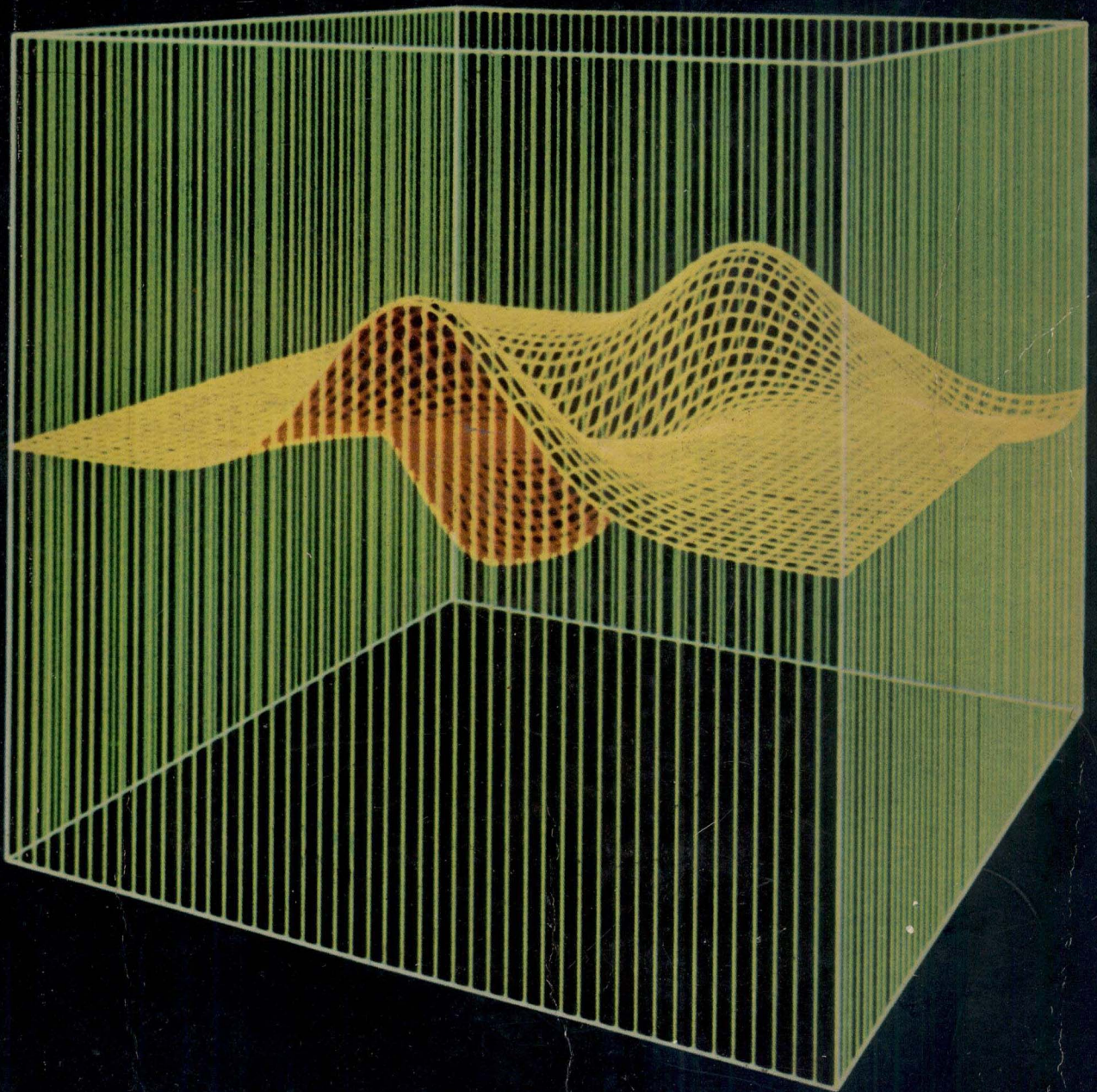


MELVIN L. PRUEITT

COMPUTER GRAPHICS

118 COMPUTER-GENERATED
DESIGNS

INCLUDING 10 IN FULL COLOR



COMPUTER GRAPHICS

118 Computer-Generated Designs

by
MELVIN L. PRUEITT

Staff Member, Los Alamos Scientific Laboratory
of the University of California

Dover Publications, Inc., New York

Copyright © 1975 by Melvin L. Prueitt.
All rights reserved under Pan American and International Copyright Conventions.

Published in Canada by General Publishing Company, Ltd., 30 Lesmill Road, Don Mills, Toronto, Ontario.

Published in the United Kingdom by Constable and Company, Ltd., 10 Orange Street, London WC 2.

Computer Graphics: 118 Computer-Generated Designs is a new work, first published by Dover Publications, Inc., in 1975.

The research and illustrations contained in this book were performed under the auspices of the United States Atomic Energy Commission.

DOVER *Pictorial Archive* SERIES

Computer Graphics: 118 Computer-Generated Designs belongs to the Dover Pictorial Archive Series. Up to ten illustrations from this book may be reproduced on any one project or in any single publication free and without special permission. Wherever possible please include a credit line indicating the title of this book, author and publisher. Please address the publisher for permission to make more extensive use of illustrations in this book than that authorized above.

The republication of this book in whole is prohibited.

International Standard Book Number: 0-486-23178-X
Library of Congress Catalog Card Number: 74-18611

Manufactured in the United States of America
Dover Publications, Inc.
180 Varick Street
New York, N.Y. 10014

INTRODUCTION

Until the present, the artist has worked by hand and eye. Today, however, man may conceive form and color for machines to translate into reality. The modern electronic computer can deftly control an electron beam and camera or manipulate a pen over paper to generate designs appealing to man. Some of these designs are simple; others make elaborate use of perspective. This book represents a sampling of such designs.

Because an artist has traditionally been defined as a creative person with sufficient hand-eye coordination to produce art, some critics would contend that computer-generated art is not art at all since the artist is not in intimate contact with his creation. But what about the individual who is deeply moved by a work of art or a view of nature's artistry and hasn't the skill to reproduce what he sees? He might be defined as a "passive artist," for he recreates within himself what he sees. He is able to relate emotionally to the color and form of the image he has generated in his mind—the image of an external object. In this sense, the passive artist may be superior to some professional artists who have the technical skill to create, but lack an emotional relationship with their creations.

The computer has brought in the day of the passive artist. It may make art possible for him; it does not make it easy. The artist may conceive of eloquent designs, but he must face the difficult task of instructing the computer to duplicate his conceptions. Computer programs are the tools that perform most of the operations in the production of computer art. Although they are invaluable in easing the difficulty of transforming the idea into the actual picture, the artist must be able to communicate his designs to the computer in terms of numbers or mathematical functions. Thus, computer art is in many ways far more difficult than other, more traditional, forms of art.

