2627

NRL Report 7588

The Theory and Construction of Multivariate Gaussian Surfaces

LUISE S. SCHUETZ AND GEORGE G. ZIPFEL, JR.

Propagation Branch
Acoustics Division

August 31, 1973

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited. DDC-A

CONTENTS

Abstract ii
Problem Status ii
Authorization ii

INTRODUCTION 1

THEORY 1

Surface Generation 1
Ensemble Statistics 2
Statistical Fluctuations 7
Conditional Height Distribution 10

THE NUMERICAL ALGORITHM 11

Description of Implementation 11
Calculation of the Surface 11
Additional Statistical Comparisons 13
Tool Travel During Milling 13
Results 18

CONCLUDING REMARKS 20

ACKNOWLEDGMENTS 22

REFERENCES 22

APPENDIX A — PROGRAM OUTLINE 23

APPENDIX B — PROGRAM LISTING 25

ABSTRACT

A procedure has been developed for the construction of random rough surfaces with statistics which approximate the ideal of multivariate Gaussian statistics with a prescribed spectrum. Estimates of the deviation of the sample surfaces from the ideal ensemble statistics are made. The procedure uses a fast-Fourier-transform algorithm to produce a numerical surface which is then used to construct a physical realization of the surface via a numerically controlled milling machine. The resulting surface is one which can be used in scattering experiments, and the degree of conformity of its statistics to those of the ideal is well understood. The use of these surfaces will eliminate a major source of uncertainty in the interpretation of controlled experiments.

PROBLEM STATUS

This is a final report on one phase of the problem; work on other phases is continuing.

AUTHORIZATION

NRL Problems S01-40 and S01-47
Projects RF 52-552-402-4070 and SF 54-552-010-14317

Manuscript submitted March 27, 1973.

THE THEORY AND CONSTRUCTION OF MULTIVARIATE GAUSSIAN SURFACES

INTRODUCTION

In the experimental study of scattering of sound from random rough surfaces, it is desirable to have surfaces whose statistical properties are as well known as possible. To compare theory and experiment, researchers generally compromise and assume a Gaussian height distribution, since they are the easiest to deal with theoretically. Often this assumption has been justified by measuring the surface and showing that it deviates only slightly from Gaussian. Even if the experimental surface has a Gaussian height distribution, this does not guarantee a multivariate Gaussian height distribution. Since other assumptions or approximations are usually made in the theoretical aspects of scattering, the removal of those assumptions concerned with surface statistics is necessary. It is the purpose of this report to outline a method to construct a surface with a multivariate Gaussian height distribution and a prescribed correlation function. The procedure that we have used results in surfaces whose degree of accuracy is relatively well known. The surfaces so constructed are to our knowledge unique.

The present report contains two main sections. The first main section contains a theoretical description of the procedure with some error estimates. The second main section contains a description of the algorithm used to numerically generate and physically form a surface. The second section also briefly discusses a specific surface with a Gaussian correlation function constructed using the program. An outline and a listing of the computer programs written for the study is given in the appendixes.

THEORY

Surface Generation

The stochastic process used to generate the surfaces is the usual (1, 2) random-phase-Fourier-series process. Specifically the surface height is the real part of a complex function of x and y (or $\mathbf{x}$), where x and y are the coordinates in a plane parallel to the average surface, that is

$$h(\mathbf{x}) = \operatorname{Re} A(\mathbf{x}) . \tag{1}$$

The function $A(\mathbf{x})$ is expressed in terms of a random-phase Fourier series

$$A(\mathbf{x}) = A_T^{-1/2} \sum_{n=-N_y}^{N_y} \sum_{m=-N_x}^{N_x} e^{i\mathbf{p}_{mn}\cdot\mathbf{x}} \sqrt{2S(\mathbf{p}_{mn})}\; e^{i\phi_{mn}} , \tag{2}$$

where

$$\mathbf{p}_{mn} = \frac{2\pi m}{L_x}\,\hat{i}_x + \frac{2\pi n}{L_y}\,\hat{i}_y \tag{3}$$

and

$$A_T = L_x L_y \ .$$

The phases ϕ_{mn} are $(2N_x + 1)(2N_y + 1)$ independent random variables uniformly distributed over the interval $(0, 2\pi)$. Each surface formed in this way is defined on a rectangle L_x by L_y and can be periodically continued outside this region. The coefficients in the series have been chosen so that $S(\mathbf{p})$ is the spectrum of the surface. The fact that $h(\mathbf{x})$ is a Gaussian process is a consequence of the central limit theorem (1). However, for completeness, the process of the preceding paragraph will be analyzed in detail. A detailed analysis is also useful in order to develop estimates of the deviations from the ideal inherent in any finite surface constructed from Eq. (1).

If one assumes multivariate Gaussian statistics, then the lowest order or two-point correlation function contains all the information necessary to specify the statistics, and the higher order correlation functions are simply related to the two-point correlation function. If the statistics are not multivariate Gaussian, then this is no longer true, and it is important to know how the correlation functions deviate from the ideal.

These deviations are due to the finite surface size and appear in two ways. One way is the deviation of the ensemble statistics from the ideal. This is treated in the next subsection, and it is found that this deviation is negligible compared to the second deviation, which is the fluctuations in the statistics of an individual surface from the ensemble statistics. This is treated in the third subsection "Statistical Fluctuations" by estimating the mean-squared fluctuations in the correlation functions as a function of the order of the correlation function (equation (22)). The fluctuations increase as the order increases, and when the fluctuations become comparable to the maximum value of the correlation function, this is taken to be the maximum usable order. A specific example is contained in the third subsection. Since the correlation functions appear in the theory of surface scattering, this estimate is of particular interest.

Ensemble Statistics

The statistical quantities of interest to us are the height correlation functions, which follow directly from the characteristic functions associated with the joint probability distributions. First let us consider the joint probability density for the complex quantity $A(\mathbf{x}_i)$, which will be represented by a two-dimensional vector $\mathbf{A}(\mathbf{x}_i)$. The usual convention of associating the imaginary part of $A(\mathbf{x}_i)$ with the 2 axis and the real part with the 1 axis will be followed.

It is easiest to consider building the surface up one term at a time, so for the time being we will think of the series in Eq. (2) as a single-parameter sum. The problem viewed in this way can be seen to be a type of random walk process.

The N-point joint probability density for $\mathbf{A}(\mathbf{x}_i)$ generated by a Fourier series of K terms will be denoted by $P_K[\mathbf{A}(\mathbf{x}_1), \ldots, \mathbf{A}(\mathbf{x}_N)]$. The density functions are most easily

found by writing an integral recursion relation which relates $P_K[\mathbf{A}(\mathbf{x}_1), \ldots, \mathbf{A}(\mathbf{x}_N)]$ to $P_{K-1}[\mathbf{A}(\mathbf{x}_1), \ldots, \mathbf{A}(\mathbf{x}_N)]$. Since each term in the series is statistically independent of the others, the recursion relation is particularly simple and is given by

$$P_K[\mathbf{A}(\mathbf{x}_1), \ldots, \mathbf{A}(\mathbf{x}_N)] = \int \Delta p_K[\mathbf{A}(\mathbf{x}_1) - \mathbf{B}(\mathbf{x}_1), \ldots, \mathbf{A}(\mathbf{x}_N) - \mathbf{B}(\mathbf{x}_N)]$$

$$\times P_{K-1}[\mathbf{B}(\mathbf{x}_1), \ldots, \mathbf{B}(\mathbf{x}_N)]\, d^2B(\mathbf{x}_1) \ldots d^2B(\mathbf{x}_N) . \tag{4}$$

This says that the probability of $\mathbf{A}(\mathbf{x})$ having a certain value at the points $\mathbf{x}_1, \ldots, \mathbf{x}_N$ with K terms in the series is the sum over the probabilities of having a certain value with $K - 1$ terms in the series times the probability that Kth term will increment the value of $\mathbf{A}(\mathbf{x})$ by $\mathbf{A}(\mathbf{x}) - \mathbf{B}(\mathbf{x})$.

Equation (4) can be solved immediately by introducing the characteristic functions associated with the joint probability densities P_K and Δp_K. The characteristic functions C_K and c_K are defined by

$$P_K[\mathbf{A}(\mathbf{x}_1), \ldots, \mathbf{A}(\mathbf{x}_N)] = \frac{1}{(2\pi)^{2N}} \int \exp\left[i \sum_{m=1}^{N} \mathbf{k}_m \cdot \mathbf{A}(\mathbf{x}_m)\right]$$

$$\times C_K(\mathbf{k}_1, \ldots, \mathbf{k}_N)\, d^2k_1 \ldots d^2k_N \tag{5}$$

and

$$\Delta p_K[\mathbf{A}(\mathbf{x}_1), \ldots, \mathbf{A}(\mathbf{x}_N)] = \frac{1}{(2\pi)^{2N}} \int \exp\left[i \sum_{m=1}^{N} \mathbf{k}_m \cdot \mathbf{A}(\mathbf{x}_m)\right]$$

$$\times c_K(\mathbf{k}_1, \ldots, \mathbf{k}_N)\, d^2k_1 \ldots d^2k_N . \tag{6}$$

Substitution of (5) and (6) into (4) gives

$$\ln C_K(\mathbf{k}_1, \ldots, \mathbf{k}_N) = \sum_{j=1}^{K} \ln c_j(\mathbf{k}_1, \ldots, \mathbf{k}_N) .$$

Rewriting this equation in the form appropriate to Eq. (2) gives

$$\ln C_{N_x N_y}(\mathbf{k}_1, \ldots, \mathbf{k}_N) = \sum_{m=-N_y}^{N_y} \sum_{n=-N_x}^{N_x} \ln c_{mn}(\mathbf{k}_1, \ldots, \mathbf{k}_N) . \tag{7}$$

The probability density $\Delta p_{mn}[\mathbf{A}(\mathbf{x}_1), \ldots, \mathbf{A}(\mathbf{x}_N)]$ and hence $c_{mn}(\mathbf{k}_1, \ldots, \mathbf{k}_N)$ can be determined as follows: Since p_{mn} corresponds to only one term in the Fourier series, the $\mathbf{A}(\mathbf{x}_i)$ differ only by a phase factor. Thus $\mathbf{A}(\mathbf{x}_2), \ldots, \mathbf{A}(\mathbf{x}_N)$ are determined once

$\mathbf{A}(\mathbf{x}_1)$ is known. Since the phase of $\mathbf{A}(\mathbf{x}_1)$ is uniformly distributed over $(0, 2\pi)$, the probability distribution for a single component surface can be expressed as

$$\Delta p_{mn}[\mathbf{A}(\mathbf{x}_1), \ldots, \mathbf{A}(\mathbf{x}_N)] = \frac{1}{2\pi D_{mn}} \prod_{j=2}^{N} \delta_2[\mathbf{A}(\mathbf{x}_j) - \mathbf{R}^j_{mn} \cdot \mathbf{A}(\mathbf{x}_1)]\, \delta_1(|\mathbf{A}(\mathbf{x}_1)| - D_{mn}) \tag{8}$$

where D_{mn} is the amplitude of the mnth term in the series given by Eq. (2), that is,

$$D_{mn} = A_T^{-1/2} \sqrt{2S(\mathbf{p}_{mn})}\,, \tag{9}$$

and $\delta_1(\mathbf{x})$ and $\delta_2(\mathbf{x})$ are the one- and two-dimensional Dirac delta functions respectively. The dyad $\mathbf{R}^j_{mn}$ produces a rotation corresponding to the phase shift induced by $\exp\,[i\mathbf{p}_{mn} \cdot (\mathbf{x}_j - \mathbf{x}_1)]$. Explicitly

$$\mathbf{V} \cdot \mathbf{R}^j_{mn} \cdot \mathbf{V}' = (V_1 V_2) \begin{bmatrix} \cos\theta^j_{mn} & -\sin\theta^j_{mn} \\ \sin\theta^j_{mn} & \cos\theta^j_{mn} \end{bmatrix} \begin{bmatrix} V_1' \\ V_2' \end{bmatrix}, \tag{10}$$

where

$$\theta^j_{mn} = \mathbf{p}_{mn} \cdot (\mathbf{x}_j - \mathbf{x}_1)\,.$$

The characteristic function $c_{mn}(\mathbf{k}_1, \ldots, \mathbf{k}_N)$ can be found by Fourier-transforming (8). The result is

$$c_{mn}(\mathbf{k}_1, \ldots, \mathbf{k}_N) = J_0\left(\left|\sum_{j=1}^{N} \mathbf{k}_j \cdot \mathbf{R}^j_{mn}\right| D_{mn}\right)$$

$$= J_0\left[D_{mn}\left\{\sum_{j,\ell=1}^{N} \mathbf{k}_j \cdot \mathbf{k}_\ell \cos[\mathbf{p}_{mn} \cdot (\mathbf{x}_j - \mathbf{x}_\ell)]\right\}^{1/2}\right]. \tag{11}$$

The characteristic function for $\mathbf{A}(\mathbf{x})$ could now be determined from (11) and (7). However the quantities that we are really interested in are the N-point correlation functions for $h(\mathbf{x}) = \mathrm{Re}\,\mathbf{A}(\mathbf{x}) = A_1(\mathbf{x})$. The density function for $A_1(\mathbf{x}_1), \ldots, A_1(\mathbf{x}_N)$ follows from that of $\mathbf{A}(\mathbf{x}_1), \ldots, A(\mathbf{x}_N)$ by integrating over $A_2(\mathbf{x}_1), \ldots, A_2(\mathbf{x}_N)$. Thus the characteristic function for $h(\mathbf{x}_1), \ldots, h(\mathbf{x}_N)$ follows from the characteristic function for $\mathbf{A}(\mathbf{x}_1), \ldots, \mathbf{A}(\mathbf{x}_N)$ by setting $\mathbf{k}_1 \cdot \hat{i}_2, \ldots, \mathbf{k}_N \cdot \hat{i}_2$ equal to 0. Thus

$$\ln C_{N_x N_y}(k_1, \ldots, k_N) =$$

$$\sum_{n=-N_y}^{N_y} \sum_{m=-N_x}^{N_x} \ln J_0 \left[D_{mn} \left\{ \sum_{j,\ell=1}^{N} k_j k_\ell \cos\left[\mathbf{p}_{mn} \cdot (\mathbf{x}_j - \mathbf{x}_\ell)\right] \right\}^{1/2} \right], \quad (12)$$

where

$$C_{N_x N_y}(k_1, \ldots, k_N) = \left\langle \exp\left[i \sum_{j=1}^{N} k_j h(\mathbf{x}_j) \right] \right\rangle .$$

Now D_{mn} is a small quantity, vanishing as $A_T^{-1/2}$ for large A_T. Since we are interested in A_T large compared to the correlation area, (12) will be expanded in a power series using

$$\ln J_0(z) \approx -\frac{1}{4} z^2 - \frac{1}{64} z^4 . \quad (13)$$

Combining (12) and (13) gives

$$\ln C_{N_x N_y}(k_1, \ldots, k_N) \approx -\frac{1}{2} \sum_{j,\ell=1}^{N} k_j k_\ell \Gamma(\mathbf{x}_j - \mathbf{x}_\ell)$$

$$-\frac{1}{16 A_T} \sum_{j,\ell=1}^{N} \sum_{j',\ell'=1}^{N} k_j k_\ell k_{j'} k_{\ell'} \Gamma_2(\mathbf{x}_j - \mathbf{x}_\ell, \mathbf{x}_{j'} - \mathbf{x}_{\ell'}) , \quad (14)$$

where

$$\Gamma(\mathbf{x}) = (2\pi)^{-2} \int \cos(\mathbf{p} \cdot \mathbf{x})\, S(\mathbf{p})\, d^2p \quad (15)$$

and

$$\Gamma_2(\mathbf{x}_1, \mathbf{x}_2) = (2\pi)^{-2} \int \cos(\mathbf{p} \cdot \mathbf{x}_1) \cos(\mathbf{p} \cdot \mathbf{x}_2)\, S^2(\mathbf{p})\, d^2p . \quad (16)$$

If the second term in (14) is neglected, then the generating function for a multivariate Gaussian distribution is obtained. Thus the second term in (14) represents the deviations from multivariate Gaussian statistics due to the finite size of A_T.

It is desirable to have an estimate of the error introduced in the correlation functions by the second term in (14). This will be done by finding the error in $\Gamma(\mathbf{x}_1, \ldots, \mathbf{x}_{2m})$

when all the points are collapsed to the same point. The measure of error will be taken to be

$$\epsilon_{2m} = \left| \frac{\langle h^{2m} \rangle - \langle h^{2m} \rangle_{\text{Gaussian}}}{\langle h^{2m} \rangle_{\text{Gaussian}}} \right| , \tag{17}$$

where $\langle\ \rangle$ means average over the distribution corresponding to (14) and $\langle\ \rangle_{\text{Gaussian}}$ means average over the distribution corresponding to the first term in (14). These quantities are most easily computed using the relationship

$$\langle e^{ikh} \rangle \approx \exp\left[-\frac{1}{2} k^2 \Gamma(0) - \frac{1}{16 A_T} k^4 \Gamma_2(0, 0) \right] . \tag{18}$$

Expanding both sides of (18) and equating powers of k gives, to $O(A_T^{-1})$,

$$\langle h^{2m} \rangle \approx \frac{(2m)!}{m!} \left[\frac{\Gamma(0)}{2} \right]^m \left\{ 1 - \frac{m(m-1)}{4} \frac{\Gamma_2(0, 0)}{\Gamma^2(0) A_T} + O(A_T^{-2}) \right\} .$$

Thus to this order

$$\epsilon_{2m} \approx \frac{m(m-1)}{4} \frac{\Gamma_2(0, 0)}{\Gamma^2(0) A_T} . \tag{19}$$

A question of more interest is: How many points can the correlation function have before the error measured by ϵ_{2m} obliterates its significance? This will be estimated by setting $\epsilon_{2m_{\max}}$ equal to 1. Thus

$$m_{\max}(m_{\max} - 1) \approx \frac{\Gamma^2(0) A_T}{\Gamma_2(0, 0)} . \tag{20}$$

As an example, (20) will be evaluated for a Gaussian correlation function

$$\Gamma(x) = \Gamma_0 e^{-x^2/L_c^2} .$$

From this and Eqs. (15) and (16) it follows that

$$\Gamma(0) = \Gamma_0$$

and

$$\Gamma_2(0, 0) = \frac{\pi}{2} \Gamma_0^2 L_C^2 .$$

Thus for this case

$$m_{\max}(m_{\max} - 1) \approx \frac{8}{\pi} \frac{A_T}{L_C^2} .$$

Specifically for $L_x = L_y = 10L_c$

$$2m_{\max} \approx 30 .$$

That is, the highest order correlation function that is significant is the 30-point function.

