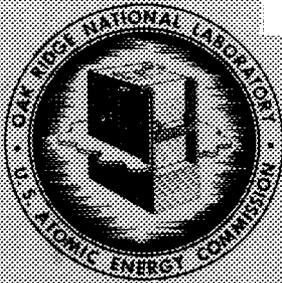




3 4456 0513460 9

CENTRAL RESEARCH LIBRARY
DOCUMENT COLLECTION

1



OAK RIDGE NATIONAL LABORATORY

operated by

UNION CARBIDE CORPORATION
NUCLEAR DIVISION



for the

U.S. ATOMIC ENERGY COMMISSION

ORNL - TM - 2018

CALCOMP PLOTTING OF X-RAY POLE FIGURES

Gordon R. Love

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
DOCUMENT COLLECTION

LIBRARY LOAN COPY

DO NOT TRANSFER TO ANOTHER PERSON

If you wish someone else to see this document, send in name with document and the library will arrange a loan.

NOTICE This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

ORNL-TM-2018

Contract No. W-7405-eng-26

METALS AND CERAMICS DIVISION

CALCOMP PLOTTING OF X-RAY POLE FIGURES

Gordon R. Love

JANUARY 1968

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee
operated by
UNION CARBIDE CORPORATION
for the
U.S. ATOMIC ENERGY COMMISSION



3 4456 0513460 9

CONTENTS

	<u>Page</u>
Introduction	1
Program POLE	4
Subroutine INPUT	6
Subroutine POLECAT	14
Subroutine SKETCH	15
Discussion and Summary	19
Conclusions	28
Acknowledgments	28
Appendix A. Program POLE	29
Appendix B. Subroutine INPUT	31
Appendix C. Subroutine POLECAT	34
Appendix D. Subroutine SKETCH	36

CALCOMP PLOTTING OF X-RAY POLE FIGURES

Gordon R. Love

INTRODUCTION

A quite general problem faced by the experimentalist is the publication of intrinsically three-dimensional plots of data in but two dimensions. A solution to this problem may be approached in two principal ways. One may try either various kinds of projections or various kinds of constant-parameter cross sections of the data. Projections range from the simple "three views" of the machine drawing through isometric to trimetric projections and ultimately to the pairs of drawings which, viewed in stereo, produce an excellent illusion of three dimensionality. An unfortunate corollary of these projection schemes is that any part of the three-dimensional figure which is not coplanar with the observation plane is more or less badly distorted by projection. Both distances and angles may be affected. Recovery of experimental data in quantitative form from these drawings is made difficult by the very manipulations which make observation of qualitative features easier and, furthermore, these drawings are hard to produce. Because it is relatively easier to construct constant-parameter cross sections and because these do not introduce ambiguity in recovering quantitative data, the latter approach has been preferred by scientists.

One may distinguish two variants of the constant-parameter plots. Drawings of the dependent variable versus one of the independent variables for various constant values of the other variable [$U(X,Y)$ vs X at $Y = Y_1, Y_2$, etc.] will be called "parametric" plots. Drawings of the loci of pairs of independent variables giving constant values of the dependent variable [Y vs X such that $U(X,Y) = U_1, U_2$, etc.] we call "contour" plots. Perhaps the most common of these latter are contour maps that show lines of constant elevation above sea level. Constant values of independent variables may be relatively easy to maintain in the laboratory, and therefore parametric graphs are produced with a minimum of data processing. Only by coincidence do they clearly

reveal significant trends in experimental data. Contour plots almost inevitably require interpolation in the set of experimental measurements and hence data reduction, but they frequently are the best possible compromise between clarity, accuracy, and convenience.

In selected cases, especially when the independent variables display some natural symmetry which is not cartesian, it might be desirable to combine contour plotting with some form of conformal mapping in order to preserve a facsimile of the experimental symmetry in the reported data. As an archetype for this class of data representations, we cite the x-ray pole figure.

Typically, a pole figure for a polycrystalline solid is determined by examining the reflected x-ray intensity for a selected crystallographic plane as a function of angular position. In ideal presentation format, then, data on x-ray reflection intensity might be presented in a quasispherical figure in which the various sphere radii correspond to reflected intensity and the angular positions of the radii represent the angles of the observations. In more practical terms, the angular coordinates of a given observation may be reduced to coordinates in a plane by use of the conventional stereographic projection and the variation in intensity may be represented in the cartesian space by drawing selected iso-intensity contours.

To guarantee adequate determination of the angular variation of x-ray intensity, that is, to assure that no peaks are missed, a large number of individual observations are made. Usually intensity measurements are made continuously along a set of relatively closely spaced paths and the positions along that path of preselected values of intensity are recorded. However, automatic conversion of the digital data is not continuous but yields numerical data taken at intervals; the raw data for pole-figure construction are recorded as an array of discrete points. Typically, over 900 such data points are incorporated into a single figure.

The crudest possible way to locate a line of constant intensity in such data is to note that it must pass between, and only between, points of higher intensity than it on one hand and points of lower intensity than it on the other. Because of the large number of data points

incorporated in a single pole figure, this criterion and "eyeball interpolation" have been used for decades to construct pole figures of satisfactory accuracy. Even so, the construction of pole figures is time-consuming and deadly boring and, therefore, a fit problem to reduce to machine-processing methods. A secondary advantage of machine data processing in this case is that the machine is able to do considerably more careful interpolation among the data should it ever be required or warranted.

The problem of machine construction of x-ray pole figures is in reality two problems. First, it is necessary to enter and/or generate within the machine a very large number of single items of data. Typically, the x- and y-coordinates of each of over 900 points in a stereographic projection of the data must be calculated and the value of intensity at each point must be read from input. Also included in this step should be provisions for adjusting the observations, specifically for background intensity corrections, normalizing the observations to a multiple of the equivalent reflection from a random sample, scaling the output to a convenient size for examination and/or publication, etc. Second, the machine must construct pole figures summarizing the input data in a form useful to the user; in particular it must draw contours that approximate as well as possible actual lines of constant reflected x-ray intensity in conventional stereographic projection. Depending on the scale of the drawing, the accuracy of the data, and the whim of the programmer, it might be advisable to construct a relatively large number of iso-intensity contours, rotate the constructed figures about reference axes, change the labeling of maxima and minima, and otherwise modify the output.

The program to be described here is, I feel, an adequate solution to the problem outlined above. It is, in addition, capable of accepting data in a large variety of input formats; that is, a large number of geometries of the data-taking procedure or a large number of possible paths through the experimental space. While it does not produce contours continuous in first derivative and therefore it might offend the purist, it at least does not permit intersection of lines of different intensity

or any similar offences against the necessary properties of contour curves. Furthermore, it operates with an absolute minimum of auxiliary control cards, can readily be adapted to automatic data-taking equipment, and is relatively economical of computer time.

I shall discuss the subroutines in roughly the order in which they are used by the program and in barely sufficient detail to make the Fortran listings, included as appendices, readable. Systematic optimization has not been attempted; careful estimates of computation time are not available; the specific language used throughout is Fortran 63; but there are no known incompatibilities with Fortran IV. The program has run in this laboratory for pole figure construction using the CDC 1604 and the Calcomp plotter for approximately one year without incident. To illustrate various features of the program, selected computer-drawn plots are included later in this report.

Program POLE

The main program of the package for drawing x-ray pole figures is POLE. It has two significant functions: it converts the angular positions of each data point from coordinates on a hemisphere to x- and y-displacements in a plane by the trigonometry of the stereographic projection; it also acts as a clearing house and traffic cop among the other subprograms of the package. Several options are available to the user of the program, and these are reflected in the multiplicity of possible paths through this program. Though we defer detailed description of the options, we can identify them as: (1) multiple data sets, (2) choice of either the Normal Direction or the Rolling Direction as principal reference direction, (3) a numerical dump of the data in pole figure format, and (4) variable number of iso-intensity contour lines. A single punched-card input record is read by POLE statement 90, which contains a 32-character title in columns 4-35 and four integer constants NDEX, KTYPE, N4, and IK(7). If NDEX, columns 1-3, is negative, the program ends normally. Zero or positive values of NDEX imply further data, and each completion of the entire program returns to statement 90, allowing option (1). If KTYPE, columns 36-45, is zero or negative, the

data to be entered subsequently will be assumed to be taken with the Rolling Direction as axis of rotation ("R.D.-data"). In this case, the random intensity will be calculated in the program. If KTYPE is greater than zero, the data is assumed to be taken with the Normal Direction as rotation axis ("N.D.-data") and the value of KTYPE will be used for random intensity. If N_4 , columns 46-55, is zero or negative, a numerical dump of the input data will not be made; if positive, option (3) will be exercised. Finally, the number of iso-intensity contours, up to ten, to be plotted is entered in IK(7).

Common arrays are established which ultimately will contain the values of observed intensity, the x- and y-coordinates of each observation, and a number of internally necessary integers. The program then calls upon subroutine INPUT to read the set of intensity data and to generate the two angles which describe the location of each datum. INPUT returns to POLE only sufficient data for one plot. Typically this is just half of all data, and it is assumed that these data are from the "Front Side" of the hemispherical figure. The program calls upon POLECAT to construct a semicircular field of radius RO inches and then calculates the projection of the data onto the plane of the plot.

Two different sets of trigonometric relations are necessary to project the data onto the plane of the plot because the origin of coordinates in the spherical coordinate system depends on whether the data are taken represent R.D.-data or N.D.-data. In the former case, the point at which both angular coordinates are zero is the rolling direction; in the latter case, that point is the normal direction. Since, in either case, we prefer the plotted figure to present the rolling direction as the "North Pole" of a semicircle centered on the normal direction, two separate conversions are required. The relations are:

(1) for R.D.-data:

$$X = RO \cdot \cos \alpha \cdot \sin \phi \cdot \Delta$$

$$Y = RO \cdot \cos \phi \cdot \Delta$$

where

$$\Delta = (1 - \sin \phi \cdot \sin \alpha) / (\sin^2 \alpha \cdot \cos^2 \phi + \cos^2 \alpha) .$$

(2) for N.D.-data:

$$x = R0 \cdot \tan (\phi/2) \cdot \cos \alpha$$

$$y = R0 \cdot \tan (\phi/2) \cdot \sin \alpha .$$

In either case, ϕ represents the polar angle measured from the reference direction, α is the azimuth angle measured clockwise from the direction labeled Transverse Direction in the plots, and R0 is the diameter of the figure desired.

