\vec{N}\cdot\vec{N})\} + h\vec{N}_v/|\vec{N}|$$

$$\vec{P}^*_{uv} = \vec{P}_{uv} + \langle\vec{N}\rangle\{h_{uv} - h_u(\vec{N}\cdot\vec{N}_v)/(\vec{N}\cdot\vec{N}) - h_v(\vec{N}\cdot\vec{N}_u)/(\vec{N}\cdot\vec{N}) - h[(\vec{N}\cdot\vec{N}_{uv})/(\vec{N}\cdot\vec{N})$$
$$+ (\vec{N}_u\cdot\vec{N}_v)/(\vec{N}\cdot\vec{N}) - 3(\vec{N}\cdot\vec{N}_u)(\vec{N}\cdot\vec{N}_v)/(\vec{N}\cdot\vec{N})^2]\} + \vec{N}_u\{h_v - h(\vec{N}\cdot\vec{N}_v)/(\vec{N}\cdot\vec{N})\}$$
$$/\,|\vec{N}| + \vec{N}_v\{h_u - h(\vec{N}\cdot\vec{N}_u)/(\vec{N}\cdot\vec{N})\}/|\vec{N}| + h\vec{N}_{uv}/|\vec{N}|$$

$$.....[33]$$

When the offset is constant, the derivatives h_u, h_v, and h_{uv} are zero. Equations [33] are consequently greatly simplified. In the general case, however, h is a function of $\vec{P}$, i.e. $h = h(\vec{P})$. The derivatives of h are

$$h_u = \text{grad}(h)\cdot\vec{P}_u$$

$$h_v = \text{grad}(h)\cdot\vec{P}_v$$

$$h_{uv} = \text{grad}(h)\cdot\vec{P}_{uv} + \text{grad}(h_v)\cdot\vec{P}_u = \text{grad}(h)\cdot\vec{P}_{uv} + \text{grad}(h_u)\cdot\vec{P}_v$$

These are inserted into equations [33]. Sabin (1968d) has pointed out that in the general case the offset is no longer along the surface normal, $\vec{N}$, because the normals for $\vec{P}$ and $\vec{P}^*$ are no longer identical. For strict accuracy it is necessary to make preliminary calculations of $\vec{P}^*_u$ and $\vec{P}^*_v$, and then to use $[\vec{P}^*_u \times \vec{P}^*_v]$ instead of $\vec{N}$. The reader is referred to Sabin (1968d) for more detailed consideration of these aspects.

The dissection process

Of all the facets of reduction, the dissection process emphasizes most clearly the practical advantages of computerized modelling. Consider as an example of its use a project involving precise determination of feeding currents produced within a brachiopod shell by valve closure, e.g. the work on *Richthofenia* by Rudwick and Cowen (1967). One of the first tasks in the project must be to study the action of the hinge structure so that the precise way in which the valves articulate can be determined. All that is required here is detailed study of one small part of the shell--but a part which is notoriously difficult to examine by conventional means because by its very nature it is obscured by the valve exterior. Were, however, the brachiopod to have been modelled first in computerized form, such obscuring features could be removed from view, together with other irrelevant structures. This type of computerized dissection has been used very effectively by Westbroek in his work on the structures of the rhynchonellid commissure (Westbroek, *et al.*, 1976).

The mechanics of dissection of a patch-based computerized model are quite straightforward. All that is necessary is that the user be able to specify which parts of the model he wishes to remove, and which parts he wishes to retain. Perhaps the simplest way of doing this is to refer to each patch by its coordinates in the mesh. In more complex systems (for example the NMG system, Sabin, 1968a), models are built up of individual parts which have alphanumeric names assigned to them by the user.

Example

Examples of plane sectioning through the bicubic B-spline model of the Cherokee data set are shown in Figure 35. In Figure 35f and 35g, the sectioning plane has been used as a fault plane, along which an offset has been specified in magnitude and direction.

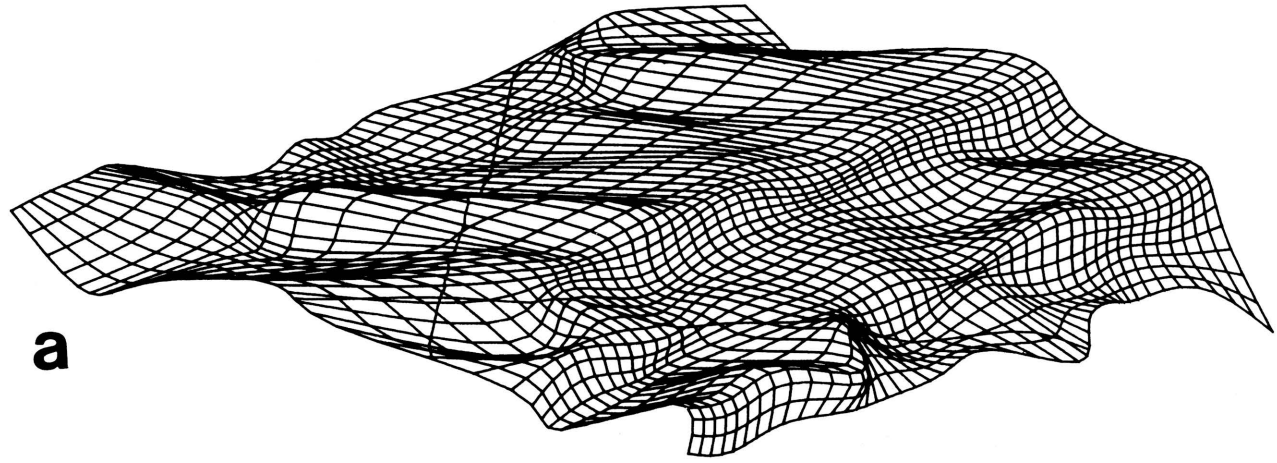
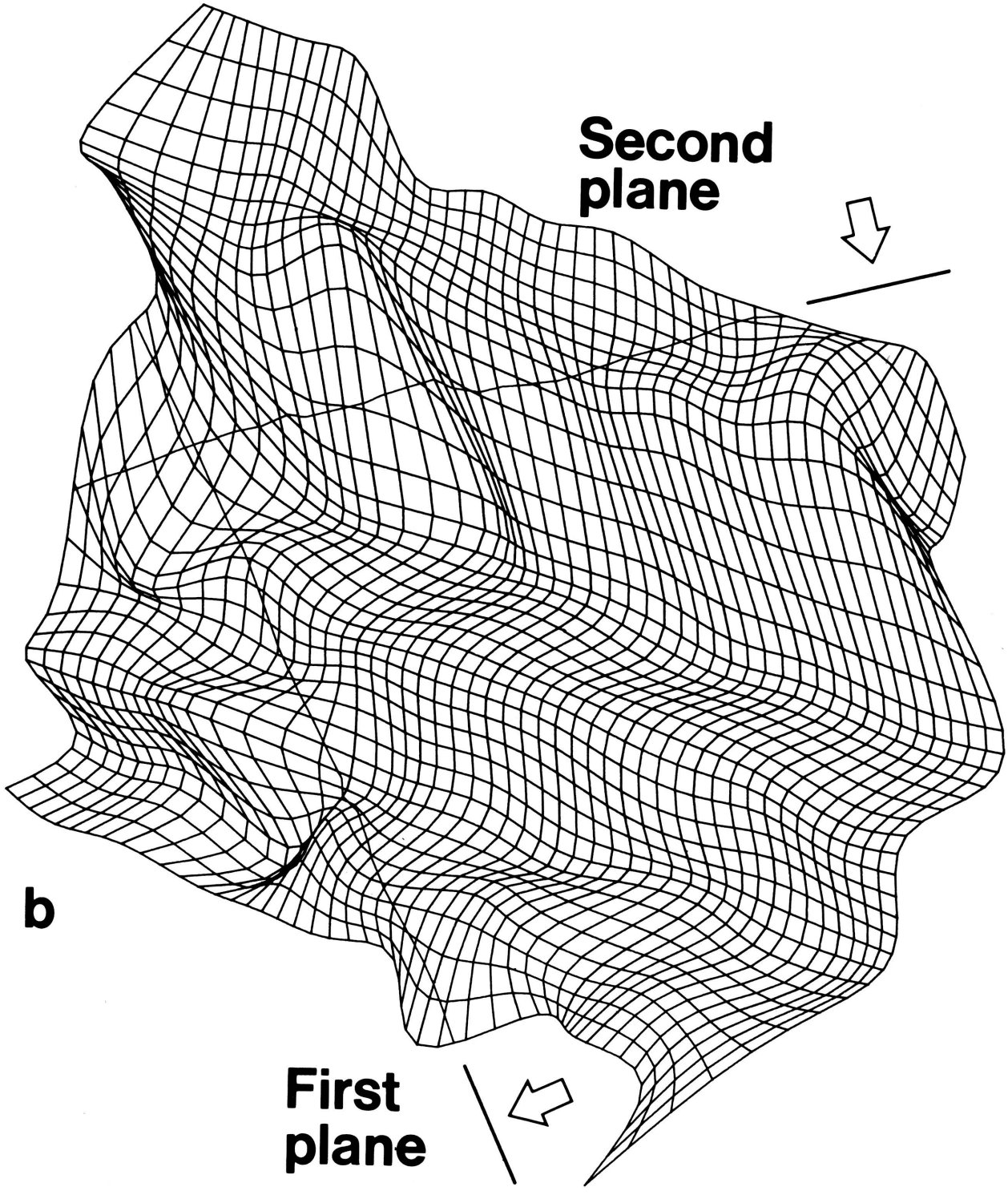
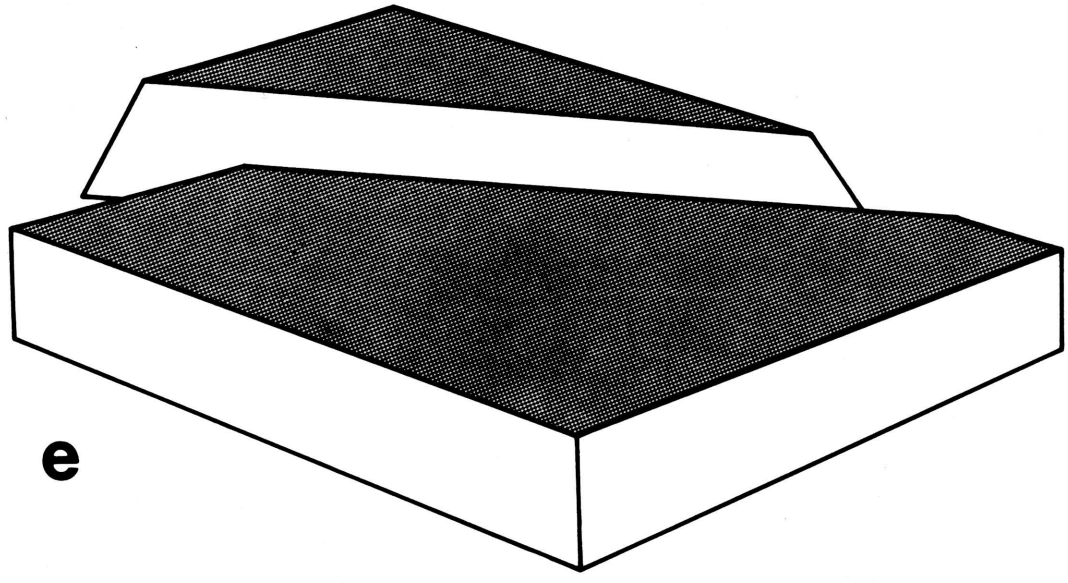


FIGURE 35. Manipulations of the bicubic B-spline model of the Cherokee data set. (a), (b) Trimetric projections of the top surface, with intersections of two vertical sectioning planes superimposed. The first plane [shown in elevation in (c)] has normal (1, 0, 0, -12); the second plane [shown in elevation in (d)] has normal (0, 1, 0, -12). (a) $\phi = -100°$, $\theta = -30°$. (b) $\phi = -160°$, $\theta = -50°$. (c), (d) Elevations of the sections shown in (a) and (b): in each case the viewing direction is given by the arrows in (b). (e) Block diagram showing fault movement to be produced in the model: the fault plane normal is (1, 0, 1, -8). (f) Trimetric projection of top surface, after faulting [cf. (e)]. $\phi = -100°$, $\theta = -30°$. (g) Trimetric projection of top surface, after faulting. $\phi = -160°$, $\theta = -50°$.

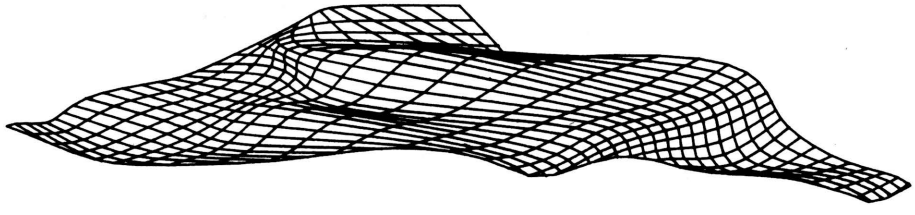

Second
plane
b
First
plane

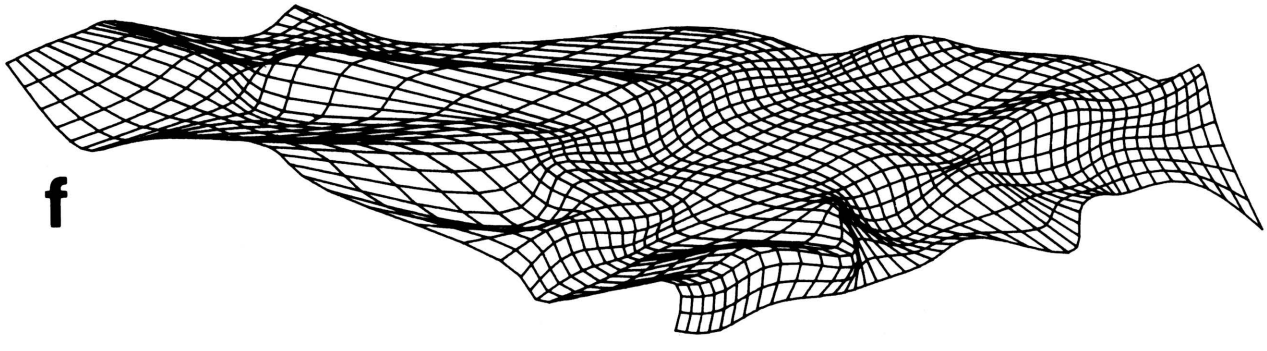
c

d



e





f

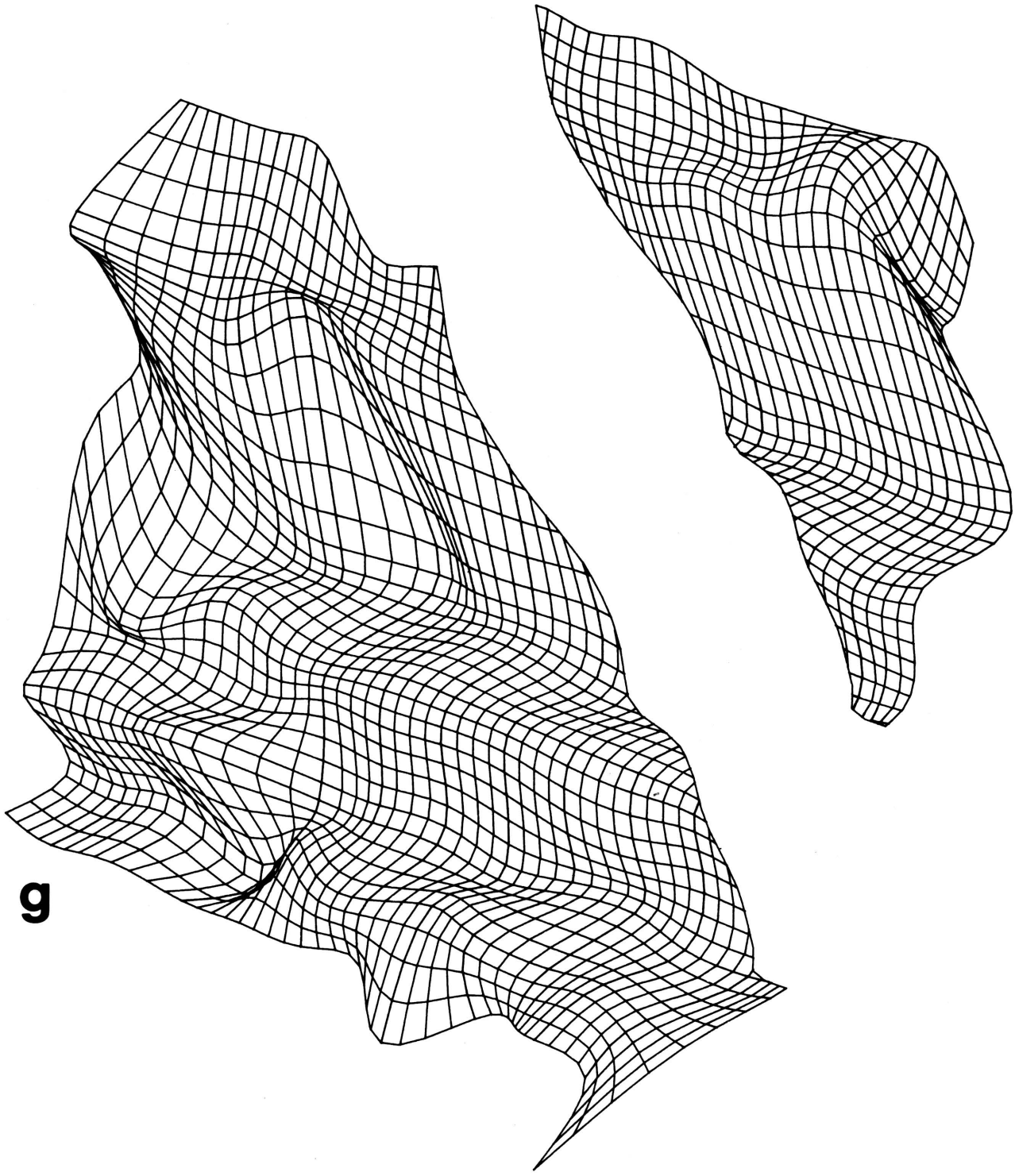


FIGURE 35. (c, d, e, f, g).

Simulation

Geology is hardly an experimental science. On the contrary, we observe today in the geological record the results of "experiments" performed millions of years ago under conditions of which we have little knowledge and over which we have no control. In trying to work out what actually happened in the geologic past, we must seek to identify processes which can produce results analogous to those found in our geological observations. This is the classic uniformitarian approach of Hutton and Lyell.

For phenomena at the smallest scale we can attempt to duplicate the field and laboratory observations by running the process under controlled conditions--the obvious example here is the use of a flume tank to imitate aspects of clastic sedimentation. When the phenomena are at a larger scale, either in their temporal or spatial dimension, the experimental aspect can be retained by developing computer simulations of the process. Harbaugh and Bonham-Carter (1970) have documented some elegant examples of this approach, and Raup's work on the simulation of shell coiling in molluscs is also noteworthy (Raup, 1966, 1967).