Actually these deviations from a Gaussian process are relatively small compared to the statistical fluctuations of the ensemble members. These fluctuations will be discussed next.

Statistical Fluctuations

The ensemble just discussed has the desired statistical properties. However each member of the ensemble has statistical properties which fluctuate about those of the whole ensemble. It is necessary to construct a measure of these fluctuations which is relatively easy to calculate and interpret.

For each surface of the ensemble we define a correlation function in the usual way:

$$\Gamma_s(\mathbf{x}_1, \ldots, \mathbf{x}_{2n}) = A_T^{-1} \int_{A_T} \prod_{i=1}^{2n} h(\mathbf{x}_i + \mathbf{x}')\, d^2x' . \tag{14}$$

Also one can define an ensemble correlation function

$$\Gamma(\mathbf{x}_1, \ldots, \mathbf{x}_{2n}) = \left\langle \prod_{i=1}^{2n} h(\mathbf{x}_i) \right\rangle . \tag{15}$$

Ensemble-averaging (14) and using the translational invariance of the ensemble gives

$$\left\langle \Gamma_s(\mathbf{x}_1, \ldots, \mathbf{x}_{2n}) \right\rangle = \Gamma(\mathbf{x}_1, \ldots, \mathbf{x}_{2n}) .$$

That is, while the correlation function Γ_s associated with one surface in the ensemble does not exactly correspond to the ensemble correlation function, the ensemble average of Γ_s does. Since only one surface is dealt with at a time in an experiment, it is useful to estimate the deviation inherent in a given surface. One way of doing this is to calculate the mean-squared deviation of $\Gamma_s(\mathbf{x}_1, \ldots, \mathbf{x}_{2n})$ from $\Gamma_1, \ldots, \mathbf{x}_{2n})$. This is difficult to calculate in general. However, if the variables $\mathbf{x}_1, \ldots, \mathbf{x}_{2n}$ are equated, then the problem considerably simplifies. For this estimate a multivariate Gaussian ensemble will be assumed. The statistical deviation will be defined as

$$\Delta_{2n} = \left. \frac{\left\langle [\Gamma_s(\mathbf{x}_i, \ldots, \mathbf{x}_{2n}) - \Gamma(\mathbf{x}_1, \ldots, \mathbf{x}_{2n})]^2 \right\rangle}{\Gamma(\mathbf{x}_1, \ldots, \mathbf{x}_{2n})^2} \right|_{\mathbf{x}_i = \mathbf{x}\ (i=1, \ldots, 2n)}$$

or

$$\Delta_{2n} = \frac{\int \left\langle h^{2n}(\mathbf{x}+\mathbf{x}')\, h^{2n}(\mathbf{x}+\mathbf{x}'')\right\rangle d^2_{x'} d^2_{x''}}{A_T^{\,2}\left\langle h^{2n}(\mathbf{x})\right\rangle^2} - 1 . \tag{16}$$

Both of the moments of $h(\mathbf{x})$ are most directly determined from the relation involving the assumption of Gaussian statistics, namely,

$$\left\langle e^{i[k_1 h(\mathbf{x}_1)+k_2 h(\mathbf{x}_2)]}\right\rangle = e^{(-\Gamma_0/2)(k_1{}^2+k_2{}^2)-\Gamma(\mathbf{x}_1-\mathbf{x}_2)k_1 k_2},$$

by expanding in powers of k_1 and k_2. This gives

$$\left\langle h^{2n}(\mathbf{x}_1)\, h^{2n}(\mathbf{x}_2)\right\rangle = [(2n)!]^2 \left(\frac{\Gamma_0}{2}\right)^{2n} \sum_{\ell=0}^{n} \left(\frac{2\Gamma_{12}}{\Gamma_0}\right)^{2\ell} \frac{1}{[(n-\ell)!]^2(2\ell)!} \tag{17}$$

and

$$\left\langle h^{2n}(\mathbf{x})\right\rangle = \frac{(2n)!}{n!}\left(\frac{\Gamma_0}{2}\right)^n \tag{18}$$

Thus the error measure is

$$\Delta_{2n} = (n!)^2 \sum_{\ell=0}^{n} \frac{2^{2\ell}}{[(n-\ell)!]^2(2\ell)!} A_T^{-1} \int_{A_T} \left[\frac{\Gamma(\mathbf{x})}{\Gamma_0}\right]^{2\ell} d^2x - 1 . \tag{19}$$

The sum will be approximated as follows. The summand as a function of ℓ is strongly peaked at some value $\ell_{\max}$. The location of this peak can be found by neglecting the ℓ dependence of

$$\int_{A_T} [\Gamma(\mathbf{x})/\Gamma_0]^{2\ell}\, d^2x ,$$

since it is a slowly varying function of ℓ by comparison. Use of Stirling's approximation gives for $\ell_{\max}$

$$2\ell_{\max} \approx n + O(1) . \tag{20}$$

Thus the error measure is approximated by

$$\Delta_{2n} \approx A_T^{-1}(n!)^2 \int_{A_T} \left[\frac{\Gamma(\mathbf{x})}{\Gamma_0}\right]^n d^2x \sum_{\ell=0}^{n} \frac{2^{2\ell}}{[(n-\ell)!]^2(2\ell)!} - 1 . \tag{21}$$

The sum can be evaluated, and the result is

$$\sum_{\ell=0}^{n} \frac{2^{2\ell}}{[(n-\ell)!]^2(2\ell)!} = \frac{(4n)!}{[(2n)!]^3} .$$

Combining this with (21) and using Stirling's approximation gives

$$\Delta_{2n} \approx 2^{2n-(1/2)} A_T^{-1} \int_{A_T} \left[\frac{\Gamma(\mathbf{x})}{\Gamma_0}\right]^{2n} d^2x - 1 . \tag{22}$$

This is the desired relation. For a Gaussian correlation function, (22) can be evaluated. Let

$$\Gamma(\mathbf{x}) = \Gamma_0 e^{-\mathbf{x}^2/L_c^2} .$$

Then (22) gives

$$\Delta_{2n} \approx 2^{2n-(1/2)} \frac{\pi}{n} \frac{L_c^2}{L_x L_y} - 1 . \tag{23}$$

It should be kept in mind that (23) is valid only for large $2n$.

Equation (22) or (23) can be used to estimate the highest order correlation function that is significant. This is done by setting $\Delta_{2n_{\max}} \approx 1$ (100% error). This gives for a Gaussian correlation function

$$\frac{2^{2n_{\max}}}{n_{\max}} \frac{L_c^2}{L_x L_y} \approx 1 \tag{24}$$

and for $L_c^2/L_x L_y = 1/100$

$$2n_{\max} = 9 .$$

Thus for this particular choice of parameter the nine-point correlation is the highest order that has any significance. Since Δ_{2n} increases rapidly as n increases, the value of $2n$ that can be considered reasonably accurate is the next lowest, or 8 for this case.

An indirect check on this estimate was made using a numerically computed surface. The argument used was as follows: Essentially $2n_{\max}$ is a measure of the highest order moment of $p(h)$ that is significant. The main way that deviations from a Gaussian distribution manifest themselves for a finite surface is in the finiteness of the range of heights. As the order of the moment increases, its error increases, since it is more sensitive to the tails of $p(h)$. A way of estimating the maximum usable moment is as follows. The $2n$th moment of h is ideally

$$\langle h^{2n} \rangle = (2\pi\Gamma_0)^{-1/2} \int_{-\infty}^{\infty} h^{2n} e^{-h^2/2\Gamma_0} \, dh .$$

The integrand peaks strongly at

$$h^2 = 2n\Gamma_0 .$$

Thus an estimate of $2n_{max}$ can be obtained by setting

$$2n_{max} = \frac{h^2_{max}}{\Gamma_0} .$$

The surface numerically constructed by the methods discussed in the next section has a Gaussian correlation function with $L_x = L_y = 10L_c = 10$ and $\Gamma_0 = L_c^2$. It was found that for this case $h_{max} \approx 3$. This gives

$$2n_{max} \approx 9 ,$$

which is in good agreement with our previous estimate.

One additional question can be asked about the method of error estimation used to derive (23) and (24): What happens to the error when the points $\mathbf{x}_1, \ldots, \mathbf{x}_{2n}$ are not taken to be equal? This question is difficult to answer exactly, but an indication of the probable answer can be found from an examination of the fluctuations in the two-point correlation function. In this case, if $\Gamma(\mathbf{x})$ is monotonically decreasing, then the fractional error increases as $|\mathbf{x}|$ increases. This is due to the fact that there is a constant mean-squared noise in $\Gamma(\mathbf{x})$, and since $\Gamma(\mathbf{x})$ decreases as $\mathbf{x}$ increases, the fractional error will increase. It is reasonable to expect that this behavior is also present in the higher order correlation functions.

Conditional Height Distribution

In the next section, where the numerical procedures used to generate and machine an actual surface are discussed, an approximation is made concerning the behavior of the surface near an extremum. To justify this approximation and to gain some insight into the nature of the surfaces generated, we will examine the conditional height distribution. The distribution of interest is the probability distribution of heights h_1 at some distance $\mathbf{x}_{12}$ from an extremum of the surface, given that the height at the extremum is h_2. Starting from the multivariate Gaussian distribution $p(h_1, h_2, \nabla h_3)$, it follows that

$$p(h_1|h_2, \nabla h_2 = 0) = \left\{\frac{\Gamma_0}{2\pi\left[\Gamma_0^2 - \Gamma(\mathbf{x}_{12})^2\right]}\right\}^{1/2} \exp\left\{-\frac{\Gamma_0}{2}\frac{\left[h_1 - \frac{\Gamma(\mathbf{x}_{12})}{\Gamma_0} h_2\right]^2}{\Gamma_0^2 - \Gamma(\mathbf{x}_{12})^2}\right\},$$

where $\Gamma_0 = \Gamma(\mathbf{x} = 0)$, and $\mathbf{x}_3 = \mathbf{x}_2$.

The chief feature revealed here is that near an extremum the surface has on the average the shape of $\Gamma(\mathbf{x})$. Since the width of the distribution is very narrow when $|\mathbf{x}_{12}|$ is smaller than a correlation length and the fractional width decreases as h_2 increases, the highest peaks near their maximum will closely resemble the correlation function.

This also provides an interesting insight into the appearance of the surface given the correlation function of the surface. In the next section we assume that the point of minimum radius of curvature of the surface coincides with the point of maximum excursion from zero. This will be a good approximation as long as the minimum radius of curvature and maximum value of the correlation function coincide at $\mathbf{x} = 0$. This is true for a Gaussian correlation function and will be true for most correlation functions of interest.

THE NUMERICAL ALGORITHM

Description of Implementation

The computer program package RANMILL was designed to implement the methods described in the preceding section and to carry out the numerically controlled milling of the surface. The RANMILL package consists of four computer programs written for a Control Data Corporation 3800 series computer. An outline of RANMILL is given in Appendix A, and a listing is given in Appendix B.

RANMILL calculates a series of random surfaces using the method described in the preceding section and chooses the optimum surface on the basis of the agreement of the theoretical and calculated height distributions. It then does a detailed statistical analysis of the optimum surface, comparing its statistical properties to those prescribed. The user can further employ the package to determine optimum tool travel during milling and, finally, to produce the input for a Bunker Ramo 3300 Incremental Continuous Path Numerical Control System (3).

Calculation of the Surface

As shown by the section containing Eqs. (1) through (3), a random surface with specified statistics can be generated from the equation

$$h(x, y) = \frac{1}{L_x L_y} \operatorname{Re}\left\{ \sum_{m=-N_x}^{N_x} \sum_{n=-N_y}^{N_y} A(m, n) \exp\left[2\pi i \left(\frac{mx}{L_x} + \frac{ny}{L_y}\right) + i\phi(m, n)\right] \right\},$$

where $A(m, n) = \sqrt{2L_x L_y S(m, n)}$ and where

N_x and N_y are the number of sampling points along the x and y axes respectively,

L_x is the length along the x axis,

L_y is the length along the y axis,

S is the power spectrum, the Fourier transform of which is the autocorrelation function of $h(x, y)$,

$\phi(m, n)$ is a set of $N_x N_y$ random phases uniformly distributed on the interval $[0, 2\pi]$ and shifted such that $\phi(0, 0) = \pi/2$, and

$A(0, 0) \cos \phi(0, 0)$ equals the mean height of the surface and is equal to 0.

This equation lends itself particularly well to computer implementation, since the double sum can be evaluated by means of the "fast Fourier transform" algorithm of Cooley and Tukey (4).

The first program in the RANMILL package, SURFACE, calculates a surface and examines its statistics. Use is made of the fast-Fourier-transform program FASTFOUR (5). This subroutine calculates the Fourier transform at equally spaced sample points, the number of which, due to storage limitations, cannot exceed 4096. Between sample points the function is evaluated in RANMILL by a bicubic interpolation algorithm which gives a continuous gradient (6). The spectral function, $S(m, n)$, is evaluated in a subroutine supplied by the user. Using FASTFOUR to obtain the Fourier transform of the spectral function, SURFACE first calculates the theoretical autocorrelation function $\Gamma(x, y)$ and thus the mean squared height of the surface $\Gamma(0, 0)$. From this information the theoretical height distribution, which is Gaussian, is easily calculable.

As shown in the preceding section, if one considers the ensemble of surfaces generated, the statistical properties of $h(x, y)$ will be the desired ones. However, since we are constructing only a single representation of the surface, its statistics are subject to statistical variations around those of the entire ensemble. To ensure the generation of a surface which is approximately ergodic, it is necessary to calculate a series of surfaces, retaining only the one which "best" represents the statistics of the ensemble. The "best" surface is chosen on the basis of the minimum mean-squared deviation from the theoretical height distribution.

A more thorough statistical analysis of the surface is then performed. The joint probability distribution of heights and the two- and three-point autocorrelation functions are directly calculated at selected points and compared to the theoretical values. For this comparison the theoretical values are based on the calculated, rather than the expected, mean-squared height. The theoretical height distribution is recalculated based on the actual mean-squared height and once again compared to the height distribution of the surface.

Finally SURFACE calculates the partial derivatives of the surface, and the three matrices $h(x, y)$, $\partial h(x, y)/\partial x$, and $\partial h(x, y)/\partial y$ evaluated at the sample points inherent in FASTFOUR are stored on magnetic tape.

The chief problem encountered in the implementation of the program was the generation of the random set $\phi(m, n)$. Equation (1) is particularly sensitive to correlations in $\phi(m, n)$. These correlations result in defects in the statistical properties of the surface. In one instance the use of an inadequate random sequence was observed to generate a surface with a height distribution which deviated significantly from the expected Gaussian.

Subroutines that generate so-called pseudorandom number sequences are available in most program libraries. In these the numbers are generated sequentially, each number being uniquely determined by its predecessor. Such number generators have the advantage that they can generate the elements of a sequence one at a time without requiring storage for the entire array. However, to a greater or lesser extent, depending on the method, sequences generated in this way show correlations and were found inadequate for use in this study.

A much better set of random numbers can be generated if the sequential requirement is relaxed and the random set is generated as an entity. We found that the nonsequential

random set generated by repeated shuffling of the set of numbers produced by a sequential generator was satisfactory. In the random-number generator presently used in SURFACE, a sequence is first generated using a multiplicative congruential pseudorandom number generator from the program library. Several passes are then made through the sequence in which elements are interchanged with others at random addresses within the array. The random addresses are generated by the same multiplicative generator that produced the original sequence. The surface which is generated by the program can be changed either by specifying a different initial number for the random-number generator or by simply shuffling more or less often.

A test was performed on the random number set generated in this way. It consisted of using the program SURFACE to generate a surface corresponding to a flat surface spectrum using a 63-by-63-point matrix. The resulting $h(x, y)$ matrix should have entries that are Gaussian distributed. It should also have a two-point autocorrelation matrix which is a Kronecker delta and a three-point autocorrelation matrix which is zero. These matrices were calculated numerically and were found to be correct within the expected uncertainty (due to counting errors inherent in the use of a finite sample).

Additional Statistical Comparisons

Using the surface matrix and the partial derivative matrices as written on magnetic tape by SURFACE, program PARSTAT calculates the probability distributions of the partial derivatives and of the gradient and compares them to the theoretical distributions. The maximum gradient of the surface and its location are also printed out. Finally PARSTAT estimates the maximum curvature of the surface. As stated in the last paragraph of the preceding main section, we assume that it occurs at either the maximum or the minimum of $h(x, y)$. The curvature is then calculated by fitting a quadratic form in that region.

Tool Travel During Milling

The path of the tool which mills the surface is determined by program POINTS 8. Like PARSTAT, this program obtains the surface matrix and the partial derivative matrices from magnetic tape. At this point the matrices may be rescaled using input scale factors.

POINTS 8 contains an interpolation subroutine, INTERP, which provides values of the surface and its derivatives between matrix points. This subroutine uses the bicubic interpolation algorithm of Coons (6), which produces a function with continuous first derivatives.

For milling, the surface is defined as a set of equally spaced parallel cuts $h(x,$ constant). The cuts consist of linear chords defined by points on the surface. The spacing between cuts and the definition of the chords depends on the desired accuracy to which the surface is to be milled. For the calculation of a chord representation, three tolerances are supplied by the user. One of these, ϵ_h, specifies allowable error in the height of the milled surface. The other two, T_x and T_y, are tolerances on the slope. These tolerances are used in calculating both the cut spacing and the chord lengths.

One milling error which depends on the cut spacing is introduced by the hemispherical shape of the milling tool. The use of such a mill traveling along parallel cuts produces

a scalloping effect. By approximating the surface at the point in question by a portion of a sphere, the error introduced by this effect can be shown to be well approximated by

$$\delta = \frac{1}{8}(\Delta y)^2(1 + h_y{}^2)\left(\frac{1}{R} + \frac{1}{R_{mf}}\right),$$

where Δy is the spacing between adjacent cuts, h_y is $\partial h(x, y)/\partial y$, R is the radius of curvature of the surface, and R_{mf} is the radius of the mill. The radius of the mill must be less than or equal to the smallest radius of curvature of the surface; therefore

$$\delta \leqslant \frac{(\Delta y)^2(1 + h_{y\max}^2)}{4R_{mf}}.$$

Then, if ϵ_h is the allowable error in the surface height,

$$\Delta y = \sqrt{\frac{4\epsilon_h R_{mf}}{1 + h_{y\max}^2}}.$$

It is conservative to replace $h_{y\max}$ by the maximum gradient of the surface S_m. Thus

$$\Delta y = \sqrt{\frac{4\epsilon_h R_{mf}}{1 + S_m{}^2}}.$$

It can be seen from this relation that, to keep the number of cuts to a minimum, R_{mf} should be taken as large as possible without exceeding the minimum radius of curvature of the surface.