If an all-number dump of the experimental data is requested, option (3), it is made in PLOT. In this case, as in all cases in which the plotting subroutines are applied systematically to the data arrays, a specialized scanning loop is used. The loop forces the plotting routine to examine increasing values of I for odd-numbered values of J and decreasing values of I for even-numbered values of J. This scanning procedure eliminates long traverses of the figure with the pen in the "up" position since it always considers adjacent points in the data array.

The balance of the program inserts titles on the figures, labels the data "Front Side" and "Back Side" appropriately, reexamines the input list for multiple data sets, times itself using ICLOCKF, and records the total machine time per plot. The central variable to correct sequencing of the subroutines and execution of the labeling steps is IK(4). A summary of the variable names used in PLOT, with brief definitions of their function, is given together with the Fortran listing of the full program in Appendix A.

Subroutine INPUT

This subroutine reads the raw values of reflected intensity, stores them in arrays in the machine for use by the other subroutines, and generates from auxiliary input information the pairs of coordinates that locate each intensity observation in terms of the experimental coordinate system. The relationships that are established within the computer among the data entries, that is, among both the position and intensity variables, are crucial for the subsequent success of the program. They merit careful study here.

The simplest form of parametric data taking, in concept, is based on taking each of the variables within the experimenter's control (in this case the position variables) and treating them as independent variables. One may then select a number of values of each variable arranged in some serial order such that $X_{i+1} > X_i$, etc., and observe the dependent variable, the intensity, for every value of Y_j at each value of X_i . These coordinates define a natural rectangular data array, and there is an observation (data point) at every point $U_{i,j}$ within that array. This data-taking scheme is very common in practice and has a number of properties that make it convenient for subsequent data processing. The two most important properties are: (1) the subscripts of each data point serve to link it directly and unambiguously with the independent variables for which it was determined; and (2) points which are stored in adjacent locations, that is, have subscripts differing by unity, are points which are "adjacent" in the experimental space, that is, have incremental differences in one or the other of the independent parameters. Both these properties are necessary to the construction of x-ray pole figures using the techniques of this program. However, the form of data taking which has been found most convenient in this laboratory does not allow completely independent selection of the two-position parameters and is not therefore directly compatible with the simple scheme for data storage outlined above.

The pole-figure goniometers in use in this laboratory are designed for scanning of reflected intensity while both angular position variables, α and ϕ , are varied continuously. The position variables are therefore no longer truly independent of one another and it is not possible to construct storage arrays in the computer in the fashion described above. However, if a value of intensity is recorded at constant intervals of time (since the rate of change of both position variables is constant) then there are sufficient regularities in the data-taking process to permit simple, useful storage of the data. Figures 1 through 3 are schematics which help to illustrate how this is done. Figure 1 shows in stereographic projection how data are taken for N.D.-data. From arbitrary starting points (usually $\phi = 0$, $\alpha = 0$ and 180° but here taken to be $\phi = 30^\circ$, $\alpha = 30$ and 210° for clearer illustration) the scanner proceeds

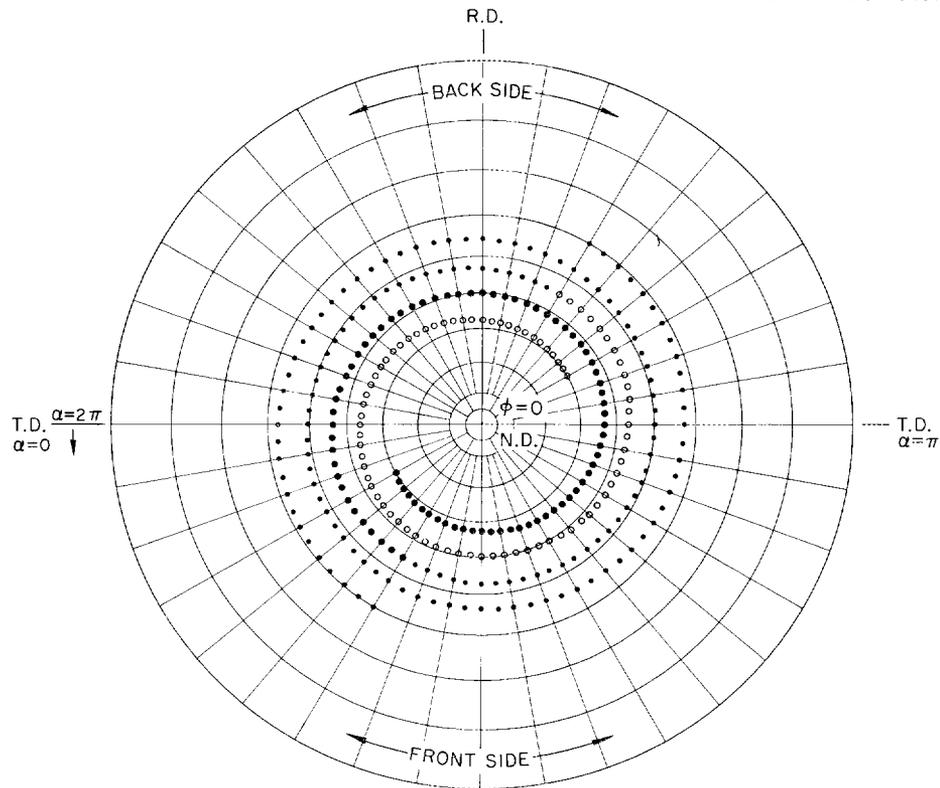


Fig. 1. Schematic Typical of Data Taken for Sheet Textures.

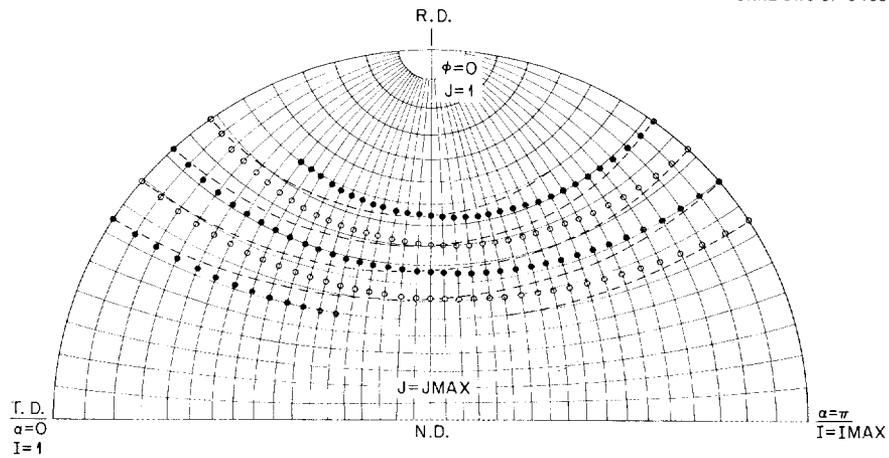


Fig. 2. Schematic Typical of Data Taken for Conventional Textures.
 "Front Side" shown bold; "Back Side," dotted.

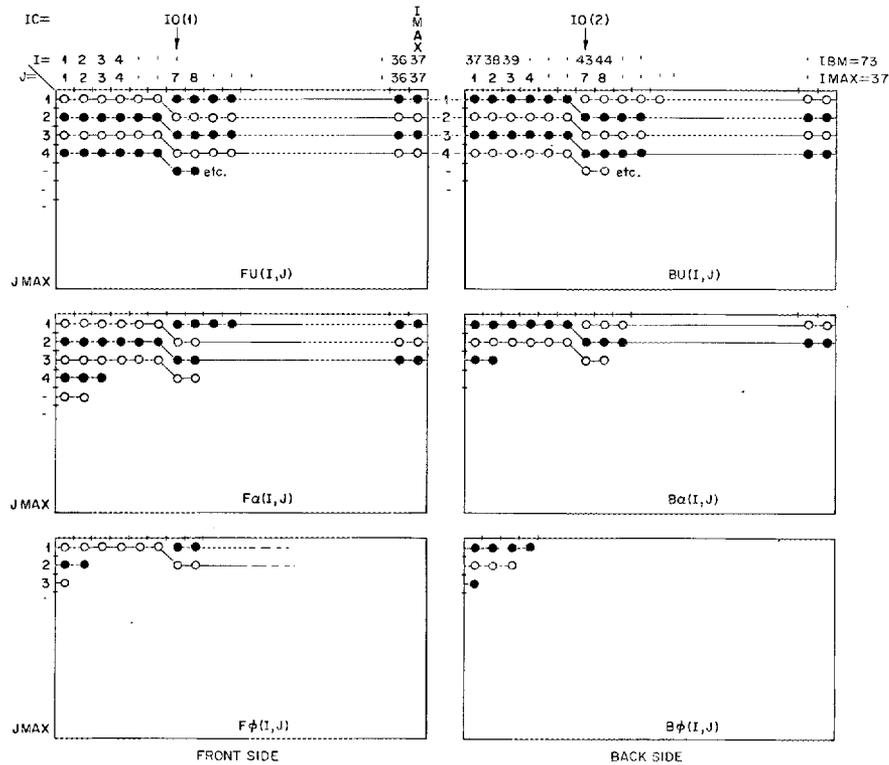


Fig. 3. Schematic Typical of Data Storage in Computer Memory. The numerical values chosen for $IO(i)$, and $IMAX$ correspond to those that would apply for either the Fig. 1 or 2 data-taking schemes.

along one or more (here two) shallow spiral paths and records single data observations at regular intervals of time. In this schematic, data are recorded at 5° intervals in α and 0.2° intervals in ϕ ; the total increase in ϕ is 7.2° per circuit and, as there are two scans, the separation of the traces in the ϕ -direction is 3.6° . Figure 2 shows the same data-taking procedure as it might appear for R.D. data. To emphasize that the differences between Figs. 1 and 2 are merely matters of the choice of projection and location of the point $\phi = 0$, we carefully preserve in both figures the number of traces, their starting points, and the relative position of successive data points on each trace.