The relevance of the simulation approach to the field of computerized modelling is clear--here, of course, we refer exclusively to simulations of morphology. Examples might include the changing form of a shingle spit (McCullagh and King, 1970), the development of land forms by glacial erosion, or the growth of porphyroblasts during metamorphism (P. K. Harvey, 1973, unpublished report, Kansas Geological Survey). Whereas a computerized model describes an object by its final shape, a morphological simulation defines instead an initial condition and a set of rules for growth. Given these, the final shape is always to some degree predictable. A corollary of this is that specification of the initial and final shapes inevitably circumscribes the growth rules. Thus a computerized model is a logical result of a simulation of morphology, just as the model itself may act as a guide to the ground rules for the simulation. Simulation and computerized modelling are in this respect no more than alternative formulations of the same basic idea.

Clearly the field of simulation is so wide and its methods so varied that it is unreasonable to devote more than a passing reference to the practical interface between computerized modelling and simulation. The important feature to note is that the two methods tend in practice to dovetail neatly together. This indeed is to be expected because a morphological simulation (whether stochastic or deterministic) is of necessity a regular process. In consequence, its results can in most cases be translated logically into the regularity of a model.

Mensuration

Throughout this paper we have made one tacit assumption, namely that the models discussed are not *ad hoc* representations of shape. They are complete mathematical descriptions, and as such can be measured with an ease and precision limited only by the sophistication of the computer system. The importance of being able to make measurements hardly needs elaboration; methods used for numerical classification and analysis of shape (for instance those referred to in the Introduction to this paper) can utilize measurements made from models just as easily as those made from real objects. Models can thus be compared with their real-world counterparts.

An example of the use of measurements derived from a computerized model has been given by Tipper (1977). In that case it was desired to study the variability of the dip and strike of individual cross-strata exposed in a single well core. Each cross-stratum was modelled by a single bicubic patch, the form of which was determined by the trace of the stratum on the periphery of the core. Because the mathematical form of each patch was known, it was possible to predict its attitude at any point. Thus the dip and strike of each stratum could be determined precisely at points within the core, where the stratum was of course unexposed. In this instance, the computerized modelling system provided a measurement facility unobtainable in any other way.

In making punctual or linear measurements on a model, one problem continually recurs. This is the problem of identifying particular points on a surface when their parametric coordinates are unknown. Clearly, if the maximum projected width of a model is to be measured, the coordinates of the leftmost and rightmost points on its surface must first be found. These will not in general be known in parametric terms. In the same way, if the thickness of a modelled sand body which will be penetrated by an almost perpendicular well of known position and attitude is to be determined, the parametric coordinates of the top and bottom points through which the well will pass must be found.

Fortunately most of these specific cases can be resolved by using a few basic techniques. These can be combined in various ways to give more complex manipulations. Two of the most useful techniques are the determination of the nearest point on a surface to a given point, $\vec{Q}$, and the determination of the point(s) of intersection of a straight line with a surface. Each of these can in fact be closely related to part of Sabin's "marching" method for plane sectioning.

The nearest point algorithm

Sabin (1968c) developed the procedure known as the nearest point algorithm. Let the surface be $\vec{P}(u,v)$, and the fixed point be $\vec{Q}$. The distance between $\vec{Q}$ and a point on the surface is

$$f = [\vec{P} - \vec{Q}] \cdot [\vec{P} - \vec{Q}]$$

The function f is to be minimized. This implies that its derivatives must be zero, i.e. that

$$f_u = 2[\vec{P} - \vec{Q}] \cdot \vec{P}_u = 0$$

and

$$f_v = 2[\vec{P} - \vec{Q}] \cdot \vec{P}_v = 0 \quad .$$

These expressions may be written in general form as

$$a = \vec{P}_u \cdot [\vec{P} - \vec{Q}] \qquad \text{and} \qquad b = \vec{P}_v \cdot [\vec{P} - \vec{Q}] \quad .$$

Their derivatives are

$$a_u = \vec{P}_u \cdot \vec{P}_u + \vec{P}_{uu} \cdot [\vec{P} - \vec{Q}] \ , \qquad a_v = \vec{P}_u \cdot \vec{P}_v + \vec{P}_{uv} \cdot [\vec{P} - \vec{Q}] \ ,$$

$$b_u = \vec{P}_v \cdot \vec{P}_u + \vec{P}_{vu} \cdot [\vec{P} - \vec{Q}] \quad (= a_v) \ , \qquad b_v = \vec{P}_v \cdot \vec{P}_v + \vec{P}_{vv} \cdot [\vec{P} - \vec{Q}] \ .$$

Assume now that an initial approximation can be given to the required point, say $P(u_0,v_0)$. The values of 'a' and 'b' at this point are a_0 and b_0 respectively. A second approximation

to the point, $P(u_0+\delta u, v_0+\delta v)$, may then be obtained by using Newton's method as before. Thus

$$\begin{bmatrix}\delta u\\ \delta v\end{bmatrix} = \begin{bmatrix}a_u & a_v\\ b_u & b_v\end{bmatrix}^{-1}\begin{bmatrix}a_0\\ b_0\end{bmatrix}$$

The minimization process continues until the change in f becomes sufficiently small. It is clear that the latter part of the sectioning algorithm is identical to this process. In that case the function 'a' was the discrepancy in step length from the ideal, and the function 'b' was the distance of the point from the sectioning plane.

The intersection of a straight line with a surface

Following the development by Sabin (1968e), let the surface be $\vec{P}(u,v)$, and the line be $[\vec{P} - \vec{Q}] \times \vec{T} = 0$. The line is a vector passing through $\vec{Q}$ with direction numbers $\vec{T}$. The perpendicular distance of a point $\vec{P}$ from the line is

$$f = \{[\vec{P} - \vec{Q}] \times \vec{T}\} \cdot \{[\vec{P} - \vec{Q}] \times \vec{T}\}$$

As before, f is to be minimized. This implies that its derivatives must be zero, i.e. that

$$f_u = [\vec{P}_u \times \vec{T}] \cdot \{[\vec{P} - \vec{Q}] \times \vec{T}\} = 0$$

and

$$f_v = [\vec{P}_v \times \vec{T}] \cdot \{[\vec{P} - \vec{Q}] \times \vec{T}\} = 0 \quad .$$

Again these are written in the general form as

$$a = [\vec{P}_u \times \vec{T}] \cdot \{[\vec{P} - \vec{Q}] \times \vec{T}\} \qquad \text{and} \qquad b = [\vec{P}_v \times \vec{T}] \cdot \{[\vec{P} - \vec{Q}] \times \vec{T}\} \quad .$$

Their derivatives are

$$a_u = [\vec{P}_u \times \vec{T}] \cdot [\vec{P}_u \times \vec{T}] + [\vec{P}_{uu} \times \vec{T}] \cdot \{[\vec{P} - \vec{Q}] \times \vec{T}\} \quad ,$$

$$a_v = [\vec{P}_u \times \vec{T}] \cdot [\vec{P}_v \times \vec{T}] + [\vec{P}_{uv} \times \vec{T}] \cdot \{[\vec{P} - \vec{Q}] \times \vec{T}\} \quad (= b_u) \quad ,$$

$$b_v = [\vec{P}_v \times \vec{T}] \cdot [\vec{P}_v \times \vec{T}] + [\vec{P}_{vv} \times \vec{T}] \cdot \{[\vec{P} - \vec{Q}] \times \vec{T}\} \quad .$$

The solution proceeds in the same way as that for the nearest point algorithm.

SUMMARY

(1) Studying the shape of complex three-dimensional objects and surfaces forms a major part of much geological work. Two approaches can be used: formal numerical analysis, and methods of surface representation. The latter methods involve the construction of mathematical (and necessarily computerized) models of the objects being studied; this approach is the more efficient when the objects have relatively complex curved surfaces.

(2) The general problem of surface representation is, given the coordinates of points assumed to lie on one simply connected surface, to describe that surface in such a way that it can be manipulated easily and precisely. Two steps are involved: the first is one of modelling, the second one of manipulating the models.

(3) The process of modelling involves defining some interpolation or approximation function that will predict the location of points on the surface away from the data points. The form of this function (the surface model) is determined both by the intrinsic nature of the surface, and by the purpose for which the model is being made.

(4) The modelling method most widely used in geology is trend surface analysis, the fitting of a certain class of surface model to sets of scattered data points. If the number of data points exactly equals the number of degrees of freedom in the model, an exact fit can be obtained. If not, then the method of least-squares can be used to give a model that is optimal under certain conditions. Because these conditions are rarely fulfilled in practice, most least-squares models are just as arbitrary as surfaces fitted by eye.

(5) Regionalized Variable Theory can also be used to generate surface models that have some optimal properties. In this approach, the interpolation function at any point is taken to be a random variate, the value of which depends on its position. If the statistics of the variate are known, its value at any given location can be estimated; if they are not (as is often the case in practice), the estimates will be quite arbitrary.

(6) For a modelling method to be generally applicable, certain minimum requirements can be prescribed. (a) It should produce surfaces that are locally smooth, i.e. that are continuous both in value and first derivative. (b) These surfaces should, if necessary, be capable of being multiple-valued in any coordinate direction. (c) The form of the model should be independent of its orientation. These requirements are best satisfied by developing models in a curvilinear coordinate system.

(7) Modelling methods can have either a global or a local basis. Global-basis methods use the same interpolation (or approximation) function throughout the model, whereas local-basis

(piecewise) methods partition the model into local domains and define separate functions (termed "surface patches") for each of these.

(8) The most convenient curvilinear coordinate representation (and one that is usable both for global- and local-basis models) organizes the data points into some form of network, usually the topological equivalent of a planar rectangular grid. Each cell in the network is quadrilateral in plan view, and its edges form the axes of a two-parameter curvilinear coordinate system. Each cell can also, if required, be used as the domain for a single surface patch.

(9) Use of this network approach reduces the surface representation problem to a standard one of bivariate interpolation (or approximation) over a quasi-planar grid. With a network of rectangular cells, four modes of interpolation are possible; each may be either Lagrangian or Hermitian, depending on the smoothness required in the final model. The most general mode, the Boolean sum, interpolates both the corner points and the edge curves of each cell. With certain restrictions made on the edge curves, either of two other modes are obtained, the lofted surfaces that interpolate only opposing edge curves, or the most restricted mode, the Cartesian product, which interpolates only the cell corner points.

(10) Modelling methods developed for the purpose of computer-aided design (CAD) can also be used for surface representation. They may be either interpolation or approximation methods, and may have either a global or a local basis. They are generally of Cartesian product (rarely Boolean sum) form.

(11) Bézier surfaces are a Cartesian product implementation of Bernstein polynomial approximation in a curvilinear coordinate system. Their global basis renders them too inflexible for many surface representation purposes.

(12) The B-spline basis can be used as an appropriate local-basis equivalent of the Bernstein basis. Cartesian product B-spline surfaces are local-basis approximants. They are computationally efficient, and very suitable for surface representation unless matching to arbitrarily shaped boundaries is required.

(13) Global-basis spline interpolation can also be used as a modelling method, and can be implemented either in a rectangular Cartesian or a curvilinear coordinate system. The use of bicubic splines requires that derivatives be specified at the network boundary; this is usually inconvenient.

(14) Coons surfaces are the most widely used CAD method. In their strict sense they are local-basis Boolean sum interpolants. The Coons bicubic patch is a particular specialization, a Cartesian product surface that can be derived from global-basis bicubic spline interpolation. A substantial advantage of the Coons formulation, and one that makes it ideal for use in surface representation, is that it can be used to blend surfaces into boundary curves of

known form. These space curves can themselves be represented by B-splines, by parabolic blended curves (Overhauser curves), or by local-basis interpolating splines (Catmull-Rom splines).

(15) Constructing surface models is not an end in itself. It is essential also that they can be manipulated in such a way as to supply answers to questions of interest. But, in contrast to the construction of models, this manipulation process has received relatively little attention.

(16) Five categories of manipulation can be identified: transformation, visualization, reduction, simulation, and mensuration.

(17) Transformation refers simply to the implementation of standard coordinate transforms, both in Euclidean and non-Euclidean coordinate systems.

(18) Visualization is the production of pictures of a model as it would be seen from different viewpoints. Three steps are involved: projection of the three-dimensional model onto the appropriate two-dimensional viewing plane, elimination of hidden lines and surfaces to enhance the realism of the image, and provision of shading to simulate natural lighting effects.

(19) Reduction includes the processes of cutting (both plane sectioning and the intersection of two curved surfaces), exfoliation, and dissection. Exfoliation is the process of stripping material layer-by-layer from all or part of a model. Dissection refers to the breaking up of a model into its constituent parts.

(20) Any morphological simulation operates by applying a set of growth rules to an initially defined shape. The final shape is always to some degree predictable. A surface model, on the other hand, by describing that final shape in detail, reflects the set of rules by which it was obtained. Simulation and modelling are little more than alternative formulations of the same basic idea.

(21) Mensuration refers to the processes of obtaining measurements from models, for example lengths, areas, and volumes.

ACKNOWLEDGMENTS

The work on which this report is based was funded by programs of the Geologic Research Section, Kansas Geological Survey. My thanks go to the Survey Director, Bill Hambleton, and to his staff, both research and technical, for their continued support. In particular I thank my Survey colleagues, Curt Conley, John Davis, John Doveton, and Jim Surber for innumerable discussions; Jo Anne DeGraffenreid and Kaye Long for grappling with the manuscript; Harold Cable, Joan Jaeger, and Chris Roche for grappling with the computer; and Carla Kuhn for producing the cover.

I thank Ian Braid (Cambridge University CAD Group), Mack C. Colt Inc. (Iola, Kansas), Ray Davies (BAC, Weybridge), George Dodd (General Motors), Walter Johnson (Ford Motor Company), and Malcolm Sabin (Kongsberg) for their assistance in obtaining unpublished reports and data.

John Davis and Lois Mansfield (Dept. Computer Science, University of Kansas) made helpful comments on the manuscript.

REFERENCES

Ahlberg, J. H., 1974, A picture-view of splines: Tech. Rept., Division of Applied Mathematics, Brown University, 115 p.

Ahlin, A. C., 1964, A bivariate generalization of Hermite's interpolation formula: Mathematics of Computation, v. 18, p. 264-273.

Ahuja, D. V., and S. A. Coons, 1968, Geometry for construction and display: IBM Systems Journal, v. 7, p. 188-205.

Akima, H., 1970, A new method of interpolation and smooth curve fitting based on local procedures: Journal of the Association for Computing Machinery, v. 17, p. 589-602.

_____, 1974, A method of bivariate interpolation and smooth surface fitting based on local procedures: Communications of the Association for Computing Machinery, v. 17, p. 18-20.

_____, 1975, Comments on 'Optimal Contour Mapping using Universal Kriging' by Ricardo A. Olea: Journal of Geophysical Research, v. 80, p. 832-834.

Anderson, W. L., 1971, Application of bicubic spline functions to two-dimensional gridded data: Rept. GD-71-022, U. S. Geological Survey, 36 p.

Barnhill, R. E., 1974, Smooth interpolation over triangles, *in* Barnhill, R. E., and R. F. Riesenfeld (eds.), Computer Aided Geometric Design: Academic Press, New York, p. 45-70.

_____, 1977, Representation and approximation of surfaces, *in* Rice, J. R. (ed.), Mathematical Software III: Academic Press, New York, p. 69-120.

Bass, N. W., 1936, Origin of the shoestring sands of Greenwood and Butler counties, Kansas: Kansas Geological Survey Bulletin 23, p. 1-135.

Bézier, P., 1972, Numerical Control: John Wiley & Sons, London, 240 p.

_____, 1974, Mathematical and practical possibilities of UNISURF, *in* Barnhill, R. E., and R. F. Riesenfeld (eds.), Computer Aided Geometric Design: Academic Press, New York, p. 127-152.

Bhattacharyya, B. K., 1969, Bicubic spline interpolation as a method for treatment of potential field data: Geophysics, v. 34, p. 402-423.

Birkhoff, G., 1969, Piecewise bicubic interpolation and approximation in polygons, *in* Schoenberg, I. J. (ed.), Proceedings of the Symposium on Approximation with Special Emphasis on Spline Functions [Madison, Wisconsin, 1969]: Academic Press, New York, p. 185-221.

_____, and C. R. de Boor, 1965, Piecewise polynomial interpolation and approximation, *in* Garabedian, H. L. (ed.), Approximation of Functions: Elsevier, Amsterdam, p. 164-190.

_____, and H. L. Garabedian, 1960, Smooth surface interpolation: Journal of Mathematics and Physics, v. 39, p. 258-268.

_____, and L. Mansfield, 1974, Compatible triangular finite elements: Journal of Mathematical Analysis and Applications, v. 47, p. 531-553.

Blackith, R. E., and R. A. Reyment, 1971, Multivariate Morphometrics: Academic Press, London, 412 p.

Blinn, J. F., 1977, Models of light reflection for computer synthesized pictures: Computer Graphics, v. 11, p. 192-198.

_____, and M. E. Newell, 1976, Texture and reflection in computer generated images: Communications of the Association for Computing Machinery, v. 19, p. 542-547.

Bookstein, F. L., 1977, The study of shape transformation after D'Arcy Thompson: Mathematical Biosciences, v. 34, p. 177-219.

Braid, I. C., 1975, The synthesis of solids bounded by many faces: Communications of the Association for Computing Machinery, v. 18, p. 209-216.

Brewer, J. A., and D. C. Anderson, 1977, Visual interaction with Overhauser curves and surfaces: Computer Graphics, v. 11, p. 132-137.

Brower, J. C., 1973, Ontogeny of a Miocene pelecypod: Journal of the International Association for Mathematical Geology, v. 5, p. 73-90.

Catmull, E., 1975, Computer display of curved surfaces: Proceedings of Conference on Computer Graphics, Pattern Recognition and Data Structure, p. 11-17.

_____, and R. Rom, 1974, A class of local interpolating splines, *in* Barnhill, R. E., and R. F. Riesenfeld (eds.), Computer Aided Geometric Design: Academic Press, New York, p. 317-326.

Chadwick, P. K., 1975, A psychological analysis of observation in geology: Nature, v. 256, p. 570-573.

_____, 1976, Visual illusions in geology: Nature, v. 260, p. 397-401.

Christopher, R. A., and J. A. Waters, 1974, Fourier series as a quantitative descriptor of miospore shape: Journal of Paleontology, v. 48, p. 697-709.

Clark, J. H., 1974, 3-D design of free-form B-spline surfaces: Tech. Rept. UTEC-CSc-74-120, Dept. of Computer Science, University of Utah, 75 p.

_____, 1976a, Designing surfaces in 3-D: Communications of the Association for Computing Machinery, v. 19, p. 454-460.

_____, 1976b, Hierarchical geometric models for visible surface algorithms: Communications of the Association for Computing Machinery, v. 19, p. 547-554.

Cliff, A. D., and J. K. Ord, 1973, Spatial Autocorrelation: Pion Ltd., London, 178 p.

Coons, S. A., 1967, Surfaces for computer-aided design of space forms: Tech. Rept. MAC-TR-41, Massachusetts Institute of Technology, 105 p.

_____, 1974, Surface patches and B-spline curves, *in* Barnhill, R. E., and R. F. Riesenfeld (eds.), Computer Aided Geometric Design: Academic Press, New York, p. 1-16.

Cottafava, G., and G. Le Moli, 1969, Automatic contour map: Communications of the Association for Computing Machinery, v. 12, p. 386-391.

Craidon, C. B., 1975, A computer program for fitting smooth surfaces to an aircraft configuration and other three-dimensional geometries: Tech. Memorandum TM X-3206, National Aeronautics and Space Administration, Washington, D. C., 110 p.

Crow, F. C., 1977, The aliasing problem in computer-generated shaded images: Communications of the Association for Computing Machinery, v. 20, p. 799-805.

Davis, P. J., 1963, Interpolation and Approximation: Blaisdell, New York, 393 p.

de Boor, C., 1962, Bicubic spline interpolation: Journal of Mathematics and Physics, v. 41, p. 212-218.