The linear chords by which each cut is approximated are also chosen on the basis of ϵ_h, as well as on the basis of the slope tolerances T_x and T_y. Chords are chosen to be of maximum length without violating the following conditions:

1. that they fall entirely within the region $[h(x, y) - \epsilon_h, h(x, y) + \epsilon_h]$,
2. that their slopes be within T_x of $\partial h(x, y)/\partial x$, and
3. that the change in $\partial h(x, y)/\partial y$ along the length of an individual chord be less than T_y.

Given a fixed starting point $[x_0, y_0, h(x_0, y_0)]$, chords of increasing length to the points $[x_0 + \Delta x, y_0, h(x_0 + \Delta x, y_0)]$, $[x_0 + 2\Delta x, y_0, h(x_0 + 2\Delta x, y_0)]$, etc. are tested against the three limiting conditions until one fails. If the first chord which is tested fails to meet one of the conditions, the chord length is repeatedly decreased by halving the x coordinate of the endpoint until all conditions are met.

A constant value of Δx is used throughout the surface. Its length is chosen such that the first chord cannot fail to meet the first condition. If the surface is approximated by a parabola, the maximum deviation of the chord from the surface is

$$\delta = \max[\,|h(x) - c(x)|\,] \approx \frac{1}{8}(\Delta x)^2 \frac{\partial^2 h(x)}{\partial x^2} ,$$

where Δx is the change in the x coordinate over the length of the chord. Thus it is conservative to set

$$\Delta x = \sqrt{\frac{8\epsilon_h}{h_{xx\max}}}$$

where

$$h_{xx\max} = \frac{\partial^2 h(x)}{\partial x^2} .$$

The first trial chord automatically satisfies condition 1 and is tested against only conditions 2 and 3. If it fails, increasingly shorter chords are tested until one does not fail. Clearly these shorter chords need not be tested against condition 1.

If the first trial chord meets conditions 2 and 3, longer chords are tried. They are tested against all three conditions. The following method is used to test whether they meet condition 1: As each chord is tested, the slopes of the corresponding chords to the top and bottom of the error band are calculated:

$$S_{n+} = \frac{\Delta h_n + \epsilon_h}{n\Delta x} , \qquad n = 1, 2, \ldots, N ,$$

$$S_{n-} = \frac{\Delta h_n - \epsilon_h}{n\Delta x} , \qquad n = 1, 2, \ldots, N .$$

As long as the slope of the chord falls between $\min(S_{n+})$ and $\max(S_{n-})$, the chord lies within the error band.

As mentioned before, a hemispherical tool is used for milling the surface. Ideally the mill should travel tangent to the chord representation of the surface, its contact point being the lines of the chords. However the mill travels in a straight line from one chord endpoint to the other, so that if $\partial h(x, y)/\partial y$ changes along a chord, and the surface in effect rotates around the mill, the contact point of the mill with the surface is not exactly a straight line. The error introduced by this effect can be reduced to any desired value by decreasing the error criterion on the y partial, T_y.

The milling tool is offset in such a way that at each point it is approximately tangent to the surface. If the chords from a given cut are considered to lie in the xz plane, then Fig. 1 illustrates the case in which

$$\left.\frac{\partial h(x, y)}{\partial y}\right|_{x_0, y_0} = 0 .$$

$\mathbf{R}_1$ and $\mathbf{R}_2$ are radius vectors of the milling tool from the point of intersection of the chords. Since the milling tool is to be tangent to the chord it is cutting, $\mathbf{R}_i$ is perpendicular to the ith chord. $\mathbf{D}$ is the vector from the point of intersection of the chords to that of the paths of the mill center.

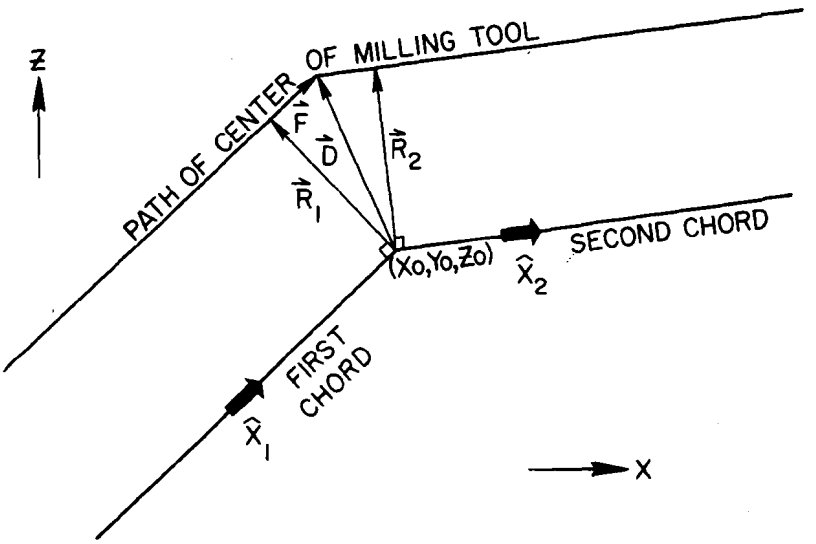

Fig. 1—The paths of the mill center are shown for the case in which the y partial derivative is zero. For this case the mill center travels in the plane of the chords. The machine tool axis is parallel to the z axis.

For the general case, where $\partial h(x, y)/\partial y$ is nonzero, $\mathbf{R}_1$ and $\mathbf{R}_2$ are rotated out of the xz plane so as to be approximately normal to $h(x, y)$ at (x_0, y_0). This causes $\mathbf{D}$ also to rotate out of the plane; however, as will be shown, the rotation is done such that $\mathbf{F}$, which is given by $\mathbf{F} = \mathbf{D} - R_1$, remains parallel to, but not in, the xz plane.

Consider two chords x_1 and x_2 with slopes S_1 and S_2 respectively which intersect at the point (x_0, y_0, z_0). The unit vectors along the chords are

$$\hat{\mathbf{x}}_1 = d_1(\hat{\mathbf{i}} + S_1\hat{\mathbf{k}}) ,$$

$$\hat{\mathbf{x}}_2 = d_2(\hat{\mathbf{i}} + S_2\hat{\mathbf{k}}) ,$$

where $d_1 \equiv \sqrt{1/(1 + S_1^2)}$ and $d_2 \equiv \sqrt{1/(1 + S_2^2)}$. If

$$z_x \equiv \left.\frac{\partial h(x, y)}{\partial x}\right|_{x_0, y_0} ,$$

$$z_y \equiv \left.\frac{\partial h(x, y)}{\partial y}\right|_{x_0, y_0} ,$$

then the plane tangent to the surface at (x_0, y_0, z_0) is given by

$$z_x x + z_y y = z$$

and a vector normal to the tangent plane is

$$\mathbf{N} = z_x\hat{\mathbf{i}} + z_y\hat{\mathbf{j}} - \hat{\mathbf{k}} .$$

Since $|S_i - z_x| < T_x$, z_x can be approximated by the slope of $(\hat{x}_1 + \hat{x}_2)$, so that

$$\mathbf{N} \approx \frac{d_1 s_1 + d_2 s_2}{d_1 + d_2}\,\hat{i} + z_y \hat{j} - \hat{k}\,.$$

Then a vector which lies in the tangent plane and is perpendicular to $\hat{x}_1 + \hat{x}_2$ is defined by

$$\mathbf{V} = \mathbf{N} \times (\hat{x}_1 + \hat{x}_2)\,,$$

and

$$\mathbf{R}_1 = \frac{(\mathbf{V} \times \hat{x}_1)}{|\mathbf{V} \times \hat{x}_1|}\,R_{\text{mill}}\,,$$

$$\mathbf{R}_2 = \frac{(\mathbf{V} \times \hat{x}_2)}{|\mathbf{V} \times \hat{x}_2|}\,R_{\text{mill}}\,.$$

By showing that the y component of $\mathbf{R}_1$ equals that of $\mathbf{R}_2$, one can show that $\Delta\mathbf{R}$ is parallel to the plane of the chords. Projecting $\Delta\mathbf{R}$ onto the plane of the chords results in Fig. 2. $\mathbf{R}_1'$ and $\mathbf{R}_2'$ are the projections of $\mathbf{R}_1$ and $\mathbf{R}_2$. It can be seen from Fig. 2 that $\mathbf{F}$ has magnitude

$$|\mathbf{F}| = \frac{0.5\,|\Delta\mathbf{R}|}{\sin(\theta/2)}$$

and direction $\hat{x}_1$. Thus

$$\mathbf{F} = \pm\hat{x}_1\,\frac{0.5\,|\Delta\mathbf{R}|}{\sin(\theta/2)}\,,$$

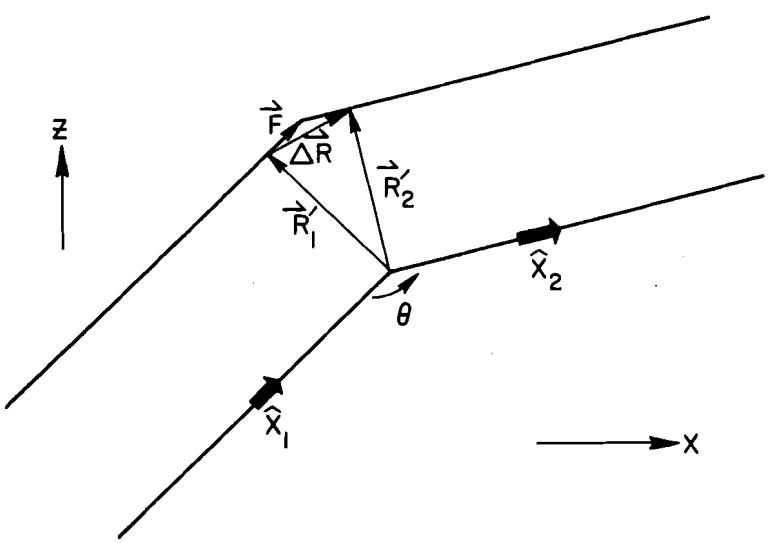

Fig. 2—The projection of the mill center path onto the plane of the chords is shown for the general case. Since the y partial derivative is nonzero, the mill center travels not in the plane of the chords but in another, parallel plane. The machine tool axis is parallel to the z axis.

where the positive sign is valid for $S_1 > S_2$ and the negative sign is used for $S_1 < S_2$. For $S_1 = S_2$, $\mathbf{F} = 0$. Finally the position of the mill center relative to (x_0, y_0, z_0) is given by

$$\mathbf{D} = \mathbf{R}_1 + \mathbf{F} .$$

It is assumed in the above derivation that the mill center paths intersect. Although, strictly speaking, this is true only if $\partial h(x, y)/\partial y$ does not change over the length of the chord, this assumption is forced to be valid to any desired accuracy by the introduction of condition 3 in the chord search, namely that the change in the y partial derivative along a chord be less than a tolerance input by the user.

To increase the similarity of the chord structure of adjacent cuts, the chord search always proceeds in the same direction, that is, in the direction of increasing x. However, to save machining time, the order of points in every other cut is reversed after it has been defined so that the milling is back and forth.

To reduce the amount of material which the finishing mill must remove, the surface is first roughed out using a cylindrical (flat-end) mill. For roughing, a rough chord representation of the surface should be used; that is, the input error tolerances should be large. The flat-end mill travels a constant distance R_z above the chord to ensure that it does not gouge the surface:

$$R_z = 1.1\ \epsilon_r + R_{mr} S_m ,$$

where R_{mr} is the radius of the mill used for roughing, ϵ_r is the error criterion used in the chord search for roughing, and S_m is the maximum gradient of the surface. The spacing of adjacent roughing cuts is 90 percent of the diameter of the roughing mill.

To decrease machining time during roughing, a graduated feed rate is used. Deeper chords are machined more slowly than those near the surface of the material. The maximum and the minimum feed rate, $F_{\max}$ and $F_{\min}$, are input quantities. The feed rate at which the nth chord is machined is given by

$$F_n = \frac{F_{\max} A}{A + \max(d_n, 0)} ,$$

where $A = F_{\min} d_{\max}/(F_{\max} - F_{\min})$, $d_{\max}$ is the maximum depth attained by the rough surface, and d_n is the maximum depth of the nth chord. In the finishing of the surface, when the amount of material being removed with each cut is smaller and more nearly constant, a constant feed rate is used.

Results

Figures 3 through 5 describe a surface which was generated and milled using RANMILL. Figure 3 is a photograph of a surface which was generated using a Gaussian spectral function

$$S(m, n) = \Gamma_0 \pi^2 L_c^2 \exp\left[-2\pi L_c^2 \left(\frac{m^2}{N_x^2} + \frac{n^2}{N_y^2}\right)\right] .$$

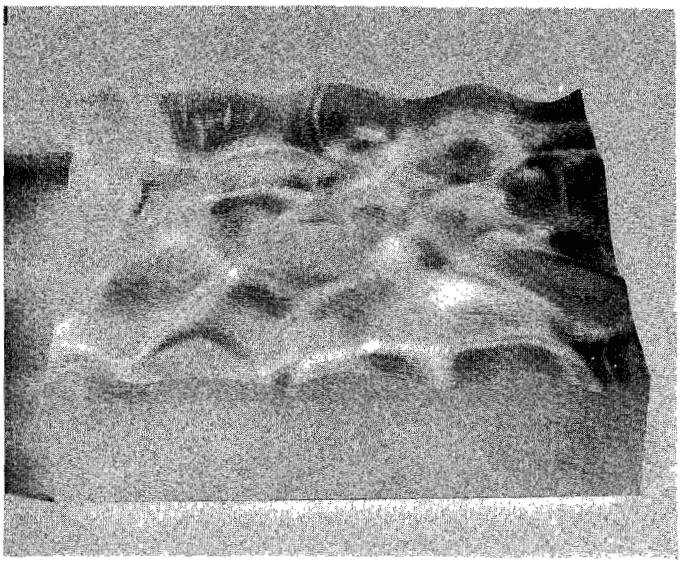

Fig. 3—Sample surface. The sample shown here is 10 by 10 inches square and has an autocorrelation function $\Gamma(x) = (1/36)e^{-x^6}$. This surface was designed using RANMILL and milled out of Lucite.

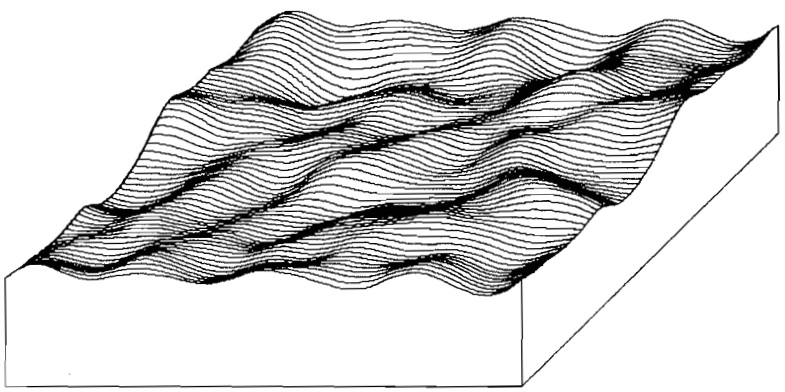

Fig. 4—An isometric computer plot of the surface shown in Fig. 3

Figure 4 is an isometric plot of the same surface. This spectrum results in a surface with a Gaussian autocorrelation function, with a correlation length L_c and mean squared height Γ_0. L_c was taken equal to 1 inch, and the mean squared height was set to 1/36 inch. The surface is 10 by 10 inches ($L_x = L_y = 10$ inches), with 64 sampling points on a side ($N_x = N_y = 64$), and was milled out of Lucite.

The actual and expected height distributions of this surface are shown in Fig. 5, along with 85-percent confidence limits. Table 1 contains the expected two-point

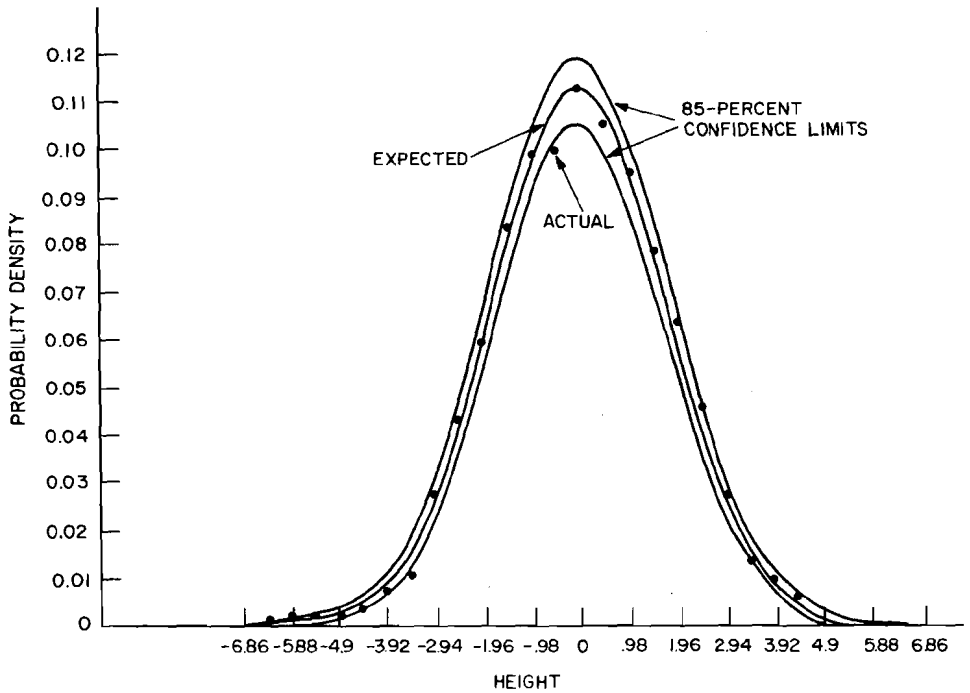

Fig. 5—The actual and expected height distribution of the numerically computed sample surface are shown. Based on Poisson counting statistics, 85 percent of the points of the calculated histogram should fall within the indicated error bands.

autocorrelation function of the surface and the deviations from it of calculated values, and Table 2 is the three-point autocorrelation function, which should be 0 everywhere. The milling tolerances used for this surface were ϵ_h = 0.004 inch, T_x = 0.06 inch, and T_y = 0.076 inch.