Some computer art consists of abstract lines on a two-dimensional surface, but there are computer programs that can create "three-dimensional" images. These are not, of course, actually three-dimensional, but by some logic network, the human brain transforms drawings in perspective into objects in the round. Consider the familiar "optical illusion" stairway which seems to invert itself periodically as the brain attempts to order an ambiguity. (In Figs. 69 and 70, both an isometric and a perspective plot of a stairway are drawn. The isometric plot is ambiguous; the perspective plot is not.)

The mathematics involved in teaching a computer to draw in perspective is relatively simple. Given a set of numbers which defines the positions of a set of

INTRODUCTION

points in space, one merely needs to “project” the points mathematically onto an imaginary screen. These points are represented by numbers that are transmitted to a plotting device for the generation of the image of the points and the lines that connect them. But *all* the points and lines end up on the plotted image, even the lines which should be hidden behind other features of the image. Compare Fig. 1 and Fig. 2. Many researchers have found that the problem of removing the superfluous lines—the “hidden-line problem”—does not have a simple solution. Most hidden-line removal programs consume far too much computer time to be used extensively.

PICTURE, a computer program which I developed at the Los Alamos Scientific Laboratory, constitutes a fast algorithm for plotting arrays of numbers in perspective with hidden lines removed. Although the program is designed primarily to display mathematical functions and experimental data, it also generates aesthetic forms in the hands of a designer. Once an array of numbers has been delivered to the program and a viewpoint specified, the program, in the framework of the computer, performs rotation, projection and hidden-line removal. This is all done in terms of numbers. Nothing remotely resembling an image is developed until the numbers, in the form of electronic pulses, reach a plotting device.

In order to produce a visual representation of a set of numbers which gives the illusion of a surface, PICTURE first generates a regular rectangular grid of lines. It is important that the grid lines be regularly spaced. Irregular spacing interferes with the ability of the human brain to interpret properly complex computer productions. The intersections of the grid lines of this two-dimensional mesh are then raised in the third dimension to heights that are proportional to the function which is to be represented over the surface. The computer could not actually present this three-dimensional form to man for observation unless it were to carve the function from a solid block, but it can do almost as well by producing a picture in perspective, which creates the illusion of three dimensions. This is done mathematically by projecting the grid lines of the surface onto an imaginary image screen placed between the viewer and the surface. The line of sight from the viewer’s eye to the intersection of two grid lines strikes the image screen at a point that becomes the intersection image. After all the intersections and connecting lines are cast onto the image screen, the image is in perspective.

What lines on the image screen should be hidden? Since there is no depth to the screen, the computer “sees” only a mass of lines on a two-dimensional surface—none are behind any others. We must go back to the rectangular grid to calculate the distance of lines from the viewpoint. Starting from the closest line on the grid and working away, the computer program compares the corresponding lines on the image screen. If a distant line is lower than a nearer line along a common line of sight, it is hidden. If the distant line is higher than all the lines between it and the viewer, it is visible. When a line is partially higher and partially lower than a nearer line, the intersection of the two is calculated and the former is cut to the appropriate length. By inverting the whole array the bottom surface may be treated similarly.

PICTURE has the options of adding a transparent box around the figure to aid the eye in judging sizes, labeling X , Y , and Z axes and the origin, placing

numbers to the left of the box to show the size of the box, and drawing lines (called side bars) from the edge of the surface to the base of the box. When side bars on the viewer's side are requested, the program automatically removes all lines from the bottom of the surface.

Color adds beauty to an image, but it also has the practical value of making the various parts of a complex form more distinguishable. The top of the surface, the side bars, and the box may all be different colors. Seven colors (including white) may be generated by mounting three complementary color filters, controlled electrically by the computer, between the cathode-ray tube and the camera. With four colors per picture and seven colors to choose from, 840 different color combinations are possible.

The number of color combinations may be increased by using a PICTURE option which defines the portion of the surface to be plotted. Then the color may be changed and another portion plotted. Thus checkerboards, stripes and other patterns are easily created. Color changes may be made according to heights of the function, which also allows the deletion of portions of the image. The designer's imagination is staggered by the possibilities presented by this large number of variables. After conceiving a form from among the infinite number of different shapes and translating the geometry into numbers, he must still choose colors and patterns. At what angle and inclination should it be viewed? From how far? How should the side bars be treated? How fine a grid should be used? Are lines running only in the X direction more suitable than lines running in both directions?

I sometimes allow the computer to choose the colors by means of a numerical random number generator. That eases the load of decision making when a large number of varied plots are to be made, but the color combinations are not always good. This illustrates the value of the computer in art. When the ordinary artist paints a scene, he may ask, "I wonder how it would look if the viewpoint were on the other side," or "How would it look if I used different colors?" He can find out only by considerable effort of repainting. The computer can produce, from a single array of numbers, images from a variety of angles and with different color combinations. The designer can then select his favorite.

There is still a gap between the designer's concept and the realization of the picture. It is the designer's responsibility to choose (or derive) the functions which can define his mental creation. He must formulate his ideas in the language of the computer—numbers. It is possible to punch all the numbers of an array into cards and read them into the computer, but that is laborious (a 50 x 50 array contains 2,500 points). It is easier to produce a simple computer calling program which first determines the numbers in repetitive loops and then calls the plotting program. The calling program uses mathematical functions to provide smooth curves.

The pictures that fill this book are visual representations of sets of numbers. Most of them resulted from the deliberate intention to create a particular form. Some turned out to be surprises when the result did not match the original conception. A few were errors. Some, near the end, resulted from the plotting of data from nature—experimental data. Even the heart of the atom—the nucleus—revealed some of its secrets in Figs. 99–102. The computer "con-

INTRODUCTION

ceived” Figs. 37–50, which were the offspring of pseudo-random numbers. But even here, the characteristics of the “random plots” were determined by man. These pictures were the summation of several bivariate normal distribution functions. The number of the functions and the width, height and position of each function were determined randomly, but the limits on these numbers and the type of function had to be set by man.

In many cases the array is displayed in different manners or with different viewing angles in more than one sequential figure. In some figures, we may note distortion resulting from a very close viewpoint. The computer viewpoint is moved down below the surface in some instances to give the observer a look at features which may be hidden when viewed from above. These are not “computer-eye” views, of course, because the computer deals blindly in numbers only.

We may marvel at the gentle flow of form and the color combinations of the surfaces or the rugged prominences of the images on the following pages, but these images are the progeny of what is perhaps a far more magnificent beauty—the beauty of a logic system. The computer program consists of a set of instructions which forms an intricate logic network designed to guide the computer in the performance of desired tasks. This beauty is seldom observed by anyone but the programmer himself, who is often exhilarated by inventing a unique twist in the complexities of logic, but cannot show the beauty of his invention to others because they cannot usually appreciate the whole. He must be resigned to view his creation alone. Only the fruits of a program such as PICTURE may be appreciated by others. It seems almost miraculous that, given any two-dimensional array of numbers, this program can create a perspective plot with hidden lines removed in exactly the right places even though it has never before “seen” that particular array of numbers. It is the complex logic system of the computer program that makes such pictures possible.