_____, 1972, On calculating with B-splines: Journal of Approximation Theory, v. 6, p. 50-62.

Delfiner, P., and J. P. Delhomme, 1975, Optimum interpolation by kriging, *in* Davis, J. C., and M. J. McCullagh (eds.), Display and Analysis of Spatial Data: John Wiley & Sons, London, p. 96-114.

Delmet, D. A., and R. L. Anstey, 1974, Fourier analysis of morphological plasticity within an Ordovician bryozoan colony: Journal of Paleontology, v. 48, p. 217-226.

Demirmen, F., 1973, Numerical description of folded surfaces depicted by contour maps: Journal of Geology, v. 81, p. 599-620.

Doveton, J. H., and A. J. Parsley, 1970, Experimental evaluation of trend surface distortions induced by inadequate data-point distributions: Transactions of the Institute of Mining and Metallurgy, Section B, v. 79, p. 197-208.

Ehrlich, R., and B. Weinberg, 1970, An exact method for characterization of grain shape: Journal of Sedimentary Petrology, v. 40, p. 205-212.

Elias, H., 1972, Identification of structure by the common-sense approach, *in* Weibel, E. R., G. Meek, B. Ralph, P. Echlin, and R. Ross (eds.), Stereology 3: Blackwell, Oxford, p. 59-68.

Ferguson, J., 1964, Multivariable curve interpolation: Journal of the Association for Computing Machinery, v. 11, p. 221-228.

Forrest, A. R., 1972a, On Coons and other methods for the representation of curved surfaces: Computer Graphics and Image Processing, v. 1, p. 341-359.

_____, 1972b, Interactive interpolation and approximation by Bézier polynomials: Computer Journal, v. 15, p. 71-79.

_____, 1974, Computational geometry--achievements and problems, *in* Barnhill, R. E., and R. F. Riesenfeld (eds.), Computer Aided Geometric Design: Academic Press, New York, p. 17-44.

Gevirtz, J. L., 1976, Fourier analysis of bivalve outlines: implications on evolution and autecology: Journal of the International Association for Mathematical Geology, v. 8, p. 151-163.

Goël, J-J., 1968, Construction of basic functions for numerical utilisation of Ritz's method: Numerische Mathematik, v. 12, p. 435-447.

Gordon, W. J., 1971, Blending-function methods of bivariate and multivariate interpolation and approximation: Society for Industrial and Applied Mathematics, Journal of Numerical Analysis, v. 8, p. 158-177.

_____, and R. F. Riesenfeld, 1974a, Bernstein-Bézier methods for the computer-aided design of free-form curves and surfaces: Journal of the Association for Computing Machinery, v. 21, p. 293-310.

_____, and _____, 1974b, B-spline curves and surfaces, *in* Barnhill, R. E., and R. F. Riesenfeld (eds.), Computer Aided Geometric Design: Academic Press, New York, p. 95-126.

Gould, S. J., 1967, Evolutionary patterns in pelycosaurian reptiles; a factor analytical study: Evolution, v. 21, p. 385-401.

_____, 1969, An evolutionary microcosm: Pleistocene and Recent history of the land snail, *P.* (*Poecilozonites*) in Bermuda: Bulletin of the Museum of Comparative Zoology, Harvard University, v. 138, p. 407-532.

Gouraud, H., 1971, Computer display of curved surfaces: Tech. Rept. UTEC-CSc-71-113, Dept. of Computer Science, University of Utah, 80 p.

Grant, F., 1957, A problem in the analysis of geophysical data: Geophysics, v. 22, p. 309-344.

Greysukh, V. L., 1966, The possibility of studying landforms by means of digital coding: Izvestiya Akademii Nauk SSSR, Series on Geography No. 4, p. 102-110 [translation, *in* Soviet Geography: American Geographical Society, 1967, v. 8, New York, p. 137-149].

Harbaugh, J. W., and F. W. Preston, 1965, Fourier series analysis in geology, *in* Short Course and Symposium on Computers and Computer Applications in Mining and Exploration: College of Mines, University of Arizona, v. 1, p. R1-R46.

_____, and D. F. Merriam, 1968, Computer Applications in Stratigraphic Analysis: John Wiley & Sons, New York, 282 p.

_____, and G. Bonham-Carter, 1970, Computer Simulation in Geology: John Wiley & Sons, New York, 575 p.

Hardy, R. L., 1971, Multiquadric equations of topography and other irregular surfaces: Journal of Geophysical Research, v. 76, p. 1905-1915.

Hessing, R. C., H. K. Lee, A. Pierce, and E. N. Powers, 1972, Automatic contouring using bicubic functions: Geophysics, v. 37, p. 669-674.

Hildebrand, F. B., 1956, Introduction to Numerical Analysis: McGraw-Hill, New York, 511 p.

Holroyd, M. T., and B. K. Bhattacharyya, 1970, Automatic contouring of geophysical data using bicubic spline interpolation, Geological Survey of Canada Paper 70-55, 40 p.

James, W. R., 1966, FORTRAN IV program using double Fourier series for surface fitting of irregularly spaced data: Kansas Geological Survey Computer Contribution 5, 19 p.

_____, 1968, Least-squares surface fitting with discontinuous functions: Tech. Rept. 8, Geography Branch, Office of Naval Research (ONR Task 389-150), 49 p.

_____, 1970, Regression models for faulted structural surfaces: Bulletin of the American Association of Petroleum Geologists, v. 54, p. 638-646.

Junkins, J. L., G. W. Miller, and J. R. Jancaitis, 1973, A weighting function approach to modeling of irregular surfaces: Journal of Geophysical Research, v. 78, p. 1794-1803.

Kaesler, R. L., and J. A. Waters, 1972, Fourier analysis of the ostracode margin: Bulletin of the Geological Society of America, v. 83, p. 1169-1178.

Kendall, M. G., and A. Stuart, 1967, The Advanced Theory of Statistics, Volume 2, 2nd Edition: Griffin, London, 690 p.

Krige, D. G., 1966, Two-dimensional weighted moving average trend surfaces for ore valuation, *in* Proceedings of Symposium on Mathematical Statistics and Computer Applications in Ore Valuation: Journal of the South African Institute of Mining and Metallurgy, p. 13-79 [discussion by Whitten, E.H.T., p. 51-55].

Krumbein, W. C., 1959, Trend surface analysis of contour-type maps with irregular control-point spacing: Journal of Geophysical Research, v. 64, p. 823-834.

Lawson, C. L., 1977, Software for C^1 surface interpolation, *in* Rice, J. R. (ed.), Mathematical Software III: Academic Press, New York, p. 161-194.

Matheron, G., 1965, Les Variables Régionalisées et Leur Estimation: Masson et Cie., Paris, 305 p.

_____, 1967, Kriging, or polynomial interpolation procedures?: The Canadian Mining and Metallurgical Bulletin, v. 60, p. 1041-1045.

_____, 1971, The theory of regionalized variables and its applications: Les Cahiers du Centre de Morphologie Mathématique de Fontainebleau, no. 5, 211 p.

McCullagh, M. J., and C.A.M. King, 1970, SPITSYM, a FORTRAN IV computer program for spit simulation: Kansas Geological Survey Computer Contribution 50, 20 p.

McQuillan, M. W., 1968, Geology of the Davis-Bronson pool, Allen and Bourbon counties, Kansas: Unpub. M.S. thesis, Kansas State University, Manhattan, 117 p.

Morse, P. M., and H. Feshbach, 1953, Methods of Theoretical Physics: McGraw-Hill, New York, 1978 p.

Newman, W. M., and R. F. Sproull, 1973, Principles of Interactive Computer Graphics: McGraw-Hill, New York, 607 p.

Olea, R. A., 1975, Optimum mapping techniques using Regionalized Variable Theory: Kansas Geological Survey Series on Spatial Analysis 2, 137 p.

Overhauser, A. W., 1968, Analytic definition of curves and surfaces by parabolic blending: Tech. Rept. SL 68-40, Scientific Research Staff, Ford Motor Company, Dearborn, Michigan, 8 p.

Payne, P. J., 1968, A contouring program for Coons' surface patches: C.A.D. Group Document 16, Cambridge University.

Peters, G. J., 1974, Interactive computer graphics application of the parametric bi-cubic surface to engineering design problems, *in* Barnhill, R. E., and R. F. Riesenfeld (eds.), Computer Aided Geometric Design: Academic Press, New York, p. 259-302.

Phong, B. T., 1975, Illumination for computer generated pictures: Communications of the Association for Computing Machinery, v. 18, p. 311-317.

Prelat, A., 1974, Statistical estimation of wildcat well outcome probabilities by visual analysis of structure contour maps of Stafford County, Kansas: Kansas Geological Survey, K.O.X. Project, 103 p.

Ramsay, J. G., 1967, Folding and Fracturing of Rocks: McGraw-Hill, New York, 568 p.

Raup, D. M., 1966, Geometric analysis of shell coiling: general problems: Journal of Paleontology, v. 40, p. 1178-1190.

_____, 1967, Geometric analysis of shell coiling: coiling in ammonoids: Journal of Paleontology, v. 41, p. 43-65.

Riesenfeld, R. F., 1973, Applications of B-spline approximation to geometric problems of computer aided design: Tech. Rept. UTEC-CSc-73-126, Dept. of Computer Science, University of Utah, 92 p.

Rogers, D. F., and J. A. Adams, 1976, Mathematical Elements for Computer Graphics: McGraw-Hill, New York, 239 p.

Rudwick, M.J.S., 1976, The emergence of a visual language for geological science, 1760-1840: History of Science, v. 14, p. 149-195.

_____, and R. Cowen, 1967, The functional morphology of some aberrant strophomenide brachiopods from the Permian of Sicily: Bolletino della Società Paleontologica Italiana, v. 6, p. 113-176.

Sabin, M. A., 1968a, Numerical master geometry: Tech. Rept. VTO/MS/146, British Aircraft Corporation, Weybridge, U. K., 11 p.

_____, 1968b, Parametric surface equations for non-rectangular regions: Tech. Rept. VTO/MS/147, British Aircraft Corporation, Weybridge, U. K., 17 p.

_____, 1968c, Two basic interrogations of parametric surfaces: Tech. Rept. VTO/MS/148, British Aircraft Corporation, Weybridge, U. K., 5 p.

_____, 1968d, Offset parametric surfaces: Tech. Rept. VTO/MS/149, British Aircraft Corporation, Weybridge, U. K., 6 p.

_____, 1968e, General interrogations of parametric surfaces: Tech. Rept. VTO/MS/150, British Aircraft Corporation, Weybridge, U. K., 5 p.

_____, 1969a, Conditions for continuity of surface normal between adjacent parametric surfaces: Tech. Rept. VTO/MS/151, British Aircraft Corporation, Weybridge, U. K., 22 p.

_____, 1969b, A 16-point bicubic formulation suitable for Multipatch surfaces: Tech. Rept. VTO/MS/155, British Aircraft Corporation, Weybridge, U. K., 8 p.

_____, 1971a, Trinomial basis functions for interpolation in triangular regions (Bézier triangles): Tech. Rept. VTO/MS/188, British Aircraft Corporation, Weybridge, U. K., 4 p.

_____, 1971b, Interrogation techniques for parametric surfaces, *in* Parslow, R. D., and R. Elliot Green (eds.), Advanced Computer Graphics: Plenum, London, p. 1095-1118.

Schoenberg, I. J., 1946, Contributions to the problem of approximation of equidistant data by analytic functions: Quarterly of Applied Mathematics, v. 4, p. 45-99, 112-141.

_____, 1967, On spline functions, *in* Shisha, O. (ed.), Inequalities: Academic Press, New York, p. 255-291.

Speight, J. G., 1968, Parametric description of land form, *in* Stewart, G. A. (ed.), Land Evaluation: Macmillan of Australia, Sydney, p. 239-250.

Sprunt, B. F., 1975, Hidden-line removal from three-dimensional maps and diagrams, *in* Davis, J. C., and M. J. McCullagh (eds.), Display and Analysis of Spatial Data: John Wiley & Sons, London, p. 198-209.

Sutherland, I. E., R. F. Sproull, and R. A. Schumacker, 1974, A characterization of ten hidden-surface algorithms: Computing Surveys, v. 6, p. 1-55.

Thompson, D'A. W., 1961, On Growth and Form: Cambridge University Press, Cambridge, 346 p. [abridged edition, Bonner, J. T. (ed.)].

Tipper, J. C., 1976, The study of geological objects in three dimensions by the computerized reconstruction of serial sections: Journal of Geology, v. 84, p. 476-484.

_____, 1977, Three-dimensional analysis of geological forms: Journal of Geology, v. 85, p. 591-611.

_____, 1978, Computerized modeling for shape analysis in geology, *in* Merriam, D. F. (ed.), Recent Advances in Geomathematics: Pergamon Press, Oxford, p. 157-170.

Van Dyke, R. J., 1975, Geology and depositional environments of the reservoir sandstone, Kincaid oil field, Anderson County, Kansas: Unpub. M.S. thesis, University of Kansas, Lawrence, 97 p.

Walsh, J. L., 1935, Interpolation and approximation by rational functions in the complex domain: American Mathematical Society Colloquium Publications, v. 20, 382 p.

Waters, J. A., 1977, Quantification of shape by use of Fourier analysis: the Mississippian blastoid genus *Pentremites*: Paleobiology, v. 3, p. 288-299.

Watson, G. S., 1972, Trend surface analysis and spatial correlation, *in* Fenner, P. (ed.), Quantitative Geology: The Geological Society of America, Spec. Paper 146, p. 39-46.

Westbroek, P., B. Hesper, and F. Neijndorff, 1976, Three-dimensional stereographic representation of serial sections: Journal of Geology, v. 84, p. 725-730.

Whittaker, E., and G. Robinson, 1944, The Calculus of Observations, 4th Edition: Blackie, London, 397 p.

Whitten, E.H.T., 1970, Orthogonal polynomial trend surfaces for irregularly spaced data: Journal of the International Association for Mathematical Geology, v. 2, p. 141-152.

_____, 1975, The practical use of trend-surface analyses in the geological sciences, *in* Davis, J. C., and M. J. McCullagh (eds.), Display and Analysis of Spatial Data: John Wiley & Sons, London, p. 282-297.

_____, and M.E.V. Koelling, 1973, Spline-surface interpolation, spatial filtering, and trend surfaces for geological mapped variables: Journal of the International Association for Mathematical Geology, v. 5, p. 111-126.

_____, and _____, 1975, Computation of bicubic-spline surfaces for irregularly spaced data: Tech. Rept. 3, U. S. Army Research Office, Durham, N. C. (Grant DA-ARO-D-31-124-72-G54), 57 p.

Wu, S-C., J. F. Abel, and D. P. Greenberg, 1977, An interactive computer graphics approach to surface representation: Communications of the Association for Computing Machinery, v. 20, p. 703-712.

Zusne, L., 1970, Visual Perception of Form: Academic Press, New York, 547 p.

APPENDIX 1

SPECIALIZED NOTATION USED IN MORE THAN ONE SECTION OF THE TEXT

PART I

Notation	Definition	Text Page(s)
$\vec{X}$	Vector	6 ff
A *or* A(h)	Blending function matrix for global-basis spline	43, 44, 45, 49
A' *or* A'(h)	Transpose of A(h)	43, 44, 45
B, $\vec{B}$	Boundary matrix for global-basis spline	43, 44, 45
$\vec{B}(u)$, $\vec{B}(u,v)$	Univariate and bivariate Bézier approximant	23, 24, 25, 27
B_x, B_y *or* $\vec{B}_x$, $\vec{B}_y$	Generalized interpolation or approximation operators, acting in the x and y (or u and v) directions respectively	22, 46, 48
C^n	Continuity class	21, 33, 36, 40, 41, 43, 48, 52, 55, 58
$\vec{D}(u)$	Univariate Catmull-Rom interpolant	53, 54, 55
$f_0(u)$, $f_0(v)$ $f_1(u)$, $f_1(v)$ $g_0(u)$, $g_0(v)$ $g_1(u)$, $g_1(v)$	Blending functions for Coons surface	46, 47, 48, 49, 53
m, n	General notation for dimensions of quadrilateral mesh	11, 23, 24, 25, 27, 28, 32, 36, 37, 40, 41, 42, 43
P_i, $P_{i,j}$ *or* $\vec{P}_i$, $\vec{P}_{i,j}$	General notation for point, referred to its position on a line or a quadrilateral mesh. Indexing is usually $0 < i < m$ and $0 < j < n$	11, 12, 23, 24, 25, 27, 28, 32, 33, 35, 36, 37, 40, 41, 42, 43, 44, 50, 51, 52, 53, 54, 55

Notation	Definition	Text Page(s)
$P(0,0)$, $P(0,1)$, $P(1,0)$, $P(1,1)$	Corner points of unit-square mesh cell, referred to rectangular Cartesian coordinates	22
$P(x,0)$, $P(x,1)$, $P(0,y)$, $P(1,y)$	Sides of unit-square mesh cell, referred to rectangular Cartesian coordinates	22
$\vec{P}(0,0)$, $\vec{P}(0,1)$, $\vec{P}(1,0)$, $\vec{P}(1,1)$	Corner points of unit-square mesh cell, referred to parametric coordinates (u,v)	27, 28, 46, 47, 48, 49, 53, 54
$\vec{P}(u,0)$, $\vec{P}(u,1)$, $\vec{P}(0,v)$, $\vec{P}(1,v)$	Sides of unit-square mesh cell referred to parametric coordinates (u,v)	46, 47, 48, 49, 51, 53
$\vec{P}_u(u,v)$, $\vec{P}_v(u,v)$	Partial derivatives with respect to u and v (the u- and v-partials), calculated at point $\vec{P}(u,v)$	28, 47, 49, 50
$\vec{P}_{uv}(u,v)$	Cross-partial derivative (twist vector), calculated at point $\vec{P}(u,v)$	28, 47, 49, 51
$\vec{Q}(u,v)$	Coons surface	46, 47, 48, 49
$\vec{R}(u)$, $\vec{R}(u,0)$, $\vec{R}(u,1)$ $\vec{R}(0,v)$, $\vec{R}(1,v)$	Univariate Overhauser interpolant	52, 53, 54
$\vec{R}(u,v)$	Overhauser-Coons surface interpolant	53
$S_{r,s}(u,v)$	Global-basis spline interpolant, referred to rectangular Cartesian coordinates (u,v), for cell with origin at $P_{r,s}$	43, 44
$\vec{S}_{r,s}(u,v)$	Global-basis spline interpolant, referred to parametric coordinates (u,v), for cell with origin at $\vec{P}_{r,s}$	44, 45

Notation	Definition	Text Page(s)
$\vec{T}_r(u)$, $\vec{T}_{r,s}(u,v)$	Univariate and bivariate B-spline approximant, defined in intervals r and r,s respectively	32, 33, 36, 37
u, v	Parameters, usually in range $0 \rightarrow 1$	10, 14, 15, 23, 24, 25, 27, 28, 32, 33, 34, 36, 37, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55
u^x, v^x	Partial derivatives of x with respect to u and v, in rectangular Cartesian coordinate system (u,v,x)	42, 43, 44
uv^x	Cross-partial derivative of x, in rectangular Cartesian coordinate system (u,v,x)	42, 44
α_i, α_{ij}	Coefficients of global-basis interpolating spline. For bicubic spline, $0 < i < 3$ and $0 < j < 3$	40, 43
$\beta(u)$	Overhauser basis function	52, 53
$\varsigma(u)$	Catmull-Rom basis function	55
θ	Viewing transformation angle	18, 28, 37, 57
$\xi(u)$, $\xi(v)$	B-spline basis functions	32, 33, 34, 36
ϕ	Viewing transformation angle	18, 28, 37, 57
$\psi(u)$, $\psi(v)$	Bernstein basis functions	23, 24, 27

PART II

Notation	Definition	Text Page(s)
$\vec{X}$	Vector	60 ff
$\langle\vec{X}\rangle$	Unit vector in direction $\vec{X}$	67, 68, 70
$\times$	Vector cross-product	68, 70, 78

Notation	Definition	Text Page(s)
$\cdot$	Vector dot-product	67, 68, 70, 77, 78
$\|X\|$, $\|\vec{X}\|$	Magnitude of X or $\vec{X}$	68, 70
$\vec{N}$, $\vec{N}_0$	Normal vector to plane or surface	67, 68, 70
$\vec{P}$, $\vec{P}_0$, $\vec{P}_1$, $\vec{P}_2$	General notation for point	67, 68, 70, 77, 78
$\vec{P}_u$, $\vec{P}_v$, $\vec{P}_{uv}$, $\vec{P}_{vu}$, $\vec{P}_{uu}$, $\vec{P}_{vv}$	Partial derivatives of $\vec{P}$	68, 70, 77, 78
$\vec{Q}$	Fixed point	67, 68, 77, 78
$\vec{T}_0$	Tangent vector to intersection	67, 68
u, v	Parameters, usually in range $0 \rightarrow 1$	66, 67, 68, 70, 77, 78
θ	Viewing transformation angle	62, 65, 72
ϕ	Viewing transformation angle	62, 65, 72

APPENDIX 2

DATA POINT COORDINATES

The 130 data points, each given a unique identifying number, are arranged on a 13-by-10 quadrilateral mesh as shown below. The coordinates of each point (in arbitrary units) are then tabulated for both the lower and upper surfaces.