The point to be made by comparison of the two figures is that, from the point of view of data storage and handling, they are basically identical. A given "row" of data, the string of data for $0 < \alpha < \pi$,

contains a constant number of single observations. Further, if we define a "column" as the sub-set of points having approximately the same value of α (in our schematic, identical values of α), then the "columns" in each array are also of identical length. In fact, if we assign the subscript \underline{I} to the position of an observation counting from left to right along one "row," and the subscript \underline{J} to "column" position from top to bottom (or from $\phi = 0$ to increasing ϕ), we may construct in computer memory rectangular arrays, shown schematically in Fig. 3, in which there is a one-for-one correspondence between each storage location and each individual datum. For a single pole figure drawing, it is necessary to construct three such arrays in machine memory — one containing intensity values and two others containing each of the position values. The method of storing data in the arrays preserves the property of "adjacency" among the experimental data; a single pair of subscripts, I and J , identifies in each array exactly the corresponding three variables which define a single observation. We have therefore preserved both the consistency of subscripting and the "adjacency" of data storage which are so useful in subsequently constructing the pole figure.

In the next several paragraphs we describe in considerable detail the decision-making steps involved in defining the limits of the arrays in machine memory, the techniques for avoiding storage of multiple data entries in single locations, and the unit logical steps of data entry. A test of the adequacy of our solutions to these problems is given by the later figures and the average machine time spent in data entry. This will be discussed later.

The storage arrays are defined using the following auxiliary information read from punched cards: ADEC, the α -increment between successive points; BDEC, the ϕ -increment; NTRACES, the number of scans made by the x-ray camera; KMAX, the total number of data points to be used from a single scan; and NTRACES values of ALPHO(K) and PHIO(K), the α - and ϕ -coordinates on the first points on each of the traces. The integer width of a single plot, that is, the maximum value of I subscript, is

$$IMAX = \pi/ADEC ,$$

except that if this division is exact and if the traces begin at an integral multiple of ADEC it is necessary to increase IMAX by one to retain all the data. The maximum value of the J subscript is

$$JMAX = (NTRACES * KMAX)/(2 * IMAX) .$$

Notice that doing integer arithmetic in this way truncates the arrays actually used for plotting to the largest available rectangular array; if the final points of an individual trace do not end just at the starting point of a trace, they are stored correctly, but will not be recalled by the plotting program. This loss of data appeared a small price to pay for the programming simplicity it introduced.

For each of the NTRACES values of ALPHO(K) an appropriate value of the starting point for the I-subscript is calculated according to

$$IO(K) = ALPHO(K)/ADEC + 1 .$$

Beyond their obvious use as the first subscript of a string of stored data values, these integers have an additional critical function. Each time that the experimental trace passes one of these values of I it is "eclipsed" by another trace, that is another trace (possibly itself) now lies between it and the "North Pole," the point $\phi = 0$. We have agreed to consider the J-subscript to indicate the number of traces "south" of the point $\phi = 0$; therefore each time one of the points $I = IO(K)$ is encountered, it is necessary to increment J to get the stored values "out of the way" of the values to be stored for the other trace. This is shown in schematic in Fig. 3. We have retained the geometry of point location and separation shown in Figs. 1 and 2, and we intend each symbol of Fig. 3 to indicate that a data value from that trace would be entered into that location of the rectangular array.

In the simplest case of a single spiral scan, the working of this sorting process is trivially obvious. The data are read in successively for one complete circuit of the sampling hemisphere and J is held constant while I is incremented by one between each point. After one circuit, when I becomes again equal to IO(1) (there is only one value of IO), J is incremented by one and another circuit of the figure is stored in the next row of the two-dimensional array. It may not be so

obvious, but it is also true that any number of interpenetrating scans can be handled with no additional logical complications. Of course, if the pitch of the spiral scan is very high, the method becomes logically cumbersome; for this reason the storage arrays for ALPHO and IO are equipped to accommodate no more than ten separate traces.

The unit operation of data entry is then as follows: All of the intensity data for a single trace ($KMAX \leq 1500$ entries) are read from input into temporary storage in GUT. ALPHA and PHI are set initially equal to ALPHO(K) and PHIO(K); for the K^{th} trace, I is set to IO(K) and J to 1 (an exception is discussed below). The value of ALPHA is compared with π and 2π , respectively, to determine the quadrant of the sphere into which the data fall, "Front Side" or "Back Side," as illustrated in Fig. 2, and the appropriate storage arrays are selected. A single value of intensity is read into its array and, immediately, the appropriate values of ALPHA and PHI are stored in their arrays according to

$$FR(I,J) = ALPHA ,$$

$$FZ(I,J) = PHI ,$$

$$FU(I,J) = GUT(K) .$$

Then I is incremented by one, ALPHA by ADEC, PHI by BDEC, K by one (to select the next variable in the buffered input), I is compared with each of the IO(K) in turn to determine whether J should be incremented and the above procedure is repeated from the point of comparison of ALPHA with π and 2π .

There are other complications which have been passed over for expositional simplicity. The I-subscript increases from 1 to IMAX twice during a complete revolution; once as the trace crosses the "Front Side" array, again as it crosses the "Back Side." So that the beginning points of the several traces are correctly identified, it is necessary that an additional integer variable, IC, be used for comparison with the IO(K). The range of IC is just twice that of I. Similarly, although ALPHA may have all values from 0. to 2π , the plots of data representing the "Back Side" are "right handed" only for values of α between 0. and π . Therefore entries in the "Back Side" arrays are

$$BR(I,J) = ALPHA - \pi; BZ(I,J) = PHI; BU(I,J) = GUT(K) .$$

Data taken at $\alpha = 0$. and π properly exist in both the "Back Side" and "Front Side" arrays. When this situation arises, the program expands the array storage to accommodate the repeated variable (the last entry in any row of the front side array becomes the first entry in the back side array, etc.). The details of this replication operation are best revealed directly in the Fortran listing of this subroutine, Appendix B; the critical statements are those defining JB and JC together with the four statements beginning at statement No. 14. A word of caution here: The decision to expand the array capacity is based on the first trace alone. If any point on that trace coincides with a limit of the figure, the arrays will be expanded and the last entry of each row will be duplicated whether or not it coincides with an edge of the plot. So long as the density of data observations is high, this will introduce only small distortions in the plots.

The two other tasks performed by this subroutine are background correction and normalization of intensity. Background is measured by setting 2θ off the Bragg peak and either making an additional spiral scan or by sampling background at several values of α and ϕ . In either event, the background correction is made by simple subtraction of that observed background intensity nearest the particular data point and the background varies sufficiently slowly that this introduces only trivial errors. In practice in this laboratory the background intensity varies so slowly that it has proved to be more convenient to enter it in modified fashion; on one or more punched cards, containing up to eight pairs of numbers, are entered fixed-point numbers representing the background correction and integers representing the maximum subscript in the input array, GUT, which has that value. The array GUT, filled with KMAX values of background intensity, is then used term-by-term to make the background subtraction. Up to 64 different values of background correction may be read.

Normalization of intensity is performed in two different ways depending on the data-acquisition method. For textures determined from spherical samples, where data are presumably available from an entire hemisphere, the random intensity is determined by elementary numerical integration of $U \sin \phi \, d\phi$ using the trapezoidal rule and the available

(background corrected) input data. Then each data point is normalized to a multiple of this random intensity. For sheet textures, since data are not available for an entire hemisphere, an external normalization constant must be supplied via KTYPE in PLOT. Typically, this is determined from measurements on a powder sample in the same x-ray geometry but, where this value is not available, we commonly enter a dummy value such that the final plots are in arbitrary units from 0. to 10.

In Appendix B, along with its Fortran listing, we list the variable names used in INPUT together with their physical significance as defined in this text.

Subroutine POLECAT

This subroutine performs three simple functions. It constructs the semicircular field on which the plot is to be drawn and labels it with the directions which are reference direction on the x-rayed sample. It also locates and labels relative maxima in the numerical input data as adjusted and normalized by INPUT. Finally, it reads the values of intensity for which iso-intensity contours are to be constructed and calls SKETCH to do the plotting. A quirk of the identification of the local maxima has proved to be useful and deserves some additional explanation.

To avoid discriminating against local maxima defined by multiple data points, points which are at least equal to all their neighbors are tentatively defined as maxima. This means that large-area local minima containing at least one point equal to all of its nearest neighbors will also be identified as "maxima." In determining the maxima, therefore, we count the number of equal nearest neighbors of each maximum and, where there are any, we label only a few of all the identical points. The decision to label a given maximum is also based on the ratio of that maximum to the first iso-intensity value entered; only if that maximum is greater than $1/2$ UISO(1) will it be recorded. As for the preceding segments the correlation between variable names in this subroutine and physical parameters of the plotting package is given, together with the Fortran listing, in Appendix C.

Subroutine SKETCH

The problem of constructing a set of iso-intensity contours throughout the whole set of observed data is formidable. Happily, it simplified to the problem of constructing the set of contours passing among a subset of only three data points and the corollary problem of choosing those three points in a consistent sequence that finally covers the whole set of observed data. The logical steps in this process are considerably simplified by the necessary properties of iso-value lines; they are continuous and do not branch.

Consider the "triangle" defined by three pairs of subscripts, I and J, such that each of its three corners represents a combination of $U_{ij}(X,Y)$, X_{ij} , and Y_{ij} . If any two corners of the "triangle" have values such that U at one is larger than, and U at the other is smaller than the value of UISO corresponding to a given iso-intensity line, then the line must pass between those points. Further, if the line enters the triangle along one side then it must exit through one of the other two sides; equivalently, if one pair of points "brackets" an iso-value line, then one and only one other pair also "brackets" the line. Obviously, since we use a finite sampling interval, there might exist pairs of points between which the true iso-value line passes twice and which therefore lie both to the same side of the contour. Such circumstances are ignored, that is, a plotted iso-value line passes but once between two points. This constitutes a limit to the resolution of the plotting technique that is more serious than the accuracy and reproducibility of the plotter pen position, for example.

Two simple functions are used in conjunction with this subroutine. The first of these, BRACE, asks whether the third of its three arguments is intermediate in magnitude between the first two of its arguments and returns BRACE = 1 if the answer is affirmative and BRACE = 0 if negative. This function is used, with two adjacent values of U_{ij} and the value of UISO, the contour value, as arguments, to determine whether the contour passes between the points. The second function, GUESS, does simple linear interpolation in terms of one of the position variables to make an estimate of the value of that variable for which the intensity U

equals UIISO. Granted that the intensities at two corners of the triangle are found to bracket a contour line, BRACE = 1, GUESS is called twice to interpolate between the known values of the position variables and thus to determine the x- and y-coordinates of the point through which the contour passes.