CONCLUDING REMARKS

A procedure has been developed for constructing random surfaces whose statistics approximate those of a multivariate Gaussian surface. Estimates of the deviations of a representative surface from the ideal have been discussed in detail. These estimates are very useful if the surface is to be used in an experiment which is to be compared closely to theory; otherwise the question of whether the theory or experiment is in error will always linger.

The numerical method used is based on a fast-Fourier-transform algorithm and requires as an input some parameters that govern the resulting surface size and the directional power spectrum of the surface (in the form of a subroutine). The resulting surface is periodically continuable, so that a larger surface can be constructed by combining replicas as if laying tiles, and has been selected so as to best represent the properties of the ensemble. The program package also produces a detailed analysis of the statistical properties of the surface and a paper tape for machining the surface on a numerically controlled milling machine. The surface generation is quite efficient, so that it is practical to select the best surface out of 100 for production.

Table 1

The values of the theoretical two-point autocorrelation function $\Gamma(\mathbf{x}) = e^{-\mathbf{x}^2}$ and the percent deviations displayed by the numerically computed sample autocorrelation function, $\Gamma_s(\mathbf{x}) = (1/A_T) \int h(\mathbf{x}+\mathbf{x}')\, h(\mathbf{x}')\, d^2x'$ shown as a function of x and y. The rms surface height and the correlation length are equal to 1. The surface area A_T is 10×10.

y	$\Gamma(\mathbf{x})$ and Percent Deviation for Various Values of x										
	0.000	0.156	0.313	0.469	0.625	0.781	0.938	1.094	1.250	1.406	1.563
0.000	1.000	0.976	0.907	0.803	0.677	0.543	0.415	0.302	0.210	0.138	0.087
	0.0%	−0.20%	−0.78%	−1.64%	−2.64%	−3.57%	−4.19%	−4.27%	−3.60%	−2.11%	0.16%
0.156	0.976	0.952	0.885	0.783	0.660	0.530	0.405	0.295	0.205	0.135	0.085
	0.09%	0.25%	0.01%	−0.55%	−1.30%	−2.07%	−2.62%	−2.72%	−2.18%	−0.88%	1.18%
0.313	0.907	0.885	0.823	0.728	0.614	0.493	0.377	0.274	0.190	0.126	0.079
	0.34%	0.84%	0.91%	0.62%	0.07%	−0.59%	−1.14%	−1.37%	−1.06%	−0.06%	1.66%
0.469	0.803	0.783	0.728	0.644	0.543	0.436	0.333	0.243	0.168	0.111	0.070
	0.67%	1.46%	1.82%	1.76%	1.36%	0.77%	0.16%	−0.25%	−0.25%	−0.35%	1.64%
0.625	0.677	0.660	0.614	0.543	0.458	0.368	0.281	0.205	0.142	0.094	0.059
	0.94%	2.00%	2.59%	2.72%	2.45%	1.89%	1.20%	0.57%	0.25%	0.41%	1.19%
0.781	0.543	0.530	0.493	0.436	0.368	0.295	0.226	0.164	0.114	0.075	0.047
	1.04%	2.30%	3.08%	3.37%	3.21%	2.67%	1.90%	1.07%	0.42%	0.14%	0.41%
0.938	0.415	0.405	0.377	0.333	0.281	0.226	0.172	0.126	0.087	0.057	0.036
	0.84%	2.27%	3.19%	3.60%	3.54%	3.04%	2.21%	1.23%	0.29%	−0.39%	−0.61%
1.094	0.302	0.295	0.274	0.243	0.205	0.164	0.126	0.091	0.063	0.042	0.026
	0.30%	1.83%	2.85%	3.38%	3.42%	2.98%	2.15%	1.07%	−0.08%	−1.07%	−1.72%
1.250	0.210	0.205	0.190	0.168	0.142	0.114	0.087	0.063	0.044	0.029	0.018
	−0.58%	1.00%	2.11%	2.74%	2.89%	2.55%	1.79%	0.69%	−0.56%	−0.99%	−2.75%
1.406	0.138	0.135	0.126	0.111	0.937	0.075	0.057	0.042	0.029	0.019	0.012
	−1.74%	−0.13%	1.05%	1.79%	2.08%	1.90%	1.27%	0.26%	−0.99%	−2.30%	−3.49%
1.563	0.087	0.085	0.079	0.070	0.059	0.047	0.036	0.026	0.018	0.012	0.008
	−3.03%	−1.40%	−0.15%	0.72%	1.19%	1.22%	0.80%	−0.03%	−1.16%	−2.45%	−3.73%

Table 2

The values of the numerically computed sample three-point autocorrelation function $\Gamma_s(\mathbf{x}, \mathbf{z}) = (1/A_T) \int h(\mathbf{x}')\, h(\mathbf{x}'+\mathbf{x})\, h(\mathbf{x}'+z)\, d^2x'$ are shown for selected values of $\mathbf{x}$ with $\mathbf{z} = 0.047\,(i_x + i_y)$. The rms surface height and the correlation length are equal to 1. The surface area A_T is 10×10. The ideal ensemble value of $\Gamma_s(\mathbf{x}, \mathbf{z})$ is 0.

y	$\Gamma_s(\mathbf{x}, \mathbf{z})$ for Various x				y	$\Gamma_s(\mathbf{x}, \mathbf{z})$ for Various x			
	2.031	4.063	6.094	8.125		2.031	4.063	6.094	8.125
2.031	−0.262	−0.249	0.010	0.137	6.094	0.046	−0.057	−0.037	0.228
4.063	0.146	0.133	0.161	−0.094	8.125	−0.161	0.052	0.127	0.063

Caruthers and Novarini (7) have also developed a procedure for numerically calculating random rough surfaces based on spatial filtering. Our procedure has several advantages, some of which are: careful statistical estimation and testing of fluctuation error, a fast algorithm, and the production of a physical realization of the surface.

ACKNOWLEDGMENTS

We thank Kenneth D. Flowers for his careful reading of our manuscript and helpful comments, Edward L. Wright for his contributions to developing the machine tool control routines, and the personnel of the machine shop, Engineering Services Division, for their cooperation during production of the surfaces. One of us (GGZ) thanks the National Research Council for its financial support.

REFERENCES

1. D. Middleton, *Introduction to Statistical Communication Theory*, McGraw-Hill, New York, 1960.
2. E.L. Wright, "Probability Density Functions of *n* Vectors Added with Random Directions," NRL Report 6922, 1969, and NRL internal memorandum, 1968.
3. *The Bunker Ramo Corporation Specification for 3300 Incremental Continuous Path Numerical Control System*, Mar. 14, 1969.
4. J.W. Cooley and J.W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. Comp. 19, 297 (1965).
5. C.D. Lowenstein, *Fast Fourier Analysis*, University of California, San Diego, June 17, 1966.
6. S.A. Coons, *Surfaces for Computer-Aided Design of Space Figures*, Massachusetts Institute of Technology, Mechanical Engineering Department, Design Division, Cambridge, Jan. 1964.
7. J.C. Novarini and J.W. Caruthers, "Numerical Modeling of Randomly Rough Surfaces with Application to Sea Surfaces," Texas A&M University, Dept. of Oceanography, Ref. No. 71-13-T, 1971.

Appendix A

PROGRAM OUTLINE

I. Program SURFACE

A. Generate a series of surfaces

1. Generate a set of random phases and shift it so that $\phi(0, 0) = \pi/2$

2. Generate the spectral function $A(m, n)$

3. Calculate the expected correlation function

4. Calculate the surface and its height distribution

5. Reshuffle the random phases and repeat until the desired number of surfaces has been generated

B. Print the surface whose height distribution deviates least from the theoretical one and store it on magnetic tape. Calculate and output its partial derivatives and its mean squared height. Search for the maximum and minimum of $h(x, y)$.

C. Check the statistics of the surface using calculated, rather than theoretical, mean-squared height in determining expected distributions.

1. Height distribution

2. Joint probability distribution

3. Autocorrelation function

4. Three-point autocorrelation function

II. Program PARSTAT

A. Calculate probability distribution of x and y partial derivatives and of the gradient and compare to expected distribution

B. Calculate maximum gradient occurring on the surface and its location

C. Estimate the radius of curvature at the surface maximum and at the minimum

III. Program POINTS 8

A. Scale the surface using input scale factors

B. For bivariate surfaces calculate a rough representation of the surface to be used in roughing it out. This step is omitted for surfaces in only one variable.

1. Calculate cut spacing

2. Search out chords

3. Displace mill a constant distance R_z above the surface

4. Calculate feed rate as a function of chord depth

5. Buffer out points onto magnetic tape

C. Define a fine chord representation of the surface to be used for the final milling.

1. Calculate cut spacing

2. Search out chords

3. Calculate mill displacements for the hemispherical mills

4. Use a constant feed rate

5. Buffer out the points onto magnetic tape

IV. Program PAPER

This is a postprocessor which reads the surface as defined by POINTS 8 from magnetic tape, puts it into format required by the 3300 and punches a paper tape.