179	178	4	8	9	183	16	17	96	97	99	105	106
177	1	7	12	182	18	19	93	185	94	102	103	104
186	3	6	11	21	40	41	42	184	107	108	109	110
5	10	22	39	38	43	44	50	59	86	87	111	113
23	28	31	36	46	48	49	58	84	89	88	112	138
25	27	30	35	52	55	57	61	83	82	91	137	176
29	32	33	53	54	62	63	81	80	135	136	175	173
65	67	75	76	78	119	125	130	139	141	171	172	162
66	68	116	118	122	124	128	129	140	169	168	166	160
69	74	115	121	123	126	127	143	144	149	150	151	152

Point no.	X	Y	Z-base	Z-top
179	10.940	29.077	2.500	2.500
178	9.300	27.297	2.500	2.500
4	9.885	26.125	2.500	2.500
8	10.148	25.648	2.563	2.563
9	10.742	25.239	2.438	2.438
183	12.620	25.317	2.563	2.563
16	12.397	23.496	2.625	2.625
17	13.240	22.600	2.500	2.500
96	14.100	21.583	2.375	2.375
97	14.981	20.827	2.500	2.500
99	15.994	19.717	2.750	2.750
105	16.872	18.807	2.563	2.563
106	17.479	17.987	2.813	2.813
177	7.420	27.497	2.631	2.881
1	7.800	26.477	2.287	2.662
7	8.293	25.473	2.369	2.744
12	9.082	24.658	2.325	2.575
182	10.220	23.677	2.262	2.887
18	10.764	22.911	2.194	3.069
19	11.698	21.983	2.563	2.563
93	12.440	21.185	2.137	2.637
185	12.640	20.577	2.425	3.113
94	13.289	20.597	2.381	2.631
102	14.341	19.240	2.525	3.213
103	15.180	18.268	2.206	2.956
104	16.017	17.445	2.563	3.125
186	5.060	26.697	2.525	2.838
3	5.646	26.050	1.850	2.600
6	6.564	25.126	2.231	2.669
11	7.245	24.243	2.363	2.988
21	8.295	23.176	2.281	2.781
40	9.213	22.229	1.963	2.463
41	10.091	21.312	2.275	2.775
42	11.038	20.374	2.594	2.719
184	11.830	19.617	2.575	2.825
107	13.663	17.661	2.244	2.806

Point no.	X	Y	Z-base	Z-top
108	14.529	16.737	2.350	2.662
109	15.225	16.026	2.313	2.500
110	16.109	15.119	2.306	2.431
5	4.048	25.300	2.181	2.619
10	4.884	24.455	2.162	2.725
22	5.668	23.597	1.988	2.863
39	6.607	22.625	1.662	2.475
38	7.621	21.606	1.906	2.469
43	8.494	20.662	2.475	2.850
44	9.426	19.740	2.706	3.081
50	10.319	18.786	2.575	2.887
59	11.052	17.706	2.444	2.756
86	12.105	16.991	2.563	2.563
87	13.045	16.051	2.300	2.738
111	14.052	15.028	2.331	3.081
113	15.120	13.952	2.363	2.738
23	3.409	23.923	2.031	2.969
28	4.233	22.878	1.906	2.469
31	5.135	21.942	1.781	2.844
36	6.028	21.054	2.156	2.969
46	6.920	20.116	2.537	2.975
48	7.807	19.228	2.500	3.063
49	8.676	18.301	2.450	3.075
58	9.556	17.372	2.319	3.006
84	10.455	16.450	2.625	2.625
89	11.350	15.574	2.813	2.813
88	12.246	14.631	2.625	3.125
112	13.765	14.054	2.344	2.781
138	13.112	10.033	2.250	2.875
25	1.744	23.176	2.269	2.769
27	2.644	22.261	2.088	2.650
30	3.536	21.375	2.106	2.731
35	4.435	20.444	2.106	2.794
52	5.324	19.513	1.794	2.606
55	6.211	18.571	1.869	2.619
57	7.128	17.652	2.456	3.019
61	8.003	16.758	2.262	2.950
83	8.909	15.829	2.625	2.625
82	9.874	14.837	2.750	2.750
91	10.826	13.870	2.363	2.925
137	11.630	11.786	2.625	3.063
176	11.452	8.529	2.063	2.813
29	1.034	21.648	2.338	3.025
32	1.919	20.705	2.088	2.838
33	2.817	19.815	2.137	2.825
53	3.665	18.851	2.250	2.750
54	4.582	17.914	2.412	2.662
62	5.397	16.967	2.150	2.838
63	6.377	16.120	2.044	2.856
81	7.299	15.176	2.044	2.794
80	7.720	14.591	2.269	2.894
135	9.231	13.232	2.344	2.844
136	10.151	12.261	1.881	3.006
175	10.961	9.090	2.213	2.775
173	11.300	6.583	2.125	2.813
65	0.952	18.304	2.294	2.669
67	1.417	16.605	2.600	3.037
75	2.328	15.708	2.500	3.125
76	3.215	14.791	1.875	2.563
78	4.110	13.899	2.075	2.825

Point no.	X	Y	Z-base	Z-top
119	4.920	13.110	2.213	2.400
125	5.583	12.417	2.313	2.313
130	6.756	11.250	2.438	2.438
139	7.625	10.300	2.213	2.900
141	8.239	8.797	2.125	3.188
171	9.439	8.472	2.500	2.500
172	10.314	7.589	2.281	2.656
162	11.392	5.297	2.188	2.500
66	0.521	17.516	2.012	2.450
68	1.043	16.124	2.750	2.750
116	1.151	13.377	2.000	2.625
118	2.475	13.287	2.219	2.781
122	3.302	12.464	2.438	2.438
124	4.285	11.478	2.313	2.313
128	5.307	10.354	2.188	2.188
129	6.134	9.520	2.006	2.381
140	7.034	8.630	2.000	2.688
169	7.873	7.731	2.094	2.656
168	8.673	6.974	2.281	3.031
166	9.475	6.140	1.894	2.581
160	11.216	4.370	2.231	2.419
69	-0.235	16.447	2.450	3.075
74	-0.403	14.007	1.981	2.731
115	0.363	13.204	2.150	2.713
121	1.855	11.697	2.438	2.438
123	2.754	10.726	2.438	2.438
126	3.563	9.352	2.375	2.375
127	4.562	8.703	2.313	2.313
143	5.417	7.999	2.000	2.625
144	6.318	7.111	2.313	2.313
149	6.380	4.799	2.162	2.475
150	7.071	4.077	1.956	2.706
151	7.985	3.323	2.369	2.744
152	8.510	2.587	2.162	2.537

APPENDIX 3

BÉZIER CONTROL POLYHEDRA

The two sets of 130 control points, for the lower and upper surfaces respectively, are arranged on identical 13-by-10 meshes:

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62	63	64	65
66	67	68	69	70	71	72	73	74	75	76	77	78
79	80	81	82	83	84	85	86	87	88	89	90	91
92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117
118	119	120	121	122	123	124	125	126	127	128	129	130

Lower surface

Point no.	X	Y	Z
1	10.94	29.08	2.50
2	170.29	173.21	11.65
3	-702.91	-609.46	-34.49
4	1855.58	1659.65	87.04
5	-3379.68	-2988.36	-129.40
6	4701.60	4248.06	154.82
7	-5005.99	-4579.75	-136.88
8	4230.22	3978.69	109.69
9	-2772.34	-2659.61	-64.67
10	1435.17	1423.78	32.01
11	-510.86	-512.87	-4.51
12	134.97	139.73	3.07
13	17.48	17.99	2.81
14	-4.59	25.79	5.00
15	-1043.00	717.24	-284.17
16	4724.09	-3243.59	1487.76
17	-12740.93	9399.39	-4296.91
18	24613.13	-19625.71	9656.67
19	-36395.34	32046.50	-17335.17
20	42688.51	-41532.00	22806.92
21	-40267.65	42844.37	-21609.69
22	30416.90	-34313.90	14983.81
23	-17766.85	20482.12	-7588.00
24	7414.35	-8311.53	2690.44
25	-1871.05	1932.44	-549.42
26	9.88	71.94	2.93
27	49.44	29.21	-6.62
28	2766.42	-2479.07	284.86
29	-12465.71	11853.09	-2047.55
30	34483.58	-33798.09	6801.92
31	-68216.28	70484.54	-18372.38
32	104196.06	-113832.88	38682.23
33	-126445.22	146131.31	-54849.38
34	123637.55	-148758.17	53020.08
35	-96150.29	117822.98	-36352.89

Point no.	X	Y	Z
36	57541.76	-69333.49	17839.89
37	-24326.79	27978.55	-5980.80
38	6310.54	-6335.53	1117.65
39	32.39	-178.71	2.28
40	-100.20	23.12	25.21
41	-4717.03	4457.32	-806.14
42	21247.53	-21028.74	5060.42
43	-59332.47	60605.92	-16047.00
44	118889.93	-126566.54	41692.28
45	-184773.21	204805.22	-84868.69
46	228773.10	-262621.19	117916.47
47	-228172.69	266877.52	-112962.45
48	180430.52	-210372.12	77636.22
49	-109241.95	123094.98	-38526.01
50	46694.82	-49307.26	13146.95
51	-12170.39	11101.48	-2477.65
52	-15.21	398.36	-2.54
53	156.05	35.23	-35.69
54	5959.38	-5200.99	2241.17
55	-26655.19	25207.42	-11815.41
56	75070.80	-73154.44	34925.59
57	-152203.89	153903.51	-82373.30
58	240805.67	-249869.46	153752.84
59	-304070.88	321220.78	-204287.42
60	309216.58	-326508.41	193311.99
61	-248126.68	257043.49	-134214.32
62	151700.40	-149793.56	68476.47
63	-65195.39	59936.86	-24379.05
64	17054.56	-13411.57	4873.06
65	69.25	-490.46	15.68
66	-144.00	-13.96	41.66
67	-5594.70	4373.74	-3419.14
68	24984.79	-21638.23	16906.34
69	-70797.92	63814.15	-48222.32
70	145471.17	-135555.37	107727.21
71	-234443.79	221546.13	-190495.05
72	302343.31	-286179.56	245881.70
73	-313565.33	291943.49	-231043.59
74	255446.47	-230223.96	161884.58
75	-157470.84	134222.17	-84357.79
76	67949.72	-53950.35	31018.23
77	-17654.30	12224.50	-6449.10
78	-61.48	463.41	-15.83
79	91.98	80.64	-25.85
80	3741.22	-2214.79	2760.10
81	-16628.86	12112.35	-13292.71
82	47387.40	-36843.29	37193.68
83	-98709.02	80496.47	-81399.79
84	162305.58	-133747.18	141616.62
85	-214179.87	175443.40	-181335.65
86	227120.90	-181045.26	170427.04
87	-188257.83	144160.22	-120430.05
88	117305.39	-84464.64	63799.11
89	-50830.04	34398.89	-24001.18
90	13168.07	-7903.59	5109.55
91	67.25	-288.70	15.96
92	-37.12	-27.26	16.91
93	-1599.11	241.46	-1372.39
94	7027.40	-2406.22	6446.55

Point no.	X	Y	Z
95	-20011.92	9178.05	-17521.07
96	42196.75	-22812.15	37630.10
97	-70843.17	41097.53	-64954.99
98	96031.83	-57310.69	82963.35
99	-104635.40	61948.03	-78029.20
100	88769.25	-50917.72	55501.57
101	-56183.49	30334.69	-29765.48
102	24583.91	-12560.65	11388.98
103	-6352.24	2940.13	-2446.97
104	-17.20	134.65	-4.58
105	10.12	32.45	-2.72
106	434.32	325.24	381.92
107	-1841.77	-808.35	-1703.59
108	5122.50	1111.60	4503.77
109	-10677.40	-218.52	-9544.83
110	17975.03	-2053.89	16492.16
111	-24630.97	5298.04	-21121.26
112	27275.25	-7481.55	19940.34
113	-23463.83	7120.60	-14259.43
114	15024.21	-4507.53	7702.55
115	-6593.17	1968.41	-2958.68
116	1717.01	-456.23	636.34
117	21.45	-27.42	4.14
118	-0.24	16.45	2.45
119	-20.01	-71.86	-29.74
120	90.82	372.47	142.62
121	-266.87	-892.05	-381.89
122	573.94	1661.82	770.28
123	-908.28	-2329.72	-1175.45
124	1136.15	2656.08	1418.34
125	-1102.70	-2420.12	-1331.66
126	856.97	1798.21	971.06
127	-486.08	-1014.84	-524.20
128	206.39	416.59	202.83
129	-38.65	-94.26	-42.53
130	8.51	2.59	2.16

Upper surface

Point no.	X	Y	Z
1	10.94	29.08	2.50
2	171.03	196.67	16.14
3	-706.32	-646.85	-38.51
4	1865.19	1706.82	84.65
5	-3399.93	-3033.95	-111.70
6	4735.74	4277.42	113.84
7	-5053.49	-4582.40	-70.93
8	4285.70	3954.35	26.27
9	-2827.12	-2618.24	21.76
10	1480.96	1379.74	-42.45
11	-543.09	-477.50	48.94
12	153.84	117.38	-28.61
13	17.48	17.99	2.81
14	-5.08	51.12	-5.51
15	-1126.49	718.31	-656.05
16	5101.79	-3374.87	2876.42
17	-13732.48	9863.56	-7022.65
18	26425.36	-20665.71	12579.61
19	-38843.78	33824.07	-18520.76
20	45202.52	-43929.20	22406.33
21	-42263.05	45407.67	-21728.65
22	31671.81	-36454.94	16426.74
23	-18408.42	21830.84	-9207.07
24	7675.95	-8899.42	3463.28
25	-1957.69	2097.51	-667.81
26	28.99	50.86	-24.26
27	56.37	-16.73	57.81
28	3423.13	-3104.14	3500.46
29	-15458.32	14893.64	-15445.13
30	42645.78	-42424.79	38541.96
31	-83960.99	88223.11	-69653.73
32	127277.20	-142263.40	100578.48
33	-153014.39	182391.91	-116462.03
34	148111.65	-185573.54	107049.30
35	-114213.51	147008.96	-76671.30
36	67984.13	-86649.20	41060.99
37	-28688.22	35013.96	-14910.22
38	7476.38	-7985.04	2901.35
39	1.20	-181.54	44.91
40	-140.67	93.13	-148.03
41	-6912.97	6908.68	-10794.83
42	31222.67	-32500.71	47804.99
43	-87018.33	92997.45	-120742.10
44	173454.99	-193124.92	219611.31
45	-267288.07	311313.07	-313221.96
46	327392.30	-398068.71	353324.17
47	-322856.20	403761.18	-314272.41
48	252910.92	-318105.14	217575.78
49	-152245.42	186332.79	-112999.88
50	64906.11	-74697.59	40182.32
51	-16993.07	16878.36	-7824.32
52	23.11	519.24	-63.05
53	265.87	-55.35	272.83
54	10630.60	-10461.23	21584.06
55	-47771.34	49674.60	-95432.76
56	134288.45	-142431.18	242031.33
57	-270572.46	296574.96	-439972.70
58	423594.20	-478430.63	620343.65
59	-527967.26	611853.80	-686151.07

Point no.	X	Y	Z
60	529741.49	-619783.85	596004.98
61	-420522.64	487032.37	-402282.64
62	255434.32	-284009.00	203886.33
63	-109502.66	113521.50	-71036.62
64	28771.29	-25454.12	13717.64
65	40.71	-839.72	97.49
66	-328.66	65.93	-347.82
67	-12688.77	12071.63	-30573.40
68	56923.14	-57627.10	134693.19
69	-161096.36	166059.66	-341084.61
70	328344.41	-346738.98	616946.81
71	-522264.24	560307.02	-859047.00
72	662662.64	-717205.63	933146.90
73	-676187.65	726692.80	-793308.60
74	543730.23	-570573.73	522968.32
75	-332685.35	332218.04	-258637.26
76	143148.54	-132957.68	88004.40
77	-37447.37	29962.19	-16649.06
78	-73.79	1031.65	-111.32
79	290.47	53.34	342.34
80	11542.34	-10223.88	31944.81
81	-51678.25	50089.24	-139963.76
82	147141.93	-145229.45	352810.45
83	-303348.04	305346.63	-633296.36
84	490225.66	-495198.02	870491.77
85	-633219.91	636248.86	-929826.39
86	657245.19	-646401.72	775248.47
87	-535478.20	508716.05	-499894.55
88	330177.77	-296473.80	241223.33
89	-142502.81	119371.25	-79902.40
90	37139.96	-27137.88	14715.36
91	117.74	-919.66	112.77
92	-175.60	-65.55	-233.97
93	-7514.18	5719.62	-24145.96
94	33509.63	-29173.82	105228.77
95	-95774.78	86404.84	-263679.61
96	199483.80	-184667.92	469450.65
97	-327208.04	302933.70	-637637.45
98	430181.53	-393218.19	671400.70
99	-454309.24	402920.29	-550702.43
100	375454.96	-319201.65	348400.99
101	-233628.64	186713.50	-164333.38
102	101331.37	-75691.22	53007.93
103	-26334.38	17378.94	-9477.11
104	-72.00	613.93	-76.27
105	59.16	79.06	103.73
106	3096.52	-1656.06	11176.65
107	-13635.64	9434.32	-48447.58
108	38866.56	-29316.23	120749.92
109	-81322.12	65059.54	-213427.27
110	134792.31	-109363.73	287094.39
111	-179676.14	144955.10	-298857.34
112	192546.99	-150947.73	241935.52
113	-161080.13	121064.89	-150642.70
114	101093.83	-71265.95	69653.00
115	-44028.02	29128.34	-21901.70
116	11446.60	-6726.14	3804.00
117	52.92	-248.68	36.92

Point no.	X	Y	Z
118	-0.24	16.45	3.07
119	-54.62	-42.75	34.37
120	148.55	322.86	-141.29
121	-345.83	-821.82	404.08
122	662.85	1578.26	-812.03
123	-989.71	-2245.79	1267.70
124	1194.38	2585.13	-1532.78
125	-1131.11	-2370.24	1457.23
126	860.04	1769.78	-1061.25
127	-475.01	-1002.42	586.77
128	192.32	413.07	-225.13
129	-28.06	-94.14	55.49
130	8.51	2.59	2.54

APPENDIX 4

B-SPLINE CONTROL POINTS

The two sets of 180 control points, for the lower and upper surfaces respectively, are arranged on identical 15-by-12 meshes. Note that, in contrast to the Bézier control polyhedra, the X and Y coordinates are the same for equivalent B-spline control points.