When such a point is identified and located, the pen of the plotter is moved to that point, pen up. Then each of the other sides of the conceptual triangle is examined, in turn, and when the point at which the contour leaves the triangle is identified and located, the pen of the plotter is moved to that point, pen down. This is the unit process of drawing the full contour plot. Note that if the iso-value line does not pass through either of the first two sides examined, it is not necessary to examine the third side. Further, note that if the contour does not pass through the first side examined but does pass through the second, it is not necessary to call BRACE to verify that it passes through the third, etc. Each segment of contour line is constructed as an isolated straight line. These lines join to within the accuracy of the plotter, an accuracy of approximately 0.005 in., when the appropriate adjacent triangle is examined.

A set of four triangles is constructed in the machine at each point in the array considered by the subroutine. A schematic representation of these triangles is given in Fig. 4 which is also the defining figure for the variables. For a given value of the subscripts I and J, defining a given storage point for each of the position variables and the intensity variable within machine memory, we define a "box" by the following relations:

$$(1) = I, J ; (2) = I+1, J ; (3) = I+1, J+1 ;$$

$$(4) = I, J+1 ; (5) = 1/4 \sum_{i=1}^4 (i) .$$

The point (5) is the mean of the first four points. Note that this is a relation among subscripts and, in fact, at each point of the "box" there exist defined values for each of the intensity and position variables necessary to make the plot. The system of numbering triangles

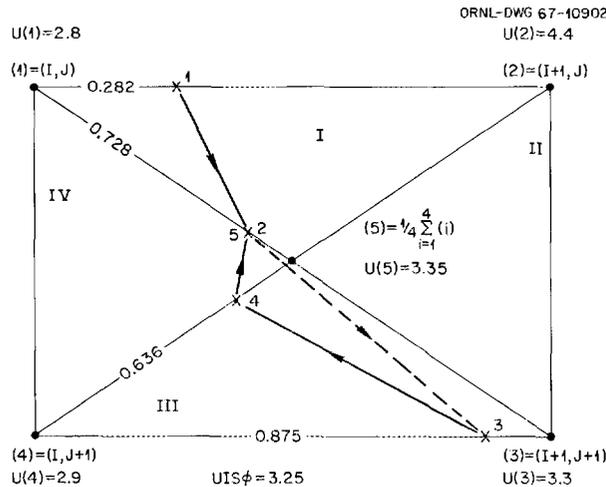


Fig. 4. Schematic of Minimum Logical Unit, "Box," for which Plot can be Constructed. Numerical values chosen for illustration only. Detailed explanation in text.

and points illustrated in Fig. 4 makes it possible to examine the component triangles in systematic fashion. Rather than attempt to decipher the numbering system, merely note that each triangle is examined in clockwise order about the common center point, point (5), and each side of each triangle is examined in counterclockwise order beginning at the side opposite point (5).

The sequence of calculations which produces a plot is best described by example referring to Fig. 4. When the "box" corresponding to I,J is first examined, values for $U(1..5)$, $R(1..5)$, and $Z(1..5)$ are copied out of the storage arrays and calculated as indicated above. It is at this point that the care exercised in subroutine INPUT to get physically adjacent points into adjacent storage locations bears fruit; because the first four of these points are physically adjacent, the fifth point is intermediate in value between the first four in a fashion that allows us artificially to increase the apparent resolution of the plots. In effect, for plotting purposes, we double the number of usable data points. The program then examines the "triangle" defined by point (5) and each of the outside edges of the "box," in turn, in the sequence defined by the Roman numerals.

For the values of intensity and UIISO indicated on the figure the sequence of calculations is as follows. Subroutine BRACE finds that points (1) and (2) bracket the iso-value line and, by linear interpolation using GUESS, that the contour should pass through a point 0.282 of the distance between R(1) and R(2) (measured in the x-direction) and 0.282 of the distance between Z(1) and Z(2) (measured in the y-direction). The pen is lifted from the paper, wherever it is, and moved to the appropriate x- and y-locations, pen up. BRACE next finds that the points (1) and (5) bracket the contour value and GUESS calculates that the contour line should pass 0.728 of the distance between the appropriate values of R and Z. The pen moves to that point in a straight line, pen down. It is not necessary to examine the side (2) to (5) because, by definition, the contour cannot pass through that side. Entering triangle II, BRACE examines the pairs of points (2) to (3) and (2) to (5) without finding a pair that bracket the contour; it is not necessary to examine the pair (3) to (5). In triangle III, BRACE verifies that the points (3) to (4) bracket the contour and GUESS directs the pen to a point 0.875 of the distance from (4) to (3) in each of the position coordinates. In this case the pen moves in the up position, as indicated by the dotted line. Once BRACE finds that the contour does not pass between the pair of points (3) and (5), GUESS is called immediately to interpolate between points (4) to (5) since the contours must pass through that side. Again the pen moves to that point pen down. Despite the fact that both ends of the segment in triangle IV have been previously identified, the side (1) to (4) is checked by BRACE to verify that the contour does not pass through that side; the side (4) to (5) is checked and the pen moved to the appropriate point, pen up (in this case no motion) and the side (1) to (5) is used, with GUESS to fix the coordinates to which the pen next proceeds, pen down.

These four triangles are examined again for each of the contour values used in the program and appropriate line segments are constructed. Only after all possible contour lines have been constructed for this "box" does the attention of the program shift to a new value of I and J.

The boxes are examined in the serpentine sequence defined in the discussion of POLE, I increasing for J odd and I decreasing for J even. The upper limits on the DO loops which scan the array are one less than the array dimensions so that undefined parameters will not be introduced as box variables. Note that it is an intrinsic property of this plotting scheme that it plots to the limits of the data and then allows the lines to end "in space." This is considered a useful feature in the particular case of sheet textures since the region over which the texture is known is visually defined by the limits of the array of contour lines. Where data exist for an entire hemisphere, the small misfit between the limits of the data array and the limits of the field constructed by POLECAT is not usually objectionable. In this regard, note Figs. 5 through 10 of this report.

Discussion and Summary

Several examples of the figures produced by this plotting package are shown in Figs. 5 through 10. Textures determined from hemispherical samples, Figs. 5 and 6, are the result of four interpenetrating scans spaced at 90° in α (or 3.6° in ϕ) apart. The all-number dump, Fig. 5, shows the helical scanning path most clearly and also illustrates the problems inherent in using such a field as the basis for a hand-drawn figure; in the region near the pole, overlap of the numbers is inevitable. In this figure, as in all plots of the numerical data itself, the numerical information is positioned on the page so that the decimal point in the tabulated value lies at the point for which the observation was made.

The data used to construct sheet textures, Figs. 7 and 8, are taken in a single helical scan with increments of 3.6° in α and 5° in ϕ . The maximum value of ϕ for which data are used is normally $\phi = 75^\circ$, the approximate extreme value of the contour lines in Fig. 8. Fake data were used in Fig. 9 to illustrate that the program can handle other data-taking formats without incident. As is typical in this laboratory for transmission pole figures, the data were considered to have been obtained from ten circular scans taken so that each scan was separated from its

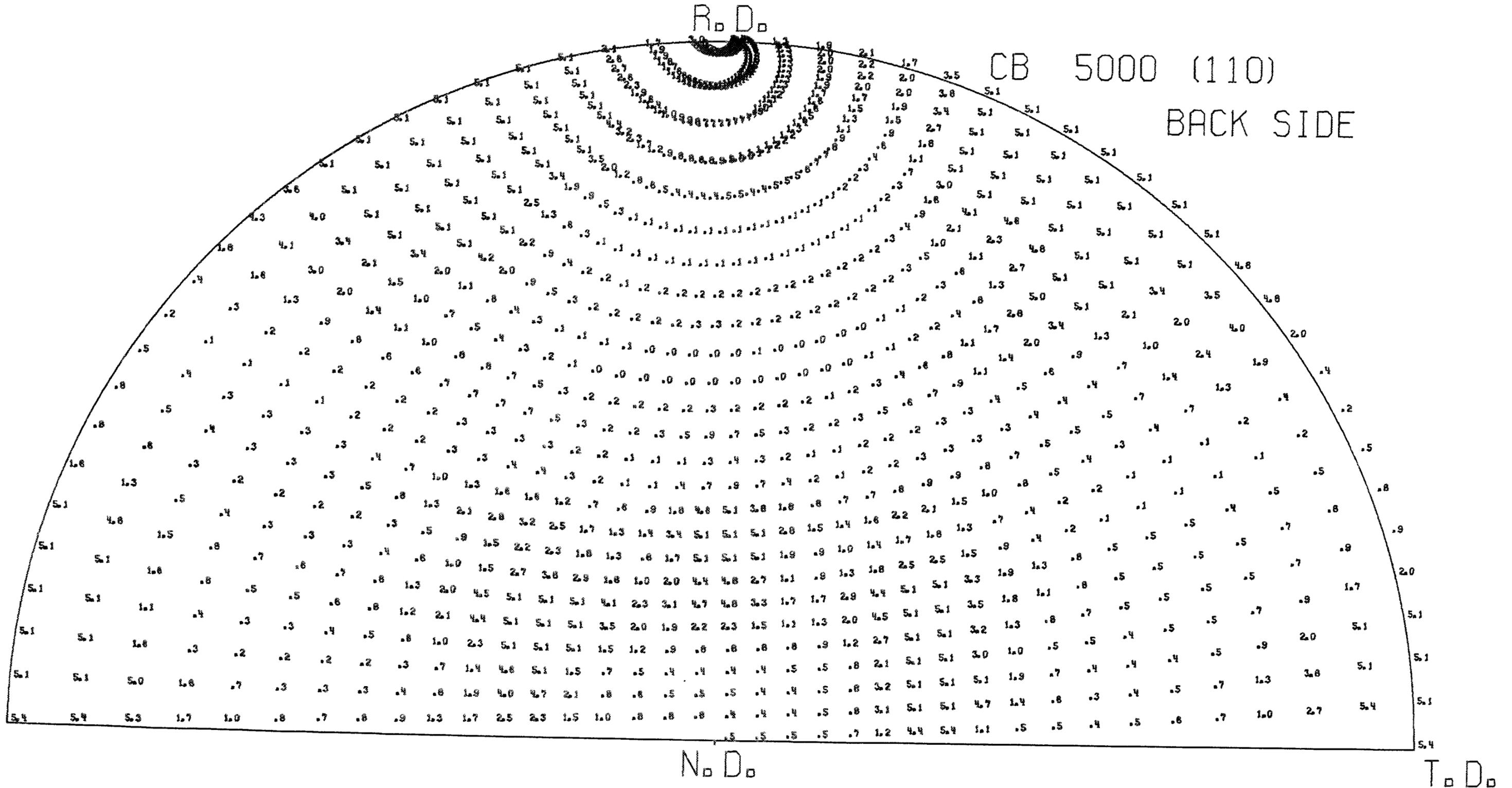


Fig. 5. All Number Dump of the Data from Which an "R.D.-Data" Pole Figure is Constructed, Illustrating in Particular the Spiral Scanning Path.

