

OAK RIDGE NATIONAL LABORATORY

operated by

UNION CARBIDE CORPORATION

for the

U.S. ATOMIC ENERGY COMMISSION



ORNL-TM-229 *ef*

OR ABS

A FORTRAN PROGRAM FOR CALCULATING SINGLE CRYSTAL ABSORPTION CORRECTIONS

D. J. Wehe
W. R. Busing
H. A. Levy

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. The information is not to be abstracted, reprinted or otherwise given public dissemination without the approval of the ORNL patent branch, Legal and Information Control Department.

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.

OR ABS

A FORTRAN PROGRAM FOR CALCULATING
SINGLE CRYSTAL ABSORPTION CORRECTIONS

by

D. J. Wehe, W. R. Busing, and H. A. Levy

April, 1962

DATE ISSUED

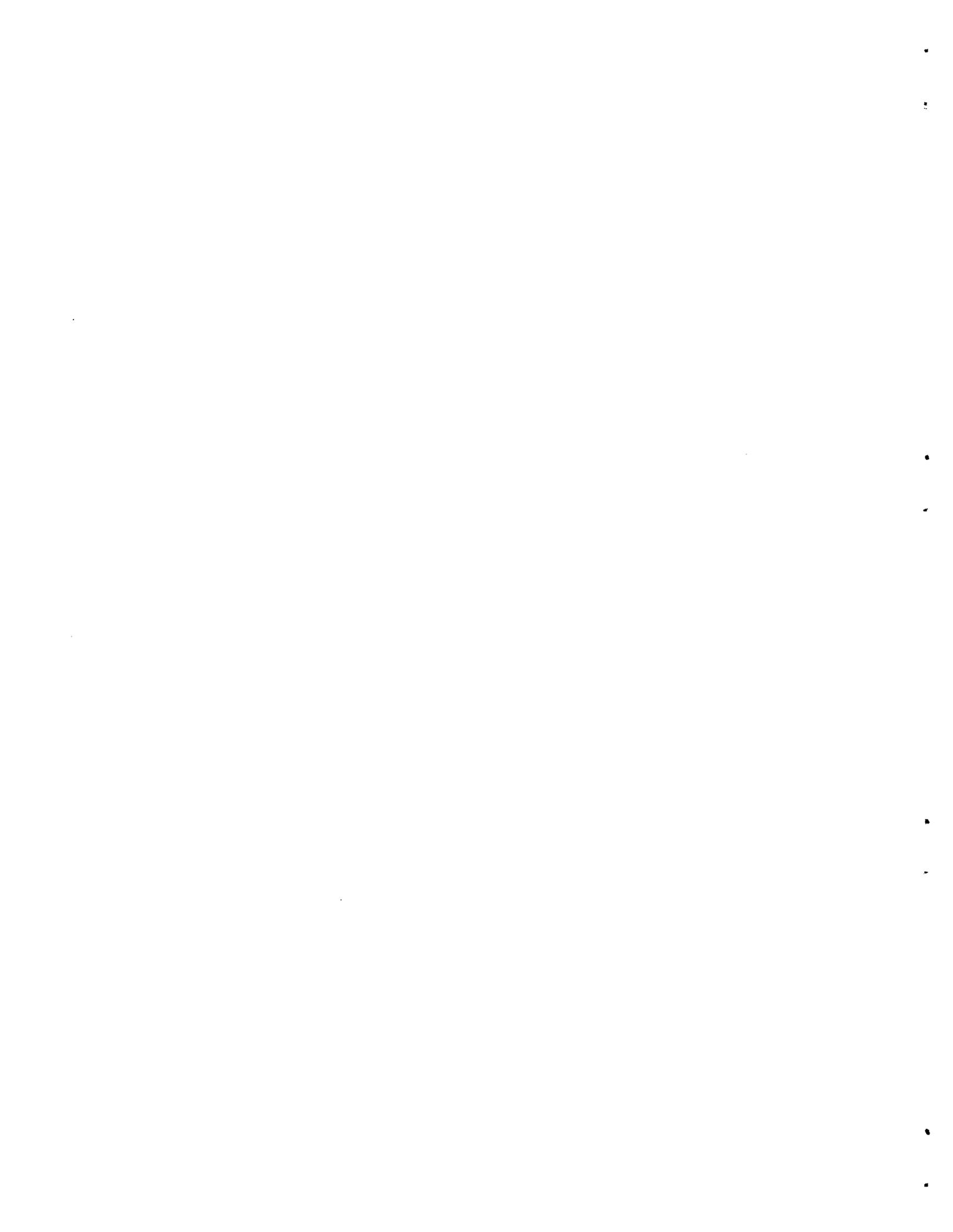
JUN 27 1962

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



CONTENTS

Abstract	1
Introduction	2
Mathematical Method	3
Outline of Program	5
Flow Chart	7
Preparation of the Program Deck	8
Details of the Subprograms	9
Subroutine SIZE	10
Subroutine CLOCK	10
Subroutine EPSR	11
Subroutine FACE	11
Subroutine ANGLE	13
Figure 1. Angles and Coordinate Systems	15
Restrictions and Machine Requirements	17
Timing	18
Data Input	18
Card Deck for Oak Ridge Installation	21
Operating Procedure for Oak Ridge Installation	21
Symbols Used	21
Storage Table for the Array DATA	24
References	25
Program Listing	26
Distribution	43



ABSTRACT

This memorandum describes a Fortran program for calculating the absorption correction to be applied to single crystal x-ray or neutron diffraction intensity measurements. Instructions are included for using the program which is available from the authors in the form of symbolic punched cards.

INTRODUCTION

The derivation of structure factors from integrated intensity measurements in X-ray or neutron diffraction usually requires a correction for absorption of radiation in the specimen. Tabulated values of this correction, expressed as the ratio of integrated intensity yielded by an absorbing specimen to that yielded by an hypothetical non-absorbing specimen of equal volume, are available for cylindrical (Bradley, 1935) and spherical (Evans and Ekstein, 1952) sample shapes. For accurate measurement of structure factors, the specimen is often reduced to one of these simple forms.

However, for a number of reasons, it frequently proves to be undesirable or difficult to shape a sample. The size and shape of available crystals may not permit cutting a sphere or cylinder large enough to give satisfactory intensities. Physical properties such as anisotropic resistance to grinding, or ease of cleavage or fracture may make shaping difficult. Furthermore, a cylindrical specimen may be used in only one orientation; while a spherical specimen obviates this difficulty, it may not permit optimum use of a beam with a long narrow cross section such as is usual in neutron diffraction spectrometers.

Several methods of correcting for absorption in samples of other shapes have been reported by Hendershot (1937), Albrecht (1939), Howells (1950) and Evans (1952), but all of them appear to be rather laborious. The availability of high-speed computers now makes possible the rapid calculation of the absorption correction factor for each reflection from a crystal of essentially arbitrary shape. The program to be described is similar to one prepared for the Oak Ridge computer, the Oracle (Busing and Levy, 1956).

MATHEMATICAL METHOD

It is assumed that the crystal is bounded by n plane surfaces and thus is described by a set of inequalities

$$a_s x + b_s y + c_s z - d_s \geq 0, \quad s = 1, 2, \dots, n, \quad (1)$$

with coefficients a_s, b_s, c_s, d_s chosen so that the inequalities are all satisfied only if the point x, y, z lies inside or on the surface of the crystal. This is a satisfactory description if there are no re-entrant angles between bounding planes, a condition which is assumed for this treatment. Cartesian coordinate axes for this description are those used to describe the angles which give the directions of the primary and diffracted beam. Two configurations are discussed under the subroutine ANGLE.

The factor to be computed is given by

$$A = \int (1/V) \exp[-\mu(r_\alpha + r_\beta)] dV, \quad (2)$$

where the integration is over the volume of the crystal, V , and where μ is the linear absorption coefficient, r_α the path length along the primary beam direction, and r_β that along the diffracted beam direction. In this program the integral is evaluated numerically using the method of Gauss (see, for example, Margenau and Murphy, 1943, p. 462) which will be described first for a one-dimensional integration. The method approximates the integral by a weighted sum of m terms:

$$\int_a^b g(x) dx \approx (b-a) \sum_{i=1}^m R_i g(x_i),$$

where

$$x_i = a + (b-a)u_i$$

and where the u_i 's and R_i 's are fractional constants which depend only on m , and their values are available for $m \leq 16$ (Lowan, Davids, and Levenson, 1942). The u_i 's determine the points, x_i , at which the integrand

is evaluated, while the R_i 's are the relative weights of the terms in the sum.