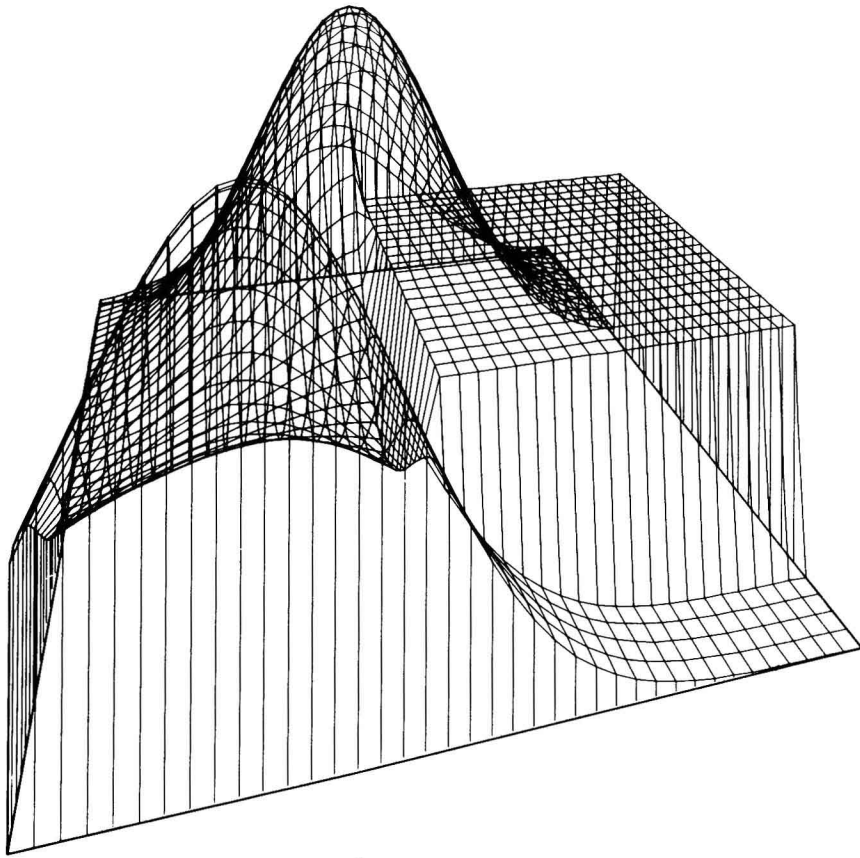
And finally there is the greatest logic system of them all—the human brain—which allows us to observe a set of lines on a sheet of paper and to envisage a three-dimensional form. This is the logic system that takes different wavelengths of light, different photon energies, and presents color to our awareness. This is the system which can tie together a sequence of static pictures isolated at points of time and award us a sense of motion. With this tool we are equipped to bridge the gap in communications left by the computer as it scrawled its lines upon a surface from the numbers the computer had thought.