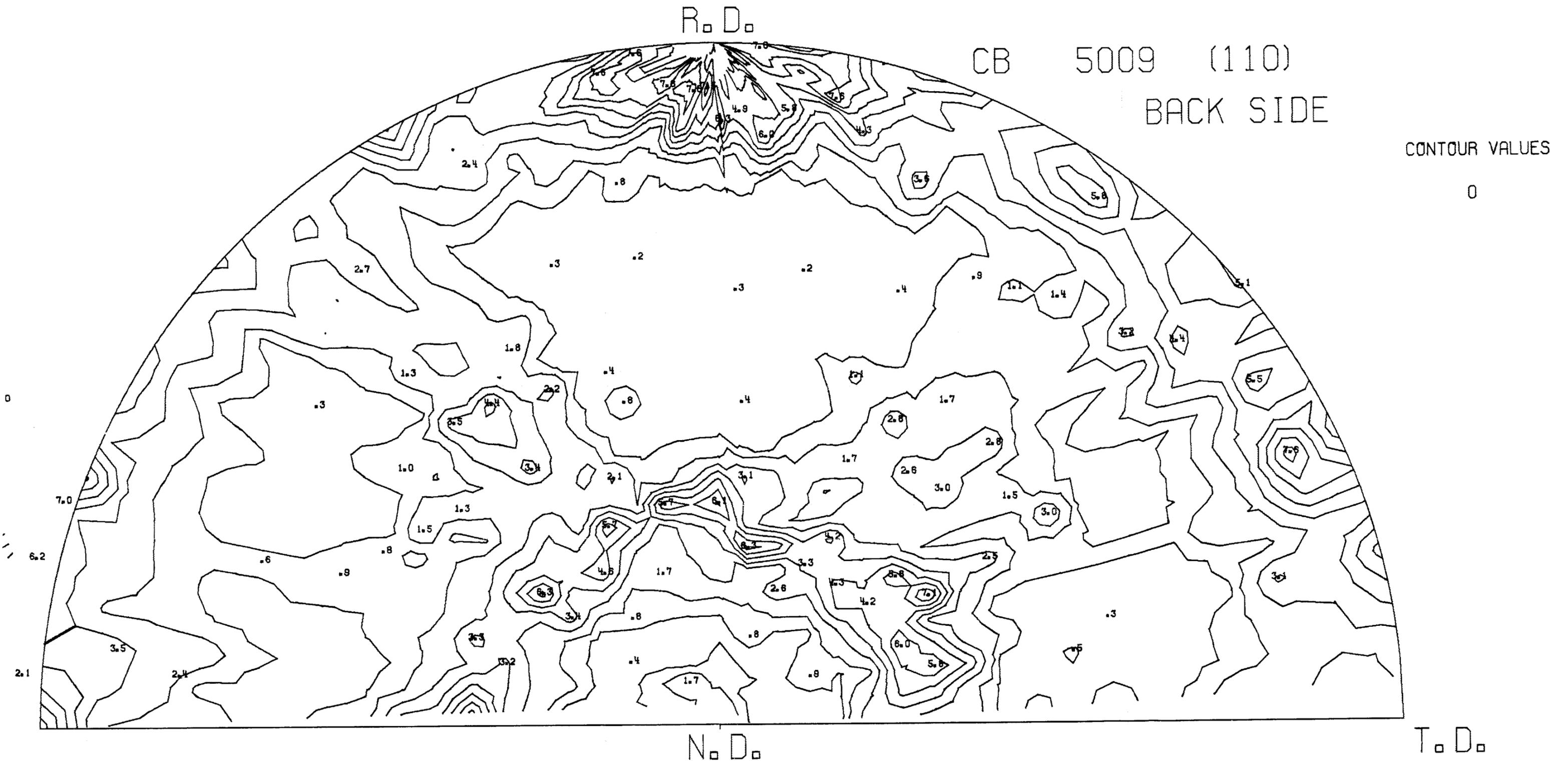
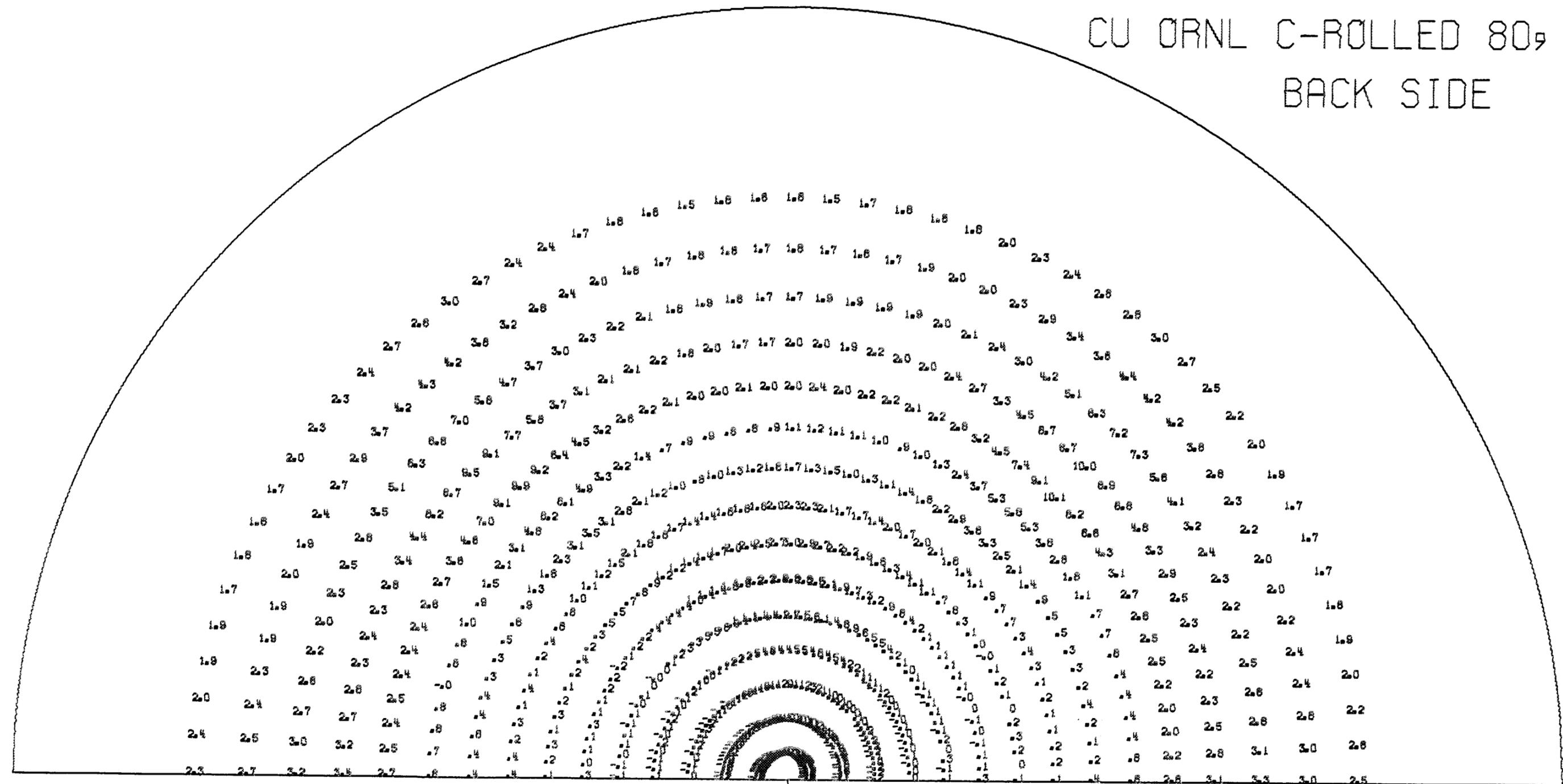


Fig. 6. Pole Figure of Typical Coarse-Grained Sample of the "R.D.-Data" Type.

R. D.

CU ORNL C-ROLLED 80, R. T. (111)
BACK SIDE



N. D.

T. D.

Fig. 7. All Number Dump of the Data from Which an "N.D.-Data" Pole Figure Is Constructed.

R. D.

CU ORNL C-ROLLED 80, R. T. (111)