For triple integration, the approximation becomes

$$\int_a^b dx \int_{c(x)}^{d(x)} dy \int_{e(x,y)}^{f(x,y)} g(x,y,z) dz \approx \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m (b-a)[d(x_i)-c(x_i)][f(x_i,y_j)-e(x_i,y_j)]R_iR_jR_k g(x_i,y_j,z_k),$$

where

$$x_i = a+(b-a)u_i, \quad (3)$$

$$y_j = c(x_i)+[d(x_i)-c(x_i)]u_j, \quad (4)$$

$$z_k = e(x_i,y_j)+[f(x_i,y_j)-e(x_i,y_j)]u_k. \quad (5)$$

Tests of the method for conditions typical of neutron diffraction studies indicate that an m of 8 is satisfactory. The problem of determining A then reduces to finding the limits of integration, a , b , $c(x_i)$, $d(x_i)$, $e(x_i,y_j)$, and $f(x_i,y_j)$ and evaluating $g(x_i,y_j,z_k) = (1/V)\exp[-\mu(r_\alpha+r_\beta)]$.

In order to find a and b , the lower and upper limits on x , the routine first takes all possible combinations of three of the n planes which bound the crystal and solves for the coordinates of their intersections. Because there are no re-entrant angles between the faces, those intersections which are not corners of the crystal must lie outside of it. These are distinguished from the corners by rejecting points which fail to satisfy any one of the inequalities (1). The smallest and largest x values in the remaining set are then taken as a and b , respectively.

For a given value of x_i , the limits on y are found in a similar way by solving the equations of all possible pairs of faces, subject to the condition that $x = x_i$. Points outside the crystal are again rejected

and the smallest and largest y values remaining are taken as $c(x_i)$ and $d(x_i)$, respectively. The limits on z are determined in an analogous way for given values of x_i and y_j .

For a point x_i, y_j, z_k , the function $\exp[-\mu(r_\alpha + r_\beta)]$ is evaluated as follows. The distance from this point to the intersection of the primary beam with a face s of the crystal (or its extension) specified by one of the inequalities (1) is given by

$$r_{\alpha s} = \frac{d_s - a_s x_i - b_s y_j - c_s z_k}{a_s \gamma_{\alpha x} + b_s \gamma_{\alpha y} + c_s \gamma_{\alpha z}} \quad (6)$$

where the γ_α 's are the direction cosines of a vector parallel to the primary beam and directed toward its source. This quantity is positive if the intersection lies toward the source of the primary beam from (x_i, y_j, z_k) and negative if it lies away from the source. The desired path length, r_α , is the smallest positive quantity of the set $r_{\alpha s}$. Similarly, the path length r_β is the smallest positive quantity in the set

$$r_{\beta s} = \frac{d_s - a_s x_i - b_s y_j - c_s z_k}{a_s \gamma_{\beta x} + b_s \gamma_{\beta y} + c_s \gamma_{\beta z}} \quad (7)$$

where the γ_β 's are the direction cosines of the diffracted beam. The exponential is then evaluated in a straightforward way.

OUTLINE OF PROGRAM

It is usually necessary to determine the values of A for many reflections from a given crystal, and, sometimes, for several values of μ . The program avoids much repetition by calculating only once the limits of integration a, b, c, d, e , and f and the points of integration x_i, y_j, z_k . These determine the m^3 numerators of Equations (6) and (7) and the m^3 weights of integration

$$R_{ijk} = (b-a)(d-c)(f-e)R_i R_j R_k ,$$

which are saved in a block of core, or on magnetic tape, if necessary.

In the second part of the program the numerators are used to determine A for each reflection.

The amount of core available for data storage is determined by a subroutine (which must be the last one loaded) which determines the length of the code, size of machine, and amount of upper core unavailable. The size of the numerator block is then computed as the largest factor of m^2 possible, considering the number of faces, number of points of integration, and number of μ . In some cases, all numbers can be stored in fast storage, and no magnetic tape is used.

The majority of the input is read by two subroutines, one of which describes the shape of the crystal (FACE) and the other of which describes the angles of the reflection (ANGLE). Thus the user may describe his crystal and reflections in any convenient manner and supply a subroutine which will convert his input into the desired data.

The calculations begin with the determination of the limits a and b as described earlier, and the values of x_i are then found from (3), using the tabulated u's. For each x_i the limits c and d are then calculated and the values of y_j found from (4). Finally, for each combination x_i, y_j the limits e and f are determined and the values z_k taken from (5). Given the coordinates $x_i, y_j,$ and z_k the n numerators of (6) and (7) may be calculated and stored, together with the weight which is determined using the tabulated values of R. The crystal volume, which is simply the sum of all the weights, is also determined in this part of the code.

In the second part of the program remaining storage is used for reflection data for several reflections. By performing the calculation

for several reflections at a time, magnetic tape handling is minimized. As each block of numerators is read from magnetic tape, the $2n$ denominators of (6) and (7) are calculated. The distances r_α and r_β are determined, and appropriate contribution added to the sums for each reflection in core.

An on-line clock helps the user to estimate the time required for each run. The clock is sampled at the beginning of the program, at the end of the first part, and at the end of the problem. The last page of output for any crystal is a log giving the crystal identification, date, n , m , number of blocks of magnetic tape used, the number of values of μ , the number of reflections, and the time required for the two parts of the program. The volume of the crystal is also reported at this time for checking purposes.

Several checks are made during the calculations. Should the program fail a check, a notation is made on the output as to the nature of the error and the extent of the calculation before the run is terminated.

PREPARATION OF THE PROGRAM DECK

There are many features of this program which depend either on the arrangement of the diffraction equipment used or on the requirements of the computing installation. For this reason it is expected that in most cases minor modifications will have to be made before the program can be used. The program is therefore being transmitted in the form of symbolic cards, although column-binary cards are also available on request.

The following is a list of the subprograms which should be included in the assembled program deck. Several of them are discussed in more

detail in the next section.

<u>Subprogram</u>	<u>Identification of Symbolic Cards</u>	<u>Remarks</u>
1. Fortran II BSS loader		
2. Main program	PL 000-336	
3. Fortran II library subroutines		Not punched auto- matically on compiling
4. GAUS Ø L	G 337-384	
5. ID Ø UT	ID 385-391	
6. Ø C Ø R	Ø C 392-407	
7. EPSR	E1 408-413	
8. CL Ø CK	C1 414-421 C2 422-488	Alternative versions
9. ANGLE	A1 489-533 A2 534-591	Alternative versions
10. FACE	F1 592-630	
11. SIZE	S 631-659	Must be last subprogram
12. Fortran II transfer card		
13. Gaussian constants		
(a) Number of points	M 660	
(b) Gaussian weights	R1-2 661-662	These constants are read by the program as data cards.
(c) Gaussian points	U1-2 663-664	

DETAILS OF THE SUBPROGRAMS

Subroutines ERROR and EXIT

The Oak Ridge installation requires that a run be terminated in one of three ways. The Fortran input routine has been modified to recognize an end of file on the input tape as the normal end of the run. An entrance

into EXIT subroutine terminates the problem in a similar fashion. An entrance into ~~ERROR~~ implies that an octal dump of core is desired. The user must supply EXIT and ~~ERROR~~ routines which conform to local standards.

Subroutine SIZE

This subroutine determines the number of cells available in core for storage of data.

The routine labelled S in columns 77-78 should be usable on a IBM 704, 709, or 7090 of any size. In the decrement of SIZE + 5 is the amount of upper core not available to the program (205 for Oak Ridge). The routine uses the fact that it is the last subprogram loaded to determine the length of the code and the part of lower core not available to the program.

If a new routine is written it must use the calling sequence

CALL SIZE (NS)

and place the number of cells available for data storage in the decrement of NS as an integer. The main program will subtract from this the number of cells used by variables and fixed arrays in ~~COMMON~~, thus determining the length of the array DATA.

Subroutine CL~~O~~CK

The routine CL~~O~~CK is entered three times for each crystal to provide the user with the date of the run and the time required for Parts I and II of the program.

Two subroutines are provided. CL~~O~~CK with the identification C1 in columns 77-78 is a dummy which places blanks in DATE and gives zero for the time in T. It may be used when no on line clock is available.

The second routine, identified by C2 in columns 77-78, reads a Chrono-log digital clock model 2704-1, which has been connected to the 716 on-line printer control panel according to drawing 2704-9 which is furnished with the clock by the Chrono-log Corporation of Philadelphia 31, Pennsylvania. In addition to the wiring shown on the drawing, a wire is run from the load switching circuit (clock cable wire 50) through a filter into the sense entry hub. The read out of the clock is under control of printer sense exit six. The date of the run is given in BCD in DATE in the form mddy, mm being the month, dd, the day, and yy, the year. The time of the run in 10's of seconds is placed in the decrement of IT.