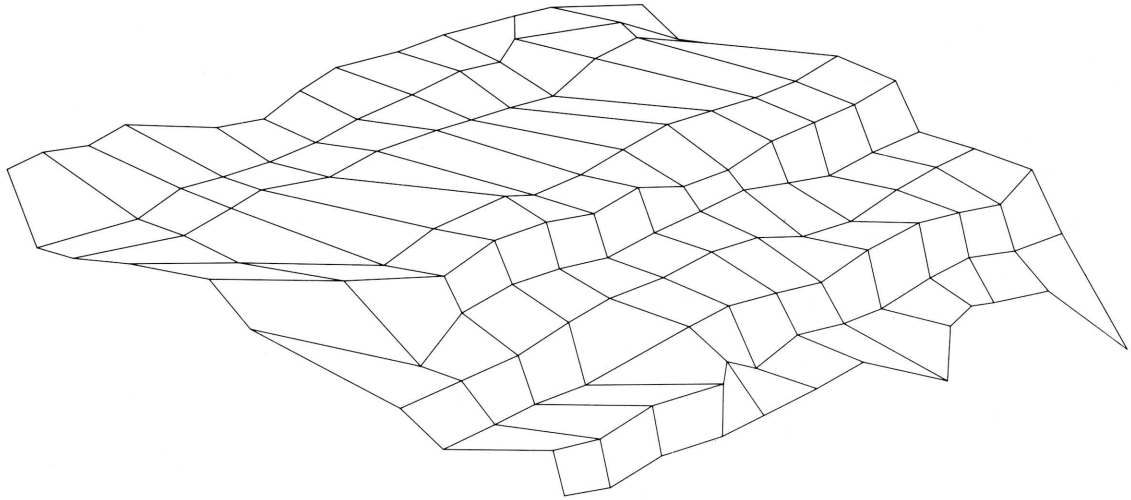
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105
106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165
166	167	168	169	170	171	172	173	174	175	176	177	178	179	180

Point no.	X	Y	Z-base	Z-top
1	20.260	33.744	1.972	1.642
2	14.688	30.906	2.324	2.033
3	9.115	28.067	2.676	2.424
4	12.049	26.634	2.485	1.802
5	11.054	27.007	3.189	3.255
6	9.307	26.581	2.484	1.668
7	16.538	29.620	3.214	1.330
8	11.567	24.418	2.128	3.146
9	13.927	24.491	3.632	2.573
10	15.729	23.006	2.009	0.615
11	17.600	19.561	2.602	3.039
12	18.187	19.659	3.072	1.763
13	19.317	18.819	3.045	1.791
14	19.416	17.902	3.036	2.158
15	19.514	16.984	3.027	2.524
16	13.196	30.966	2.491	2.491
17	10.940	29.077	2.500	2.500
18	8.684	27.188	2.509	2.509
19	10.123	25.952	2.462	2.462
20	10.134	25.753	2.642	2.642
21	10.231	24.925	2.349	2.349
22	13.396	25.981	2.591	2.591
23	11.905	23.055	2.664	2.664
24	13.366	22.777	2.502	2.502
25	14.071	21.437	2.327	2.327
26	14.949	20.974	2.440	2.440
27	16.021	19.631	2.912	2.912
28	16.933	18.806	2.413	2.413
29	17.479	17.987	2.813	2.813
30	18.025	17.168	3.213	3.213
31	6.132	28.187	3.009	3.340
32	7.192	27.248	2.676	2.967
33	8.253	26.309	2.343	2.595
34	8.197	25.270	2.440	3.122

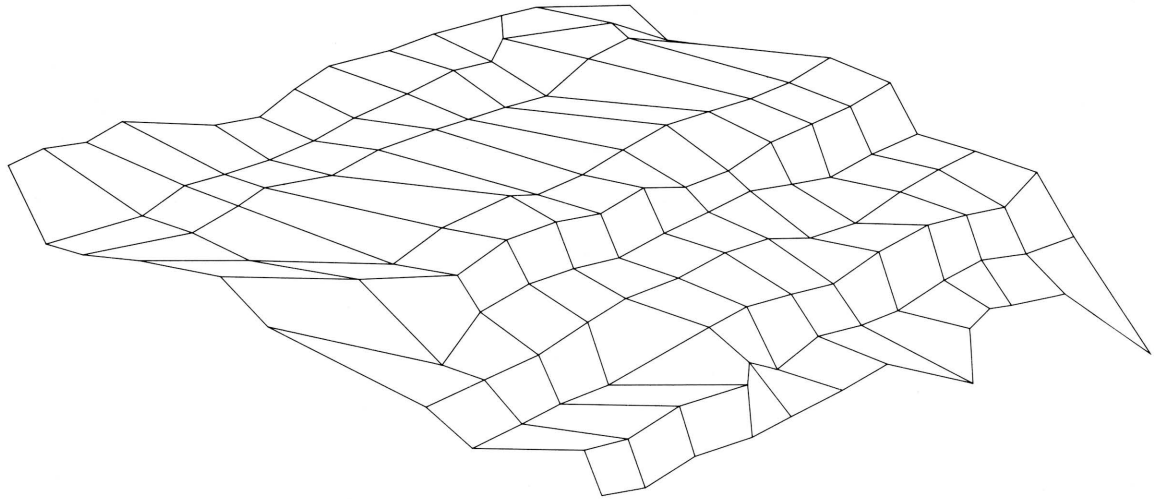
Point no.	X	Y	Z-base	Z-top
35	9.213	24.498	2.094	2.029
36	11.154	23.269	2.213	3.030
37	10.254	22.341	1.969	3.853
38	12.243	21.691	3.201	2.182
39	12.805	21.063	1.373	2.432
40	12.414	19.867	2.645	4.039
41	12.297	22.386	2.279	1.842
42	13.854	19.602	2.752	4.060
43	14.549	18.793	1.782	3.036
44	15.542	18.072	2.590	3.468
45	16.536	17.352	3.398	3.901
46	4.633	27.304	4.066	3.396
47	4.810	26.912	2.582	2.917
48	4.987	26.521	1.099	2.437
49	6.632	25.375	2.409	2.286
50	6.700	24.702	2.766	3.747
51	7.815	23.494	2.728	2.976
52	9.174	22.600	1.460	1.649
53	9.615	21.707	1.878	3.047
54	10.985	20.315	2.971	2.634
55	11.339	20.850	2.830	2.559
56	15.065	15.882	1.838	3.228
57	15.092	16.095	2.481	2.370
58	15.834	15.735	2.370	1.979
59	16.454	14.393	2.205	2.063
60	17.073	13.052	2.040	2.148
61	2.861	26.175	1.719	1.865
62	3.927	25.284	2.145	2.394
63	4.993	24.394	2.570	2.923
64	5.395	23.698	2.146	3.243
65	6.619	22.568	0.967	1.757
66	7.753	21.541	1.217	2.024
67	8.373	20.583	2.809	3.065
68	9.505	19.645	3.142	3.293
69	10.375	18.923	2.608	2.775
70	10.905	17.044	2.113	2.981
71	11.500	17.807	2.896	2.182
72	12.746	16.621	1.914	2.469
73	13.233	14.674	2.448	3.900
74	15.297	15.068	2.425	2.865
75	17.362	15.462	2.403	1.829
76	3.059	24.887	2.126	4.530
77	3.772	23.751	1.925	3.221
78	4.484	22.614	1.723	1.912
79	5.561	21.577	1.344	2.966
80	6.342	20.840	2.482	3.349
81	7.182	19.848	3.439	3.303
82	8.139	19.055	2.660	3.356
83	8.918	18.096	2.195	2.949
84	9.840	17.107	2.100	3.245
85	10.674	16.429	2.769	2.363
86	11.950	15.377	2.859	2.892
87	12.010	13.695	2.951	3.780
88	15.499	15.824	1.999	2.036
89	13.077	9.045	2.271	2.906
90	10.654	2.267	2.543	3.776
91	0.541	24.231	2.511	2.327
92	1.441	23.251	2.343	2.537
93	2.341	22.271	2.175	2.746

Point no.	X	Y	Z-base	Z-top
94	3.218	21.580	2.357	2.697
95	4.170	20.522	2.063	2.700
96	5.041	19.631	0.931	2.437
97	5.948	18.666	1.302	2.074
98	6.852	17.737	3.169	3.250
99	7.612	16.944	2.089	3.021
100	8.997	15.678	2.825	2.347
101	9.333	14.833	3.021	2.859
102	10.683	14.233	2.130	2.458
103	11.120	12.058	3.313	3.662
104	11.068	8.948	1.991	2.762
105	11.016	5.839	0.668	1.863
106	-0.176	22.803	2.817	3.787
107	0.928	22.302	2.319	3.246
108	2.032	21.801	1.821	2.704
109	2.763	20.444	1.814	2.625
110	3.601	19.706	2.497	3.051
111	4.617	18.739	3.148	2.357
112	5.264	17.673	2.243	3.344
113	6.534	16.837	1.663	2.949
114	7.687	15.674	1.722	2.763
115	6.691	16.038	2.120	3.186
116	10.052	13.864	2.629	2.334
117	10.034	14.029	1.053	3.797
118	11.318	8.059	2.147	2.313
119	11.364	6.335	2.145	2.923
120	11.409	4.610	2.143	3.534
121	1.112	20.227	2.224	2.091
122	1.052	17.429	2.411	2.630
123	0.991	14.631	2.597	3.168
124	2.771	15.705	3.308	3.894
125	3.207	13.649	1.145	1.706
126	4.256	13.016	1.775	3.603
127	5.112	12.137	2.178	1.907
128	5.080	12.149	2.483	2.227
129	6.830	10.251	2.938	2.291
130	8.178	9.712	2.308	2.767
131	7.435	6.929	1.778	4.200
132	9.650	7.582	3.171	1.304
133	10.191	8.157	2.451	3.104
134	11.278	5.210	2.181	2.422
135	12.365	2.264	1.910	1.740
136	-0.622	17.607	-0.114	1.602
137	0.577	17.805	1.803	2.249
138	1.777	18.003	3.719	2.895
139	0.359	11.373	1.071	2.024
140	2.780	14.317	2.529	3.450
141	3.048	12.568	2.713	1.966
142	4.277	12.035	2.299	2.464
143	5.618	10.136	2.103	2.067
144	6.105	9.806	1.750	2.213
145	6.897	8.606	1.816	3.032
146	8.391	8.767	2.028	1.990
147	8.934	7.454	2.528	3.981
148	9.266	6.715	1.229	2.177
149	11.877	4.606	2.261	2.389
150	14.488	2.496	3.293	2.600
151	0.121	19.390	3.076	3.530
152	-0.235	16.447	2.450	3.075
153	-0.591	13.504	1.824	2.620

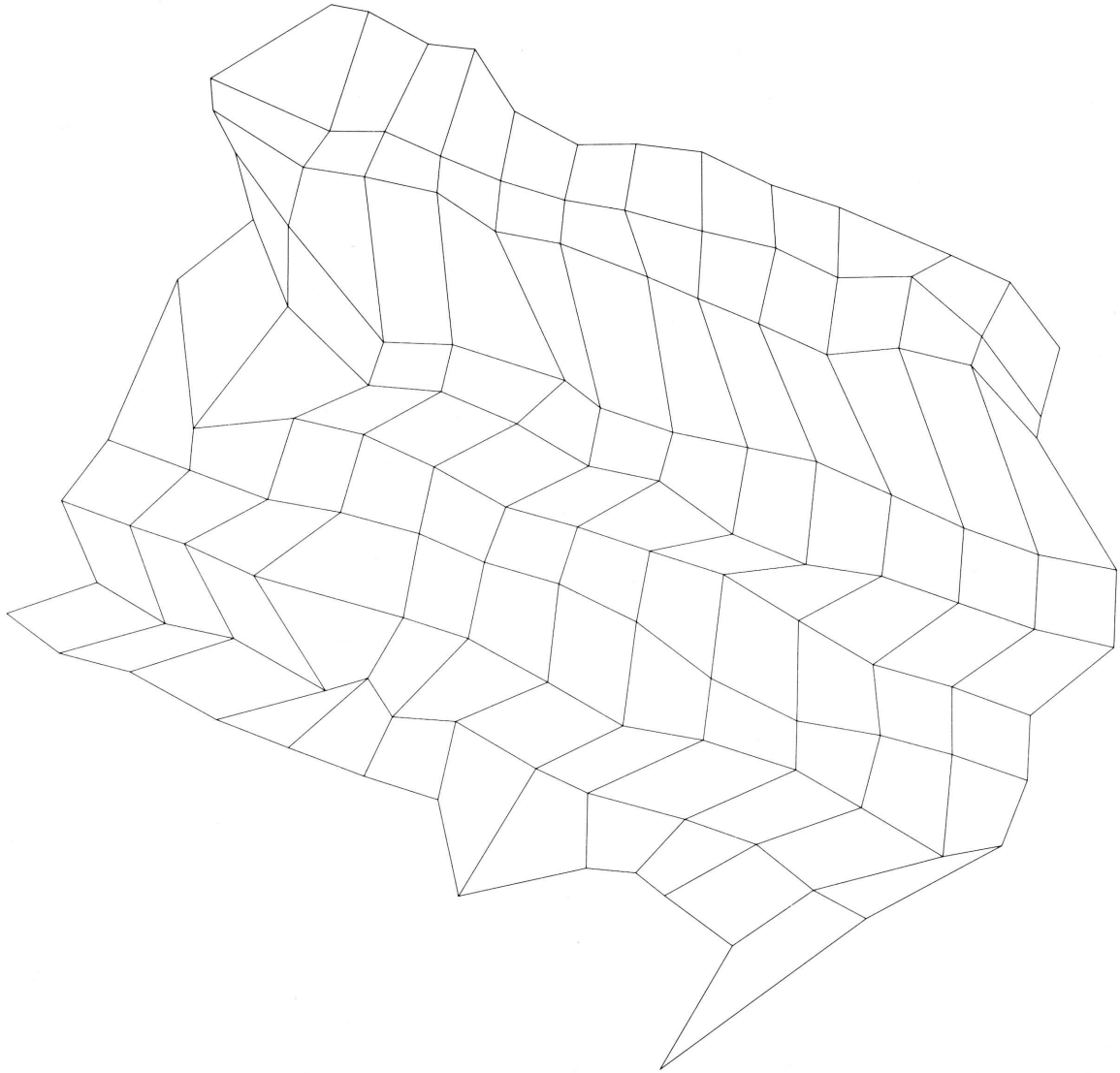
Point no.	X	Y	Z-base	Z-top
154	0.179	13.577	2.141	2.832
155	2.052	11.410	2.511	2.331
156	2.743	10.966	2.444	2.473
157	3.499	9.084	2.341	2.407
158	4.640	8.812	2.442	2.151
159	5.313	7.886	1.769	2.868
160	6.610	7.638	2.480	2.128
161	6.156	4.229	2.187	2.497
162	7.045	4.239	1.743	2.735
163	8.089	3.278	2.577	2.798
164	8.510	2.587	2.162	2.537
165	8.931	1.896	1.747	2.276
166	0.863	21.173	6.266	5.458
167	-1.047	15.089	3.097	3.901
168	-2.958	9.006	-0.072	2.344
169	-0.001	15.782	3.212	3.639
170	1.324	8.503	2.493	1.212
171	2.438	9.363	2.175	2.979
172	2.720	6.132	2.383	2.349
173	3.662	7.488	2.781	2.235
174	4.521	5.966	1.789	3.522
175	6.322	6.669	3.144	1.224
176	3.922	-0.308	2.346	3.003
177	5.156	1.024	0.958	1.490
178	6.911	-0.159	3.926	3.419
179	5.143	0.568	2.063	2.685
180	3.374	1.296	0.200	1.952