BACK SIDE

CONTOUR VALUES

9.50

1.00

2.00

3.00

4.00

6.00

8.00

.50

N. D.

T. D.

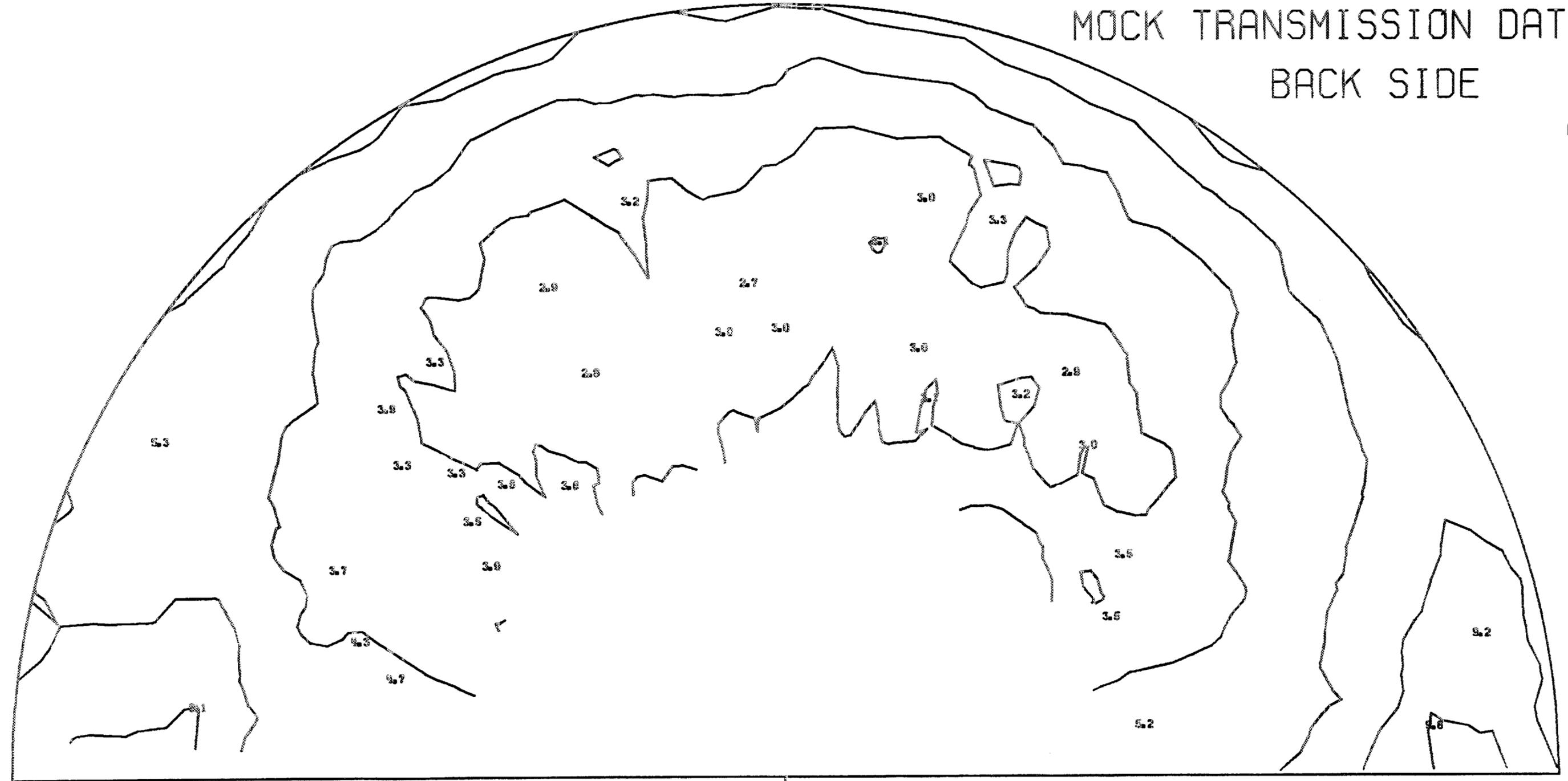
Fig. 8. Pole Figure of Typical Fine-Grained Sample of the "N.D.-Data" Type.

R₀ D₀

MOCK TRANSMISSION DATA (111) BACK SIDE

CONTOUR VALUES

- 0.50
- 1.00
- 2.00
- 3.00
- 4.00
- 6.00
- 8.00
- 9.50



N₀ D₀

T₀ D₀

Fig. 9. Pole Figure Typical of One Obtained from Transmission X-Rays of Sheet.

CU ORNL C-ROLLED 80₉ R₀T₀ (111)
BACK SIDE

CONTOUR VALUES

9.50

8.00

4.00

2.00

1.00

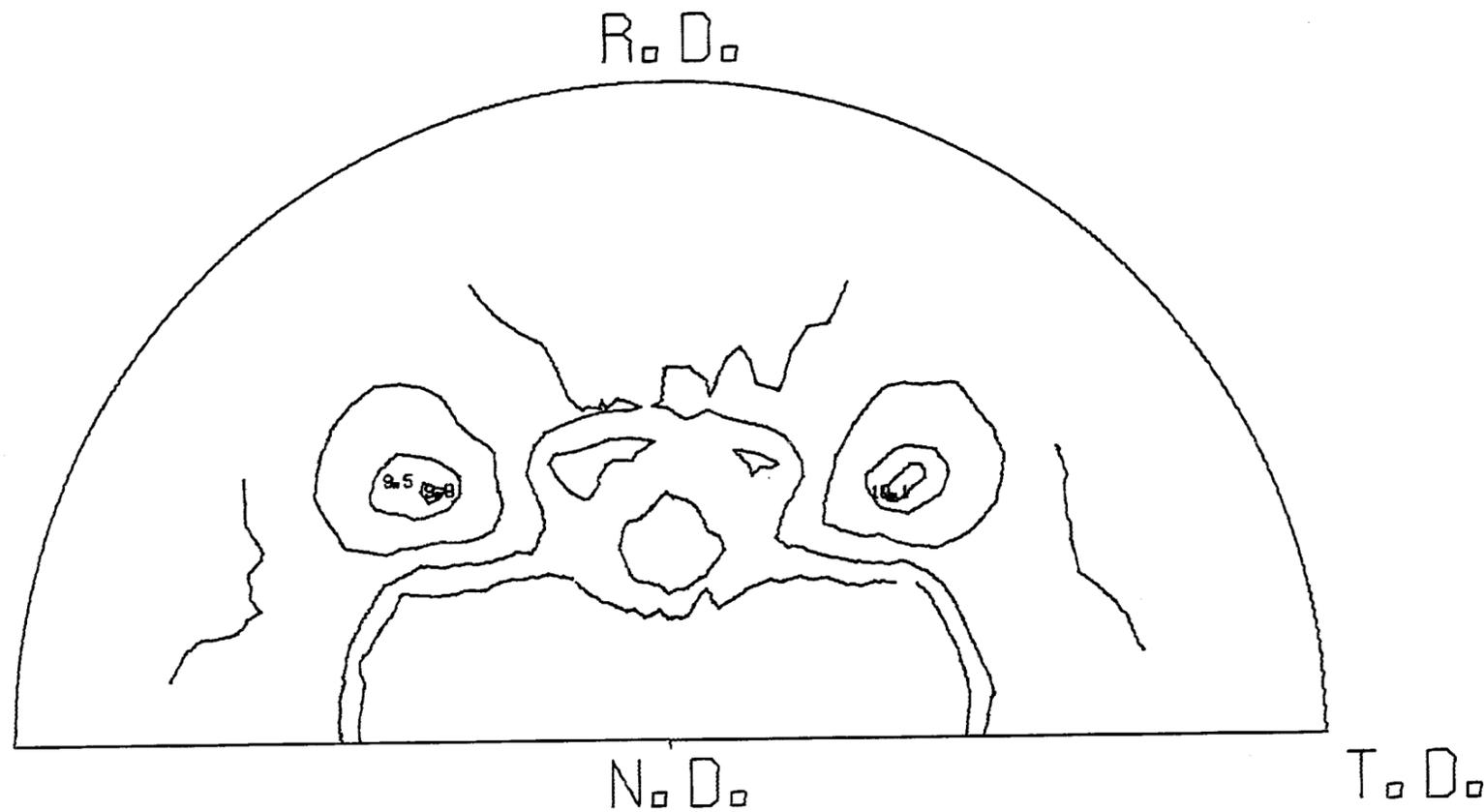


Fig. 10. The Pole Figure of Fig. 8 Reduced to One-Half Size before Drawing.

neighbors by 5° in ϕ and single observations on a scan were separated by 3.6° in α . Again the contours end where the data ended at $\phi = 45^\circ$.

Working drawings of the size included in the gatefolds are highly desirable. They are not satisfactory for preparation of slides or submission for publication. By reducing both the number of iso-value lines and the scale of the plots, as drawn, changes which do not affect the labeling, figures more convenient for publication can be generated as; for example, Fig. 10.

For a "normally" complex pole figure, the total computational time required to convert a string of 1500 data points to two figures of the size shown in the gatefolds (one "Front Side," one "Back Side," no all-number output) is 3.5 min. Of this time: 22 sec elapses during compilation of the binary deck representing the Fortran program; 31 sec are consumed by the manipulations of subroutine INPUT; outlining the field of the drawing, identifying and labeling local maxima in the field, converting the angular position variables to cartesian coordinates, and the miscellaneous operations of POLE and POLECAT require 15 sec per plot; 52 sec are then required for actually generating the instructions that produce each plot. The all-number dump of the input data consumes an additional 55 sec per plot (it is not recommended except in dire cases). Exceptionally intricate figures, as those obtained from coarse-grained crystallized samples, may require up to 75 sec per plot as the computational time is roughly proportional to total line length on the plot. For array storage of up to 3500 data points (convenient array storage in the laboratory has proven to be $IMAX = 50$, $JMAX = 35$ for both "front side" and "back side" arrays), the available storage capacity of the CDC 1604 is just adequate without temporary storage on tape or overlay.

Plotting time on Calcomp plotters averages to 10 min per plot, 20 min per data set, provided no number dump is used.

Table 1 summarizes the kind and format of inputs to the computer and the order in which they are required. In addition to the tape of plotting instructions, the only output is a time record at the completion of INPUT, the entry to SKETCH, and the completion of each plot. The library subroutines (available on demand to users of the CDC 1604 at this location) are also listed in Table 1.

Table 1. Auxiliary Inputs to Program POLE

Card 1:

Columns 1-3	NDEX, an integer
Columns 4-35	RUN, a Hollereith string, the title of the plot
Columns 36-45	KTYPE, an integer
Columns 46-55	N4, an integer
Columns 56-65	IK(7), an integer

These integers described in text, Subroutine POLE

Card 2: Format, five 10-column entries in order;

ADEC, the α -decrement between entries
 BDEC, the ϕ -decrement between entries
 NTRACES, the integer number of traces
 KMAX, the integer number of points to be used per trace
 ALEPH, the diameter in centimeters of the plot to be drawn

Card 3: Format, eight 10-column entries by pairs:

ALPHO and PHIO, the initial points on the TRACES traces

The Data: taken in this listing from auxiliary magnetic tape
 which, in turn, was constructed from an on-line
 Talley punched-paper recorder

Card 4: Format, sixteen 5-column entries by pairs:

BKG and IK, an observed background correction and the maximum
 scan value for which it is applicable

Card 5: Format, eight 10-column entries:

UIISO, the value of intensity for which a contour is to be
 constructed

Note: Cards 3, 4, and 5 may represent more than a single card,
 as necessary

Auxiliary subroutines required from the computer library:

ICLOCKF, references a 1000-cycle clock to time program execution
 PLOT, PLOTS, NUMBER, and SYMBOL, the set of subroutines which
 generate instructions for the Calcomp plotter
 SIN, COS, TAN, the trigonometric subroutines used in performing
 the stereographic projection

CONCLUSIONS

A Fortran 63 program has been written which satisfactorily converts x-ray reflected intensity data from a serial string of numerical values to a contour map, in stereographic projection, of intensity as a function of angular coordinates relative to the specimen.