Appendix B

PROGRAM LISTING

```
      PROGRAM DIMENS
      DIMENSION H(64,64),GM(64,64),CPH(64,64),A(64,64),DAX(64,64),PHH(64
     1,64),DHX(64,64),X(1024),Y(1024),MFF(3),C(64,64),AA(64,64)
      REAL LNTWO
      LOGICAL INTER
      COMPLEX,A,DAX,CPH,C
      EQUIVALENCE(H,CPH,A,DAX),(PHH,DHX),(A(2049),GM)
      LNTWO=LOGF(2.)
      READ 901,SPECT
C  FOR A TWO-DIMENSIONAL SURFACE,USE NFY=0.
20    READ 900,NFX,NFY,IFR
      IF(EOF,60) 50,21
21    CONTINUE
      IF(IFR.NE.0 )GO TO 26
      CALL BACKFILE(1)
      DO 24 LLL=1,1000
      READ(1),SCR
      IF(EOF,1) 26,23
23    CALL SKIPFILE(1)
24    CONTINUE
26    CALL BACKFILE(1)
      NX=2*(NFX+1)  $ NY=2*(NFY+1)
      IF(NFY.EQ.0) NY=1
      IF(NFX.EQ.0) NX=1
      FMX=LOGF(FLOAT(NX))/LNTWO
      FMY=LOGF(FLOAT(NY))/LNTWO
      MFF(1)=FMX
      MFF(2)=FMY
      MFF(3)=0
      IF(ABS(FMX-MFF(1)).LT.1E-10) GO TO 10
      IF(ABS(FMY-MFF(2)).LT.1E-10) GO TO 10
      PRINT 902
      STOP
10    CONTINUE
      IF(MFF(1).GE.MFF(2))GO TO 11
      INTER=.TRUE.
      MSCR=MFF(1)
      MFF(1)=MFF(2)
      MFF(2)=MSCR
11    IF(MFF(1).GT.10) MFF(1)=10
      IF(MFF(1)+MFF(2).GT.12) MFF(2)=12-MFF(1)
      NX=2**MFF(1)
      NY=2**MFF(2)
      CALL SURFACE(H,GM,CPH,A,DAX,PHH,DHX,C,AA,X,Y,MFF,NX,NY,INTER,SPECT
     1)
      GO TO 20
900   FORMAT(3I5)
901   FORMAT(A8)
902   FORMAT(*0INPUT MATRIX DIMENSIONS,CALCULATED AS 2(NFX +1) OR 2(NFY+
     1 1), ARE NOT POWERS OF TWO.*)
50    STOP
      END
```

```
      SUBROUTINE SURFACE(H,GM,CPH,A,DAX,PHH,DHX,C,AA,X,Y,MFF,NX,NY,INTER
     1,SPECT)
      DIMENSION MFF(3),SV1(1024),SV2(1024),PHY(4096),PHI(4096)
      DIMENSION H(NX,NY),GM(64,64),CPH(NX,NY),A(NX,NY),DAX(NX,NY),PHH(NX
     1,NY),DHX(NX,NY),C(NX,NY),AA(NX,NY),X(NX),Y(NY)
      COMMON/4/SKIP,SQERR,NNN
      COMMON/CONSTANT/PI,TWOPI,PISQ,SQR2,LNTWO
      COMMON/SPEC/XLT,YLT
      COMPLEX,A,DAX,CC,C,FACTOR,SCR,CPH
      LOGICAL SKIP
      LOGICAL INTER
      REAL LNTWO
      LNTWO=LOGF(2.)
      SQR2=1.414213562373095
      PI=3.14159265358979 3
      TWOPI= 2.*PI
      PISQ=PI**2
      ISURF=0
5     READ 900,XLT,YLT,FNSHUF,DELP,NNN,NSURF
      IF(.NOT.INTER) GO TO 6
      SCR=XLT
      XLT=YLT
      YLT=SCR
6     CONTINUE
      IF(NY.EQ.1) YLT=1.
      CONST=XLT*YLT
      NSHUF=FNSHUF
      FNSHUF=0.
      PRINT 901,XLT,YLT,NSHUF,NX,NY,NNN,NSURF
      READ 9001,INIT
      SKIP=.TRUE.
      SURFA  =0.
      IF(INIT.NE.0) PRINT 9014,INIT
      IF(INIT.NE.0) CALL RANFSET(INIT)
      PRINT 941,DELP
      NFX=NX/2.-1. $ NFY=NY/2.-1.
      IF(NY.EQ.1) NFY=0
      NXM=NX-1  $NYM=NY-1
      NFXP=NFX+1  $NFYP=NFY+1
      N1S=NX*NY
      N1MS=NXM*NYM
      IF(NYM.EQ.0) N1MS=NXM
      SCR=SETSPEC(FLOAT(NX),FLOAT(NY))
C CALCULATE THEORETICAL CORRELATION FUNCTION
      DO 20 NPP=1,NY
      NP=NPP-1
      DO 20 MPP=1,NX
      MP=MPP-1
      CPH(MPP,NPP)=SPECTRUM((MP-NFX),(NP-NFY))
      AA(MPP,NPP)=SQRT(2.*XLT*YLT*REAL(CPH(MPP,NPP)))
20    C(MPP,NPP)=0.
      C(NFXP,NFYP)=1.
      IF(NY.EQ.1) GO TO 44
      DO 43 I=1,NXM
43    CPH(I,NY)=AA(I,NY)=0.
44    DO 45 I=1,NXM
45    X(I)=(I-1)*XLT/FLOAT(NX)
      DO 451 I=1,NYM
      CPH(NX,I)=AA(NX,I)=0.
```

```
451   Y(I)=(I-1)*YLT/FLOAT(NY)
      AA(NX,NY)=0.
      CPH(NX,NY)=0.
      X(NX)=(NX-1)*XLT/FLOAT(NX)
      Y(NY)=(NY-1)*YLT/FLOAT(NY)
      CALL SETUP (0,MFF,SV1,SV2,0,IFERR)
      IF(IFERR.EQ.0) PRINT 903
      IF(IFERR.EQ.1) GO TO 798
      CALL FASTFOUR(C,MFF,SV1,SV2,+1,IFERR)
      IF(IFERR.EQ.0) PRINT 905
      IF(IFERR.EQ.1) GO TO 799
      CALL FASTFOUR(CPH,MFF,SV1,SV2,+1,IFERR)
      IF(IFERR.EQ.0) PRINT 905
      IF(IFERR.EQ.1) GO TO 799
      DO 21 NPP=1,NY
      DO 21 MPP=1,NX
21    PHH(MPP,NPP)=REAL(CPH(MPP,NPP)/(C(MPP,NPP)*CONST))
      PRINT 911
      JLY=NY/2
      IF(JLY.EQ.0) JLY=1
      CALL PRINTV(PHH,X,Y,NX,NY,NX/2,JLY,1)
      GAMO=PHH(1,1)
      DO 30 NZ=1,N1MS
30    PHY(NZ)= TWOPI*RANF(-1)
305   DO 315 NZZ=1,NSHUF
      DO 31 NZ=1,N1MS
      IADD    =RANF(-1)*(N1MS-1)+1
      PHINZ=PHY(NZ)
      PHY(NZ)=PHY(IADD)
31    PHY(IADD)=PHINZ
      FNSHUF=FNSHUF+1.
315   CONTINUE
      IF(SKIP) GO TO 39
384   IF(SURFA  .EQ.0.) GO TO 801
      PRINT 9016,SURFA
      DO 385 ICONT=1,N1MS
385   PHY(ICONT)=PHI(ICONT)
39    CONTINUE
      TIM=TIMEF(.1)
      PRINT 9017,TIM
      ICT00=NFXP+NXM*NFY
      DP00= .5*PI-PHY(ICT00)
      ICONT=0
      DO 40 NPP=1,NY
      NP=NPP-1
      DO 40 MPP=1,NX
      MP=MPP-1
      IF((MPP.NE.NX.OR.NX.EQ.1).AND.(NP.NE.NY.OR.NY.EQ.1)) ICONT=ICONT+1
394   RANANG=PHY(ICONT)+DP00
395   A(MPP,NPP)=AA(MPP,NPP)*CEXP(CMPLX(0.,RANANG))
40    CONTINUE
      PRINT 920,A(NFXP,NFYP)
      CALL FASTFOUR(A,MFF,SV1,SV2,+1,IFERR)
      IF(IFERR.EQ.0) PRINT 905
      IF(IFERR.EQ.1) GO TO 799
1000  CONTINUE
      SUM=0.
      SUM2=0.
      DO 50 JYP=1,NY
```

```
      DO 50 JXP=1,NX
      CC=C(JXP,JYP)
      A(JXP,JYP  )=A(JXP,JYP  )/CC
1001  H(JXP,JYP)=REAL(A(JXP,JYP  ))/CONST
      SUM=SUM+H(JXP,JYP)
      SUM2=SUM2+H(JXP,JYP)**2
50    CONTINUE
      SUM=SUM/N1S
      SUM2=SUM2/N1S
      XGAMO=SUM2-SUM**2
      PRINT 930,SUM,SUM2,XGAMO,GAMO
      HMAX=HMIN=0.
      DO 60 JYP=1,NY
      DO 60 JXP=1,NX
      IF(H(JXP,JYP).GT.HMAX) HMAX=H(JXP,JYP)
      IF(H(JXP,JYP).LT.HMIN) HMIN=H(JXP,JYP)
 60   CONTINUE
      PRINT 907,HMAX,HMIN
      IF (.NOT.SKIP) GO TO 1010
      CALL STAT(H,GM,PHH,X,Y,HMAX,HMIN,NX,NY,GAMO,XLT,YLT,INTER)
      DELT= SQERR
      PRINT 9015,FNSHUF,DELT
1003  IF(DELT.GE.DELP) GO TO 38
      DO 35 MN=1,N1MS
35    PHI(MN)=PHY(MN)+DP00
      SURFA  =FNSHUF
      DELP=DELT
38    CONTINUE
      ISURF= ISURF+1
      IF(ISURF.GE.NSURF) SKIP=.FALSE.
      IF (SKIP) GO TO 305
      GO TO 384
1010  CONTINUE
      WRITE(1),XLT,YLT,GAMO,XGAMO,NX,NY,SPECT,INIT,SURFA
      PRINT 908
      CALL PRINTV(H,X,Y,NX,NY,NX,NY,1)
      WRITE(1) ((H(JXP,JYP),JXP=1,NX),H(1,JYP),JYP=1,NY),(H(JXP,1),JXP=1
     1,NX),H(1,1)
      CALL STAT(H,GM,PHH,X,Y,HMAX,HMIN,NX,NY,XGAMO,XLT,YLT,INTER)
      FACTOR=CMPLX(0.,TWOPI)
      DO 730 I=1,2
      IF(I.EQ.1.AND.NX.EQ.1) GO TO 730
      IF(I.EQ.2.AND.NY.EQ.1) GO TO 730
      ICONT=0
      DO 710 NPP=1,NY
      NP=NPP-1
      DO 710 MPP=1,NX
      MP=MPP-1
      IF((MPP.NE.NX.OR.NX.EQ.1).AND.(NP.NE.NY.OR.NY.EQ.1)) ICONT=ICONT+1
      IF(I.EQ.1) DAX(MPP,NPP)=FLOAT(MP-NFX)*FACTOR*AA(MPP,NPP)*CEXP(CMPL
     1X(0.,PHY(ICONT)))/XLT
      IF(I.EQ.2) DAX(MPP,NPP)=FLOAT(NP-NFY)*FACTOR*AA(MPP,NPP)*CEXP(CMPL
     1X(0.,PHY(ICONT)))/YLT
710   CONTINUE
      CALL FASTFOUR(DAX,MFF,SV1,SV2,1,IERR)
      IF (IERR.EQ.0) PRINT 905
      IF (IERR.EQ.1) GO TO 799
      DO 720 JYP=1,NY
      DO 720 JXP=1,NX
```

```
      CC=C(JXP,JYP)
      DAX(JXP,JYP  )= DAX(JXP,JYP  )/CC
      DHX(JXP,JYP)= REAL(DAX(JXP,JYP  ))/CONST
720   CONTINUE
      IF(I.EQ.1) PRINT 909
      IF(I.EQ.2) PRINT 910
      CALL PRINTV(DHX,X,Y,NX,NY,NX,NY,1)
      WRITE(1) ((DHX(JXP,JYP),JXP=1,NX),DHX(1,JYP),JYP=1,NY),(DHX(JXP,1)
     1,JXP=1,NX),DHX(1,1)
730   CONTINUE
      ENDFILE 1
      ENDFILE 1
      GO TO 800
801   PRINT 940,DELP
      GO TO 800
798   PRINT 904 $ GO TO 800
799   PRINT 906
900   FORMAT(4E10.4,4I5)
9001  FORMAT(I13)
901   FORMAT(*1RANDOM SURFACE GENERATOR. */* PERIOD IN X= *,F8.3/
     1* PERIOD IN Y= *,F8.3/* NO. OF SHUFFLES PER TRIAL SURFACE= *,I4/
     2* NUMBER OF SAMPLING POINTS*
     3* ALONG X-AXIS= *,I4,* ALONG Y-AXIS= *,I4/
     4* NUMBER OF INTERVALS PER STANDARD DEVIATION USED IN CALCULATING H
     5EIGHT DISTRIBUTION= *,I4/,* NUMBER OF TEST SURFACES= *,I4)
9014  FORMAT(*ORANDOM NUMBER GENERATOR WAS INITIALIZED TO *,I14)
9015  FORMAT(*OFOR *,F8.1,* SHUFFLES DELTA=*,F13.5)
9016  FORMAT(*OOPTIMAL SURFACE RESULTED FROM*,F8.1,* SHUFFLES*)
9017  FORMAT(*OTHE TIME IN MILLISECONDS IS*,F14.1)
903   FORMAT(*OSUCCESSFUL CALL TO SETUP*)
904   FORMAT(*OUNSUCCESSFUL CALL TO SETUP*)
905   FORMAT(*OSUCCESSFUL CALL TO FASTFOUR*)
906   FORMAT(*OUNSUCCESSFUL CALL TO FASTFOUR*)
907   FORMAT(1H0,//,* MAXIMUM H(X,Y)=*,F13.5,* MINIMUM H(X,Y)= *,F13.5)
908   FORMAT(1H0,//,* THE FOLLOWING MATRIX IS H(X,Y)*)
909   FORMAT(1H0,//,* THE FOLLOWING MATRIX IS DHX(X,Y)*)
910   FORMAT(1H0,//,* THE FOLLOWING MATRIX IS DHY(X,Y)*)
911   FORMAT(1H0,//,* THE FOLLOWING MATRIX IS THE THEORETICAL AUTO-CORRE
     1LATION FUNCTION*)
920   FORMAT(*1A(NFX+1,NFY+1)= *,C(E13.5,E13.5))
930   FORMAT(*OHBAR=*,E12.5,*  H SQUARED BAR=*,E12.5,*  CALCULATED MEAN
     1SQUARED HEIGHT=*,E12.5,*  EXPECTED MEAN SQUARED HEIGHT=*,E12.5)
940   FORMAT(*ONO SURFACE WITH SQERR OF LESS THAN*,E10.4,* WAS FOUND*)
941   FORMAT(*OSEARCH FOR A SURFACE WITH SQERR BETTER THAN*,E12.4)
800   RETURN
      END



      FUNCTION SPECTRUM(M,N)
      COMMON/SPEC/XLT,YLT
      COMMON/CONSTANT/PI,TWOPI,PISQ,SQR2,LNTWO
      REAL LNTWO
      SPECTRUM=GAMAO*PI*CLS*EXP(-PISQ*((CLX*M/XLT)**2+(CLY*N/YLT)**2))
      RETURN
      ENTRY SETSPEC
      READ 900,GAMAO,CLX,CLY
      CLS=CLX*CLY
      IF(YLT.EQ.1) GAMAO=GAMAO/(SQRT(PI)*CLX)
      SPECTRUM=1.
900   FORMAT(8E10.4)
      RETURN
      END
```

```
      SUBROUTINE STAT(H,GM,PHH,X,Y,HMAX,HMIN,NX,NY,MGAMO,XLT,YLT,INTER)
      DIMENSION IX(2,10),HNP(64),HNM(64),UNC(64)
      DIMENSION PH(64),N(64)
      DIMENSION H(NX,NY),GM(64,64),PHH(NX,NY),X(NX),Y(NY)
      DIMENSION XX(10),YY(10)
      COMMON/4/SKIP,SQERR,NNN
      COMMON/CONSTANT/PI,TWOPI,PISQ,SQR2,LNTWO
      REAL MGAMO
      REAL LNTWO
      LOGICAL SKIP
      LOGICAL INTER
      SQERR=0.
      N1S=NX*NY
      DO 10 I=1,64
10    N(I)=0
      DELH=SQRT(2.*MGAMO)/FLOAT(NNN)
      HMM=ABS(HMIN)
      IF(HMAX.GT.HMM) HMM=HMAX
      NN=HMM/DELH+1.
      IF(NN.GT.31) NN=31
      HBOT=(-FLOAT(NN)-.5)*DELH
      NN=2*NN+1
      PRINT 901
      DO 20 JYP=1,NY
      DO 20 JXP=1,NX
      IC1=(H(JXP,JYP)-HBOT)/DELH+1.
      IF(IC1.LE.0) GO TO 20
      IF(IC1.GT.63) GO TO 20
      N(IC1)=N(IC1)+1
20    CONTINUE
      DO 30 I=1,NN
      HINT= HBOT+(FLOAT(I)-.5)*DELH
      FACTOR= 1./SQRT(2.*MGAMO)
      PH(I)= .5*(ERF((HINT+.5*DELH)*FACTOR)-ERF((HINT-.5*DELH)*FACTOR))
      UNC(I)=SQR2/(SQRT(PH(I)*N1S))
      TERM=N(I)/N1S-PH(I)
      PH(I)=TERM/PH(I)
      SQERR=SQERR+TERM*TERM
30    PRINT 902,HINT    ,N(I),PH(I),UNC(I)
      IF(SKIP) RETURN
305   CONTINUE
C  CORRELATION FUNCTION
200   UNCC=2.0 *MGAMO/SQRT(FLOAT(N1S))
      PRINT 907,UNCC
      E=MGAMO*.01
      DO 205 J1=1,NX
      NCX=J1
205   IF(PHH(J1,1).LT.E) GO TO 206
206   DO 207 J2=1,NY
      NCY=J2
207   IF(PHH(1,J2).LT.E) GO TO 208
208   CONTINUE
      NCX=MIN0(64,NCX)
      NCY=MIN0(64,NCY)
      DO 221 J2=1,NCY
      DO 220 J1=1,NCX
      GM(J1,J2)=0.
      DO 210 I1=1,NX
      DO 210 I2=1,NY
```

```
      K1=J1+I1-1  $ IF(K1.GT.NX) K1=K1-NX
      K2=J2+I2-1  $ IF(K2.GT.NY) K2=K2-NY
      TERM=H(K1,K2)*H(I1,I2)
210   GM(J1,J2)=GM(J1,J2)+TERM
      GM(J1,J2)=GM(J1,J2)/FLOAT(N1S)
      GM(J1,J2)=GM(J1,J2)-PHH(J1,J2)*MGAMO/PHH(1,1)
220   CONTINUE
221   CONTINUE
      CALL PRINTV(GM,X,Y,64,64,NCX,NCY,1)
C  JOINT PROBABILITY DISTRIBUTION
      DO 199 LLL=1,10
      READ 904,(IX(I,LLL),I=1,2)
      IF(IX(1,LLL).EQ.0.AND.IX(2,LLL).EQ.0) GO TO 1991
      IF(.NOT.INTER) GO TO 222
      SCR=IX(1,LLL)
      IX(1,LLL)=IX(2,LLL)
      IX(2,LLL)=SCR
222   CONTINUE
      PRINT 905,(IX(I,LLL),I=1,2)
      C12=PHH(IX(1,LLL)+1,IX(2,LLL)+1)/PHH(1,1)
      DEL1=SQRT(2.*MGAMO*(1.-C12))/FLOAT(NNN)
      DEL2=SQRT(2.*MGAMO*(1.+C12))/FLOAT(NNN)
      DEL1=DEL1*2.
      DEL2=DEL2*2.
      HMM=HMM*SQR2
      NN1=HMM/DEL1+1.
      NN2=HMM/DEL2+1.
      IF(NN1.GT.31) NN1=31
      IF(NN2.GT.31) NN2=31
      HBOT1=(-FLOAT(NN1)-.5)*DEL1
      HBOT2=(-FLOAT(NN2)-.5)*DEL2
      NN1=2*NN1+1
      NN2=2*NN2+1
      DO 100 I=1,NN1
      DO 100 J=1,NN2
100   GM(I,J)=0.
      DO 110 J=1,NY
      DO 110 I=1,NX
      IS=I+IX(1,LLL)
      JS=J+IX(2,LLL)
      IF(IS.GT.NX) IS=IS-NX
      IF(JS.GT.NY) JS=JS-NY
      HPL=(H(I,J)+H(IS,JS))/SQR2
      HMI=(H(I,J)-H(IS,JS))/SQR2
      IC1=(HMI-HBOT1)/DEL1+1.
      IC2=(HPL-HBOT2)/DEL2+1.
      IF(IC1.LE.0.OR.IC2.LE.0) GO TO 110
      IF(IC1.GT.63.OR.IC2.GT.63) GO TO 110
      GM(IC1,IC2)=GM(IC1,IC2)+1.
110   CONTINUE
      DO 115 I=1,NN1
115   HNM(I)=HBOT1+(FLOAT(I)-.5)*DEL1
      DO 116 I=1,NN2
116   HNP(I)=HBOT2+(FLOAT(I)-.5)*DEL2
      CALL PRINTV(GM,HNM,HNP, 64, 64,NN1,NN2,9)
      DO 121 J=1,NN2
      HINP=HNP(J)
      DO 121 I=1,NN1
      HINM=HNM(I)
```

```
      SCRATCH= EXP((-(HINP*HINP)/(1.+C12)-(HINM*HINM)/(1.-C12))/(+2.*MGA
     1MO))/(2.*PI*MGAMO*SQRT(1.-C12*C12))*DEL1*DEL2
121   GM (I,J)=      GM(I,J)/(FLOAT(N1S  )*SCRATCH )-1.
      PRINT 906
      CALL PRINTV(GM,HNM,HNP, 64, 64,NN1,NN2,9)
199   CONTINUE
1991  CONTINUE
C     3-POINT CORRELATION FUNCTION
      UNCC=SQRT(30.*MGAMO**3/N1S)
      DO 399 K2=1,10
      READ 904,(IX(K1,K2),K1=1,2)
      IF(IX(1,K2).EQ.0.AND.IX(2,K2).EQ.0) GO TO 400
      IF(.NOT.INTER) GO TO 1992
      SCR=IX(1,K2)
      IX(1,LLL)=IX(2,K2)
      IX(2,K2)=SCR
1992  CONTINUE
      PRINT 908,(IX(K1,K2),K1=1,2),UNCC
      NSTA=FLOAT(NX)/5. +1.
      NSTB=FLOAT(NY)/5. +1.
      NCX=NX/NSTA
      NCY=NY/NSTB
      ICO1=0
      DO 306 I=1,NCX
306   XX(I)=(I  )*NSTA*XLT/FLOAT(NX)
      DO 307 I=1,NCY
307   YY(I)=(I  )*NSTB*YLT/FLOAT(NY)
      SUM=0.
      DO 320 J1=NSTA,NX,NSTA
      ICO2=0
      ICO1=ICO1+1
      DO 320 J2=NSTB,NY,NSTB
      ICO2=ICO2+1
      GM(ICO1,ICO2)=0
      DO 310 I1=1,NX
      DO 310 I2=1,NY
      L1=J1+I1
      L2=J2+I2
      IF(L1.GT.NX) L1=L1-NX
      IF(L2.GT.NY) L2=L2-NY
      M1=IX(1)+I1
      M2=IX(2)+I2
      IF(M1.GT.NX) M1=M1-NX
      IF(M2.GT.NY) M2=M2-NY
      TERM=H(M1,M2)*H(L1,L2)*H(I1,I2)
310   GM(ICO1,ICO2)=GM(ICO1,ICO2)+TERM
      GM(ICO1,ICO2)=GM(ICO1,ICO2)/ N1S
320   SUM=SUM+GM(ICO1,ICO2)**2
      SUM=SUM/25.
      CALL PRINTV(GM,XX,YY,64,64,NCX,NCY,1)
      PRINT 909,SUM
399   CONTINUE
901   FORMAT(*0*,7X,*PROBABILITY DISTRIBUTION OF H*,/*0*,8X,*H*,12X,*N(H
     1)*,5X,*HIST/P(H)-1.*,6X,*EST. ERROR*)
902   FORMAT(1H ,E13.6,3X,I8,7X,F11.5,4X,F11.5)
903   FORMAT(1H0,*MAXIMUM DEVIATION=*,E13.6)
905   FORMAT(1H0,///,* THE FOLLOWING MATRIX IS THE HISTOGRAM*,
     1                * OF H+,H-(X=H-,Y=H+) FOR A SHIFT OF (*,2I10,*)*)
904   FORMAT(2I10)
```

```
906   FORMAT(1H1,20(/),*0THE DEVIATION FROM THEORETICAL OF THE ABOVE MAT
     1TRIX IS GIVEN ON THE FOLLOWING PAGES*)
907   FORMAT(1H1,///,*0THE FOLLOWING MATRIX IS THE ERROR IN THE AUTO-COR
     1RELATION FUNCTION OF H(X,Y), EXPECTED UNC.=*,E13.6)
908   FORMAT(1H0,///,*0THE FOLLOWING MATRIX IS THE THREE-POINT AUTO-CORR
     1ELATION FUNCTION FOR THE SHIFTS (X,Y) AND (*,2I10,*),UNC=*,E13.6)
909   FORMAT(*0MEAN-SQUARED DEVIATION FROM ZERO OF THE CALCULATED POINTS
     1 OF THE THREE-POINT CORRELATION FUNCTION= *,E13.5)
400   RETURN
      END



      SUBROUTINE PRINTV (B,X,Y,NXD,NYD,NX,NY,IFORM)
      DIMENSION B(NXD,NYD),X(NXD),Y(NYD)
      DIMENSION FORM(17)
      DATA (FORM=136H(1H ,1X,F9.5,3X,10(E11.4,1X)    )(1H0,5X,*Y*,7X,10(*
     1X=*,F9.5,1X))(1H ,1X,F8.4,3X,10(F11.5,1X)    )(1H0,5X,*H+ *,5X,10(*
     1H-=*,F8.4,1X)      ))
      JFORM=IFORM+4
      IP09=0
      IF(NX.LE.10) GO TO 30
      IPR=NX/2-5
      IPRNT=IPR/10
      IPRNT=IPR-10*IPRNT
      PRINT FORM(JFORM),(X(I),I=1,IPRNT)
      DO 10 JYP=1,NY
10    PRINT FORM(IFORM),Y(JYP),(B(JXP,JYP),JXP=1,IPRNT)
      IPRNTP=IPRNT+1
      NL=NX-9
      IF(IPRNTP.GT.NL) GO TO 30
      DO 20 I=IPRNTP,NL,10
      IP09=I+9
      PRINT FORM(JFORM),(X(J),J=I,IP09)
      PRINT FORM(IFORM),(Y(JYP),(B(JXP,JYP),JXP=I,IP09),JYP=1,NY)
20    CONTINUE
      IF(IP09.EQ.NX) GO TO 50
30    IP10=IP09+1
      PRINT FORM(JFORM),(X(J),J=IP10,NX)