The calling sequence is CALL CLOCK (IT, DATE). The routine, in general, must place 6 BCD characters in DATE and an integer in the decrement of IT. The main program will report elapsed time as $IT - IT_1$ (IT_1 obtained on an earlier entry).

Subroutine EPSR

When determining the corners of the crystal, it is necessary, because of rounding errors, to accept a point as on the crystal surface if

$$a_s x + b_s y + c_s z - d_s + \epsilon \geq 0$$

for some small $\epsilon > 0$. If the program is not giving correct results, it is possible that an examination of the corners of the crystal that were found would indicate that an adjustment of the tolerance ϵ is required.

The main program reads an input parameter EPS. The subroutine EPSR must calculate EPS1 which is to be used as ϵ in the above inequality. The subprogram identified by E1 in columns 77-78 simply sets EPS1 equal to EPS.

Subroutine FACE

This subroutine provides the program with the 4n coefficients a_s , b_s , c_s , and d_s defined in Equation (1). The subroutine includes

the reading of input, the calculation of coefficients, and the reporting of input or coefficients, if desired.

The subroutine identified by F1 in columns 77-78 reads the following input cards via tape 10:

Card 1: Columns 1-5: N, the number of faces of the crystal, $N < 100$
6-10: ~~NOF~~. A number other than zero implies that the input is to be reported.

Cards 2, ..., N + 1: Columns 1-10 a_s
11-20 b_s As defined in Equation (1)
21-30 c_s
31-40 d_s

The coefficients a_s , b_s , c_s , d_s must be given with references to a coordinate system which is consistent with the angle description (see below). If ~~NOF~~ is not zero, the title card read in the main program and a list of the coefficients a, b, c, and d are reported on tape 9.

To write a different subroutine FACE, the following conventions should be observed:

- (1) The subroutine is entered with the statement CALL FACE (IDENT, N, DATA, ~~NOF~~, DATE).
- (2) The subroutine should include the reading and reporting of all data needed to define the crystal faces. Output on tape K may be headed with the title and date of run by using the statement CALL ~~IDOUT~~ (IDENT, DATE, K).
- (3) The number of faces is stored by the subroutine as an integer in the decrement of N.
- (4) The $4n$ coefficients are stored in DATA (~~NOF~~) through DATA (~~NOF~~ + $4N - 1$) in the order a_1, b_1, c_1, d_1, a_2 , etc. At least

2N cells beyond DATA (NCDF + 4N - 1) are available to the sub-routine for temporary storage.

- (5) If sense lights 1 and 2 are used, they must be turned off before returning to the main program.

Subroutine ANGLE

This subroutine gives the main program an identification for each reflection and calculates the direction cosines $\gamma_{\alpha x}$, $\gamma_{\alpha y}$, $\gamma_{\alpha z}$, $\gamma_{\beta x}$, $\gamma_{\beta y}$, and $\gamma_{\beta z}$ used in Equations (6) and (7). Two versions of this subroutine have been provided.

The subprogram identified by A1 in columns 77-78 deals with reflections in one zone. Let the z-axis be parallel to the zone axis being studied, and let the x-axis lie in a crystal plane which will be used as a reference plane for all the reflections being studied. The y-axis is chosen to give a right handed coordinate system. The configuration of the crystal and the spectrometer for a given reflection is specified by θ , the Bragg angle, and χ , the interplanar angle with respect to the reference plane. (See Busing and Levy, 1957, Fig. 1.) From these the routine calculates

$$\alpha = \chi + \theta + \pi$$

and

$$\beta = \chi - \theta$$

which define the reverse directions of the primary beam and the forward direction of the diffracted beam. The direction cosines required by the main program are

$$\gamma_{\alpha x} = \cos \alpha$$

$$\gamma_{\alpha y} = \sin \alpha$$

$$\gamma_{\alpha z} = 0$$

$$\gamma_{\beta x} = \cos \beta$$

$$\gamma_{\beta y} = \sin \beta$$

$$\gamma_{\beta z} = 0.$$

Input for each reflection consists of a BCD identification and the angles θ and χ . This input will be reported as BCD output on tape 6 if desired.

The subprogram identified A2 in columns 77-78 is for use with the Oak Ridge automatic neutron diffractometer or with the General Electric single crystal orienter. The geometry for each reflection is defined by three angles: θ , the Bragg angle, and χ and ϕ , the orienter settings.

The crystal may be mounted on the orienter in an arbitrary way (but the values of χ and ϕ depend on the way in which it is mounted). The coordinate system to which the crystal faces are referred is chosen as shown in Fig. 1. Assume the instrument to be in position for a hypothetical reflection for which $\chi = \phi = 0$ and θ has a small positive value. The x-axis is then chosen as the normal to this hypothetical reflecting plane, and the z-axis is taken parallel to the ϕ -axis in the direction away from the goniometer mount. The y-axis is then parallel to the χ -axis in a direction chosen so that the coordinate system will be right-handed. For the Oak Ridge diffractometer (Fig. 1a) the y-axis will point in the general direction of the primary and diffracted beams, while for the General Electric orienter (Fig. 1b) the opposite will be true. (Note that for the General Electric orienter the roles of the primary and diffracted beams are interchanged but this has no effect on the calculation.)

The subprogram evaluates the required direction cosines as follows:

$$\begin{aligned} \gamma_{\alpha x} &= \cos \chi \cos \phi \sin \theta + \sin \phi \cos \theta & \gamma_{\beta x} &= \cos \chi \cos \phi \sin \theta - \sin \phi \cos \theta \\ \gamma_{\alpha y} &= \cos \chi \sin \phi \sin \theta - \cos \phi \cos \theta & \gamma_{\beta y} &= \cos \chi \sin \phi \sin \theta + \cos \phi \cos \theta \\ \gamma_{\alpha z} &= \sin \chi \sin \theta & \gamma_{\beta z} &= \sin \chi \sin \theta. \end{aligned}$$

Input for the subroutine consists of a BCD identification and the angles θ , χ , and ϕ . If desired, the identification and input angles will be reported as BCD output on tape 6.

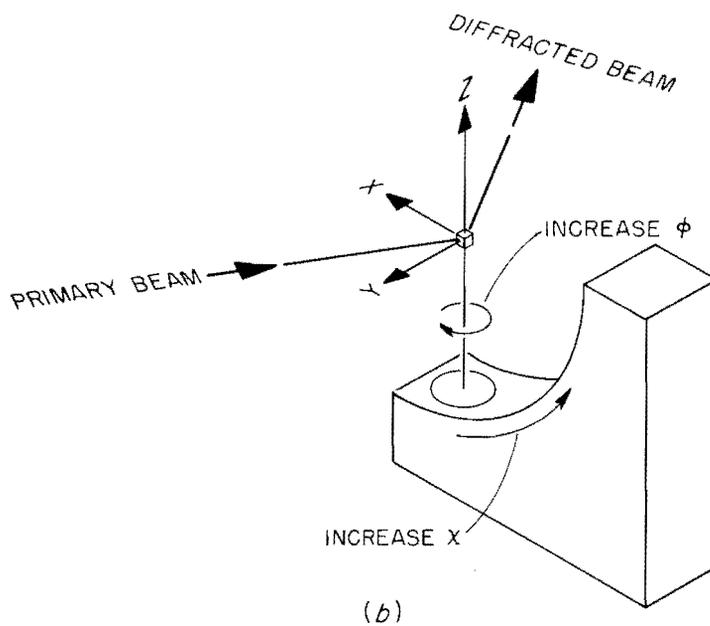
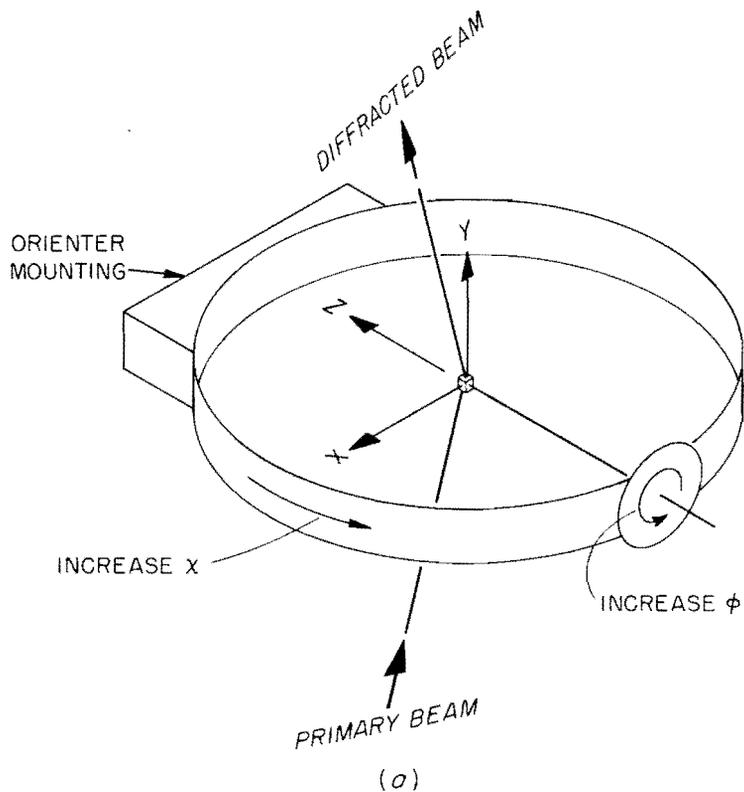


Figure 1. The angles and coordinate systems for (a) the Oak Ridge automatic diffractometer and (b) the General Electric single crystal orienter. The axes to which the crystal faces are referred are shown for $\chi = \phi = 0$.