Los Alamos
November, 1974

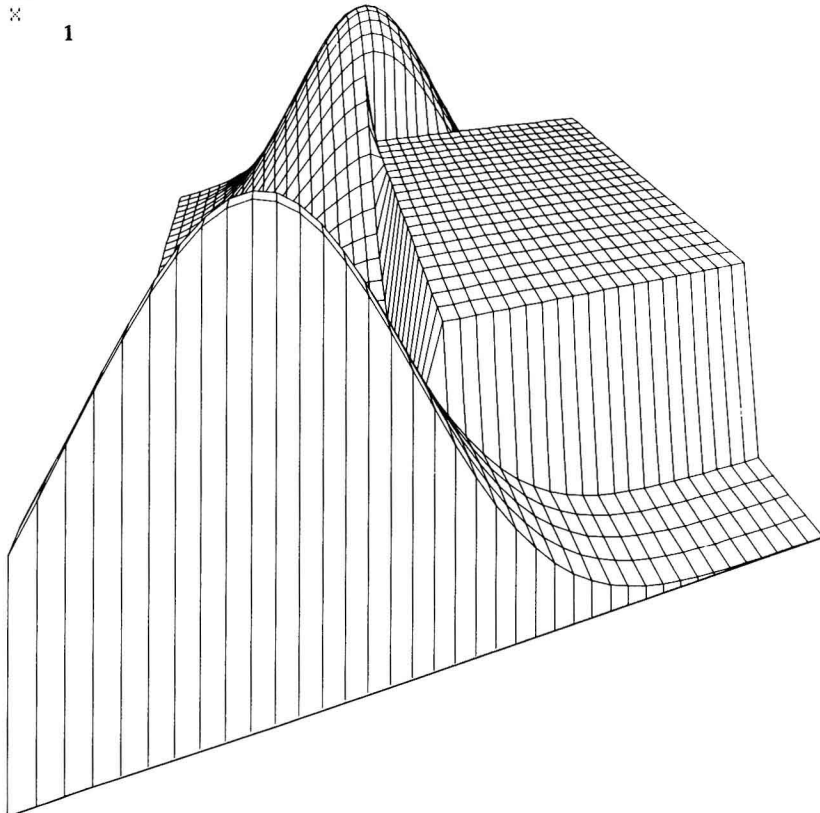
MELVIN L. PRUEITT

COMPUTER GRAPHICS

118 Computer-Generated Designs

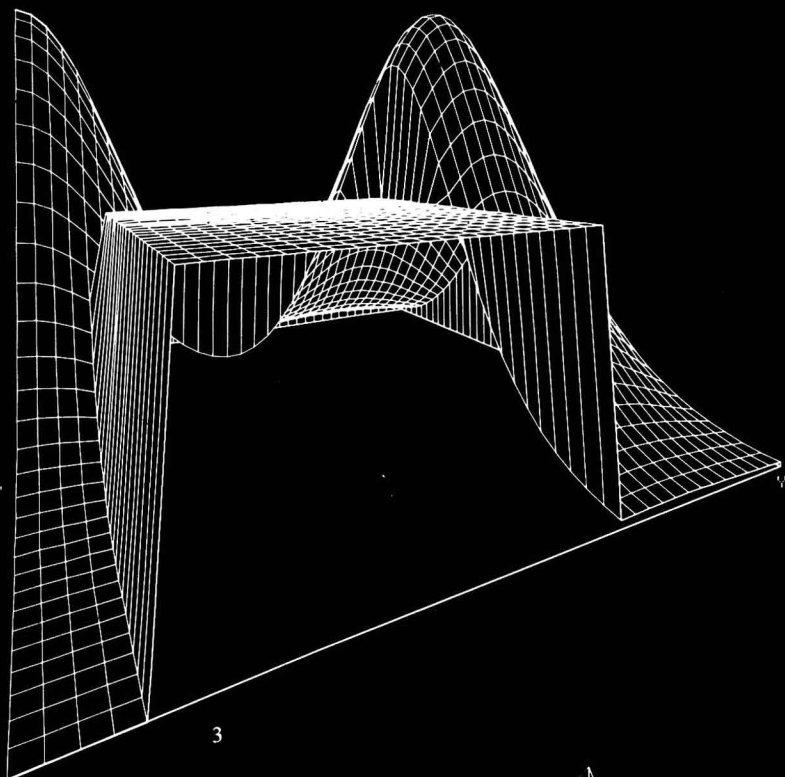


1

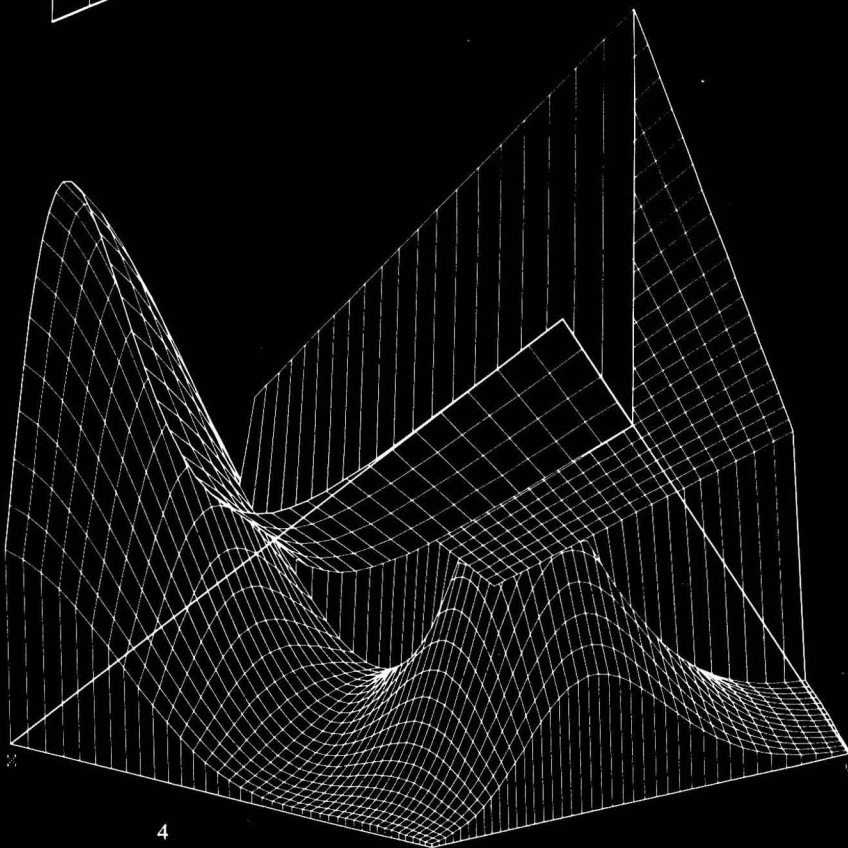


2

1, 2. Hidden-line removal. The value of removing lines which should be hidden if the figure is to appear to be in the round is made apparent by comparing Figure 1, which contains hidden lines, with Figure 2, in which the hidden lines have been removed. The human visual system can quickly assess the scene represented in Figure 2, but Figure 1 causes confusion.