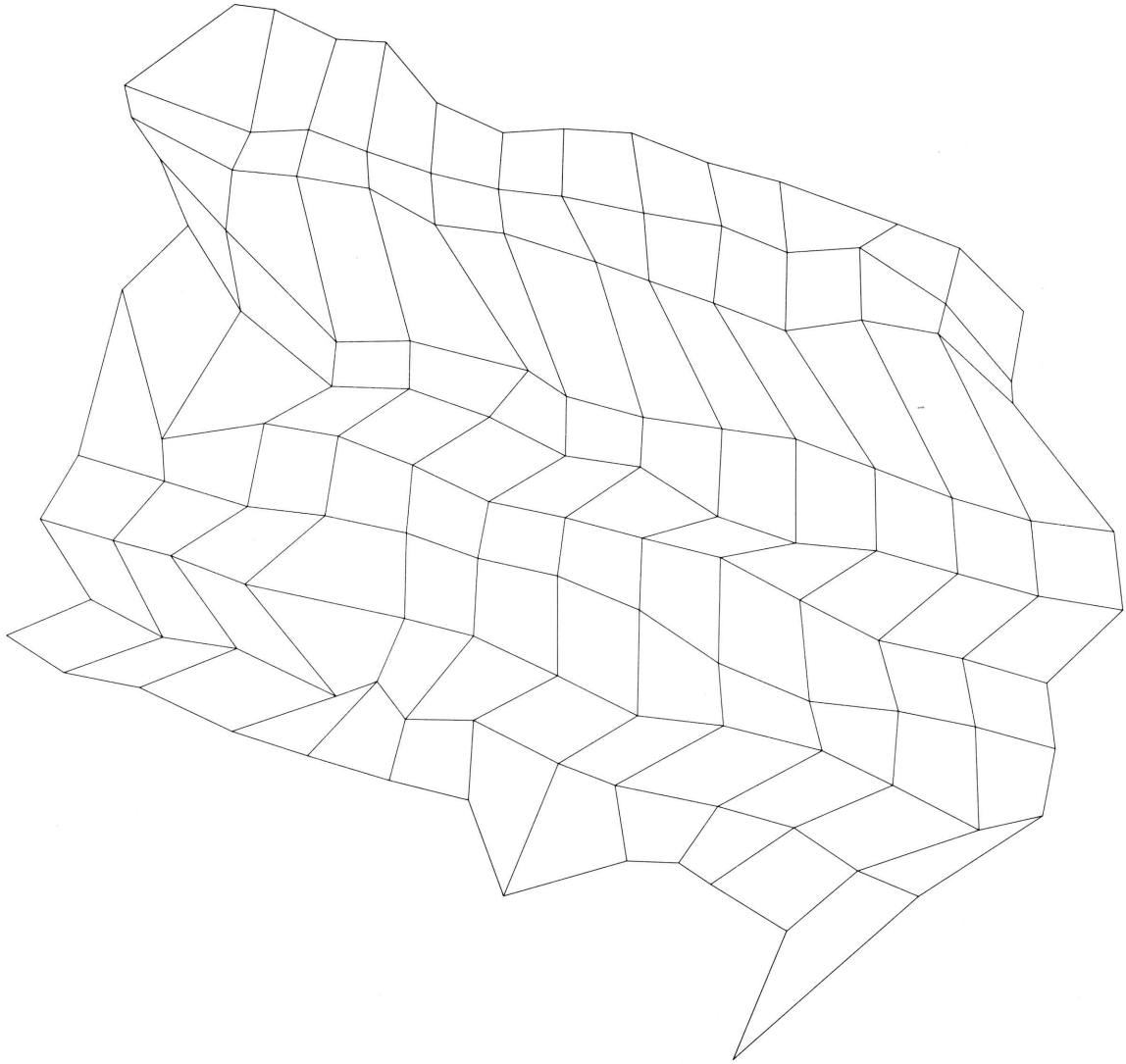
Stereo-pair 1-left



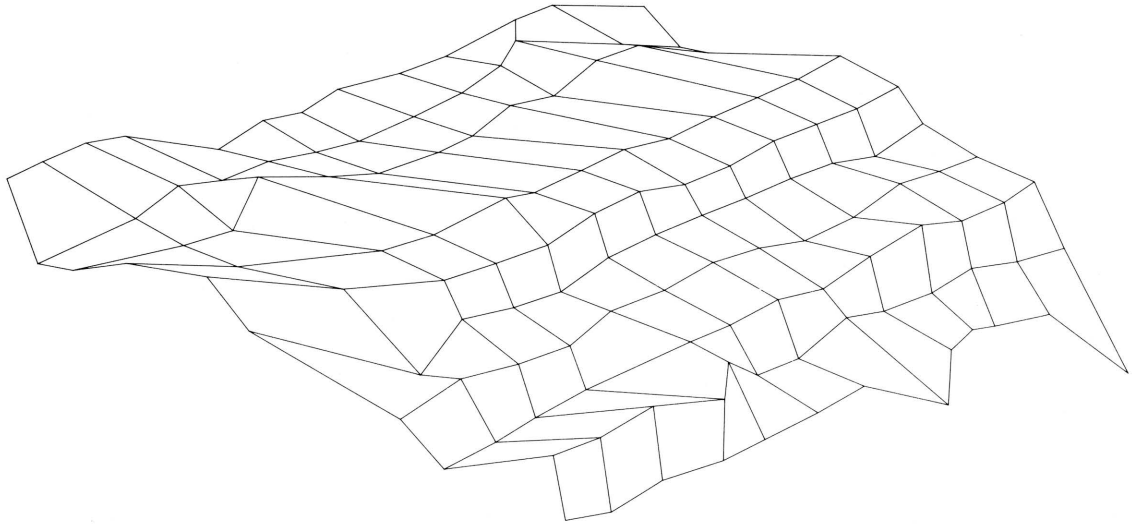
Stereo-pair 1-right



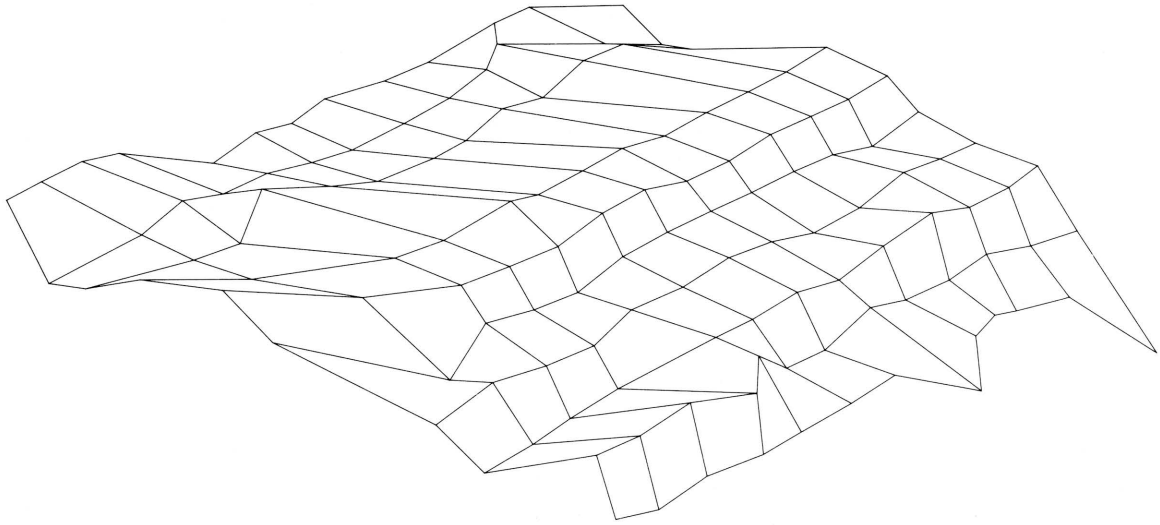
Stereo-pair 2-left



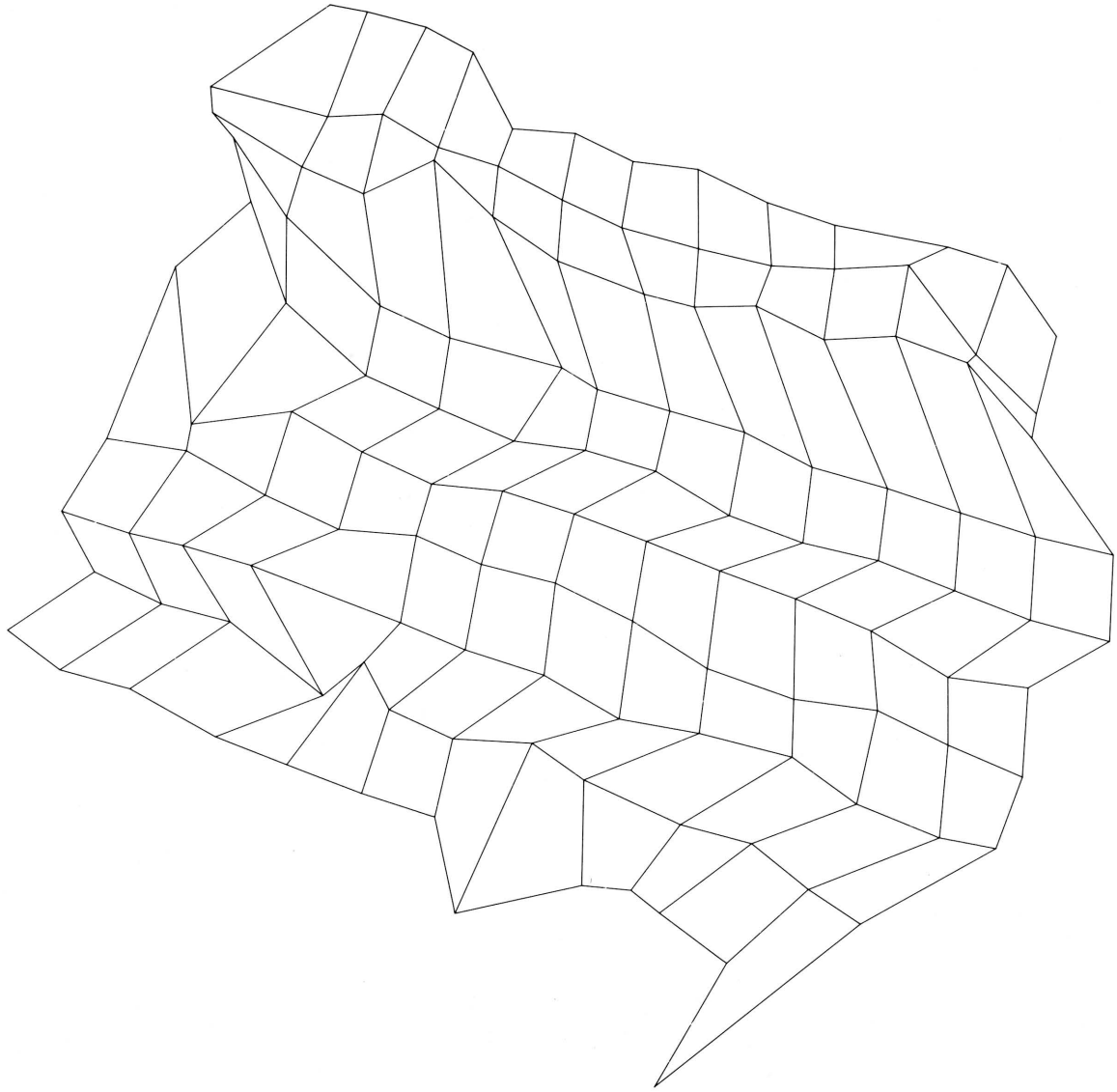
Stereo-pair 2-right



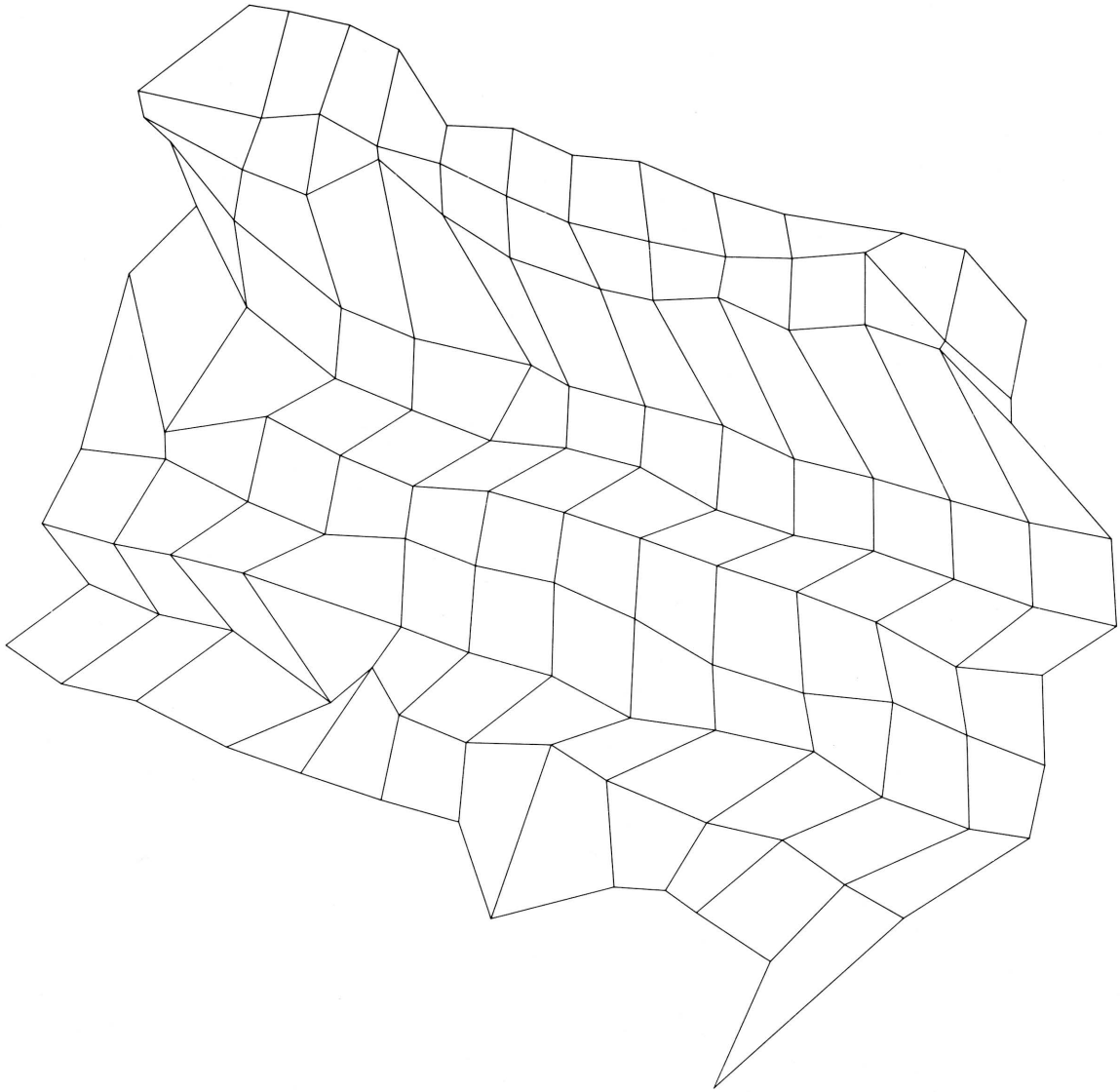
Stereo-pair 3-left



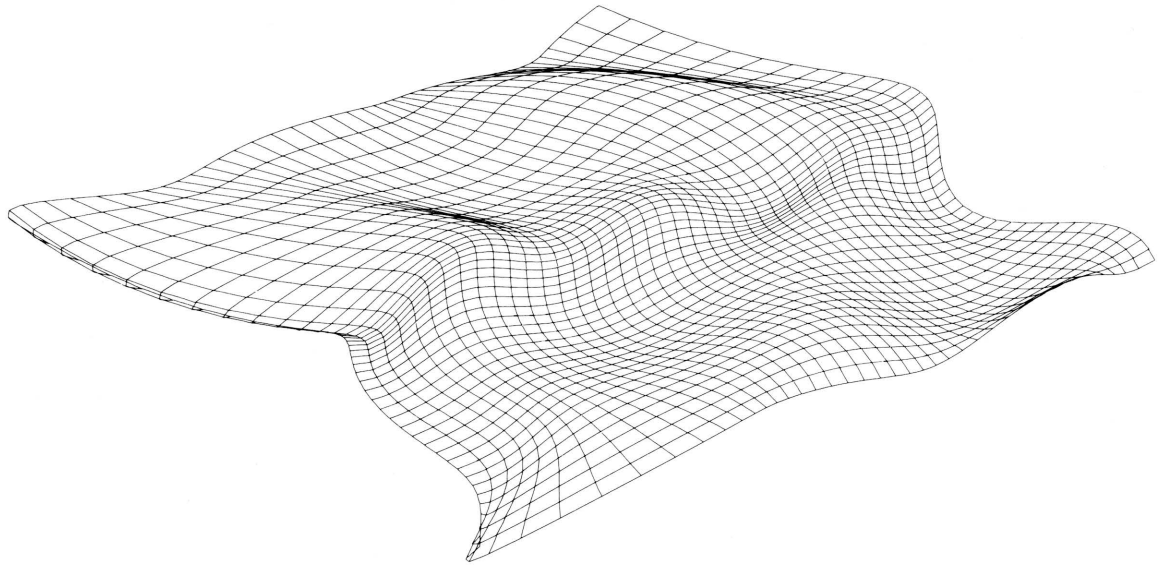
Stereo-pair 3-right



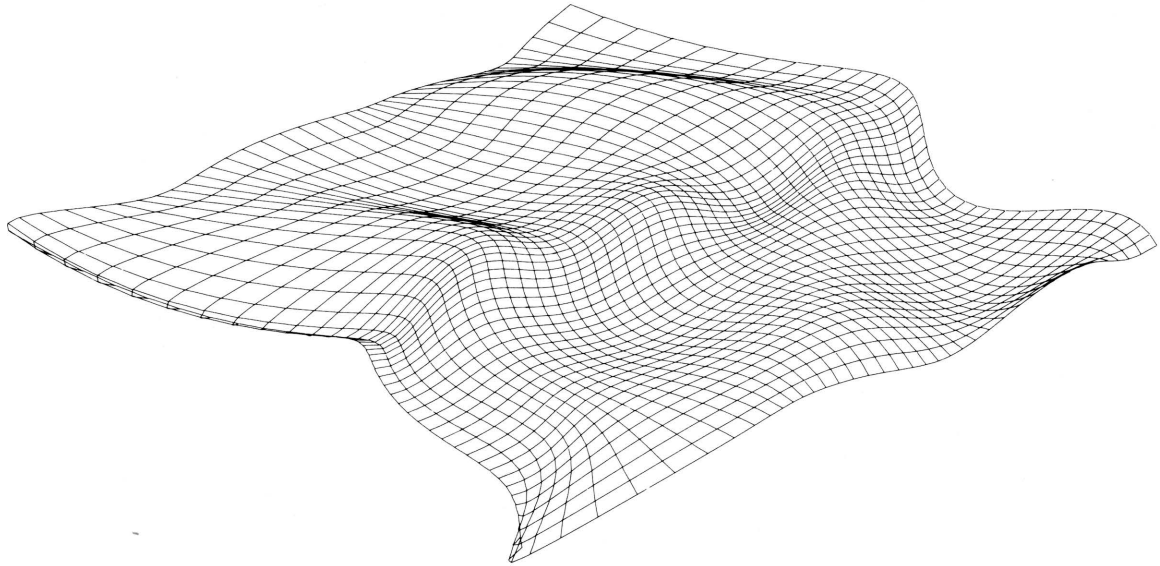
Stereo-pair 4-left



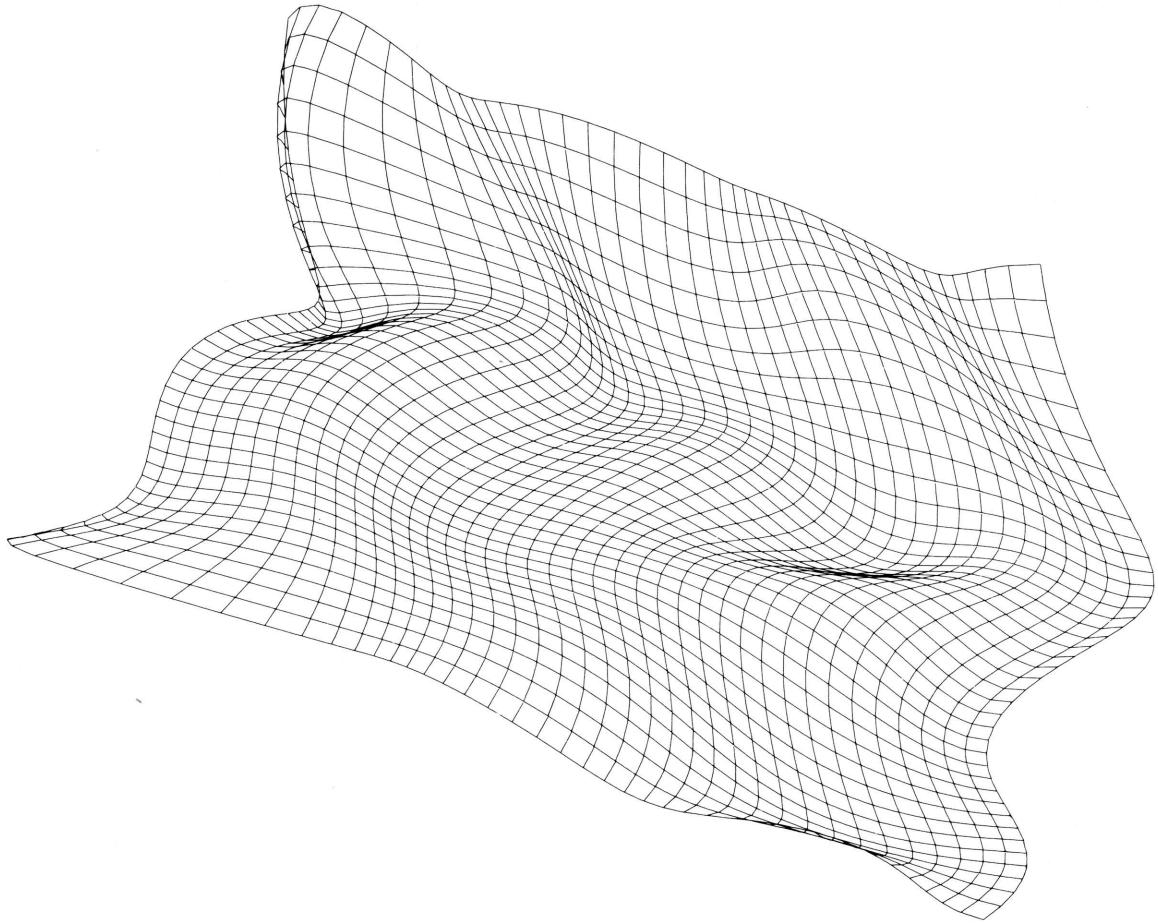
Stereo-pair 4-right