To write ANGLE subroutines for other crystal orientations, the following specifications must be adhered to.

(1) Entry is by the statement

```
CALL ANGLE (IDENT, DATA, IANG, NANGL, DATE, NANGL1).
```

The arguments are defined below.

(2) The routine must read all of the input needed to define the reflections. If the user wishes to report this input or other results, this must be done within the subroutine. Tape 6 is rewound only once at the beginning of the main program and may be used for output by ANGLE.

(3) The crystal identification is found in the twelve word BCD array IDENT, and the date is in BCD in DATE. These may be reported by the subroutine on tape K, together with a skip to Channel 1, by the statement

```
CALL IDOUT (IDENT, DATE, K)
```

(4) The first entry into ANGLE may be distinguished from subsequent entries, since NANGL1 is zero only for the first entry.

(5) The program is more efficient on small machines if the data for several reflections are in core as a group. The maximum number of reflections that can be processed in such a group is found in NANGL on the first entry only. The subroutine should set NANGL to the actual number of reflections for which the angle data has been stored in core.

(6) Data for the reflections must be stored in the one-dimensional array DATA beginning at DATA (IANG). At exit, the data should be stored in groups of seven words in the following order:

1st word: Six BCD characters to identify the following reflection

2nd word: γ_{0x})

3rd word:	$\gamma_{\alpha y}$	}	direction cosines for primary beam
4th word:	$\gamma_{\alpha z}$		
5th word:	$\gamma_{\beta x}$	}	direction cosines for diffracted beam
6th word:	$\gamma_{\beta y}$		
7th word:	$\gamma_{\beta z}$		

NANGL words of storage in DATA following the reflection data may be used as temporary storage.

(7) When the data for the last reflection has been read into core, the subroutine must turn on sense light 1.

RESTRICTIONS AND MACHINE REQUIREMENTS

The following restrictions and requirements apply to the program as written. Most of them can be changed by making the appropriate modification. Equipment needed:

- (1) The source program may be assembled and run on an IBM 704, 709, or 7090. At least 8,192 words of storage are required.
- (2) At most 4 tape units are required. One for scratch (tape 2), one for input (tape 10), and two for output. (Main output is on tape 9, but reflection angles are put out via tape 6, if desired).
- (3) A card-to-tape unit is used to prepare the input tape.
- (4) All output is printed off-line.
- (5) An on-line clock is used for timing.

Program restrictions:

- (1) Uses sense lights 1 and 2.
- (2) Does not use sense switches.
- (3) Output gives 56 lines per page. It should be listed under program control. Automatic overflow switch should be off if the carriage tape has less than 56 lines.

(4) Program does not rewind input or output tapes. No end of file is written on the main output tape by the program at the end of a run.

TIMING

The machine time for a problem may be estimated from the expression

$$T = (C_1 N + C_2)R .$$

Here T is the time in seconds, N is the number of crystal faces, and R is the number of reflections for which the correction is to be calculated. Preliminary tests on an IBM 704 computer show that $C_1 = 1.1$ seconds per face per reflection, and $C_2 = 2.0$ seconds per reflection if intermediate magnetic tape storage is used, or $C_2 = 0$ when magnetic storage is not needed. Usually magnetic storage will be needed with an 8K memory but not with a 32 K memory.

DATA INPUT

1. Title card. FORMAT (12A6)

cols.

1-72 Any 72 Hollerith characters. This title will head each section of the output.

2. Card control. FORMAT (I5, E15.8, I5)

cols.

1-5 NMU, the number of absorption coefficients. $1 \leq \text{NMU} \leq 6$.

6-20 EPS, the increment used to compensate for round-off error in determining the limits of integration. For crystals of dimensions of the order of 0.1 and face equations with coefficients of the order of 1 an EPS of 10^{-6} (i.e. 1.0E-6) will usually be satisfactory.

21-25 $K\phi = 0$ if output of coordinates of crystal corners is not desired.
 $K\phi = 1$ to obtain this output.

3. Absorption coefficients. FORMAT (6E10.8)

cols.
1-10 MU(K), the absorption coefficients to be used in calculating
11-20 A for each reflection. The number of MU(K) must be NMU given
etc. on the control card above.

4. Input for FACE routine F1.

a. Face control card. FORMAT (2I5)

col.
1-5 N, the number of faces of the crystal. $4 \leq N < 100$
6-10 N~~OP~~ = 0 if output of the coefficients a, b, c, d are not
desired; N~~OP~~ = 1 if this output is desired.

b. Coefficients of planes describing faces of crystal. FORMAT (4E10.8)

The number of cards is equal to N, the number of faces of the crystal.

col.
1-10 a.
11-20 b.
21-30 c.
31-40 d.

Note that the coordinate system to which these equations refer must be consistent with the angle description used below.

5. Input for ANGLE routine A1. (Used for zero level reflections).

a. Angle control card FORMAT (I1)

Column 1 = 1 if the input angles are to be reported.

Column 1 = 0 or blank, if this output is not desired.

b. Reflection angles and identification. FORMAT (2(A6, 2F3.3))

Two reflections are placed on each card. For a definition of the reference system used in measuring the angles, see the discussion of the subroutine.

col.
1-6 6 BCD characters to identify the following reflection
7-14 θ , the Bragg angle
15-22 χ , the interplanar angle
23-28 6 BCD characters to identify the following reflection
29-36 θ , the Bragg angle
37-44 χ , the interplanar angle

Angles are given in degrees. The list of reflections is terminated with six zeros in the identification field following the last χ .

6. Input for ANGLE routine A2. (Used with three-dimensional orienter).

a. Angle control card ~~F~~ORMAT(11)

Column 1 = 1 if the input angles are to be reported.

Column 1 = 0 or blank if this output is not desired.

b. Reflection angles and identification ~~F~~ORMAT (2(A6, 3F8.3))

Two reflections are placed on each card. For a definition of the reference system used in measuring the angles, see the discussion of the subroutine.

col.
1-6 6 BCD characters identifying the following reflection.
7-14 θ , the Bragg angle
15-22 χ } orienter angles
23-30 ϕ }
31-36 6 BCD characters identifying the following reflection.
37-44 θ , the Bragg angle
45-52 χ } orienter angles
53-60 ϕ }

Angles are given in degrees. The list of reflections is terminated with six zeros in the identification field following the last ϕ .

CARD DECK FOR OAK RIDGE INSTALLATION

1. Monitor control cards
2. Program deck
3. Data
 - a. Title card
 - b. Control card
 - c. Absorption coefficients
 - d. Input for FACE routine
 - e. Input for ANGLE routine
4. Repeat (3) if absorption coefficients are to be calculated for more than one crystal in a single monitor job.

OPERATING PROCEDURE FOR OAK RIDGE INSTALLATION

When preparing the monitor control cards the user should call for tapes on units 2 and 6. If either ANGLE routines A1 or A2 are used and the angle output is called for, the number of the files on tape 6 to be printed is equal to the number of crystals run. If the angle output has not been called for, there will be nothing on tape 6. The program rewinds tape 6 regardless of which ANGLE routine is being used.

SYMBOLS USED

- | | |
|-------|--|
| EPS | input parameter. It is used in the calculation of ϵ , the tolerance used in the determination of the corners. |
| NS | amount of core storage available in data block. |
| NP | $NP * M * (N+1)$ words are written in each block of magnetic tape in the first part of the program. |
| NANGL | number of reflections to be processed at one time. |
| N | number of planes bounding crystal. |
| V | volume of crystal. |

V1 volume of crystal computed with weights read from magnetic tape; used as a check.

IDENT BCD array containing crystal identification.

IT1 clock time at beginning of program; time for part I.

IT2 clock time at end of part I of program; time for part II.

IT3 clock time at end of program.

NANGL1 total number of reflections processed.

NMU number of values of μ to be used.

NC~~Ø~~F DATA subscript of first coefficient a_1 of planes describing crystal.