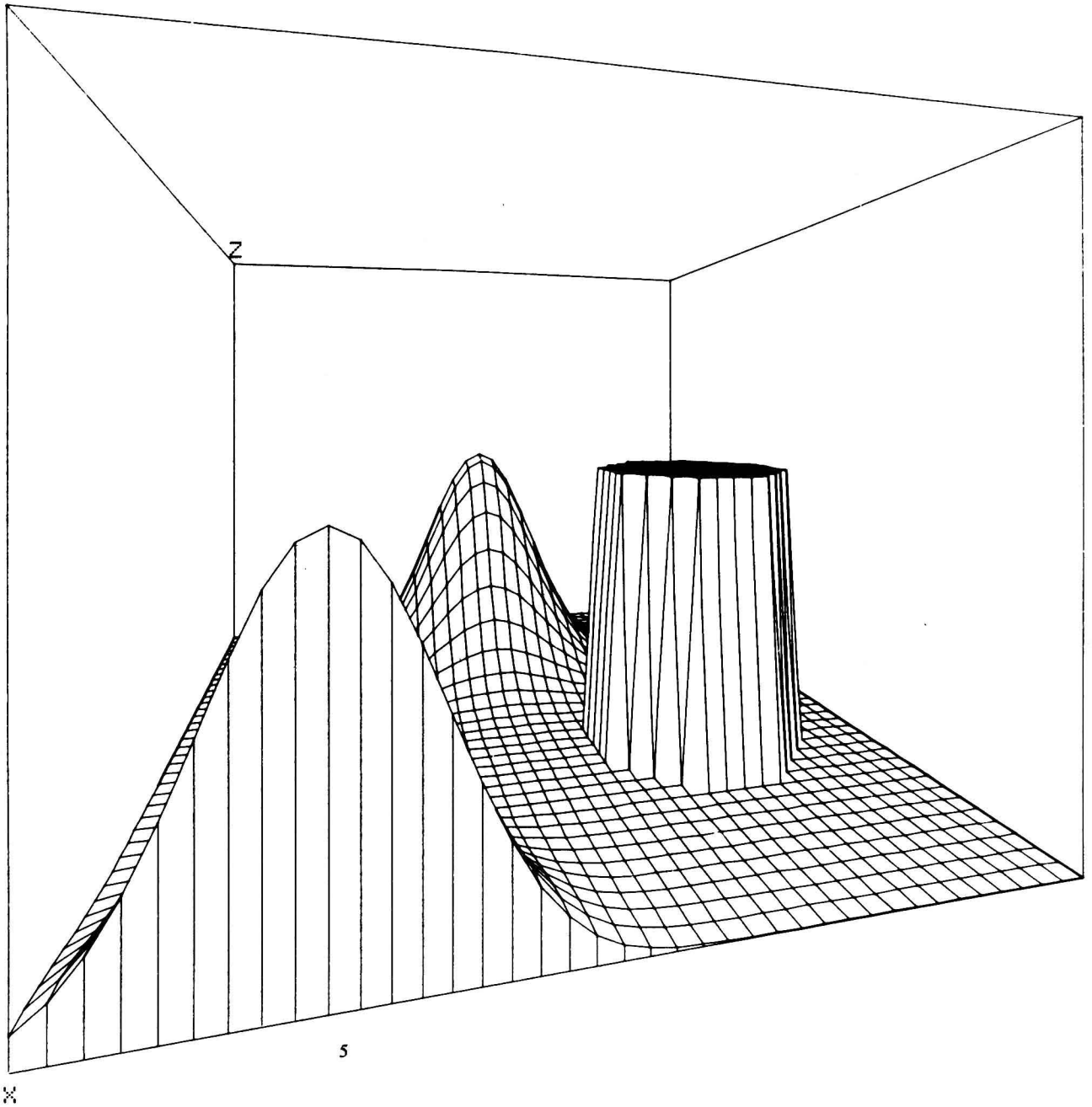


3

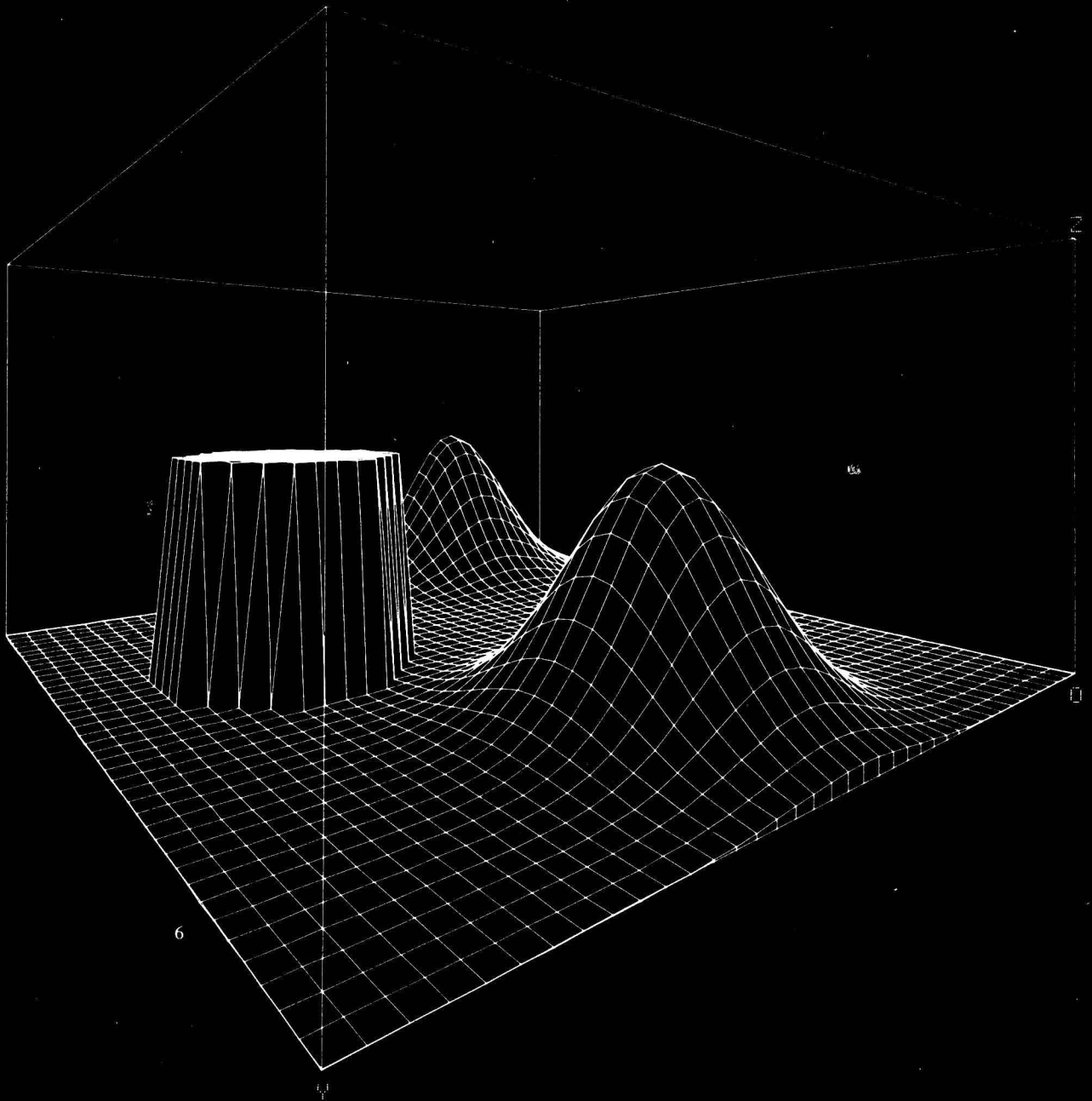


4

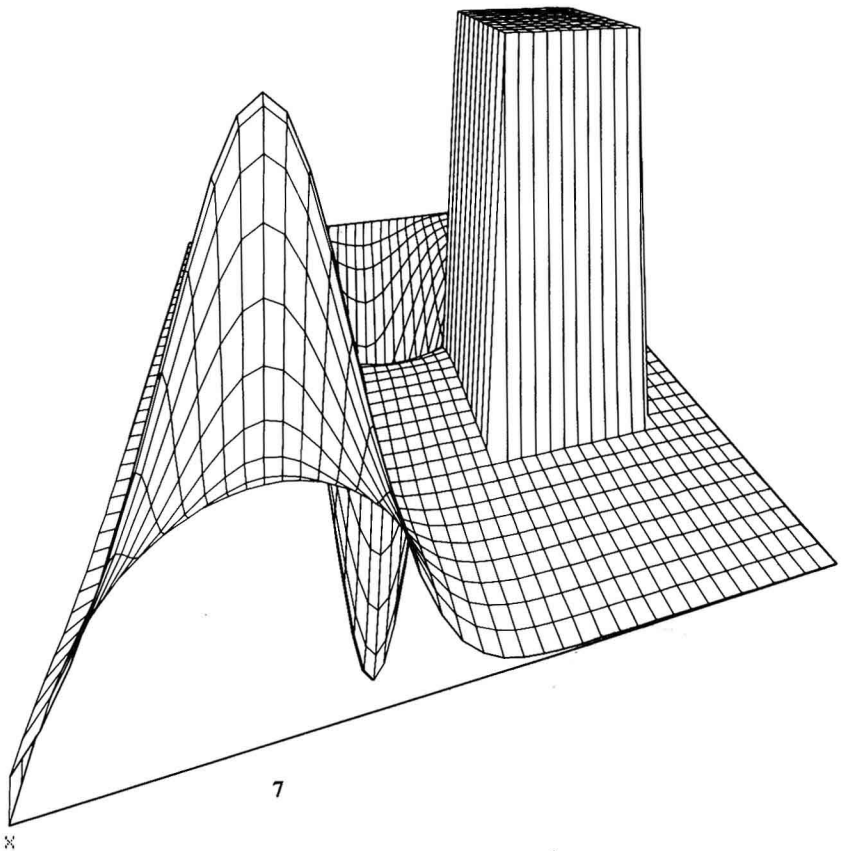
3, 4. The same array as in Figure 1, plotted from different angles. Figure 4 provides a view from below. These arrays were generated by adding a bivariate normal distribution function $F(X, Y) = C \exp\{-[(X - X_1)^2 + (Y - Y_1)^2] / W\}$ at two different places and a step function.