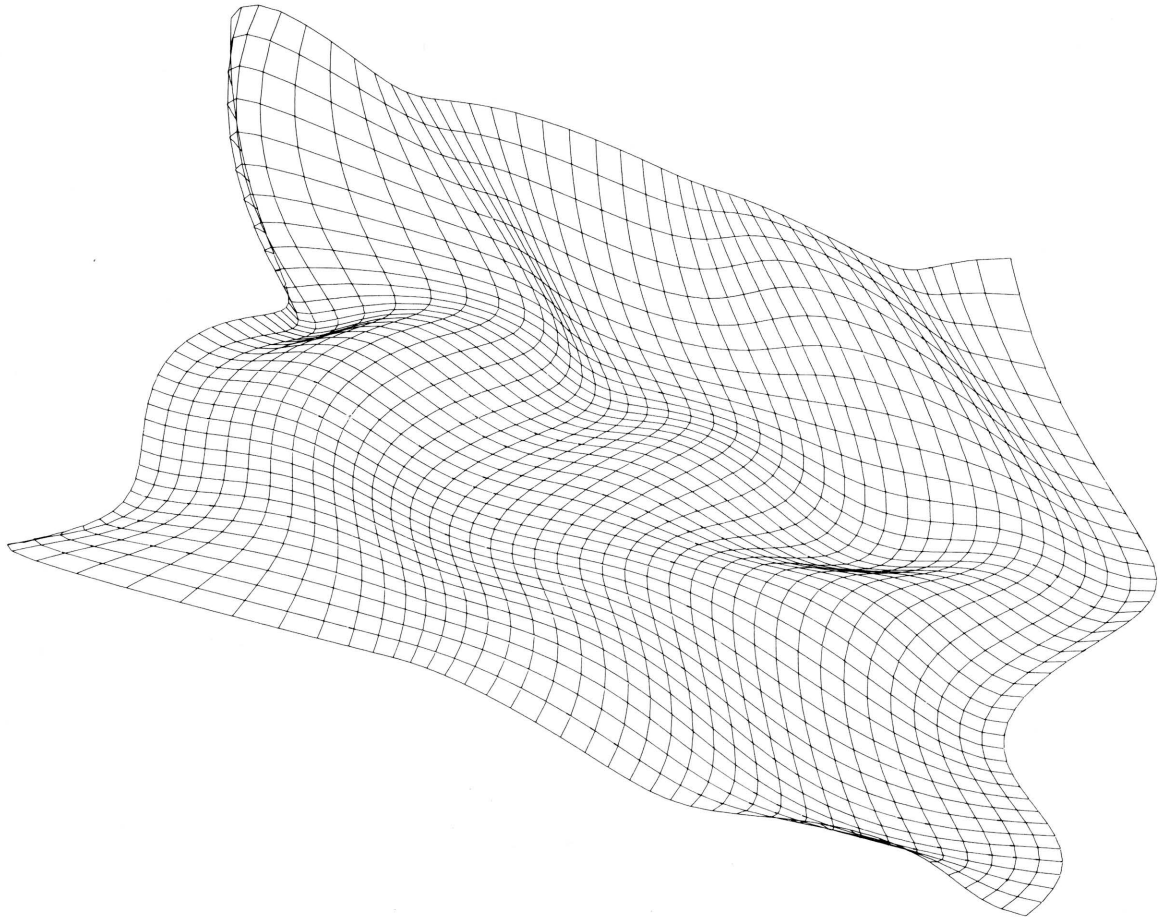
Stereo-pair 5-left



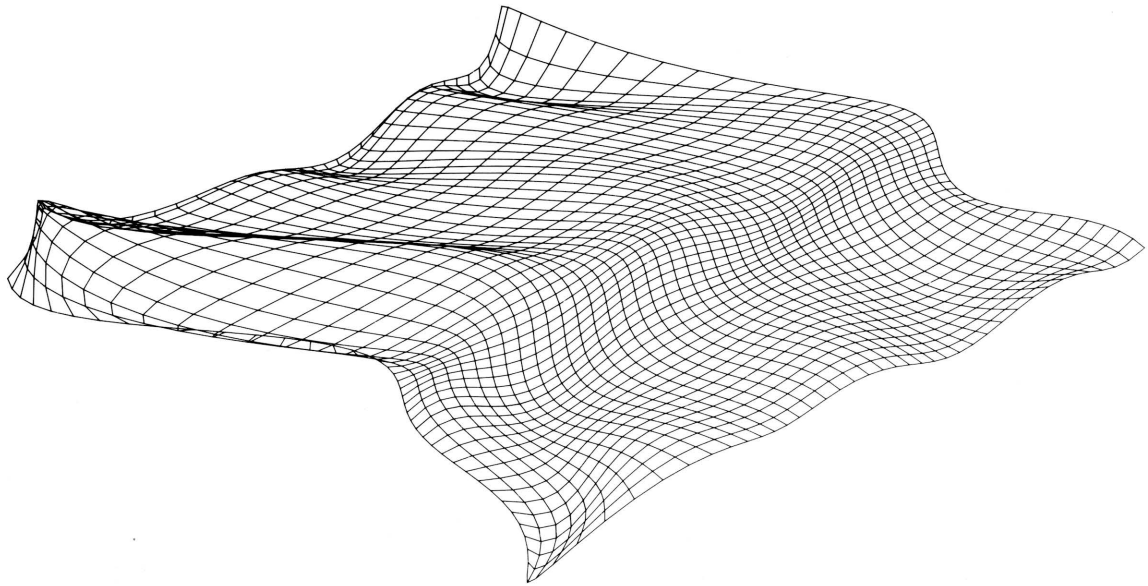
Stereo-pair 5-right



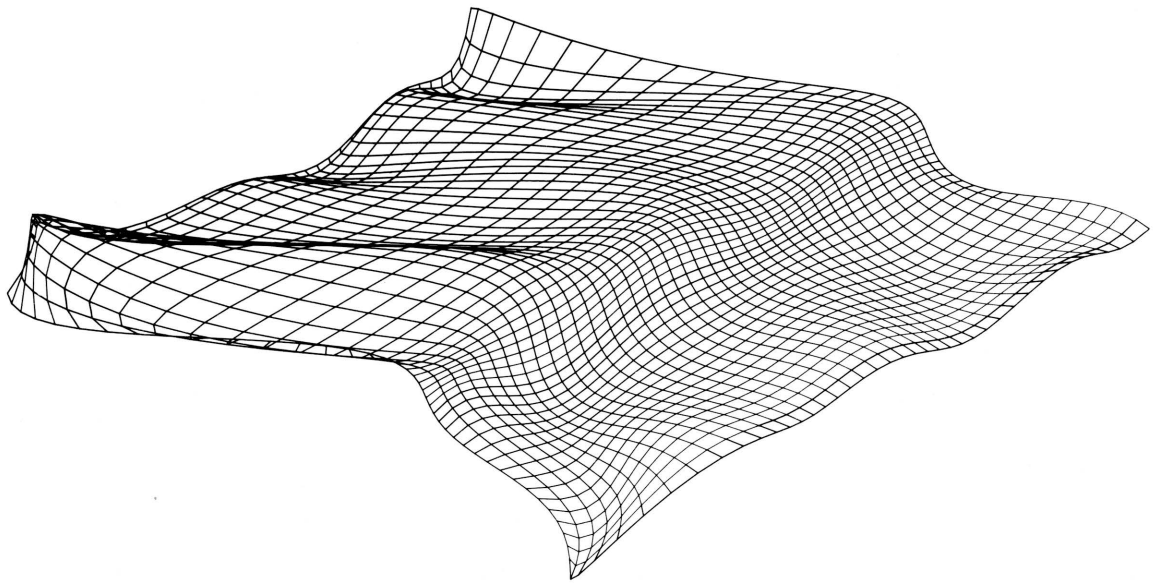
Stereo-pair 6-left



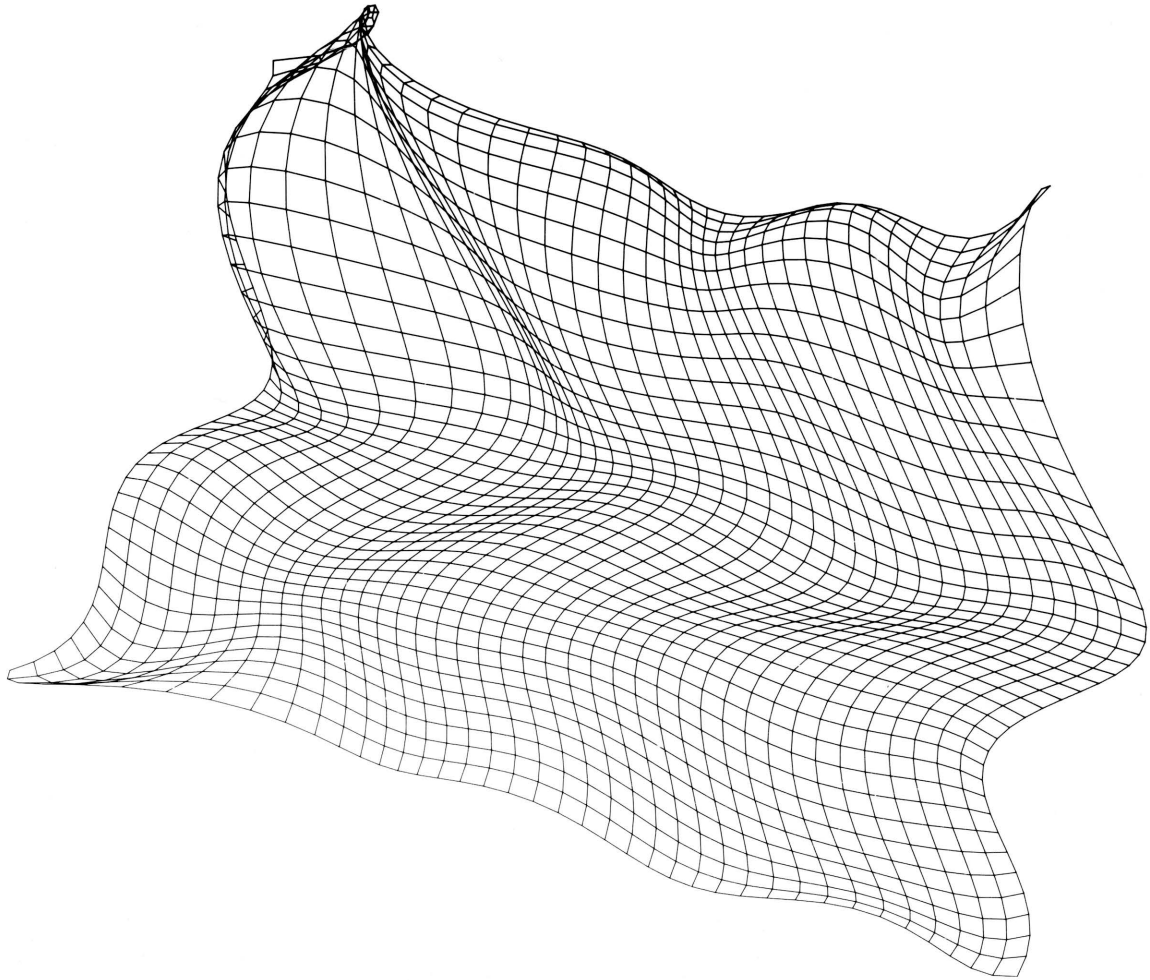
Stereo-pair 6-right



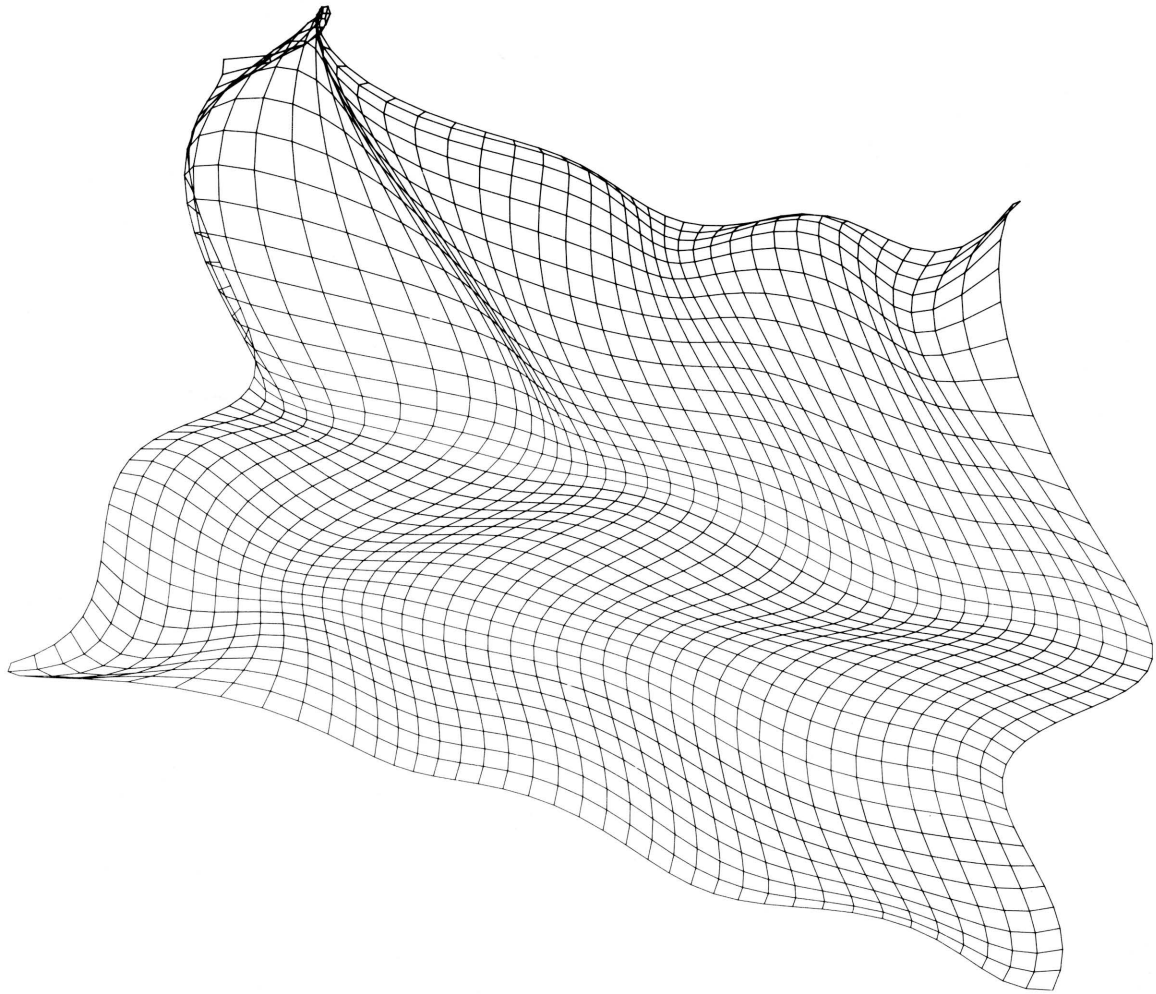
Stereo-pair 7-left



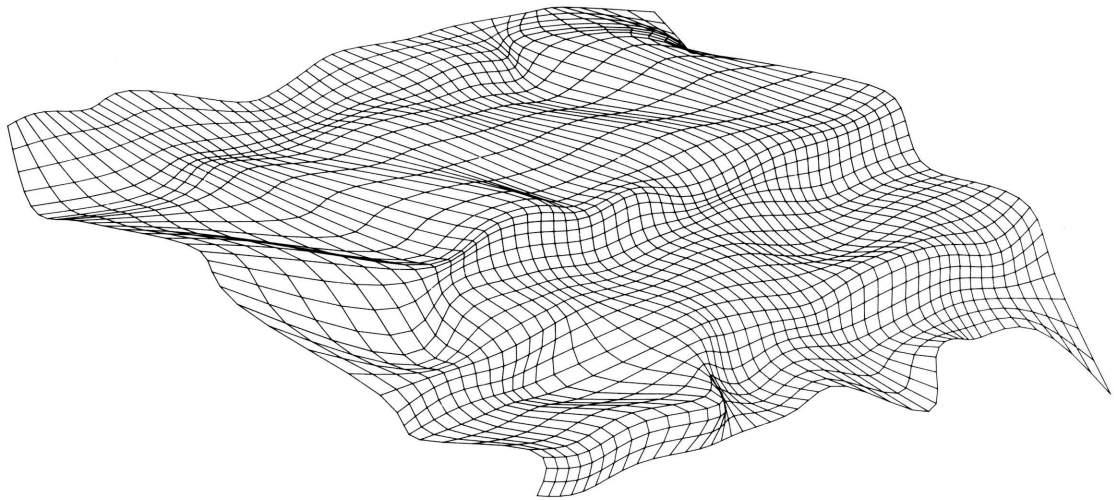
Stereo-pair 7-right



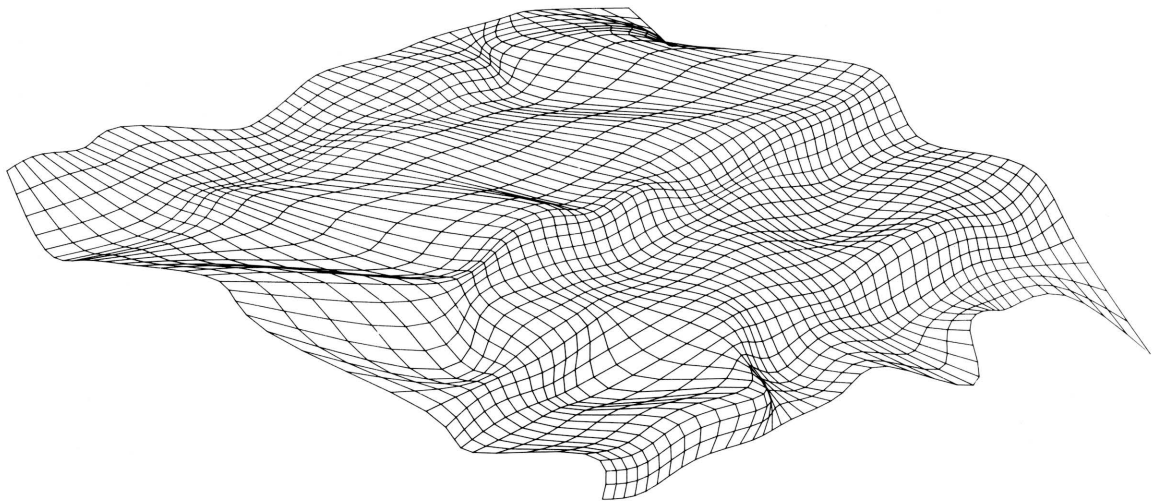
Stereo-pair 8-left



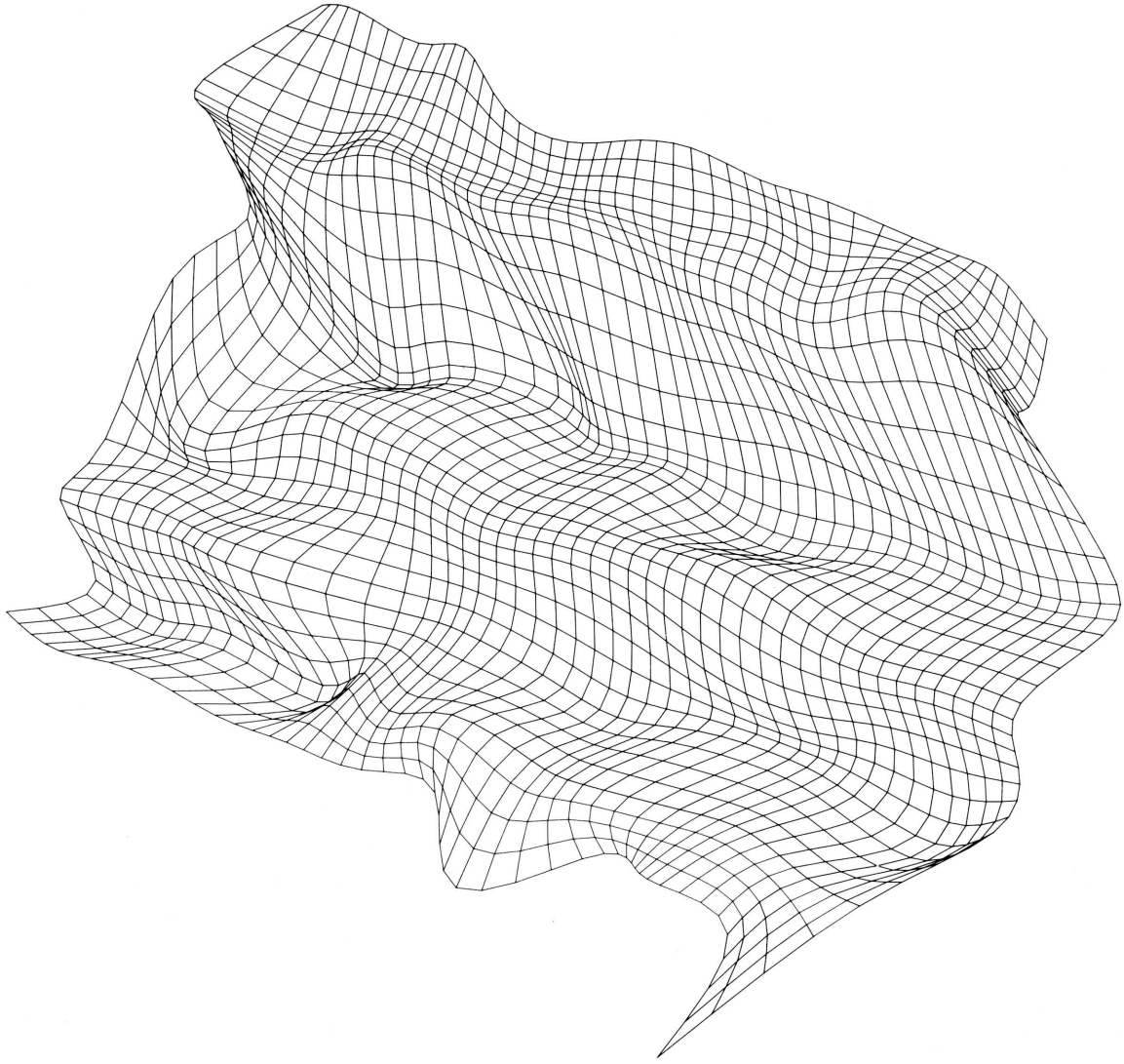
Stereo-pair 8-right



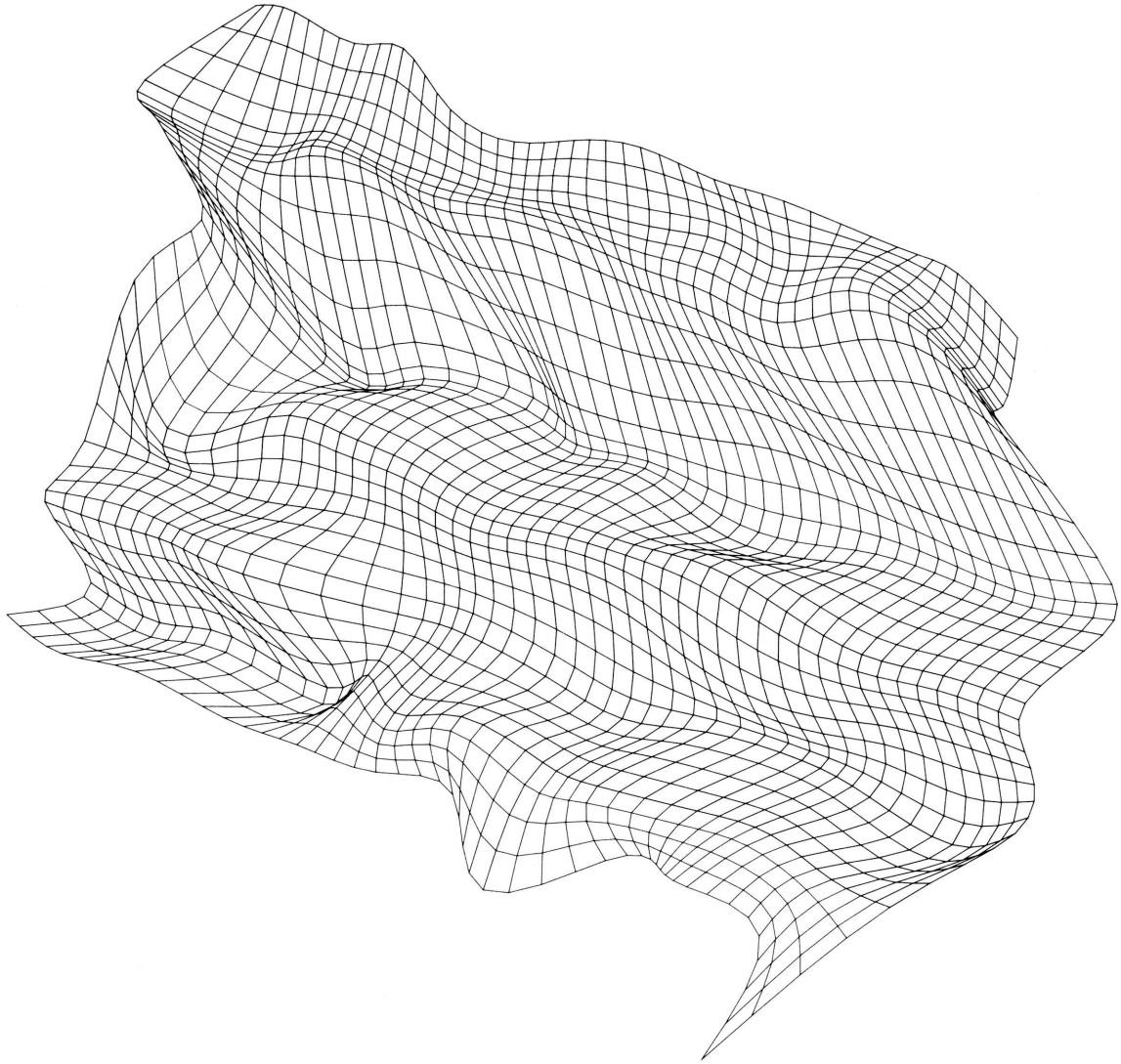