NEND DATA subscript of last coefficient d_N of planes describing crystal.

NUM DATA subscript of first numerator of $r_{\alpha s}$ in core.

NUMED DATA subscript of last numerator of $r_{\alpha s}$ in core.

JC~~Ø~~R the number of words to be reported in one line crystal corner output.

IANG DATA subscript of first angle identification.

ISUM DATA subscript of first absorption correction factor.

IDIF1 DATA subscript of the first of the N differences $d_s - a_s x_i$ (Part 1); DATA subscript of the first of the N denominators $a\gamma_{\alpha s} + b\gamma_{\alpha s} + c\gamma_{\alpha s}$ (Part 2).

IDIF2 DATA subscript of the first of the N differences $d_s - a_s x_i - b_s y_j$ (Part 1); DATA subscript of the first of the N denominators $a\gamma_{\beta s} + b\gamma_{\beta s} + c\gamma_{\beta s}$ (Part 2).

T temporary storage used by Gaussian elimination routine.

TA current value of $r_{\alpha s}$.

TB current value of $r_{\beta s}$.

X current value of x coordinate of point of integration.

RA r_α

RB r_β

WX $(b-a) x_i$

Y current value of y coordinate of point of integration.

WY $(c-d) (a-b) x_i y_j$

Z equivalent to I6, current value of z coordinate of point of integration.

A a, lower limit of integration on x axis.

B b, upper limit of integration on x axis.

DIF1 $b-a$; equivalent to B.

C $c(x_i)$, lower limit of integration on y axis.

D $d(x_i)$, upper limit of integration on y axis.

DIF2 $d(x_i) - c(x_i)$, equivalent to D.

E $e(x_i, y_j)$ lower limit of integration on z axis.

F $f(x_i, y_j)$ upper limit of integration on z axis.

VMU equivalent to DATA; storage for values of μ .

DATA name of block of storage which contains coefficients of faces, direction cosines of angles, values of $d_s - a_s x_i - b_s y_j - c_s z_k$, etc. The subscripts for each block are calculated.

NDATA equivalent to DATA; fixed point name for DATA.

IX DATA subscript for first index of corners of crystal.

R Gaussian weights.

U Gaussian points.

IP

IQ DATA subscripts used in determining corners of crystal.

IR

I IP1 and IQ1

I1, I2, I3, I4, I5, I6, I7, J, K, K1 Dummy subscripts for DATA.

EPS1 tolerance ϵ used in determining corners of crystal.

KØ output option on the reporting of the corners of the crystal.

M number of Gaussian points used in integration.

K~~Ø~~NST number of words of C~~Ø~~M~~Ø~~N storage used excepting DATA block.
 NP1 number of blocks on magnetic tape.

STORAGE TABLE FOR THE ARRAY DATA*

<u>Number of words</u>		<u>Subscript</u>
NMU ≤ 6	μ_1, \dots, μ_{NMU}	1 NMU
4 * N	$a_s, b_s, c_s, d_s, (s = 1, \dots, n)$	NC Ø F NEND
NP*M*(N+1)	$d_s - a_s x_i - b_s y_j - c_s z_k$ $(b-a)(d-c)(f-e)R_i R_j R_k$ for $s = 1, \dots, n$ $i, j, k = 1, \dots, m$	NUM NUMED
N	(Part 1) $d_s - a_s x_i$ ($s = 1, \dots, n$) (Part 2) $a_s \gamma_{\alpha x} + b_s \gamma_{\alpha y} + c_s \gamma_{\alpha z}$	IDIF1 IDIF2-1
N	(Part 1) $d_s - a_s x_i - b_s y_j$ ($s = 1, \dots, n$) (Part 2) $a_s \gamma_{\beta x} + b_s \gamma_{\beta y} + c_s \gamma_{\beta z}$	IDIF2 IX-1
12	(Part 1) Indices of planes and coordinates of corners	IX IX+11
7*NANGL	(Part 2) $Id, \gamma_{\alpha x}, \gamma_{\alpha y}, \gamma_{\alpha z}, \gamma_{\beta x}, \gamma_{\beta y}, \gamma_{\beta z}$ for each reflection	IANG ISUM-1
NMU*NANGL	(Part 2) $A_1 (\mu_k), A_2 (\mu_k) \dots$	ISUM

*DATA is equivalent to VMU and NDATA.

REFERENCES

- Albrecht, G. (1939). Rev. Sci. Instrum. 10, 221.
- Bradley, A. J. (1935). Proc. Phys. Soc. 47, 879.
- Busing, W. R. and Levy, H. A. (1957). Acta Cryst. 10, 180.
- Evans, H. T. (1952). J. Appl. Phys. 23, 663.
- Evans, H. T. and Ekstein, M. G. (1952). Acta Cryst. 5, 540.
- Hendershot, O. P. (1937). Rev. Sci. Instrum. 8, 324.
- Howells, R. G. (1950). Acta Cryst. 3, 366.
- Lowan, A. N., Davids, N. and Levenson, A. (1942). Bull. Amer. Math. Soc. 48, 739.
- Margenau, H. and Murphy, G. M. (1943). The Mathematics of Physics and Chemistry. New York: Van Nostrand.

C	ABSORPTION CALCULATION, 3-DIMENSIONAL	P1	000
	COMMON EPS,NS,NB,NP,NANGL,N,V,V1,IDENT,IT1,IT2,IT3,NANGL1,NMU,	P1	001
	2NCOF,NEND,NUM,NUMED, IDIF1, IDIF2, IANG, ISUM, JCOR,	P1	002
	3T,X,WX,Y,WY,Z,A,B,C,D,E,F,DIF1,DIF2,DIF3,TA,TB,RA,RB,	P1	003
	4VMU,DATA	P1	004
	DIMENSIONIDENT(12),T(12),R(16),U(16),VMU(1),DATA(1),NDATA(1)	P1	005
	EQUIVALENCE(EPS,DUM4),(NS,DUM5),	P1	006
	1(NB,DUM6),(NP,DUM7),(NANGL,DUM8),(N,DUM9),(V,DUM10),(V1,DUM11),	P1	007
	2(IDENT,DUM12),(IT1,DUM13),(IT2,DUM14),(IT3,DUM15),(NANGL1,DUM16),	P1	008
	3(NMU,DUM17),(IANG,DU33),(ISUM,DU34),	P1	009
	4(NCOF,DUM19),(NEND,DUM20),(NUM,DUM21),(NUMED,DUM22),(JCOR,DUM23),	P1	010
	5(IDIF1,DU31),(IDIF2,DU32),(T(3),X),(T(6),WX),(T(9),Y),(T(12),WY),	P1	011
	6(Z,I6),(A,DUM24),(B,DIF1),(C,DUM25),(D,DIF2),(E,DUM26),(F,DIF3),	P1	012
	7(IP1,IQ1,I),(T(1),TA),(T(2),TB),(T(4),RA),(T(5),RB),	P1	013
	8(DATA,VMU,NDATA)	P1	014
	READINPUTTAPE10,1003,M	P1	015
	IF(M-16)9001,9001,9002	P1	016
9001	READINPUTTAPE10,1005,(U(I),I=1,M),(R(I),I=1,M)	P1	017
	REWIND 6	P1	018
	LINES =50	P1	019
	SENSELIGHT0	P1	020
	CALLSIZE(NS)	P1	021
3213	CALLCLOCK(IT1,DATE)	P1	022
	READINPUTTAPE10,1001,(IDENT(I),I=1,12)	P1	023
	READINPUTTAPE10,1003,NMU,EPS,KO	P1	024
	IF(NMU-6)314,314,315	P1	025
314	NCOF=NMU+1	P1	026
	READINPUTTAPE10,1004,(VMU(I),I=1,NMU)	P1	027
	CALLFACE(IDENT,N,DATA,NCOF,DATE)	P1	028
9020	CALLEPSR(EPS,EPS1)	P1	029
	I1=NS-KONST-NMU-6*N-12	P1	030
	NP=I1/(M*(N+1))	P1	031
	I2=M**2	P1	032
316	IF(XMODF(I2,NP))318,312,318	P1	033
318	NP=NP-1	P1	034
	GO TO316	P1	035
312	IF(NP)8313,8313,311	P1	036
311	NANGL=(I1-NP*M*(N+1)+12)/(7+NMU)	P1	037
	IF(NANGL-2)317,317,313	P1	038
317	IF(NP-1)8313,8313,318	P1	039
313	NB=I2/NP	P1	040
	NEND=NCOF+4*N-1	P1	041