      DO 40 JYP=1,NY
40    PRINT FORM(IFORM),Y(JYP),(B(JXP,JYP),JXP=IP10,NX)
50    RETURN
      END
```

```
      PROGRAM MAIN1
      DIMENSION H(65,65),HX(65,65),HY(65,65),XX(1025),YY(1025)
      READ 900,FNX,FNY
      JXP=FNX+1.
      JYP=FNY+1.
      CALL PARSTAT8(H,HX,HY,XX,YY,JXP,JYP)
900   FORMAT(8E10.4)
      STOP
      END
      SUBROUTINE PARSTAT8(H,HX,HY,XX,YY,JXP,JYP)
      DIMENSION H(JXP,JYP),HX(JXP,JYP),HY(JXP,JYP),XX(JXP),YY(JYP)
      COMMON/INFO/SZ,SCALE,RMILL,JX,JY,XIPP,YIPP
      DIMENSION G(64,64),NX(31),NY(31),NG(31),N(31),X(31),Y(31)
      DIMENSION PRX(31),PRG(31),ER1(31),ER2(31),ER3(31),PRY(31)
      DIMENSION K(2,2,2)
      DIMENSION SC(2,4)
      DATA(PI=3.14159)
      SZ=SCALE=1.
      READ 910,XLT,YLT,GAMO,XGAMO
      READ 924,SPECT,INIT,SURFA
      JX=JXP-1
      JY=JYP-1
      N1S=JX*JY
      DX=XLT/FLOAT(JX)
      DY=YLT/FLOAT(JY)
      CALL GETH10(H,HX,HY,XX,YY,XLT,YLT,GAMO,XGAMO,JXP,JYP,SPECT,INIT,
     1SURFA)
      DO 10 I=1,31
      N(I)=16-I
10    NX(I)=NY(I)=NG(I)=0.
      GMAX=0.
      HA=-1E+20
      HB=+1E+20
      DO 20 J=1,JY
      DO 20 I=1,JX
      IF(H(I,J).LE.HA) GO TO 11
      HA=H(I,J)
      K(1,1,1)=I
      K(1,1,2)=J
11    IF(H(I,J).GE.HB) GO TO 12
      HB=H(I,J)
      K(2,1,1)=I
      K(2,1,2)=J
12    CONTINUE
      G(I,J)=HX(I,J)**2+HY(I,J)**2
      IF(G(I,J).LE.GMAX) GO TO 20
      GMAX=G(I,J)
      IMAX=I
      JMAX=J
20    CONTINUE
      IS=IMAX-1
      JS=JMAX-1
      F=SQRT(GMAX)
      PRINT 900,F,IS,JS
C  FIRST SUBSCRIPT =1 FOR MAXIMUM,=2 FOR MINIMUM.
C  SECOND SUBSCRIPT =1 AT MAX OR MIN,=2 AT ADJACENT POINT.
C  THIRD SUBSCRIPT =1 FOR X DIMENSION,=2 FOR Y DIMENSION.
      DO 22 I=1,2
      K(I,2,1)=K(I,1,1)+1
```

```
      K(I,2,2)=K(I,1,2)+1
      IF(ABS(H(K(I,2,1),K(I,1,2))).LT.ABS(H(K(I,1,1)-1,K(I,1,2))))
     1K(I,2,1)=K(I,1,1)-1
      IF(K(I,2,1).EQ.0)K(I,2,1)=JX
      IF(JY.EQ.1) GO TO 21
      IF(ABS(H(K(I,1,1),K(I,2,2))).LT.ABS(H(K(I,1,1),K(I,1,2)-1)))
     1K(I,2,2)=K(I,1,2)-1
      IF(K(I,2,2).EQ.0) K(I,2,2)=JY
21    CONTINUE
      DO 210 J=1,2
      SC(J,1)=ABS(HX(K(I,1,1),K(I,J,2))-HX(K(I,2,1),K(I,J,2)))
      IF(JY.EQ.1) GO TO 210
      SC(J,2)=ABS(HX(K(I,J,1),K(I,1,2))-HX(K(I,J,1),K(I,2,2)))
      SC(J,3)=ABS(HY(K(I,J,1),K(I,1,2))-HY(K(I,J,1),K(I,2,2)))
      SC(J,4)=ABS(HY(K(I,1,1),K(I,J,2))-HY(K(I,2,1),K(I,J,2)))
210   CONTINUE
      HXX=AMAX1(SC(1,1),SC(2,1))
      IF(JY.NE.1) GO TO 211
      R=DX/HXX
      GO TO 212
211   HXY=AMAX1(SC(1,2),SC(2,2))
      HYY=AMAX1(SC(1,3),SC(2,3))
      HYX=AMAX1(SC(1,4),SC(2,4))
      HXX=HXX/DX
      HYY=HYY/DY
      HXY=HXY/DY
      HYX=HYX/DX
      HXY=AMAX1(HYX,HXY)
      R=.5*(HXX+HYY+SQRT(HXX**2+HYY**2+4.*HXY**2))
      R=1./R
212   IS=K(I,1,1)-1
      JS=K(I,1,2)-1
      IF(I.EQ.1) PRINT 904,HA,IS,JS,R
      IF(I.EQ.2) PRINT 905,HB,IS,JS,R
22    CONTINUE
      SUMX=SUMY=SUMG=0
      DO 23 I=1,JX
      DO 23 J=1,JY
      SUMX=SUMX+HX(I,J)**2
      SUMY=SUMY+HY(I,J)**2
23    SUMG=SUMG+G(I,J)
      SUMX=2.*SUMX/N1S
      SUMY=2.*SUMY/N1S
      SUMG=   SUMG/N1S
      DELG=GMAX/15.
      DELF=F/15.
      GMAX=15.5*DELG
      FMAX=15.5*DELF
      STP=SQRT(PI)
      DO 25 I=1,16
      X(I)=N(I)*DELF
      Y(I)=N(I)*DELG
      PRX(I)=EXP(-X(I)**2/SUMX)/(STP*SQRT(SUMX))*DELF
      PRY(I)=EXP(-X(I)**2/SUMY)/(STP*SQRT(SUMY))*DELF
25    PRG(I)=EXP(-Y(I)    /SUMG)/ SUMG*DELG
      DO 26 I=17,31
      X(I)=N(I)*DELF
      Y(I)=1E+10
      PRX(I)=PRX(32-I)
```

```
      PRY(I)=PRY(32-I)
26    PRG(I)=0.
      DO 30 I=1,JX
      DO 30 J=1,JY
      IC=(-HX(I,J)+FMAX)/DELF+1.
      NX(IC)=NX(IC)+1
      IC=(-HY(I,J)+FMAX)/DELF+1.
      NY(IC)=NY(IC)+1
      IC=(GMAX-G(I,J))/DELG+1.
      NG(IC)=NG(IC)+1
30    CONTINUE
      NG(16)=2.*NG(16)
      DO 40 I=1,16
      ER1(I)=NX(I)/(N1S*PRX(I))-1.
      ER2(I)=NY(I)/(N1S*PRY(I))-1.
      ER3(I)=NG(I)/(N1S*PRG(I))-1.
40    CONTINUE
      DO 50 I=17,31
      ER1(I)=NX(I)/(N1S*PRX(I))-1.
      ER2(I)=NY(I)/(N1S*PRY(I))-1.
      ER3(I)=0.
50    CONTINUE
      PRINT 901
      PRINT 902,(N(I),X(I),Y(I),NX(I),NY(I),NG(I),PRX(I),PRY(I),PRG(I),
     1ER1(I),ER2(I),ER3(I),I=1,31)
900   FORMAT(*0MAXIMUM MAGNITUDE OF GRADIENT=*,E13.5,* OCCURS AT (*,I5,*
     1,*,I5,*)*//)
901   FORMAT(1H0,42X,*HISTOGRAMS*,15X,*PROBABILITIES*, 23X,*ERRORS*/
     1, *0INTERVAL*,7X,*PARTIAL  GRAD-SQUARED*,4X,* HX*,4X,*HY*,2X,*GRAD
     2SQ*,5X,*HX*,8X,*HY*,6X,*GRADSQ*,9X, *HX*,8X,*HY*,6X,*GRADSQ*/)
902   FORMAT(1H ,I8,2F14.4,3I6,3X,3E10.2,3X,3E10.2)
903   FORMAT(2F10.4)
904   FORMAT(*0THE MAXIMUM OF THE SURFACE =*,F12.6,* AND OCCURS AT(*,2I5
     1,*) .  ESTIMATED RADIUS OF CURVATURE AT THIS POINT=*,F12.6)
905   FORMAT(*0THE MINIMUM OF THE SURFACE =*,F12.6,* AND OCCURS AT(*,2I5
     1,*) .  ESTIMATED RADIUS OF CURVATURE AT THIS POINT=*,F12.6)
910   FORMAT(8F10.4)
924   FORMAT(A8,I13,E10.3)
      RETURN
      END
```

```
      SUBROUTINE GETH10(H,HX,HY,X,Y,XLTS,YLTS,GAMOS, XGAMOS,NXP,NYP,SPEC
     1TS,INITS,SURFAS)
C  THIS VERSION TO BE USED WITH PARSTAT8
      DIMENSION X(NXP),Y(NYP),H(NXP,NYP),HX(NXP,NYP),HY(NXP,NYP)
      COMMON/INFO/SZ,SCALE,RMILL,NX,NY,XIPP,YIPP
      DATA(EP=1E-6)
      CALL SKIPFILE(1)
      NXS=NXP-1
      NYS=NYP-1
      DO 10 I=1,NXP
10    X(I)=FLOAT(I-1)
      DO 15 I=1,NYP
15    Y(I)=FLOAT(I-1)
      DO 200 K=1,100
      READ(1) XLT,YLT,GAMO,XGAMO,NX,NY,SPECT,INIT,SURFA
      IF(EOF,1)800,50
50    CONTINUE
      PRINT 900,XLT,YLT,GAMO,XGAMO,NX,NY,SPECT,INIT,SURFA
      PRINT 900,XLTS,YLTS,GAMOS,XGAMOS,NXS,NYS,SPECTS,INITS,SURFAS
900   FORMAT(1H ,4E18.11,2I6,1X,I13,E18.11)
      IF(SPECT.NE.SPECTS.OR.NX.NE.NXS.OR.NY.NE.NYS.OR.INIT.NE.INITS)
     1GO TO 100
      IF(ABS(XLT-XLTS).GT.EP.OR.ABS(YLT-YLTS).GT.EP.OR.
     1ABS(GAMO-GAMOS).GT.EP.OR.ABS(SURFA-SURFAS).GT.EP) GO TO 100
      GO TO 300
100   CALL SKIPFILE(1)
200   CONTINUE
300   CONTINUE
      READ(1) ((H(I,J),I=1,NXP),J=1,NYP)
      READ(1) (( HX(I,J),I=1,NXP),J=1,NYP)
      IF(NY.EQ.1) GO TO 305
      READ(1) (( HY(I,J),I=1,NXP),J=1,NYP)
      GO TO 310
305   DO 306 I=1,NXP
306   HY(I,1)=HY(I,2)=0.
310   XIPP=1.
      YIPP=1.
      DO 400 I=1,NXP
      DO 400 J=1,NYP
      H(I,J)=SZ*H(I,J)
      HX(I,J)=HX(I,J)*SZ*XIPP/SCALE
      IF(NY.EQ.1) GO TO 400
      HY(I,J)=HY(I,J)*SZ*YIPP/SCALE
400   CONTINUE
C HX AND HY ARE IN UNITS OF DELTA H PER POINT.H IS IN UNITS OF LENGTH
      CALL PRINTV(H ,X,Y,NXP,NYP,NXP,NYP,1)
      CALL PRINTV(HX,X,Y,NXP,NYP,NXP,NYP,1)
      IF(NY.EQ.1) GO TO 801
      CALL PRINTV(HY,X,Y,NXP,NYP,NXP,NYP,1)
      GO TO 801
800   PRINT 901
      STOP
901   FORMAT(*0NO SURFACE WITH DESIRED PARAMETERS ON TAPE*)
801   RETURN
      END
```

```
      SUBROUTINE PRINTV (B,X,Y,NXD,NYD,NX,NY,IFORM)
C  THIS VERSION TO BE USED WITH POINTS8 AND PARSTAT8
      DIMENSION B(NXD,NYD),X(NXD),Y(NYD)
      DIMENSION FORM(17)
      DATA (FORM=136H(1H ,1X,F9.5,3X,10(E11.4,1X)    )(1H0,5X,*Y*,7X,10(*
     1X=*,F9.5,1X))(1H ,1X,F8.4,3X,10(F11.5,1X)    )(1H0,5X,*H+ *,5X,10(*
     1H-=*,F8.4,1X)      ))
      JFORM=IFORM+4
      IP09=0
      IF(NX.LE.10) GO TO 30
      IPR=NX/2-5
      IPRNT=IPR/10
      IPRNT=IPR-10*IPRNT
      PRINT FORM(JFORM),(X(I),I=1,IPRNT)
      DO 10 JYP=1,NY
10    PRINT FORM(IFORM),Y(JYP),(B(JXP,JYP),JXP=1,IPRNT)
      IPRNTP=IPRNT+1
      NL=NX-9
      IF(IPRNTP.GT.NL) GO TO 30
      DO 20 I=IPRNTP,NL,10
      IP09=I+9
      PRINT FORM(JFORM),(X(J),J=I,IP09)
      PRINT FORM(IFORM),(Y(JYP),(B(JXP,JYP),JXP=I,IP09),JYP=1,NY)
20    CONTINUE
      IF(IP09.EQ.NX) GO TO 50
30    IP10=IP09+1
      PRINT FORM(JFORM),(X(J),J=IP10,NX)
      DO 40 JYP=1,NY
40    PRINT FORM(IFORM),Y(JYP),(B(JXP,JYP),JXP=IP10,NX)
50    RETURN
      END
```

```
      PROGRAM POINTS8
C   THIS PROGRAM SEARCHES OUT  THE CHORDS ALWAYS BEGINNING FROM THE SAME EDGE AND
C   MOVING IN THE SAME DIRECTION. IT THEN REVERSES EVERY OTHER CUT SO THAT
C   CUTTING IS STILL DONE IN ALTERNATING DIRECTIONS.
      COMMON/SEARCH/XLT,YLT,GAMO,XGAMO,NXP,NYP,SPECT,INIT,SURFA
      COMMON/INFO/SZ,SCALE,RMILL,NX,NY,XIPP,YIPP
      DIMENSION TITL(10),XC(4,500),S(500),ZY(500),DIS(3)
      DIMENSION OXC(3)
      LOGICAL HIGHER,RETURN,ROUGH
      PI=3.141592653589793
      READ 900,XLT,YLT,GAMO,XGAMO,FNX,FNY
      READ 924,SPECT,INIT,SURFA
      READ 900,SZ,SCALE,GRADM,ZMAX,ZMIN
C USE OF A NEGATIVE SZ GIVES REFLECTION OF ORIGINAL SURFACE
      NX=FNX
      NY=FNY
      WRITE(5),XLT,YLT,GAMO,XGAMO,NX,NY,SPECT,INIT,SURFA
      NXP=NX+1
      NYP=NY+1
      CALL CALLGETH(0.,0.,A,B,C)
      GRADM=ABS(SZ)*GRADM/SCALE
      XLT=XLT*SCALE
      YLT=YLT*SCALE
      IF(SZ.LT.0.) GO TO 1301
      ZMIN=SZ*ZMIN
      ZMAX=ZMAX*SZ
      GO TO 1302
1301  ZMAX=ZMIN*SZ
      ZMIN=ZMAX*SZ
1302  CONTINUE
      DZ=1.1*ZMAX
      XGAMO=XGAMO*SZ**2
      REWIND 5
      READ 9016,TITL
      IF(EOF,60) 300,14
14    READ 900,DIAMR ,EPSR,TAR,TBR,FMAX,FMIN
      READ 900 ,DIAMF ,EPSF,TAF,TBF,FDRTF
      READ 900,DIAMCUT,FDRTC
      RMILLR=.5*DIAMR
      RMILLF=.5*DIAMF
      RCUT=.5*DIAMCUT
      PRINT 920,TITL
      PRINT 901,XLT,YLT,XGAMO,GRADM,NX,NY,SZ,SCALE
      XO=YO=0.
      XEDGE=XLT+2.*(2.1*RMILLF+.1)
      YEDGE=YLT+2.*(2.1*RMILLF+.1)
      FT=RMILLF*GRADM/SQRT(GRADM**2+1)
      RZ=GRADM*RMILLR+1.1*EPSR
      EPS=EPSR
      TA=TAR
      TB=TBR
      RMILL=RMILLR
      ICUT=0
      ROUGH=.TRUE.
      DY=1.8*RMILL
      IF(NY.EQ.1) GO TO 26
      PRINT 9019
16    SCR=1./DY
      PRINT 902,RMILL,EPS,TA,FMAX,FMIN,SCR
```

```
      DX=SQRT(8.*RMILL*EPS)
      Y=YO
      ICOUNT=0
      REEL=1.
161   PRINT 921,REEL
      RETURN=.FALSE.
      BUFFER OUT(5,1)(TITL(1),TITL(10))
      BUFFER OUT(5,1)(ROUGH,ROUGH)
      BUFFER OUT(5,1)(REEL,REEL)
162   IF(UNIT,5) 162,163
163   CONTINUE
      CALL SETUP3(XO,Y,1.,FMAX)
17    CALL INTERP(XO,Y,ZO,ZXO,ZY(1))
      STP=ZXO+TA
      STM=ZXO-TA
      ICUT=ICUT+1
      XC(1,1)=XO
      XC(2,1)=Y
      XC(3,1)=ZO
      PRINT 903
      DO 21 IPT=1,499
      SIP=STP
      SIM=STM
      IPTP=IPT+1
      FAC=1.
      IHALF=0
      XC(2,IPTP)=Y
      HIGHER=.FALSE.
      XT=XC(1,IPT)
18    XT=XT+DX
19    CALL INTERP(XT,Y,ZT,ZXT,ZYT)
      STPT=ZXT+TA
      STMT=ZXT-TA
      DELX=XT-XC(1,IPT)
      ST=(ZT-XC(3,IPT))/DELX
      DEPS=EPS/DELX
      AMPT=AMIN1(AMPT*HIGHER+SIP*(1-   HIGHER),SIP,STPT,ST+DEPS)
      AMMT=AMAX1(AMMT*HIGHER+SIM*(1-   HIGHER),SIM,STMT,ST-DEPS)
      IF(AMPT.LT.ST.OR.AMMT.GT.ST) GO TO 20
      IF(ABS(ZY(IPT)-ZYT      ).GT.TB) GO TO 20
      IF(XT.LT.XLT+XO) GO TO 191
      XC(1,IPTP)=XLT+XO
      CALL INTERP(XLT+XO,Y,XC(3,IPTP),ZX,ZY(IPTP))
      S(IPTP)=(XC(3,IPTP)-XC(3,IPT))/(XC(1,IPTP)-XC(1,IPT))
      GO TO 22
191   HIGHER=.TRUE.
      XC(1,IPTP)=XT
      XC(3,IPTP)=ZT
      STP=STPT
      STM=STMT
      S(IPTP)=ST
      ZY(IPTP)=ZYT
199   IF(FAC.NE.1.) GO TO 21
      GO TO 18
20    IF(HIGHER) GO TO 21
      FAC=.5*FAC
      XT=XC(1,IPT)+FAC*DX
      IF(IHALF.GT.5) PRINT 922,IHALF,IPT,XC(1,IPT),AMPT,SIP,STPT,AMMT,SI
     1M,STMT,ST,DEPS
```

```
      IHALF=IHALF+1
      IF(IHALF.LE.35) GO TO 19
      STOP
21    CONTINUE
22    IPT=IPTP-1
      S(1)=S(IPTP)
      DO 230 I=1,IPT
      IF(.NOT.ROUGH) GO TO 225
      XC(3,I)=XC(3,I)+RZ
      GO TO 230
225   J=I+1
      CALL DISP(S(I),S(J),ZY(I),DIS(1),DIS(2),DIS(3))
      DO 23 K=1,3
23    XC(K,I)=XC(K,I)+DIS(K)
230   CONTINUE
237   DO 238 J=1,500
2371  IF(UNIT,5) 2371,2372
2372  IPT=IPT+1
      XC(1,IPT)=XC(1,J)+XLT
      XC(2,IPT)=XC(2,J)
      XC(3,IPT)=XC(3,J)
      S(IPT)=S(J)
      ZY(IPT)=ZY(J)
      IF(XC(1,IPT).GE.XEDGE) GO TO 239
238   CONTINUE
239   CONTINUE
C  THE FOLLOWING LOOP REVERSES ORDER OF EVERY OTHER CUT TO PRODUCE RETURN CUT.
      IF(.NOT.RETURN) GO TO 232
      DO 231 I=1,IPT
      J=IPT -I+1
      IF(I.GE.J) GO TO 232
      SCR=S(I)
      S(I)=S(J)
      S(J)=SCR
      SCR=ZY(I)
      ZY(I)=ZY(J)
      ZY(J)=SCR
      DO 231 K=1,3
      SCR=XC(K,I)
      XC(K,I)=XC(K,J)
231   XC(K,J)=SCR
232   SCR=AMAX1(ABS(OXC(1)-XC(1,1)),ABS(OXC(2)-XC(2,1)),ABS(OXC(3)-XC(3,
     11)))
      IF(ROUGH) A=FMIN*ABS(ZMIN-DZ)/(FMAX-FMIN)
      XC(3,1)=XC(3,1)-DZ
      IF(ICOUNT.EQ.0) SCR=ABS(10.*XC(3,1))
      XC(4,1)=FMIN/SCR
      IF(SCR.LE..015) XC(4,1)=100.
      BUFFER OUT(5,1)(XC(1,1),XC(4,1))
      J=1
      PRINT 904,ICUT,J,(XC(LLL,J),LLL=1,4),S(J),ZY(J)
      DO 2321 LLL=1,3
2321  OXC(LLL)=XC(LLL,IPT)
      DO 235 J=2,IPT
      I=J-1
      XC(3,J)=XC(3,J)-DZ
      SLIDE=AMAX1(ABS(XC(1,J)-XC(1,I)),ABS(XC(2,J)-XC(2,I)),ABS(XC(3,J)-
     1XC(3,I)))
      XC(4,J)=FMAX/SLIDE
```

```
      IF(ROUGH) XC(4,J)=XC(4,J)*A/(A+ABS(AMIN1(XC(3,J),XC(3,I),0.)))
      IF(SLIDE.LE..015) XC(4,J)=100.
233   IF(UNIT,5) 233,234
234   BUFFER OUT(5,1)(XC(1,J),XC(4,J))
      PRINT 904,ICUT,J,(XC(LLL,J),LLL=1,4),S(J),ZY(J)
235   CONTINUE
      ICOUNT=ICOUNT+IPT
2391  IF(UNIT,5)2391,2392
2392  IF(Y.GE.YEDGE) GO TO 25
      RETURN=.NOT.RETURN
      Y=Y+DY
      IF(ICOUNT.GE.2200) GO TO 24
      GO TO 17
24    CALL FINISH3(XC(1,IPT),XC(2,IPT),XC(3,IPT),FMAX)
      ENDFILE 5
      PRINT 906,REEL,ICOUNT
      REEL=REEL+1.
      ICOUNT=0
      GO TO 161
25    CALL FINISH3(XC(1,IPT),XC(2,IPT),XC(3,IPT),FMAX)
      ENDFILE 5
      ENDFILE 5
      PRINT 906,REEL,ICOUNT
      IF(.NOT.ROUGH) PRINT 908,TITL
      IF(.NOT.ROUGH) GO TO 300
      PRINT 907,TITL
26    XO=YO=2.1*RMILLF
      IF(NY.EQ.1) XO=YO=0.
      XEDGE=XO+XLT+.2+FT
      YEDGE=YO+YLT+.2+FT
      IF(NY.EQ.1) YEDGE=0.
      ICUT=0
      ROUGH=.FALSE.
      EPS=EPSF
      TA=TAF
      TB=TBF
      RMILL=RMILLF
      FMIN=FMAX=FDRTF
      DY=SQRT(4.*RMILL*EPS/(1.+GRADM**2))
      PRINT 9020
      GO TO 16
300   IF(NY.EQ.1) GO TO 715
      XCUT=YCUT=XO+.1
      REEL=1.
      ICUT=8HCUTOUT
      SCRX=XCUT+RCUT+XLT
      SCRY=YCUT+RCUT+YLT
      PRINT 9021
      BUFFER OUT(5,1)(TITL(1),TITL(10))
      BUFFER OUT(5,1)(ICUT,ICUT)
      BUFFER OUT(5,1)(REEL,REEL)
301   IF(UNIT,5)301,302
302   XC(1)=XCUT-RCUT
      XC(2)=-4.
      XC(3)=0.
      XC(4)=FDRTC/4.
      PRINT 923,(XC(LLL),LLL=1,4)
      BUFFER OUT(5,1)(XC(1),XC(4))
700   IF(UNIT,5)700,701
```

```
701   XC(3)=-5.
      XC(4)=FDRTC/5.
      PRINT 923,(XC(LLL),LLL=1,4)
      BUFFER OUT (5,1)(XC(1),XC(4))
702   IF(UNIT,5)702,703
703   XC(2)=SCRY
      XC(4)=FDRTC/ABS(SCRY+4.)
      PRINT 923,(XC(LLL),LLL=1,4)
      BUFFER OUT (5,1)(XC(1),XC(4))
704   IF(UNIT,5)704,705
705   XC(1)=SCRX
      XC(4)=FDRTC/ABS(2.*RCUT+XLT)
      PRINT 923,(XC(LLL),LLL=1,4)
      BUFFER OUT (5,1)(XC(1),XC(4))
706   IF(UNIT,5)706,707
707   XC(2)=XCUT-RCUT
      PRINT 923,(XC(LLL),LLL=1,4)
      BUFFER OUT (5,1)(XC(1),XC(4))
708   IF(UNIT,5)708,709
709   XC(1)=YCUT-RCUT
      PRINT 923,(XC(LLL),LLL=1,4)
      BUFFER OUT (5,1)(XC(1),XC(4))
710   IF(UNIT,5)710,711
711   XC(3)=0.
      XC(4)=FDRTC/5.
      PRINT 923,(XC(LLL),LLL=1,4)
      BUFFER OUT (5,1)(XC(1),XC(4))
712   IF(UNIT,5)712,713
713   XC(4)=FDRTC/ABS(XC(1))
      XC(1)=XC(2)=0.
      PRINT 923,(XC(LLL),LLL=1,4)
      BUFFER OUT (5,1)(XC(1),XC(4))
714   IF(UNIT,5)714,715
715   ENDFILE 5
      ENDFILE 5
      REWIND 5
900   FORMAT(8E10.4)
901   FORMAT(*0SCALED SURFACE PARAMETERS ARE AS FOLLOWS*/
     15X,*PERIOD IN X=*,F10.4,1X,* INCHES*/
     15X,*PERIOD IN Y=*,F10.4,1X,* INCHES*/
     15X,*MEAN SQUARE HEIGHT=*,F10.4,1X, * INCHES*/
     15X,*MAXIMUM GRADIENT=*,F12.6/
     15X,*NUMBER OF GRID POINTS PER PERIOD IN X=*,I5/
     15X,*NUMBER OF GRID POINTS PER PERIOD IN Y=*,I5//
     1* THE ORIGINAL SURFACE WAS SCALED AS FOLLOWS*/
     15X,*Z-AXIS*,F12.6/
     15X,*X AND Y AXES*,F12.6/)
9016  FORMAT(10A8)