Stereo-pair 9-left



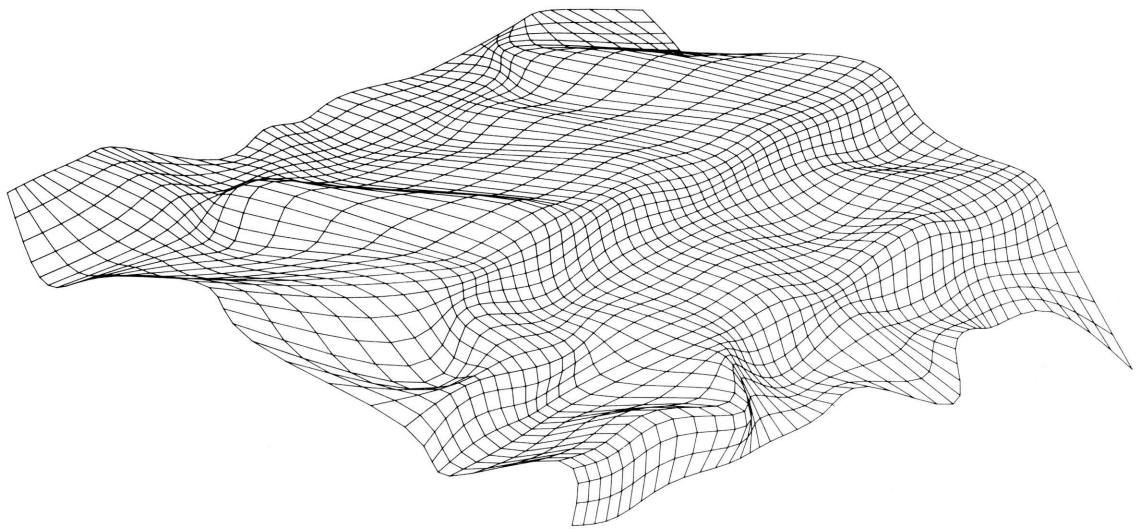
Stereo-pair 9-right



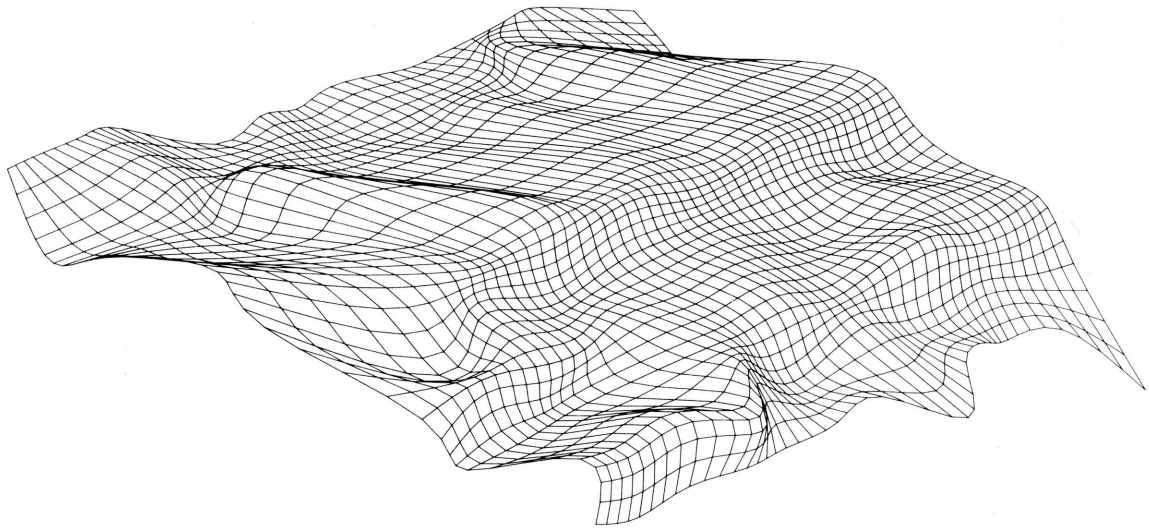
Stereo-pair 10-left



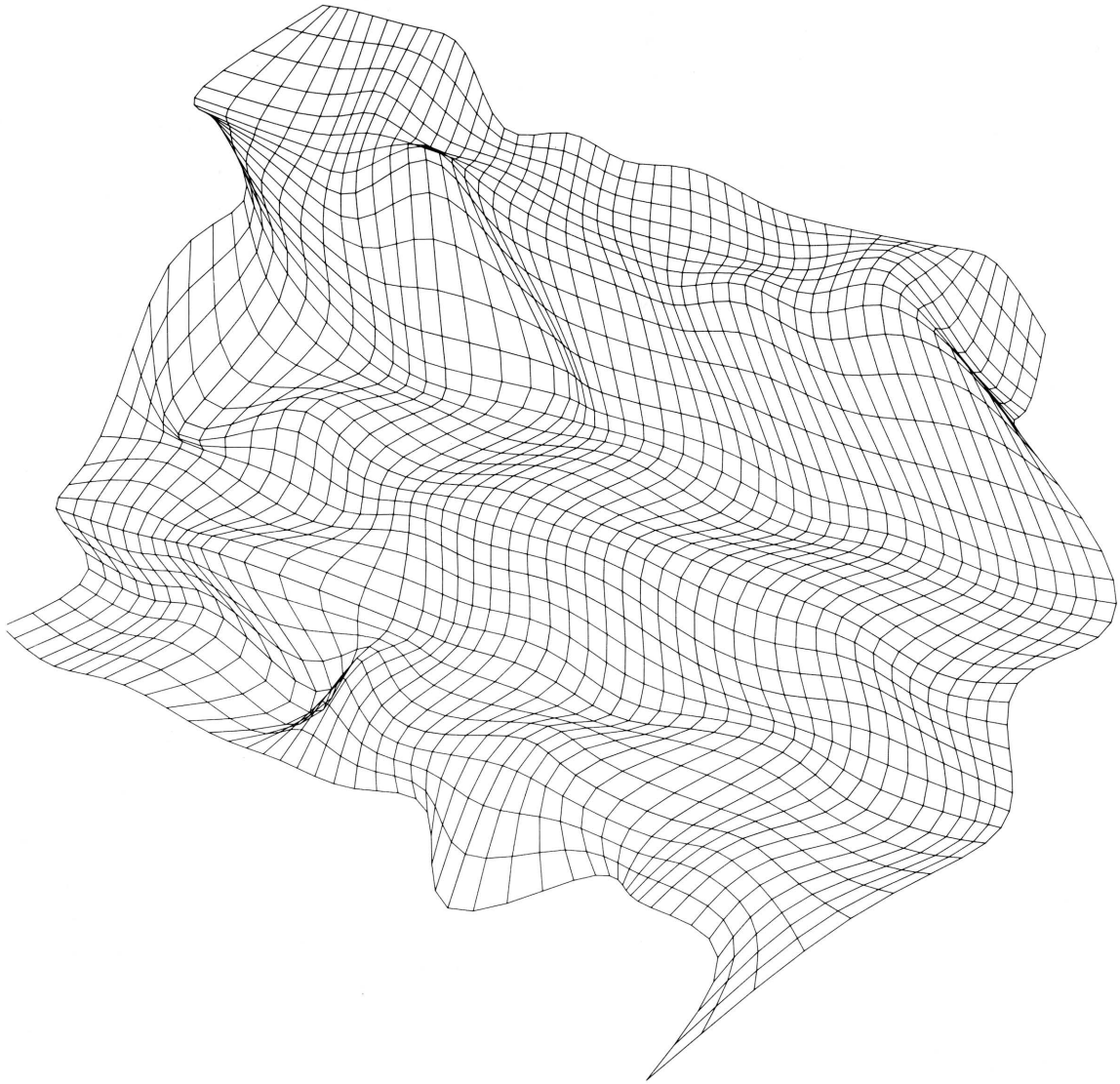
Stereo-pair 10-right



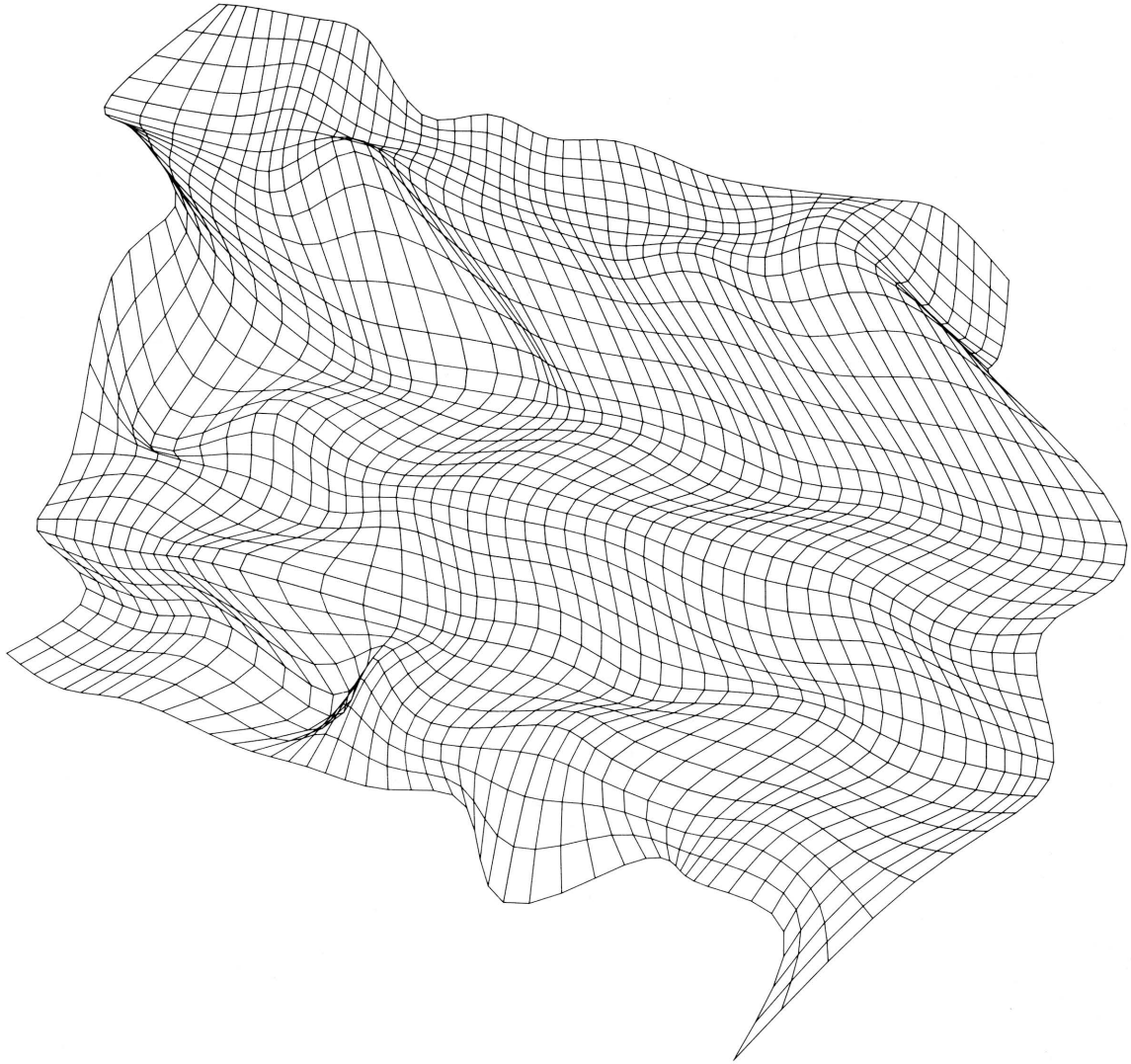
Stereo-pair 11-left



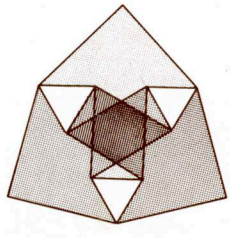
Stereo-pair 11-right



Stereo-pair 12-left



Stereo-pair 12-right



Number Four
Series on Spatial Analysis
Kansas Geological Survey
1930 Avenue A - Campus West
Lawrence, Kansas U.S.A. 66044