	NUM=NEND+1	P1	042
	NUMED=NP*M*(N+1)+NEND	P1	043
	IDIF1=NUMED+1	P1	044
	IDIF2=IDIF1+N	P1	045
	IX=IDIF2+N	P1	046
	IANG=IX	P1	047
	ISUM=NANGL*7+IANG	P1	048
C	DETERMINE COEFFICIENTS OF CORNERS AND REPORT	P1	049
C	CALCULATE A, B, B-A	P1	050
	IF(NB-1)7006,7006,7007	P1	051
7007	REWIND2	P1	052
7006	IF(KO)60,60,61	P1	053
61	CALLIDOUT(IDENT,DATE,9)	P1	054
	WRITEOUTPUTTAPE 9,2001	P1	055
	NLINES=LINES	P1	056
60	A=10.0E37	P1	057
	B=-A	P1	058
	JCOR=0	P1	059
	I1=N-2	P1	060
	DO27IP=1,I1	P1	061
	IP1=IP+1	P1	062
	I1=N-1	P1	063
	DO26IQ=IP1,I1	P1	064
	IQ1=IQ+1	P1	065
	DO54IR=IQ1,N	P1	066
	I1=NCOF+4*(IP-1)	P1	067
	DO24K=1,12,3	P1	068
	T(K)=DATA(I1)	P1	069
24	I1=I1+1	P1	070
	I1=NCOF+4*(IQ-1)	P1	071
	DO23K=1,12,3	P1	072
	T(K+1)=DATA(I1)	P1	073
23	I1=I1+1	P1	074
	I1=NCOF+4*(IR-1)	P1	075
	DO22K=1,12,3	P1	076
	T(K+2)=DATA(I1)	P1	077
22	I1=I1+1	P1	078
	CALLGAUSOL(T,3)	P1	079
	IF(SENSELIGHT2)54,69	P1	080
69	DO28K=NCOF,NEND,4	P1	081
	IF(DATA(K)*T(10)+DATA(K+1)*T(11)+DATA(K+2)*T(12)-DATA(K+3)+EPS1)5	P1	082
	14,28,28	P1	083
28	CONTINUE	P1	084

	IF(A-T(10))50,50,51	P1	085
51	A=T(10)	P1	086
50	IF(T(10)-B)52,52,53	P1	087
53	B=T(10)	P1	088
52	IF(KO)54,54,62	P1	089
62	I1=IX+JCOR	P1	090
	NDATA(I1)=IP	P1	091
	NDATA(I1+1)=IQ	P1	092
	NDATA(I1+2)=IR	P1	093
	DO20K=3,5	P1	094
	I1=JCOR+K+IX	P1	095
20	DATA(I1)=T(K+7)	P1	096
	JCOR=JCOR+6	P1	097
	IF(JCOR-12)54,29,29	P1	098
29	CALLOCOR(DATA,IX,JCOR)	P1	099
54	CONTINUE	P1	100
26	CONTINUE	P1	101
27	CONTINUE	P1	102
	IF(KO)57,57,64	P1	103
64	IF(JCOR)57,57,58	P1	104
58	CALLOCOR(DATA,IX,JCOR)	P1	105
57	DIF1=B-A	P1	106
	IF(DIF1)56,56,55	P1	107
55	CONTINUE	P1	108
	V =0.0	P1	109
	NP1=NP	P1	110
	K1=NUM	P1	111
	DO107I=1,M	P1	112
	X=A+DIF1*U(I)	P1	113
	WX=R(I)*DIF1	P1	114
	I2=IDIF1	P1	115
	DO70I1=NCOF,NEND,4	P1	116
	DATA(I2)=DATA(I1+3)-DATA(I1)*X	P1	117
70	I2=I2+1	P1	118
C	CALCULATE C, D, D-C	P1	119
	C=10.0E37	P1	120
	D=-C	P1	121
	I20=N-1	P1	122
	DO71IP=1,I20	P1	123
	IP1=IP+1	P1	124
	DO76IQ=IP1,N	P1	125
	I1=NCOF+1+4*(IP-1)	P1	126

	DO73K=1,4,3	P1	127
	T(K)=DATA(I1)	P1	128
73	I1=I1+1	P1	129
	I1=NCOF+1+4*(IQ-1)	P1	130
	DO74K=1,4,3	P1	131
	T(K+1)=DATA(I1)	P1	132
74	I1=I1+1	P1	133
	I1=IDIF1+IP-1	P1	134
	T(7)=DATA(I1)	P1	135
	I1=IDIF1+IQ-1	P1	136
	T(8)=DATA(I1)	P1	137
	CALLGAUSOL(T,2)	P1	138
	IF(SENSELIGHT2)76,68	P1	139
68	I2=IDIF1	P1	140
	DO75I1=NCOF,NEND,4	P1	141
	IF(DATA(I1+1)*T(7)+DATA(I1+2)*T(8)-DATA(I2)+EPS1)76,75,75	P1	142
75	I2=I2+1	P1	143
	IF(C-T(7))77,77,78	P1	144
78	C=T(7)	P1	145
77	IF(T(7)-D)76,76,79	P1	146
79	D=T(7)	P1	147
76	CONTINUE	P1	148
71	CONTINUE	P1	149
	DIF2=D-C	P1	150
	IF(DIF2)72,72,80	P1	151
80	CONTINUE	P1	152
	DO91J=1,M	P1	153
	Y=C+DIF2*U(J)	P1	154
	WY=WX*DIF2*R(J)	P1	155
	I2=IDIF1	P1	156
	I3=IDIF2	P1	157
	DO90I1=NCOF,NEND,4	P1	158
	DATA(I3)=DATA(I2)-DATA(I1+1)*Y	P1	159
	I2=I2+1	P1	160
90	I3=I3+1	P1	161
	E=10.0E37	P1	162
	F=-E	P1	163
	I3=IDIF2	P1	164
	DO93I1=NCOF,NEND,4	P1	165
	Z=DATA(I3)/DATA(I1+2)	P1	166
	I4=IDIF2	P1	167
	DO94I2=NCOF,NEND,4	P1	168
	IF(DATA(I2+2)*Z-DATA(I4)+EPS1)93,94,94	P1	169

94	I4=I4+1	P1	170
	IF(E-Z)96,96,97	P1	171
97	E=Z	P1	172
96	IF(Z-F)93,93,98	P1	173
98	F=Z	P1	174
93	I3=I3+1	P1	175
	DIF3=F-E	P1	176
	IF(DIF3)99,99,100	P1	177
100	CONTINUE	P1	178
	DO101K=1,M	P1	179
	Z=E+DIF3*U(K)	P1	180
	I2=IDIF2	P1	181
	DO102I1=NCOF,NEND,4	P1	182
	DATA(K1)=DATA(I2)-DATA(I1+2)*Z	P1	183
	IF(DATA(K1))103,103,104	P1	184
104	DATA(K1)=0.0	P1	185
103	K1=K1+1	P1	186
102	I2=I2+1	P1	187
	DATA(K1)=WY*DIF3*R(K)	P1	188
	V=V+DATA(K1)	P1	189
101	K1=K1+1	P1	190
	IF(NB-1)91,91,7002	P1	191
7002	NP1=NP1-1	P1	192
	IF(NP1)105,105,91	P1	193
105	WRITETAPE2,(DATA(I1),I1=NUM,NUMED)	P1	194
	K1=NUM	P1	195
	NP1=NP	P1	196
91	CONTINUE	P1	197
107	CONTINUE	P1	198
	IF(NB-1)7004,7004,7005	P1	199
7005	ENDFILE2	P1	200
	REWIND2	P1	201
7004	CALLCLOCK(IT2,DATE)	P1	202
C	END OF PART I, BEGINNING OF PART II	P1	204
	CALLIDOUT(IDENT,DATE,9)	P1	205
	WRITEOUTPUTTAPE9,2003,(VMU(I),I=1,NUM)	P1	206
	NANGL1=0	P1	207
211	CALLANGLE(IDENT,DATA,IANG,NANGL,DATE,NANGL1)	P1	208
	V1=0.0	P1	209
	NANGL1=NANGL1+NANGL	P1	210
	K1 =N+1	P1	211
	I3=NANGL*7-1+IANG	P1	212