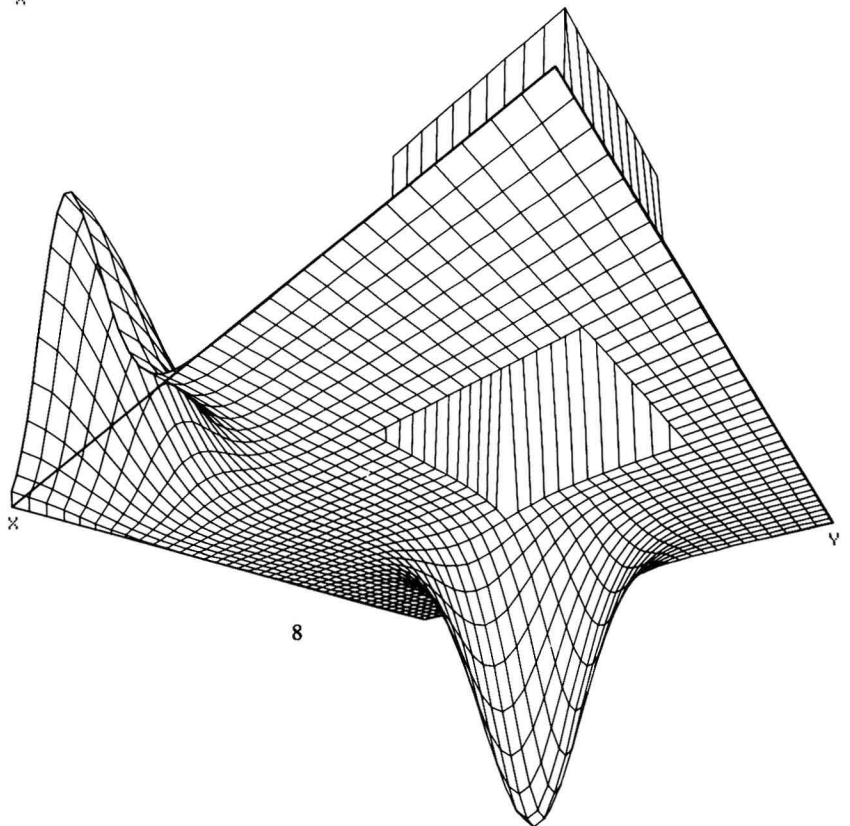
5. Two small bivariate normal distribution functions and a cylindrical column. The transparent "box" drawn around the plotting space helps the eye to judge distance and size. The X and Z axes of the coordinate system are labeled in the figure; the Y axis is invisible in this plot.

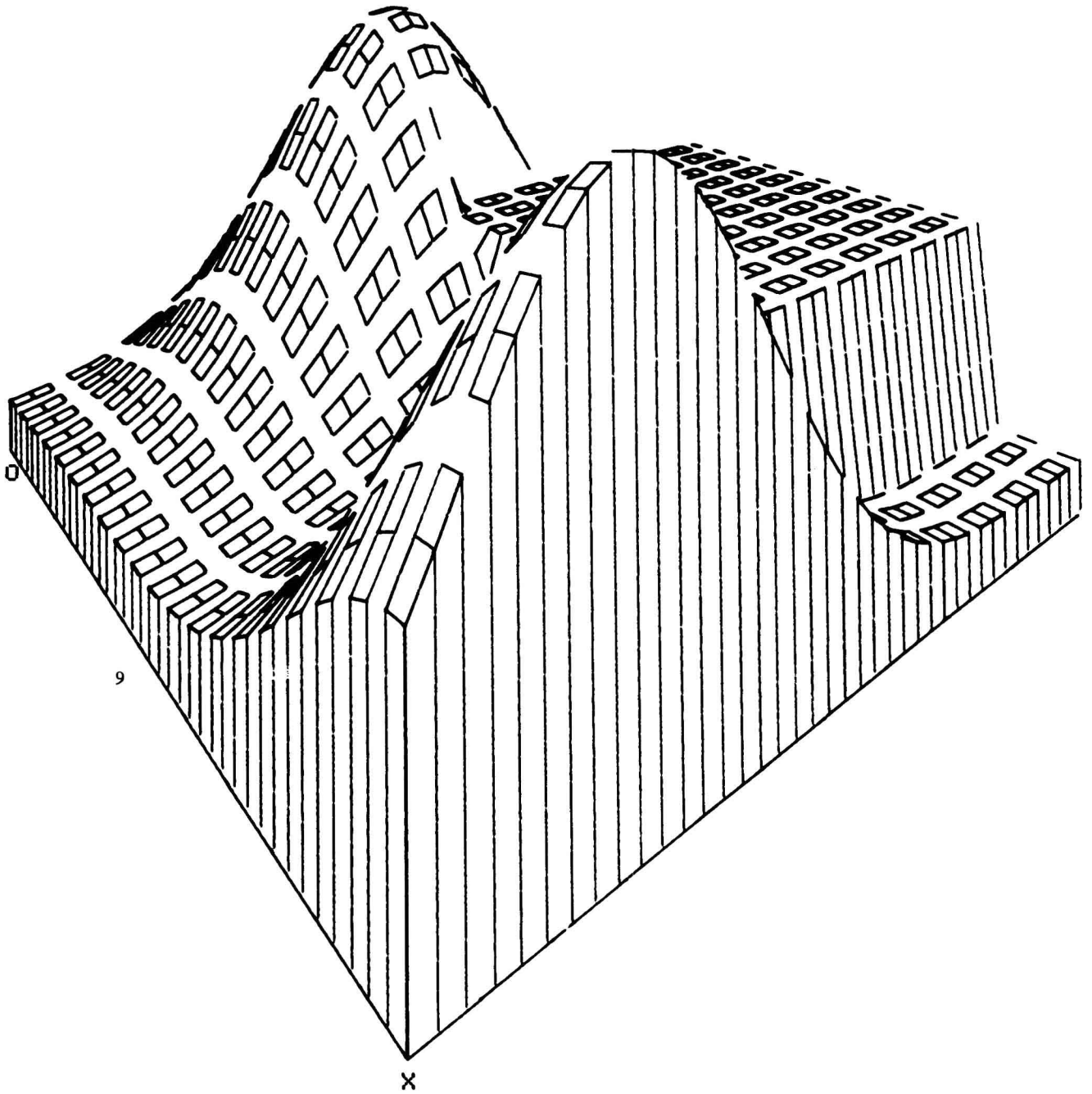


6. Two small bivariate normal distribution functions and a cylindrical column. The column is not perfectly circular because of the coarseness of the mesh.