The principal goals of the program were minimization of the number of auxiliary inputs and construction of high-resolution working figures; both these goals have been satisfactorily met.

A multitude of possible input formats can be used, but all are expected to be simply related to one or more helical scans during which both variables change simultaneously. Both pole figures having the origin of coordinates at the rolling direction and textures in which the origin is the normal direction are acceptable inputs.

The general features of map drawing, location of points bracketing a given value and linear interpolation to approximate its exact position, should be applicable to a wide variety of data-processing problems.

ACKNOWLEDGMENTS

As teachers, willing helpers, and extremely patient listeners throughout this work, I extend grateful appreciation to M. T. Harkrider and N. B. Alexander of the ORNL Mathematics Panel.

APPENDIX A. PROGRAM POLE^a

```

DIMENSION          PHI(50,25),A(50,25),
1 THETA(50,25),IK(20),RUN(4)
COMMON A,THETA,PHI,IK,AX
IK(9)=ICLOCKF(BI)
CALL PLOTS (AX,5000,25)
CALL PLOT(10.,0.,-3)
90 READ 25,NDEX,RUN,KTYPE,N4,IK(7)
25 FORMAT (13,4A8,3I10)
IK(8)=KTYPE
IF(NDEX+1)100,100,6
6 IK(4)=1
92 CALL INPUT(ALEPH)
IK(10)=(ICLOCKF(BI)-IK(9))/60
PRINT 101,IK(10)
101 FORMAT (17H END INPUT, TIME=,I4)
R0=ALEPH/2.54
35 CALL POLECAT(RUN,N4,R0)
IMAX=IK(5) & JMAX=IK(6)
IK(4)=IK(4)+1
IK4=IK(4)
GO TO (34,34,36,34,36)IK4
34 IF(KTYPE)44,44,45
45 DO46 I=1,IMAX
DO46 J=1,JMAX
ASIN=SINF(THETA(I,J))
ACOS=COSF(THETA(I,J))
PTAN=TANF(PHI(I,J)/2.)
THETA(I,J)= R0 *PTAN*ACOS
46 PHI(I,J)= R0 *PTAN*ASIN
GO TO 39
44 DO31 I=1,IMAX
DO 31 J=1,JMAX
ASIN=SINF(THETA(I,J))
ACOS=COSF(THETA(I,J))
PSIN=SINF(PHI(I,J))
PCOS=COSF(PHI(I,J))
DEL=(1.-PSIN*ASIN)/(ASIN**2*PCOS**2+ACOS**2)
THETA(I,J)= R0 *ACOS*PSIN*DEL
31 PHI(I,J)= R0 *PCOS*DEL
39 IF (N4)35,35,33
33 DO32 J=1,JMAX
DO 32 IM=1,IMAX
JM=J-2*(J/2) & IF(JM.GT,0)22,21
22 I=IMAX - IM + 1 & GO TO 32
21 I=IM
32 CALL NUMBER (THETA(I,J)=,I2,PHI(I,J),,07,A(I,J),0,0,6H(F4,I))
36 CALL SYMBOL(3,0,7,5,0.28,RUN,0,0,32)
IF (IK(4)-3)41,41,42

```

APPENDIX A (continued)

```

41 CALL SYMBOL(5.0,6.9,0.28,10HFRENT SIDE,0.0,10)
   GOT043
42 CALL SYMBOL(5.0,6.9,0.28,9HBACK SIDE,0.0,9)
43 CALL PLOT(20.,0.,-3)
   IK(10)=(ICLOCKF(RI)-IK(9))/60
   PRINT 102,IK(10)
102 FORMAT (16H END PLOT, TIME=,I4)
   IK4=IK(4)
   GO TO (35,35,92,35,90)IK4
100 CALL EXIT
   END POLE

```

^aCorrespondence list:

<u>Program Variable</u>	<u>Called in Text</u>	<u>Significance</u>
A	U	Intensity of x-ray reflection
PHI	Y or ϕ^*	y-axis position of observation
THETA	X or α^*	x-axis position of observation
IK		Array for communicating constants between subroutines
RUN	title	Hollereith title of 32 characters
KTYPE	KTYPE	See text

*The same arrays are used, initially, for the coordinates of a single datum in the spherical coordinate system appropriate to the data-taking system and, finally, for the coordinates of that same datum in the cartesian system appropriate to making the plot.

APPENDIX B. SUBROUTINE INPUT^a

```

      DIMENSION FU(50,25),RU(50,25),FR(50,25),BR(50,25),FZ(50,25),
      IGUT(1500),      IK(20) ,BZ(50,25) ,ALPH0(10),PHI0(10),I0(10),
      2BKG(65),LK(65)
      COMMON FU,FR,FZ,IK
      M1=IK(4)
      GO TO (30,30,35,35)M1
30  READ 120, ADEC,BDEC,NTRACES,KMAX,ALEPH
120  FORMAT (2F10.2,2I10,F10.2)
      ADEC=ADEC/57.2957795
      BDEC=BDEC/57.2957795
      READ 102,(ALPH0(N),PHI0(N),N=1,NTRACES)
102  FORMAT (8F10.2)
      DO50 N=1,NTRACES
      PHI0(N)=PHI0(N)/57.2957795
      ALPH0(N)=ALPH0(N)/57.2957795
50  I0(N)=XFIXF(ALPH0(N)/ADEC+.00005)+1
      MAXM=NTRACES+1
      IMAX=XFIXF((3.1415926)/ADEC+.00005)+1
      ITOP=IMAX-1
      IBM=XFIXF((6.2831852)/ADEC+.00005)+1
      AI2=IBM-I0(1) $ AI0=IMAX-I0(1)
      ALIM=ALPH0(1)+AI0*ADEC-3.1415926
      IF(ABSF(ALIM).LT,.001)51,52
51  JB=2 $ IBM=IBM+1 $ JMAX=(KMAX* NTRACES)/(2*ITOP)
      DO 60 N=1,NTRACES
      IF(I0(N)-IMAX)60,61,61
61  I0(N)=I0(N)+1
60  CONTINUE $ GO TO 53
52  JB=1      $ IMAX=IMAX-1 $ IBM=IBM-1
      JMAX=(KMAX*NTRACES)/(2*IMAX)
53  AL0M=ALPH0(1)+AI2*ADEC-6.2831852
      IK(5)=IMAX $ IK(6)=JMAX
      IF(ABSF(AL0M).LT,.001)54,55
54  JC= 2      $ GO TO 56
55  JC= 1
56  DO 100 M=1,NTRACES
      N=M
      READ (4,105)((IDUMP,GUT(K)),K=1,KMAX)
105  FORMAT (17,F6.0)
88  READ(4,104)
104  FORMAT (1X)
      IF(E0F,4)21,88
21  ALPHA=ALPH0(N)      $ PHI=PHI0(N)
      IF (ALPHA-3.1415926)1,2,2
1  IC=0 $ GO TO 3
2  ALPHA=ALPHA-3.1415926 $ IC=IMAX
3  I=I0(N)=1 $ IC=IC+1
      IF(BDEC)40,40,41

```

APPENDIX B (continued)

```

40 J=N-1 $ GO TO 42
41 J=0
42 DO 19 K=1, KMAX
13 I=1+1 $ IC =IC+1
   DO 15 N=1, NTRACES
     IF (IC-IB(N)) 15, 18, 15
15 CONTINUE           $ GO TO 23
18 J=J+1
23 IF (J-JMAX) 10, 10, 100
10 IF (IC-IMAX) 12, 14, 17
17 IF (IC-IBM) 11, 16, 20
12 FU(I,J)=GUT(K) $ FR(I,J)=ALPHA $ FZ(I,J)=PHI $ GO TO 29
11 BU(I,J)=GUT(K) $ BR(I,J)=ALPHA $ BZ(I,J)=PHI $ GO TO 29
20 IC=IC - IBM $ GO TO 17
14 FU(I,J)=GUT(K) $ FR(I,J)=ALPHA $ FZ(I,J)=PHI
   I=0 $ ALPHA = ALPHA - 3.1415926 $ GO TO (29,13) JB
16 BU(I,J)=GUT(K) $ BR(I,J)=ALPHA $ BZ(I,J)=PHI
   IC=0 $ I=0 $ ALPHA =ALPHA - 3.1415926 $ GO TO (29,13) JC
29 ALPHA=ALPHA+ADEC $ PHI=PHI+BDEC
19 CONTINUE
100 CONTINUE
22 K=1
   N=8 $ NI=1
82 READ 103, (BKG(M), LK(M), M=NI, N)
103 FORMAT (8(F5.0, 15))
   DO 80 M=2, N
     IF (LK(M) .LT. KMAX) 80, 81
80 CONTINUE
   NI=NI+8 $ N=N+8 $ GO TO 82
81 N=M
   DO 70 M=2, N
     LK1=LK(M-1) $ LK2=LK(M)
     DO 70 K=LK1, LK2
70 GUT(K)=BKG(M)
     DO 26 J=1, JMAX, NTRACES
       DO 25 I=1, ITOP
         DO 24 N=1, NTRACES
           IF (J+N-1-JMAX) 43, 43, 25
43 BU(I, J+N-1)=BU(I, J+N-1)-GUT(K+ITOP)
24 FU(I, J+N-1)=FU(I, J+N-1)-GUT(K)
25 K=K+1
       DO 27 N=1, NTRACES
         IF (J+N-1-JMAX) 44, 44, 26
44 BU(IMAX, J+N-1)=BU(IMAX, J+N-1)-GUT(K-1+ITOP)
27 FU(IMAX, J+N-1)=FU(IMAX, J+N-1)-GUT(K-1)
       K=K+ITOP
26 CONTINUE
   KTYPE=IK(8)
   IF (KTYPE.GT.0) 62, 63

```

APPENDIX B (continued)

```

62 SUM=KTYPE          $ GO TO 73
63 SUM=0.
   DO 37 J=1,JMAX
   DO 37 I=1,IMAX
37 SUM=SUM+.5*BDEC*((FU(I,J)+FU(I+1,J))*SINE(.5*(FZ(I,J)+FZ(I+1,
I)))+(BU(I,J)+BU(I+1,J))*SINE(.5*(BZ(I,J)+BZ(I+1,J))))
SUM=SUM/(NTRACES*BZ(IMAX,JMAX))
73 DO 71 J=1,JMAX
   DO 71 I=1,IMAX
   FU(I,J)=FU(I,J)/SUM
71 BU(I,J)=BU(I,J)/SUM
72 RETURN
35 DO 36 J=1,JMAX
   DO 36 I=1,IMAX
   FU(I,J)=BU(I,J) $ FR(I,J)=BR(I,J)
36 FZ(I,J)=BZ(I,J)
   RETURN
   END