	I2=ISUM+NMU*NANGL-1	P1	213
	DO200I1=ISUM,I2	P1	214
200	DATA(I1)=0.0	P1	215
	DO201I1=1,NB	P1	216
	IF(NB-1)7001,7001,7003	P1	217
7003	READTAPE2,(DATA(I2),I2=NUM,NUMED)	P1	218
7001	I6=ISUM	P1	219
	DO202I2=IANG,I3,7	P1	220
	I9=IDIF1	P1	221
	I8=IDIF2	P1	222
	DO203I4=NCOF,NEND,4	P1	223
	DATA(I9)=DATA(I4)*DATA(I2+1)+DATA(I4+1)*DATA(I2+2)+DATA(I4+2)*DATA(I2+3)	P1	224
	DATA(I8)=DATA(I4)*DATA(I2+4)+DATA(I4+1)*DATA(I2+5)+DATA(I4+2)*DATA(I2+6)	P1	225
	I9=I9+1	P1	226
	I8=I8+1	P1	227
203	DO215I7=NUM,NUMED,K1	P1	228
	I5=I7	P1	229
	RA=10.0E37	P1	230
	RB=RA	P1	231
	I9=IDIF2-1	P1	232
	I8=IDIF2	P1	233
	DO208I4=IDIF1,I9	P1	234
	IF(DATA(I4))911,213,213	P1	235
911	TA=DATA(I5)/DATA(I4)	P1	236
	IF(RA-TA)213,213,205	P1	237
205	RA=TA	P1	238
213	IF(DATA(I8))216,217,217	P1	239
216	TB=DATA(I5)/DATA(I8)	P1	240
	IF(RB-TB)217,217,207	P1	241
207	RB=TB	P1	242
217	I5=I5+1	P1	243
	I8=I8+1	P1	244
208	CONTINUE	P1	245
	RA=RA+RB	P1	246
	I10=I6	P1	247
	DO206I4=1,NMU	P1	248
	DATA(I10)=DATA(I10)+EXP(-VMU(I4)*RA)*DATA(I5)	P1	249
206	I10=I10+1	P1	250
215	CONTINUE	P1	251
	I6=I6+NMU	P1	252
202	CONTINUE	P1	253
		P1	254
		P1	255

	D08204 I4=NUM,NUMED,K1	P1	256
	I5=I4+N	P1	257
8204	V1=VI+DATA(I5)	P1	258
201	CONTINUE	P1	259
	IF(NB-1)7009,7009,7010	P1	260
7010	REWIND2	P1	261
7009	I6=ISUM+NANGL*NMU-1	P1	262
	D08021 I4=ISUM,I6	P1	263
8021	DATA(I4)=DATA(I4)/V	P1	264
	IF(V-V1)209,210,209	P1	265
210	I2=ISUM	P1	266
	I5=IANG+7*NANGL-1	P1	267
	D0400 I1=IANG,I5,7	P1	268
	I3=I2+NMU-1	P1	269
	NLINES=NLINES-1	P1	270
	IF(NLINES)240,240,241	P1	271
240	NLINES=LINES	P1	272
	CALLIDOUT(IDENT,DATE,9)	P1	273
	WRITEOUTPUTTAPE9,2003,(VMU(I),I=1,NMU)	P1	274
241	WRITE OUTPUT TAPE9,2008,DATA(I1),(DATA(I4),I4=I2,I3)	P1	275
400	I2=I3+1	P1	276
	IF(SENSELIGHT1)212,211	P1	277
212	CALLCLOCK(IT3,DATE)	P1	278
	IT1=IT2-IT1	P1	279
	IT2=IT3-IT2	P1	280
	CALLIDOUT(IDENT,DATE,9)	P1	281
	WRITEOUTPUTTAPE9,2093,V	P1	282
	WRITEOUTPUTTAPE9,2004,N,NANGL1,NMU,IT1,IT2	P1	283
	GOTO8213	P1	284
C	ERROR RETURNS	P1	285
315	CALLIDOUT(IDENT,DATE,9)	P1	286
	WRITEOUTPUTTAPE9,2010,NMU	P1	287
	CALLEXIT	P1	288
56	CALLIDOUT(IDENT,DATE,9)	P1	289
	WRITEOUTPUTTAPE9,2011,A,DIF1	P1	290
	CALLERROR	P1	291
72	CALLIDOUT(IDENT,DATE,9)	P1	292
	WRITEOUTPUTTAPE9,2012,C,DIF2	P1	293
	CALLERROR	P1	294
99	CALLIDOUT(IDENT,DATE,9)	P1	295
	WRITEOUTPUTTAPE9,2013,E,DIF3	P1	296
	CALLERROR	P1	297

209	CALLIDOUT(IDENT,DATE,9)	P1	298
	WRITE OUTPUTTAPE9,2020,V,V1	P1	299
	CALL ERROR	P1	300
9021	CALLIDOUT(IDENT,DATE,9)	P1	301
	WRITEOUTPUTTAPE9,2092,N	P1	302
	CALLEXIT	P1	303
9002	WRITEOUTPUTTAPE9,2090,M	P1	304
	CALLEXIT	P1	305
9313	CALLIDOUT(IDENT,DATE,9)	P1	306
	WRITEOUTPUTTAPE9,2313,N,M,NMU,NS,NP	P1	307
	CALLEXIT	P1	308
2313	FORMAT(36H0MEMORY IS TOO SMALL FOR PARAMETERS./4H0N =I3,3HM =I3,5HP1	P1	309
	1NMU =I2,19H0AVAILABLE STORAGE =I6,4HNP =I5)	P1	310
2090	FORMAT(21H1M IS TOO LARGE. M = I5)	P1	311
2092	FORMAT(5H0N = I5,15H IS TOO LARGE.)	P1	312
2093	FORMAT(10H0VOLUME = F15.8)	P1	313
1001	FORMAT(12A6)	P1	314
1002	FORMAT(1I1)	P1	315
1003	FORMAT(I5,E15.8,I5)	P1	316
1004	FORMAT(6E10.8)	P1	317
1005	FORMAT(4F10.8)	P1	318
2001	FORMAT(34H0COORDINATES OF CORNERS OF CRYSTAL/52H0INDEX OF PLANES FP1	P1	319
	1ORMING CORNER, FOLLOWED BY X, Y, Z//)	P1	320
2003	FORMAT(33H0ABSORPTION CORRECTION FACTOR, A/6H0MU = 6(E10.5)//)	P1	321
2004	FORMAT(5H0N = I5,3X,19HNUMBER OF ANGLES = I5,3X,15HNUMBER OF MU = P1	P1	322
	1I2//29H TIME USED IN PARTS 1 AND 2 = 2I10,3X,18HIN 10S OF SECONDS.P1	P1	323
	2)	P1	324
2008	FORMAT(1H01A6,3X,6F12.8)	P1	325
2010	FORMAT(29H0INPUT ERROR. NUMBER OF MU =I5,24HWHICH IS GREATER THANP1	P1	326
	1 6.)	P1	327
2011	FORMAT(62H0ERROR IN CALCULATION OF UPPER AND LOWER LIMITS (B, A).P1	P1	328
	1 A = E15.8,8H, B-A = E15.8/19H0CORE DUMPS FOLLOW.)	P1	329
2012	FORMAT(62H0ERROR IN CALCULATION OF UPPER AND LOWER LIMITS (D, C). P1	P1	330
	1 C = E15.8,8H, D-C = E15.8/19H0CORE DUMPS FOLLOW.)	P1	331
2013	FORMAT(62H0ERROR IN CALCULATION OF UPPER AND LOWER LIMITS (F, E). P1	P1	332
	1 E = E15.8,8H, F-E = E15.8/19H0CORE DUMPS FOLLOW.)	P1	333
2020	FORMAT(58H0ERROR. VOLUME FROM CORE IS NOT EQUAL TO VOLUME FROM TAP1	P1	334
	1PE/11H0V(CORE) = E15.8,10HV(TAPE) = E15.8/14H0DUMP FOLLOWS.)	P1	335
	END (0,1,0,0,0)	P1	336

C	SUBROUTINE GAUSOL(A,N)	G	337
	SUBROUTINE GAUSOL(A,N)	G	338
	DIMENSION A(3,4)	G	339
C	SOLUTION OF SYSTEM OF EQUATIONS BY GAUSSIAN ELIMINATION	G	340
C	USING INTERCHANGES	G	341
C	A = NAME OF MATRIX (ORIGINAL MATRIX IS DESTROYED)	G	342
C	MATRIX IS STORED BY COLUMNS, VECTOR FOLLOWS	G	343
C	N = ORDER OF MATRIX	G	344
C	(N MUST BE LESS THAN OR EQUAL DIMENSION STATEMENT)	G	345
C	SOLUTION IS FOUND IN A(1,N+1) THROUGH A(N,N+1)	G	346
	IF(SENSELIGHT2)20,20	G	347
20	N2=N-1	G	348
	N1=N+1	G	349
	DO9 I1=1,N2	G	350
	L=0	G	351
	R=A(I1,I1)	G	352
	I11=I1+1	G	353
	DO2 IR=I11,N	G	354
	IF(ABSF(R)-ABSF(A(IR,I1)))3,2,2	G	355
3	R=A(IR,I1)	G	356
	L=IR	G	357
2	CONTINUE	G	358
	IF(L)4,4,5	G	359
5	DO6 IJ=I1,N1	G	360
	X=A(L,IJ)	G	361
	A(L,IJ)=A(I1,IJ)	G	362
6	A(I1,IJ)=X	G	363
4	DO8 I2=I11,N	G	364
	T=A(I2,I1)/A(I1,I1)	G	365
	DO7 I3=I11,N1	G	366
7	A(I2,I3)=A(I2,I3)-A(I1,I3)*T	G	367
8	CONTINUE	G	368
9	CONTINUE	G	369
	A(N,N1)=A(N,N1)/A(N,N)	G	370
	DO11 I=1,N2	G	371
	I1=N-I	G	372
	I2=I1+1	G	373
	DO10 I3=I2,N	G	374
10	A(I1,N1)=A(I1,N1)-A(I1,I3)*A(I3,N1)	G	375
11	A(I1,N1)=A(I1,N1)/A(I1,I1)	G	376
	IFDIVIDECHECK15,14	G	377
15	SENSELIGHT2	G	378