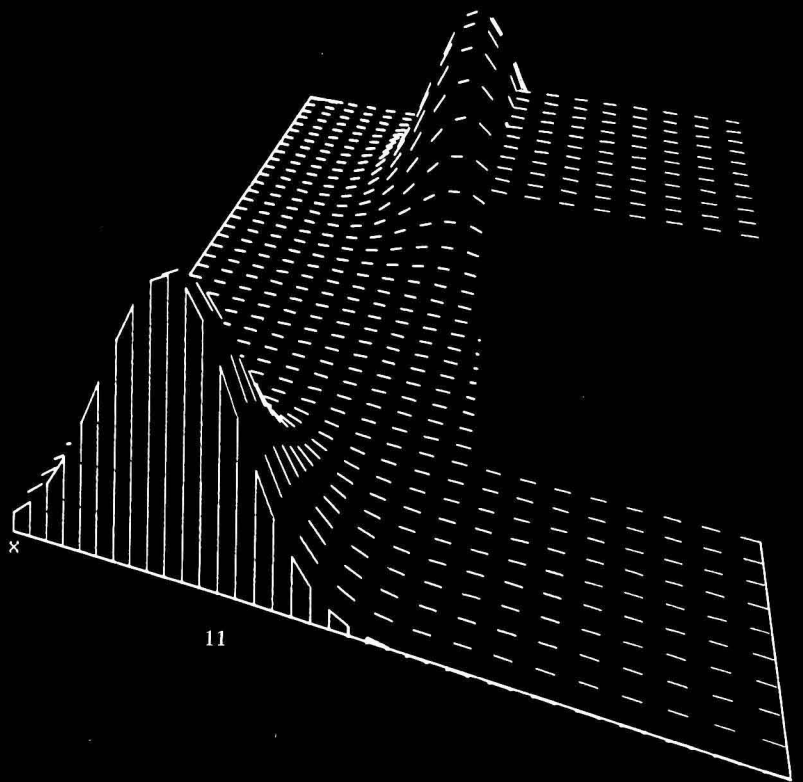
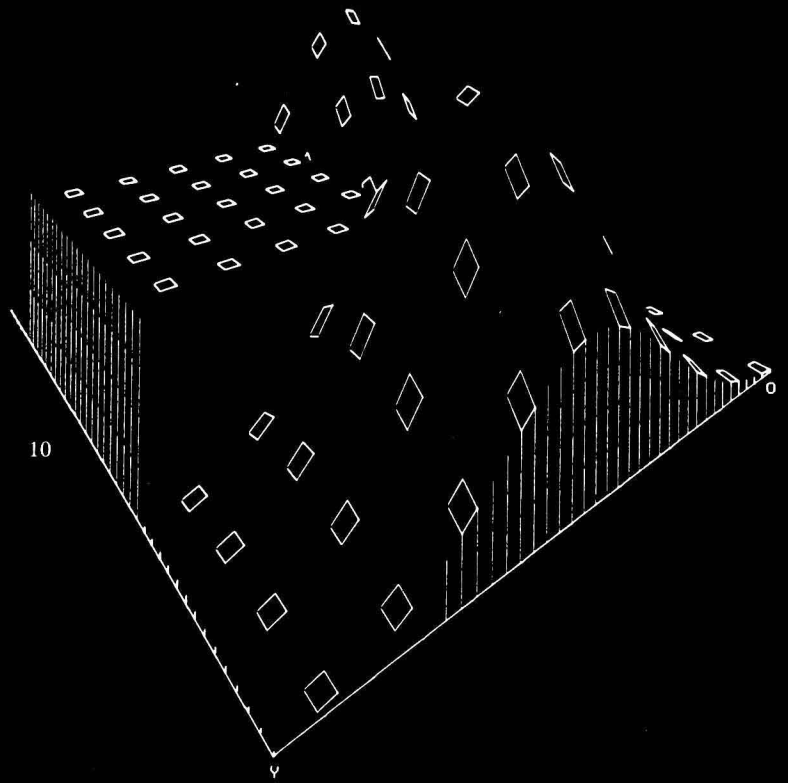


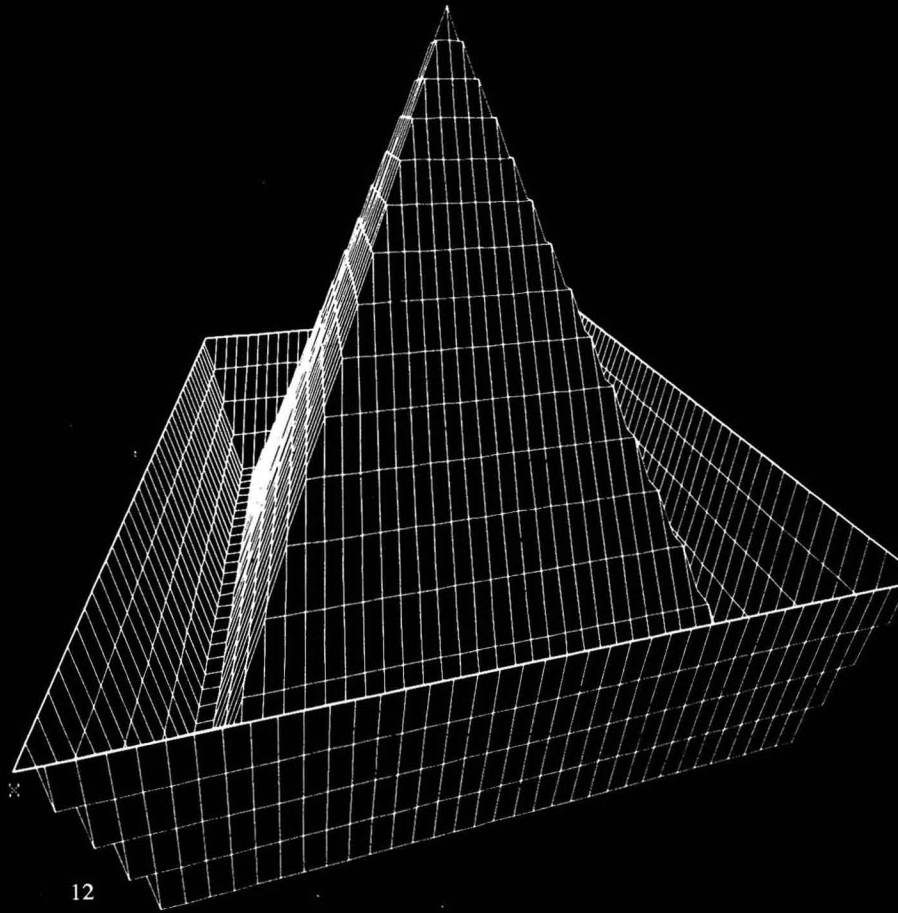
7, 8. A positive and a negative bivariate normal distribution function with a rectangular column (Fig. 7) viewed from above. Figure 8 shows the same array as Figure 7, but rotated and viewed from below.