```

^aCorrespondence list:

Name in Program	Name in Text	Significance
FU	U "Front Side"	Intensity observed for $0 \leq \alpha \leq +80$
BU	U "Back Side"	Intensity observed for $180 \leq \alpha \leq 360$
FR	α "Front Side"	α -coordinate of observation
BR	α "Back Side"	Same
FZ	ϕ "Front Side"	ϕ -coordinate of observation
BZ	ϕ "Back Side"	Same
ALPH ϕ	ALPH ϕ	α -coordinate of first point in a trace
PHI ϕ	PHI ϕ	ϕ -coordinate of first point in a trace
I ϕ	I ϕ	I-subscript of first point in a trace
GUT	GUT	Temporary storage location for input data
ALPHA	ALPHA	Initially, the α -coordinate of an observation stored, as generated, in FR or BR
PHI	PHI	Initially, the ϕ -coordinate of an observation stored in FZ or BZ
ALEPH	RO	Diameter of plots, in centimeters

APPENDIX C. SUBROUTINE POLECAT^a

```

DIMENSION U(50,25),X(50,25),Y(50,25),IK(20) ,IC(50),JC(50)
  I,RUN(4),UIS0(10)
  COMMONU,X,Y,IK
  LI=IK(7)
  IF(N4)14,14,32
32 IK4=IK(4)
  GO TO (12,9,12,8)IK4
14 IF (IK(4)-1)9,9,8
  9 READ I,(UIS0(L),L=1,LI)
  1 FORMAT (8F10.0)
  8 CALL PLOT(0,0,-1,0,3)
  CALL PLOT(0,0,0,0,3)
  R=R0 $ R2=R**2 $ IZ=R/.015-1,
  CALL PLOT(-R,0,0,2)
  J=1
  BX=-R
22 DO 21 I=1,IZ
  BX=BX+0.015
  BY=SQRTF(R2-BX**2)
21 CALL PLOT (BX,BY,2)
  IF(J-1)30,30,31
30 CALL PLOT (0.,R,2)
  CALL SYMBOL (-.35,R+.125,.28,4HR.D.,0.,4)
  CALL PLOT (0.,R,3)
  BX=0.
  J=J+1
  GO TO 22
31 CALL PLOT (R,0.,2)
  CALL SYMBOL (R+.125,-.4,0.28,4HT.D.,0.,4)
  CALL PLOT (R,0.,3)
  CALL PLOT (0.0,0,0,2)
  CALL PLOT (0.0,-.05,2)
  CALL SYMBOL (-0.35,-0.4,0.28,4HN.D.,0.,4)
  IK4=IK(4)
  GO TO (12,13,12,13)IK4
12 RETURN
13 IMAX=IK(5)-1 $ JMAX=IK(6)-1
  UMAX=.5*UIS0(1)
  LMAX=0
  2 DO 20 J=2,JMAX
  DO 20 II=2,IMAX
  JJ=J-2*(J/2) $ IF(JJ.GT.0)41,42
41 I=IMAX-II+1 $ GO TO 43
42 I=II
43 K=0
  IF(U(I,J)-UMAX)20,20,26
26 IF (U(I,J)-U(I-1,J))20,37,27
37 K=K+1

```

APPENDIX C (continued)

```

27 IF (U(I,J)-U(I+1,J))20,38,28
38 K=K+1
28 IF (U(I,J)-U(I,J+1))20,33,23
33 K=K+1
23 IF (U(I,J)-U(I,J+1))20,34,35
34 K=K+1
35 IF (K.GT.0)36,24
36 DO 39 L=1,LMAX
    LK=(IC(L)-I)**2 $ LJ=(JC(L)-J)**2
    IF (LK.GT.9)39,40
40 IF (LJ.GT.9)39,20
39 CONTINUE
    LMAX=LMAX+1 $ IC(LMAX)=I $ JC(LMAX)=
24 CALL NUMBER (X(I,J)=.12,Y(I,J),0.07,L(I,J),0.0,6H(F4.1))
20 CONTINUE
15 CALL SKETCH(UISØ)
    CALL SYMBOL (R+.125,6.5,.14,14HCONTOUR VALUES,0.,14)
    Y=6.5
    DO 50 L=1,L1
    Y=Y-.3
50 CALL NUMBER (R+.35,Y,.14,UISØ(L),0.,6H(F5.2))
RETURN
END

```

^aCorrespondence list:

Name in Program	Name in Text	Significance
U	U	Observed x-ray intensity
X	X	x-coordinate of datum
Y	Y	y-coordinate of datum
IK	IK	Dummy array for intercommunication of integers among subroutines
IC		Array established to "remember" locations of identical intensity maxima, I-subscript
JC		Same as IC, but for J-subscript
UISØ	UISØ	Value of intensity for which contour is desired

APPENDIX D. SUBROUTINE SKETCH^a

```

DIMENSION A(50,25),B(50,25),D(50,25),IK(20)
I,U(5),R(5),Z(5),C(10),UISO(10)
COMMON A,B,D,IK
N4=IK(7)
IMAX=IK(5)-1;JMAX=IK(6)-1
IK(10)=(ICLOCKF(B1)-IK(9))/60
PRINT 2,IK(10)
2 FORMAT (18H ENTER SKETCH, T=,I4)
DO1 IP=1,N4
1 C(IP)=UISO(IP)
DO26 J=1,JMAX
J2=J-2*(J/2)
DO26 I3=1,IMAX
IF(J2.EQ.0)41,42
42 I=I3 $ GO TO 15
41 I=IMAX-I3+1
15 U(1)=A(I,J) $ R(1)=B(I,J) $ Z(1)=D(I,J)
U(2)=A(I+1,J) $ R(2)=B(I+1,J) $ Z(2)=D(I+1,J)
U(3)=A(I+1,J+1) $ R(3)=B(I+1,J+1) $ Z(3)=D(I+1,J+1)
U(4)=A(I,J+1) $ R(4)=B(I,J+1) $ Z(4)=D(I,J+1)
U(5)=0.25*(U(1)+U(2)+U(3)+U(4)) $ R(5)=0.25*(R(1)+R(2)+R(3)+R(4))
Z(5)=0.25*(Z(1)+Z(2)+Z(3)+Z(4))
DO26 K=1,N4
DO26 M5=1,4
I1=M5 $ I2=M5+1 $ IF(I2.GT.4)28,27
28 I2=I2-4
27 IF(BRACE(U(I1),U(I2),C(K)),GT.0.)21,20
21 X=GUESS(C(K),U(I1),U(I2),R(11),R(12))
Y=GUESS(C(K),U(I1),U(I2),Z(11),Z(12))
CALL PLOT (X,Y,3)
IF(BRACE(U(I1),U(I5),C(K)),GT.0.)22,23
22 X=GUESS(C(K),U(I1),U(I5),R(11),R(I5))
Y=GUESS(C(K),U(I1),U(I5),Z(11),Z(I5))
GO TO 24
23 X=GUESS(C(K),U(I2),U(I5),R(I2),R(I5))
Y=GUESS(C(K),U(I2),U(I5),Z(I2),Z(I5))
24 CALL PLOT (X,Y,2) $ GO TO 26
20 IF(BRACE(U(I1),U(I5),C(K)),GT.0.)25,26
25 X=GUESS(C(K),U(I1),U(I5),R(I1),R(I5))
Y=GUESS(C(K),U(I1),U(I5),Z(I1),Z(I5))
CALL PLOT (X,Y,3) $ GO TO 23
26 CONTINUE
19 RETURN
END
FUNCTION BRACE (A,B,X)
IF(A-X)2,2,1
2 IF(X-B)3,4,4
1 IF(X-B)4,3,3
3 BRACE=1.

```

```
RETURN
4 BRACE =0.
RETURN
END
FUNCTION GUESS (A,B,C,D,E)
GUESS = D+(E-D)*(A-B)/(C-B)
RETURN
END
```

^aCorrespondence list:

<u>Name in Program</u>	<u>Name in Text</u>	<u>Significance</u>
A	U	Intensity of x-ray reflection
B	X	x-coordinate of datum
D	Y	y-coordinate of datum
U	U	The subset of intensity data corresponding to a particular "box" as described in the text
R	R	As for U, the x-coordinate
Z	Z	As for U, the y-coordinate
X		Actual x-coordinate of pen position during plotting
Y		Actual y-coordinate of pen position during plotting



INTERNAL DISTRIBUTION

- | | | | |
|-------|--------------------------------|--------|---------------------|
| 1-3. | Central Research Library | 33. | J. L. Gregg |
| 4-5. | ORNL - Y-12 Technical Library | 34. | M. T. Harkrider |
| | Document Reference Section | 35-37. | M. R. Hill |
| 6-25. | Laboratory Records Department | 38. | D. O. Hobson |
| 26. | Laboratory Records, ORNL R. C. | 39. | W. J. Hulsey (Y-12) |
| 27. | ORNL Patent Office | 40-43. | G. R. Love |
| 28. | N. B. Alexander | 44. | H. G. MacPherson |
| 29. | B. S. Borie | 45. | C. J. McHargue |
| 30. | G. E. Boyd | 46. | J. C. Ogle |
| 31. | J. E. Cunningham | 47. | D. A. Sundberg |
| 32. | J. H. Frye, Jr. | 48. | R. O. Williams |

EXTERNAL DISTRIBUTION

- 49. D. F. Cope, RDT, SSR, AEC, Oak Ridge National Laboratory
- 50. R. W. Gould, University of Florida, Gainesville, Florida
- 51. L. K. Jetter, Kaiser Aluminum, Spokane, Washington 99215
- 52. W. J. Larkin, AEC, Oak Ridge Operations
- 53. E. E. Stansbury, the University of Tennessee
- 54. E. A. Starke, Jr., Georgia Institute of Technology, Atlanta, Ga.
- 55. D. K. Stevens, AEC, Washington
- 56. Laboratory and University Division, Oak Ridge Operations
- 57-71. Division of Technical Information Extension