9019  FORMAT(////*0ROUGHING*)
9020  FORMAT(*1FINISHING*)
9021  FORMAT(*1CUT OUT*/)
902   FORMAT(*0MILL RADIUS=*,F10.4/
     1* EPSILON=*,E13.4/
     1* SLOPE CRITERION=*,E13.4/
     1* FEEDRATE VARIES BETWEEN *,F10.2,* AND *,F10.2/
     1* NUMBER OF CUTS PER INCH=*,F10.2)
903   FORMAT(1H0,3X,*CUT  POINT*,3X,*X*,12X,*Y*,12X,*Z*,10X,* FRN *, 8X,
     1*SLOPE*,10X,*ZY*)
904   FORMAT(1X,2I5,9E13.5)
```

```
906   FORMAT(*0END OF REEL*,F3.0,*   THIS REEL CONTAINS*,I5,* POINTS.*)
907   FORMAT(*0THIS REEL COMPLETES ROUGHING OF *,5X,10A8)
908   FORMAT(*0THIS REEL COMPLETES FINISHING OF *,5X,10A8)
920   FORMAT(1H1,10A8)
921   FORMAT(*0REEL NO. *,F3.0)
922   FORMAT(*0HALVED*,I6,  * TIMES.*,I5,*TH CHORD.X=*,F10.4/
     1* AMPT,SIP,STPT=*,3F12.6/* AMMT,SIM,STMT=*,3F12.6/* ST,DEPS=*,2F12
     2.6)
923   FORMAT(1H ,10E13.5)
924   FORMAT(A8,I13,E10.3)
      STOP
      END



      SUBROUTINE SETUP3(XO,YO,ZO,F)
C  THIS VERSION TO BE USED WITH POINTS8
      DIMENSION XC(4)
      XC(1)=XO
      XC(2)=YO
      XC(3)=0.
      SCR=AMAX1(ABS(XO),ABS(YO))
      XC(4)=F/SCR
      IF(SCR.LE..015) XC(4)=100.
      BUFFER OUT (5,1) (XC(1),XC(4))
506   IF(UNIT,5) 506,606
606   CONTINUE
      GO TO 100
      ENTRY FINISH3
      XC(1)=XO
      XC(2)=YO
      XC(3)=0.
      XC(4)=F/ABS(ZO)
      IF(ABS(ZO).LE..015)XC(4)=100.
      BUFFER OUT (5,1) (XC(1),XC(4))
507   IF(UNIT,5) 507,607
607   CONTINUE
      SCR=AMAX1(ABS(XO),ABS(YO))
      XC(4)=F/SCR
      IF(SCR.LE..015) XC(4)=100.
      XC(1)=0.
      XC(2)=0.
      BUFFER OUT (5,1) (XC(1),XC(4))
508   IF(UNIT,5) 508,608
608   CONTINUE
100   RETURN
      END
```

```
      SUBROUTINE DISP(S1,S2,ZY,DX,DY,DZ)
C  THIS VERSION TO BE USED WITH POINTS8
      COMMON/INFO/SZ,SCALE,RMILL
      S1S=S1*S1
      S2S=S2*S2
      C1P=1./SQRT(S1S+1.)
      C2P=1./SQRT(S2S+1.)
      D2=(C1P+C2P)/(2.*C1P)
      D1=(C1P+C2P)/(2.*C2P)
      ZY1=ZY*D1
      ZY2=ZY*D2
      C1=1./SQRT(S1S+ZY2**2+1.)
      C2=1./SQRT(S2S+ZY1**2+1.)
      DX=-S1*C1*RMILL
      DY=-ZY2*C1*RMILL
      DZ=(C1-1.)*RMILL
      IF(ABS(S1-S2).LE.1E-10) RETURN
      ZS1=ZY2*ZY1+1.
      DXP=RMILL*C1P*SQRT((1.-C1*C2*(S1*S2+ZS1))/(1.+C1P*C2P*(S1*S2+1.)))
      IF(S1.LT.S2) DXP=-DXP
      DZP=S1*DXP
      DX=DX+DXP
      DZ=DZ+DZP
      RETURN
      END


      SUBROUTINE INTERP(XX,YY,Z,ZX,ZY)
C  THIS VERSION TO BE USED WITH POINTS8
C        INTERPOLATES TO XX,YY ON THE N1 BY N1 SURFACE H WITH GRADIENT
C        (HX,HY).  H IS EXTENDED PERIODICALLY IF XX,YY OUT OF RANGE
      DIMENSION BU(4),BUP(4),F(2),FO(2),FP(2)
      COMMON/INFO/SZ,SCALE,RMILL,JX,JY,XIPP,YIPP
      COMMON/SEARCH/XLT,YLT,GAMO,XGAMO,JXP,JYP,SPECT,INIT,SURFA
      DIMENSION H(4225,1),HX(4225,1),HY(4225,1),XA(1025),YA(1025)
      JXP=JX+1
      JYP=JY+1
      X=XX/XIPP
      Y=YY/YIPP
      IF(X.LT.0.) X=X+JX*100
      IF(Y.LT.0.) Y=Y+JY*100
      NX=X
      NY=Y
      U=X-NX
      W=Y-NY
      NX=MOD(NX,JX)
      NY=MOD(NY,JY)
      NX=NX+1
      NY=NY+1
      K=1
      DO 25 I=1,2
      DO 20 J=1,2
      GO TO (5,10),I
5     V=U
      NYP=NY+J-1
C        THESE STATEMENTS TAKE SUBSCRIPTS MODULO 64 FOR PERIODICITY
      IF(NYP.EQ.JYP) NYP=1
      NXP=NX+1
      IF(NXP.EQ.JXP) NXP=1
      NNN=(NYP-1)*JXP
      HA=H(NX+NNN)
      HB=H(NXP+NNN)
      GA=HX(NX+NNN)
      GB=HX(NXP+NNN)
      GO TO 15
```

```
10    V=W
      NXP=NX+J-1
      IF(NXP.EQ.JXP) NXP=1
      NYP=NY+1
      IF(NYP.EQ.JYP) NYP=1
      HA=H(NXP+(NY-1)*JXP)
      HB=H(NXP+(NYP-1)*JXP)
      GA=HY(NXP+(NY-1)*JXP)
      GB=HY(NXP+(NYP-1)*JXP)
15    A=2.*(HA-HB)+GA+GB
      B=3.*(HB-HA)-GA-GA-GB
C        BU AND BUP ARE THE SURFACE AND SLOPE ALONG THE BOUNDARIES
C        OF THE SQUARE.  F IS F1, FO IS FO, AND FP IS F1'=-FO'
      BU(K)=HA+V*(GA+V*(B+V*A))
      BUP(K)=GA+V*(B+B+3.*V*A)
20    K=K+1
      F(I)=V*V*(3.-V-V)
      FO(I)=1.-F(I)
25    FP(I)=6.*V*(1.-V)
      NYP=NY+1
      IF(NYP.EQ.JYP) NYP=1
      HC=H(NX+(NYP-1)*JXP)
      HD=H(NX+(NY-1)*JXP)
      Z=BU(3)*FO(1)+BU(4)*F(1)+BU(1)*FO(2)+BU(2)*F(2)-HD*FO(1)*FO(2) -
     1   HC*FO(1)*F(2)-HA*F(1)*FO(2)-HB*F(1)*F(2)
      ZX=BUP(1)*FO(2)+BUP(2)*F(2)+FP(1)*(BU(4)-BU(3)+FO(2)*(HD-HA) +
     1   F(2)*(HC-HB))
      ZY=BUP(3)*FO(1)+BUP(4)*F(1)+FP(2)*(BU(2)-BU(1)+FO(1)*(HD-HC) +
     1   F(1)*(HA-HB))
      ZX=ZX/XIPP
      ZY=ZY/YIPP
      RETURN
      ENTRY CALLGETH
      CALL GETH(H,HX,HY,XA,YA,XLT,YLT,GAMO,XGAMO,JXP,JYP,SPECT,INIT,SURF
     1A)
      RETURN
      END
```

```
  SUBROUTINE GETH(H,HX,HY,X,Y,XLTS,YLTS,GAMOS, XGA
 1,INITS,SURFAS)
HIS VERSION TO BE USED WITH POINTS8
  DIMENSION X(NXP),Y(NYP),H(NXP,NYP),HX(NXP,NYP),H
  COMMON/INFO/SZ,SCALE,RMILL,NX,NY,XIPP,YIPP
  DATA(EP=1E-6)
  CALL SKIPFILE(1)
  NXS=NXP-1
  NYS=NYP-1
  DO 10 I=1,NXP
  X(I)=FLOAT(I-1)
  DO 15 I=1,NYP
  Y(I)=FLOAT(I-1)
  DO 200 K=1,100
  READ(1) XLT,YLT,GAMO,XGAMO,NX,NY,SPECT,INIT,S
  IF(EOF,1)800,50
  CONTINUE
  IF(SPECT.NE.SPECTS.OR.NX.NE.NXS.OR.NY.NE.NYS
 1GO TO 100
  IF(ABS(XLT-XLTS).GT.EP.OR.ABS(YLT-YLTS).GT.E
 1ABS(GAMO-GAMOS).GT.EP.OR.ABS(SURFA-SURFAS).(
  GO TO 300
  CALL SKIPFILE(1)
  CONTINUE
  CONTINUE
  READ(1) ((H(I,J),I=1,NXP),J=1,NYP)
  READ(1) (( HX(I,J),I=1,NXP),J=1,NYP)
  IF(NY.EQ.1) GO TO 305
  READ(1) (( HY(I,J),I=1,NXP),J=1,NYP)
  GO TO 310
  DO 306 I=1,NXP
  HY(I,1)=HY(I,2)=0.
  XIPP=XLT/FLOAT(NX)
  YIPP=YLT/FLOAT(NY)
  DO 400 I=1,NXP
  DO 400 J=1,NYP
  H(I,J)=SZ*H(I,J)
  HX(I,J)=HX(I,J)*SZ*XIPP/SCALE
  IF(NY.EQ.1) GO TO 400
  HY(I,J)=HY(I,J)*SZ*YIPP/SCALE
  CONTINUE
; AND HY ARE IN UNITS OF DELTA H PER POI
  CALL PRINTV(H ,X,Y,NXP,NYP,NXP,NYP,1
  CALL PRINTV(HX,X,Y,NXP,NYP,NXP,NYP,
  IF(NY.EQ.1) GO TO 801
  CALL PRINTV(HY,X,Y,NXP,NYP,NXP,NYP
  GO TO 801
  PRINT 901
  STOP
  FORMAT(4E10.3,I7,I13,E10.3)
  FORMAT(*ONO SURFACE WITH DESI
  RETURN
  END
```

```
      SUBROUTINE PRINTV (B,X,Y,NXD,NYD,NX,NY,IFORM)
C  THIS VERSION TO BE USED WITH POINTS8
      DIMENSION B(NXD,NYD),X(NXD),Y(NYD)
      DIMENSION FORM(17)
      DATA (FORM=136H(1H ,1X,F9.5,3X,10(E11.4,1X)    )(1H0,5X,*Y*,7X,10(*
     1X=*,F9.5,1X))(1H ,1X,F8.4,3X,10(F11.5,1X)    )(1H0,5X,*H+ *,5X,10(*
     1H-=*,F8.4,1X)      ))
      JFORM=IFORM+4
      IP09=0
      IF(NX.LE.10) GO TO 30
      IPR=NX/2-5
      IPRNT=IPR/10
      IPRNT=IPR-10*IPRNT
      PRINT FORM(JFORM),(X(I),I=1,IPRNT)
      DO 10 JYP=1,NY
10    PRINT FORM(IFORM),Y(JYP),(B(JXP,JYP),JXP=1,IPRNT)
      IPRNTP=IPRNT+1
      NL=NX-9
      IF(IPRNTP.GT.NL) GO TO 30
      DO 20 I=IPRNTP,NL,10
      IP09=I+9
      PRINT FORM(JFORM),(X(J),J=I,IP09)
      PRINT FORM(IFORM),(Y(JYP),(B(JXP,JYP),JXP=I,IP09),JYP=1,NY)
20    CONTINUE
      IF(IP09.EQ.NX) GO TO 50
30    IP10=IP09+1
      PRINT FORM(JFORM),(X(J),J=IP10,NX)
      DO 40 JYP=1,NY
40    PRINT FORM(IFORM),Y(JYP),(B(JXP,JYP),JXP=IP10,NX)
50    RETURN
      END
```

```
      PROGRAM PAPER2
      DIMENSION CO( 4,5000),N(10),I(264),NX(3),IX(7)
      DIMENSION ITITL(11),JTITL(10)
      DATA(NDEOF=0)
      DATA (NN     =105B),(N( 1)=040B)
     1      (NG     =147B),(N( 2)=001B),
     2      (NX(1)  =067B),(N( 3)=002B),
     3      (NX(2)  =070B),(N( 4)=023B),
     4      (NX(3)  =051B),(N( 5)=004B),
     5      (NPLUS  =160B),(N( 6)=025B),
     6      (NMINUS=100B),(N( 7)=026B),
     7      (NF     =166B),(N( 8)=007B),
     8      (NM     =124B),(N( 9)=010B),
     9      (NDBLK  =200B),(N(10)=031B),
     A      (NSPACE=020B),(NEOR=013B)
      DATA(ITITL(11)=8H REEL NO )
      NNPOINTS=4999
      REWIND 5
      NYY=1
      READ 901,(JTITL(LLL),LLL=1,8),SROUGH,SREEL
      PRINT 900,(JTITL(LLL),LLL=1,8),        SREEL
901   FORMAT(8A8,I2,F3.1,E10.4)
900   FORMAT(1H0,8A8,    F3.1,E10.4)
1     BUFFER IN(5,1)(ITITL(1),ITITL(10))
11    IF(UNIT,5) 11,12,96,97
12    BUFFER IN(5,1)(ROUGH,ROUGH)
121   IF(UNIT,5)121,122,97
122   BUFFER IN(5,1)(REEL,REEL)
13    IF(UNIT,5)13,14,97
14    CONTINUE
      PRINT 900,(ITITL(LLL),LLL=1,8)        ,REEL
      DO 206 LLL=1,8
      IF(JTITL(LLL).EQ.ITITL(LLL)) GO TO 206
      NDEOF=3
      GO TO 96
206   CONTINUE
      IF(SROUGH.EQ.ROUGH) GO TO 207
      DO 209 LLL=1,20
      CALL SKIPFILE (5)
      BUFFER IN(5,1)(ITITL(1),ITITL(2))
208   IF(UNIT,5)208,209,96,97
209   CONTINUE
207   CONTINUE
2     IF(SREEL.EQ.REEL) GO TO 205
      CALL SKIPFILE (5)
      GO TO 1
205   CONTINUE
      MASK=77B
      DO  3 J=1,11
      IXT=10000000000000 B
      DO  3 M=1,8
      ICO=ITITL(J)/IXT
      ICO=ICO.AND.MASK
      IXT=IXT/100B
      CALL SYMBOL(ICO)
3     CONTINUE
      CALL SYMBOL(48)
      ICO=INT(REEL/10.)
      CALL SYMBOL(ICO)
```

```
      ICO=INT(REEL)-ICO*10
      CALL SYMBOL(ICO)
      CALL SYMBOL(58)
      DO 47 IJK=1,NNPOINTS
      BUFFER IN (5,1) (CO(1,IJK),CO(4,IJK))
39    IF(UNIT,5)39,4,48,97
4     NPOINTS=IJK
47    CONTINUE
48    CONTINUE
      IF(NYY.NE.1) GO TO 484
      DO 481 IJK=1,NPOINTS
      CO(2,IJK)=CO(3,IJK)
481   CO(3,IJK)=0.
484   CONTINUE
      NPOINTS=NPOINTS+1
      DO 485 KKK=1,3
485   CO(KKK,NPOINTS)=CO(KKK,NPOINTS-1)
      CO(4,NPOINTS)=100.0
      DO 49 KKK=1,200
49    I(KKK)=NSPACE
      I(201)=NEOR
      I(202)=NDBLK
      BUFFER OUT (1,1) (I(1),I(202))
491   IF(UNIT,1)491,492
492   CONTINUE
      IA=0 $ IB=0 $ IC=1
      L=1
      IJK=1
      ISEQ=1
5     I(L)=NN
      L=L+1
      IF(IA.EQ.0) GO TO 10
      I(L)=N(IA+1)
      L=L+1
10    IF(IA.EQ.0.AND.IB.EQ.0) GO TO 20
      I(L)=N(IB+1)
      L=L+1
20    I(L)=N(IC+1)
      L=L+1
      IF(ISEQ.NE.1) GO TO 25
      I(L)=NG
      I(L+1)=N(10)
      I(L+2)=N(1)
      I(L+3)=NG
      I(L+4)=N(2)
      L=L+5
25    DO 40 INDEX=1,3
      I(L)=NX(INDEX)
      L=L+1
      ICO=CO(INDEX,IJK)*1E+4
      ISIGN=ICO/IABS(ICO)
      ICO=IABS(ICO)
      IF(ISIGN) 31,32,32
31    I(L)=NMINUS
      GO TO 33
32    I(L)=NPLUS
```

```
      IX(K)=ICO/TEN
      ICO=ICO-IX(K)*TEN
331   TEN=TEN/10.
      IF(IX(1).EQ.0) GO TO 34
      I(L)=N(IX(1)+1)
      L=L+1
34    IF(IX(2).EQ.0.AND.IX(1).EQ.0) GO TO 35
      I(L)=N(IX(2)+1)
      L=L+1
35    IF(IX(3).EQ.0.AND.IX(2).EQ.0.AND.IX(1).EQ.0) GO TO 36
      I(L)=N(IX(3)+1)
      L=L+1
36    I(L)=N(IX(4)+1)
      I(L+1)=N(IX(5)+1)
      I(L+2)=N(IX(6)+1)
      I(L+3)=N(IX(7)+1)
      L=L+4
40    CONTINUE
      I(L)=NF
      L=L+1
      ICO=CO(4,IJK)*10
      TEN=1E+6
      DO 401 K=1,7
      IX(K)=ICO/TEN
      ICO=ICO-IX(K)*TEN
401   TEN=TEN/10.
      DO 405 K=1,6
      K=K
      IF(IX(K).NE.0) GO TO 41
405   CONTINUE
41    DO 42 LLL=K,6
      I(L)=N(IX(LLL)+1)
      L=L+1
42    CONTINUE
      I(L)=N(IX(7)+1)
      L=L+1
      IF(IJK.EQ.NPOINTS) GO TO 70
      IF(ISEQ.NE.1) GO TO 50
      I(L)=NM
      I(L+1)=N(4)
      L=L+2
50    I(L)=NDBLK
      IJK=IJK+1
      IF(L.LE.220) GO TO 60
      BUFFER OUT (1,1) (I(1),I(L))
501   IF(UNIT,1)501,502
502   CONTINUE
      L=0
60    L=L+1
      ISEQ=ISEQ+1
      IF(ISEQ.EQ.1000) ISEQ=1
      ICO=ISEQ
      IA=ICO/100.
      ICO=ICO-IA*100.
      IB=ICO/10.
      ICO=ICO-IB*10.
      IC=ICO
      GO TO 5
70    I(L)=NM
```

```
      I(L+1)=N(6)
      I(L+2)=NDBLK
      L=L+2
      BUFFER OUT (1,1) (I(1),I(L))
701   IF(UNIT,1)701,702
702   CONTINUE
      L=1
      ISEQ=ISEQ+1
      IF(ISEQ.EQ.1000) ISEQ=1
      ICO=ISEQ
      IA=ICO/100.
      ICO=ICO-IA*100.
      IB=ICO/10.
      ICO=ICO-IB*10.
      IC=ICO
      I(L)=NN
      L=L+1
      IF(IA.EQ.0) GO TO 71
      I(L)=N(IA+1)
      L=L+1
71    IF(IA.EQ.0.AND.IB.EQ.0) GO TO 72
      I(L)=N(IB+1)
      L=L+1
72    I(L)=N(IC+1)
      I(L+1)=NF
      I(L+2)=N(3)
      I(L+3)=N(1)
      I(L+4)=N(1)
      I(L+5)=N(1)
      I(L+6)=NM
      I(L+7)=N(4)
      I(L+8)=N(1)
      I(L+9)=NDBLK
      L=L+10
      DO 80 KKK= L,202
80    I(KKK)=NSPACE
      BUFFER OUT (1,1) (I(1),I(202))
      GO TO 95
96    NDEOF=NDEOF+1
      IF(NDEOF.LE.2) GO TO 1
      PRINT 910
910   FORMAT(*0NO SUCH FILE ON TAPE*)
      GO TO 95
97    PRINT 920
920   FORMAT(*0IRRECOVERABLE PARITY ERROR ON UNIT 5*)
95    REWIND 5
      STOP
      END
```

```
      SUBROUTINE SYMBOL(IISW)
      DIMENSION I(264)
      DATA (L=0)
      ISW=IISW
      IF(ISW.EQ.0) ISW=38
      IF(L.EQ.0.AND.ISW.EQ.58 ) RETURN
      IF(ISW.GT.30) GO TO 401
      GO TO (1,2,3,4,5,6,7,8,9,9,9,9,9,9,9,9,17,18,19,20,21,22,23,24,25,
     125,25,25,25,25),ISW
401   ISW=ISW-30
      GO TO (32,32,33,34,35,36,37,38,39,40,41,41,41,41,41,41,41,48,48,50
     1,51,52,53,54,55,56,57,58),ISW
C  A=21B=17,B=22B=18,C=23B=19,D=24B=20,E=25B=21,F=26B=22,G=27B=23,H=30B=24,ETC
1     IA=00B$IB=22B$IC=76B$ID=02B$IE=00B$GO TO 400                          1
2     IA=56B$IB=52B$IC=52B$ID=52B$IE=22B$GO TO 400                          2
3     IA=42B$IB=42B$IC=52B$ID=52B$IE=24B$GO TO 400                          3
4     IA=70B$IB=10B$IC=10B$ID=76B$IE=10B$GO TO 400                          4
5     IA=72B$IB=52B$IC=52B$ID=52B$IE=44B$GO TO 400                          5
6     IA=34B$IB=52B$IC=52B$ID=52B$IE=04B$GO TO 400                          6
7     IA=40B$IB=40B$IC=56B$ID=60B$IE=40B$GO TO 400                          7
8     IA=24B$IB=52B$IC=52B$ID=52B$IE=24B$GO TO 400                          8
9     IA=20B$IB=52B$IC=52B$ID=52B$IE=34B$GO TO 400                          9
17    IA=36B$IB=50B$IC=50B$ID=50B$IE=36B$GO TO 400                          A
18    IA=76B$IB=52B$IC=52B$ID=52B$IL=24B$GO TO 400                          B
19    IA=34B$IB=42B$IC=42B$ID=42B$IE=24B$GO TO 400                          C
20    IA=76B$IB=42B$IC=42B$ID=42B$IE=34B$GO TO 400                          D
21    IA=76B$IB=52B$IC=52B$ID=52B$IE=42B$GO TO 400                          E
22    IA=76B$IB=50B$IC=50B$ID=50B$IE=40B$GO TO 400                          F
23    IA=34B$IB=42B$IC=42B$ID=52B$IE=54B$GO TO 400                          G
24    IA=76B$IB=10B$IC=10B$ID=10B$IE=76B$GO TO 400                          H
25    IA=42B$IB=42B$IC=76B$ID=42B$IE=42B$GO TO 400                          I
32    IA=IB=IC=ID=IE=10B$GO TO 400                                          -
33    IA=44B$IB=42B$IC=74B$ID=40B$IE=40B$GO TO 400                          J
34    IA=76B$IB=10B$IC=24B$ID=42B$IE=00B$GO TO 400                          K
35    IA=76B$IB=02B$IC=02B$ID=02B$IE=02B$GO TO 400                          L
36    IA=76B$IB=20B$IC=10B$ID=20B$IE=76B$GO TO 400                          M
37    IA=76B$IB=20B$IC=10B$ID=04B$IE=76B$GO TO 400                          N
38    IA=34B$IB=42B$IC=42B$ID=42B$IE=34B$GO TO 400                          O
39    IA=76B$IB=50B$IC=50B$ID=50B$IE=20B$GO TO 400                          P
40    IA=34B$IB=42B$IC=46B$ID=36B$IE=01B$GO TO 400                          Q
41    IA=76B$IB=50B$IC=50B$ID=54B$IE=22B$GO TO 400                          R
48    IA=IB=IC=ID=IE=0  $GO TO 400                                          BLANK
50    IA=72B$IB=52B$IC=52B$ID=52B$IE=56B$GO TO 400                          S
51    IA=40B$IB=40B$IC=76B$ID=40B$IE=40B$GO TO 400                          T
52    IA=76B$IB=02B$IC=02B$ID=02B$IE=76B$GO TO 400                          U
53    IA=70B$IB=04B$IC=02B$ID=04B$IE=70B$GO TO 400                          V
54    IA=76B$IB=04B$IC=10B$ID=04B$IE=76B$GO TO 400                          W
55    IA=42B$IB=24B$IC=10B$ID=24B$IE=42B$GO TO 400                          X
56    IA=40B$IB=20B$IC=16B$ID=20B$IE=40B$GO TO 400                          Y
57    IA=42B$IB=46B$IC=52B$ID=62B$IE=42B$GO TO 400                          Z
400   I(L+1)=IA
      I(L+2)=IB
      I(L+3)=IC
      I(L+4)=ID
      I(L+5)=IE
      I(L+6)=0
      L=L+6
      IF(L.LE.248) RETURN
58    BUFFER OUT(1,1) (I(1),I(L))
581   IF(UNIT,1)581,582
582   CONTINUE
      L=0
      RETURN
      END
```