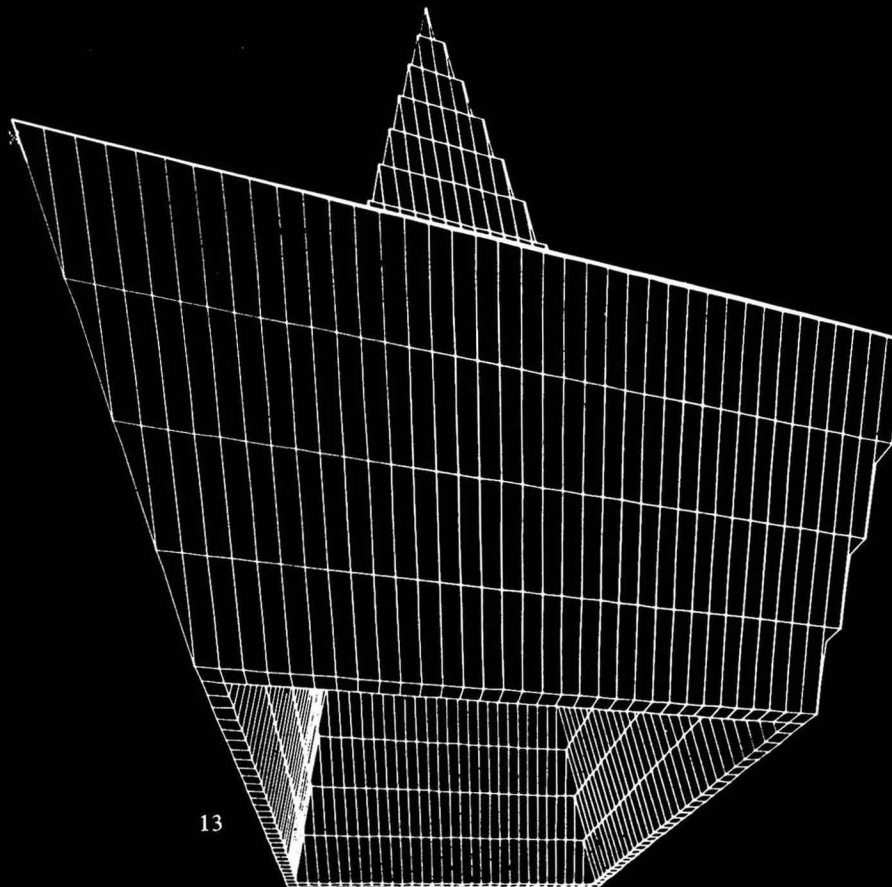


9-11. Patterned surface of the array in Figure 1. Different patterns on the same plot can be used to differentiate various areas.



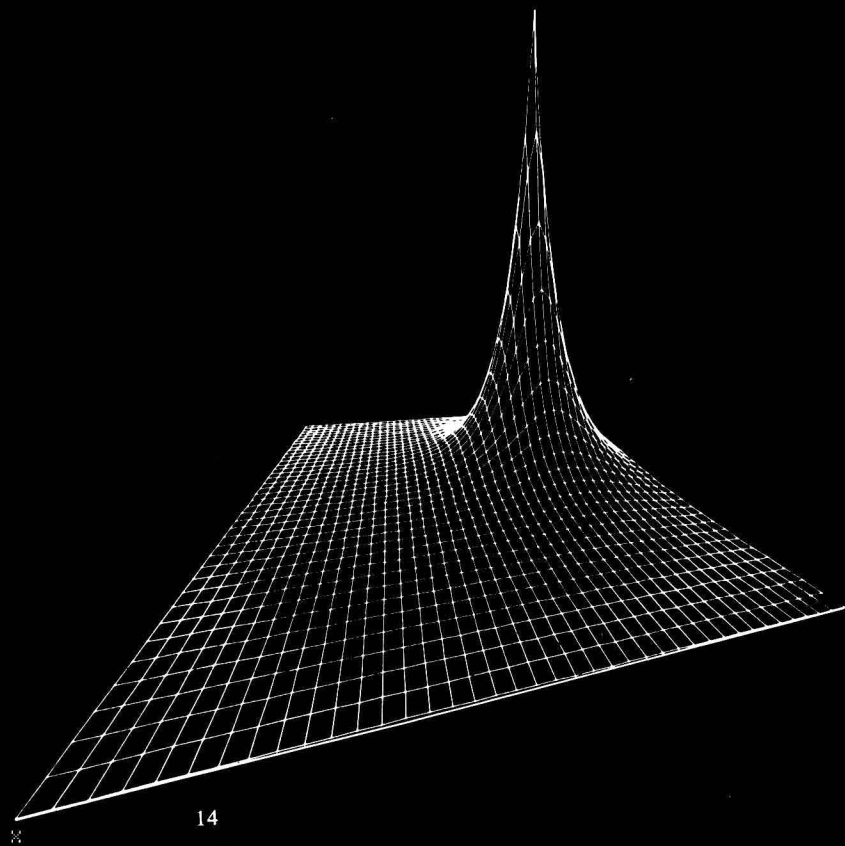


12

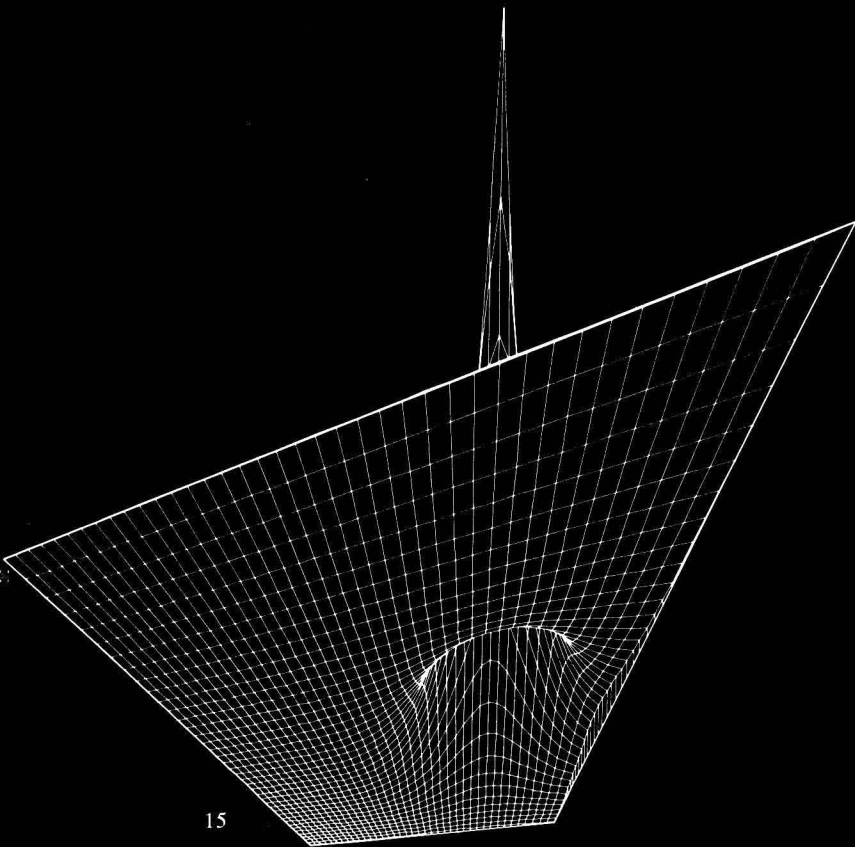


13

12, 13. Sunken pyramid, constructed from a set of intersecting planes. Figure 13 is viewed from below.



14



15

14, 15. Spires. Exponential decay
 $F(X,Y) = C \exp \left\{ -(|X - X_1| + |Y - Y_1|) / W \right\}$.