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Research Laboratory Washington, D.C. 20390	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

THE THEORY AND CONSTRUCTION OF MULTIVARIATE GAUSSIAN SURFACES

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Final report on one phase of continuing NRL problems.

5. AUTHOR(S) *(First name, middle initial, last name)*

Luise S. Schuetz and George G. Zipfel, Jr.

6. REPORT DATE August 31, 1973	7a. TOTAL NO. OF PAGES 58	7b. NO. OF REFS 7
8a. CONTRACT OR GRANT NO. NRL Problems S01-40 and S01-47 b. PROJECT NO. RF 52-552-402-4070 c. SF-54-552-010-14317 d.	9a. ORIGINATOR'S REPORT NUMBER(S) NRL Report 7588 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Department of the Navy Office of Naval Research Arlington, Va. 22217

13. ABSTRACT

A procedure has been developed for the construction of random rough surfaces with statistics which approximate the ideal of multivariate Gaussian statistics with a prescribed spectrum. Estimates of the deviation of the sample surfaces from the ideal ensemble statistics are made. The procedure uses a fast-Fourier-transform algorithm to produce a numerical surface which is then used to construct a physical realization of the surface via a numerically controlled milling machine. The resulting surface is one which can be used in scattering experiments, and the degree of conformity of its statistics to those of the ideal is well understood. The use of these surfaces will eliminate a major source of uncertainty in the interpretation of controlled experiments.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

Security Classification

Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Random rough surfaces						
Multivariate statistics						
Random surface algorithm						
Numerically controlled milling						

DD FORM 1 NOV 65 1473 (BACK)
(PAGE 2)

Security Classification

*U.S. GOVERNMENT PRINTING OFFICE: 1973 542-165/Z83 1-3