	14 IF ACCUMULATOR OVERFLOW12,17	G	379
	12 SENSELIGHT2	G	380
	17 IFQUOTIENTOVERFLOW16,13	G	381
	16 SENSELIGHT2	G	382
	13 RETURN	G	383
	END(0,1,0,0,1)	G	384
C	REPORT CRYSTAL IDENTIFICATION	ID	385
	SUBROUTINE IDOUT(IDENT,DATE,N)	ID	386
	DIMENSION IDENT(12)	ID	387
	WRITEOUTPUTTAPEN,2001,(IDENT(I),I=1,12),DATE	ID	388
2001	FORMAT(1H112A6,41X,1A6)	ID	389
	RETURN	ID	390
	END(0,1,0,0,0)	ID	391
C	REPORT ONE LINE OF ONE OR TWO CORNERS OF A CRYSTAL	OC	392
C	FOR 3-DIMENSIONAL PROBLEM	OC	393
C	FIRST THREE NUMBERS REPORTED ARE INTEGERS WHICH DESIGNATE	OC	394
C	THE PLANES WHICH FORM CORNERS,	OC	395
C	SECOND THREE NUMBERS ARE THE X, Y, Z COORDINATES OF THE CORNER.	OC	396
C	NOTATION X = LOCATION OF ARRAY OF 12 NUMBERS	OC	397
C	IX = SUBSCRIPT OF FIRST NUMBER OF ARRAY X	OC	398
C	JCOR = 6*I, WHERE I IS THE NUMBER OF CORNERS TO REPORT	OC	399
	SUBROUTINEOCOR(X,IX,JCOR)	OC	400
	DIMENSION X(1)	OC	401
	JCOR =JCOR+IX-1	OC	402
	WRITE OUTPUT TAPE9,2002,(X(K),K=IX,JCOR)	OC	403
	JCOR=0	OC	404
2002	FORMAT(1H 2(3I3.0,3F13.9,4X))	OC	405
	RETURN	OC	406
	END(0,1,0,0,0)	OC	407
C	DUMMY EPSR SUBROUTINE	E1	408
C	TO CALCULATE EPS1 = TOLERANCE FOR CORNERS OF CRYSTAL	E1	409
	SUBROUTINEEPSR(EPS,EPS1)	E1	410
	EPS1=EPS	E1	411
	RETURN	E1	412
	END(0,1,0,1,0)	E1	413

C	DUMMY CLOCK SUBROUTINE		C1	414
C	PLACES 0 IN TIME, BLANKS IN DATE		C1	415
	SUBROUTINE CLOCK(IT,DATE)		C1	416
S BLK	ALF		C1	417
	IT =0		C1	418
	DATE=BLK		C1	419
	RETURN		C1	420
	END(0,1,0,0,0)		C1	421
*	CLOCK SUBROUTINE		C2	422
	FUL	PROGRAM CARD	C2	423
	ORG 0	PROGRAM CARD	C2	424
	MZE 0,0,4	9L	C2	425
	PZE 0	9R	C2	426
	PZE T+1	8L	C2	427
	PZE 0	8R	C2	428
	BCD 1CLOCK	7L	C2	429
	PZE 0	7R	C2	430
	ORG 0		C2	431
	REL		C2	432
	CLA 1,4		C2	433
	STA ADD		C2	434
	CLA 2,4		C2	435
	STA ADD+2		C2	436
	SXD T,4		C2	437
	TSX CLOCK,4		C2	438
	HTR *+1		C2	439
	ALS 18		C2	440
ADD	STO 0		C2	441
	CLA DATE		C2	442
	STO 0		C2	443
	LXD T,4		C2	444
	TRA 3,4		C2	445
DATE	BSS 1		C2	446
ORCLK5	REM		C2	447
	REM READ CHRONO-LOG CLOCK ROUTINE. ERROR RETURN IF CLOCK FAILS		C2	448
	REM BINARY TIME (10 SEC.=1 UNIT) IN AC. FOR USE WITH CHRONO-LOG		C2	449
	REM DIGITAL CLOCK MODEL 2704-1 WIRED TO 716 PRINT ECHO ENTRIES.		C2	450
CLOCK	RPR		C2	451
	SPR 6	SENSE THE CLOCK	C2	452
	STZ DATE		C2	453

SXD	#+35,2		C2	454
STZ	COMMON		C2	455
SXD	COMMON+1,1		C2	456
LXD	#+13,1		C2	457
TNX	#+15,1,2		C2	458
CPY	DATE		C2	459
CPY	COMMON		C2	460
TQP	*-3		C2	461
LXD	#+15,2		C2	462
LXA	*+1,1		C2	463
PXD	3		C2	464
RQL	1		C2	465
TQP	*+5		C2	466
RQL	1		C2	467
TNX	*+6,1,1		C2	468
TQP	*-2	ERROR IF MQ MINUS	C2	469
TXL	*+3,,47		C2	470
ADD	*+14,2		C2	471
TIX	*-7,1,1		C2	472
TXI	*+12,4,1	ERROR RETURN	C2	473
TNX	*+11,2,1		C2	474
TRA	*+5,2		C2	475
TXI	*-10,1,9		C2	476
TXI	*-11,1,5		C2	477
TXI	*-12,1,9		C2	478
TXI	*-13,1,5		C2	479
DEC	3600,360,60,6,1		C2	480
LXD	COMMON+1,1		C2	481
LXD	*+3,2		C2	482
SPT			C2	483
TRA	2,4		C2	484
TXL	*-2		C2	485
COMMON	BSS 2		C2	486
T	HTR 0		C2	487
	END 0		C2	488
C	SUBROUTINE ANGLE (IDENT,DATA,IANG,NANGL,DATE,NANGL1)		A1	489
C	USING ONLY TWO ANGLES, THETA AND CHI		A1	490
C	ZERO IS STORED FOR REMAINING DIRECTION COSINES		A1	491
	SUBROUTINE ANGLE (IDENT,DATA,IANG,NANGL,DATE,NANGL1)		A1	492
	DIMENSION IDENT(12),DATA(1)		A1	493
	K1=0		A1	494

I3=IANG+7*NANGL-1	A1	495
IF(NANGL1)30,31,30	A1	496
31 READINPUTTAPE10,1002,NOP	A1	497
30 DO10I2=IANG,I3,14	A1	498
I4=I2+13	A1	499
READINPUTTAPE10,1001,(DATA(I1),DATA(I1+1),DATA(I1+2),I1=I2,I4,7)	A1	500
DO13I5=I2,I4,7	A1	501
IF(DATA(I5)) 13,12,13	A1	502
13 K1=K1+1	A1	503
IF(NANGL-K1-2) 11,10,10	A1	504
12 SENSELIGHT1	A1	505
NANGL=K1	A1	506
GOTO11	A1	507
10 CONTINUE	A1	508
11 I3=IANG+7*NANGL-1	A1	509
IF(NOP)17,16,15	A1	510
15 NOP=-1	A1	511
CALLIDOUT(IDENT,DATE,6)	A1	512
WRITEOUTPUTTAPE6,2001	A1	513
17 WRITEOUTPUTTAPE6,2002,(DATA(I1),DATA(I1+1),DATA(I1+2),I1=IANG,I3,7)	A1	514
1) IF(SENSELIGHT1)20,16	A1	515
20 SENSELIGHT1	A1	516
END FILE 6	A1	517
16 DO14I2=IANG,I3,7	A1	518
DATA(I2+6)=(DATA(I2+2)-DATA(I2+1))*0.0174533	A1	519
DATA(I2+3)=(DATA(I2+2)+DATA(I2+1))*0.0174533+3.1415926	A1	520
DATA(I2+1)=COSF(DATA(I2+3))	A1	521
DATA(I2+2)=SINF(DATA(I2+3))	A1	522
DATA(I2+3)=0	A1	523
DATA(I2+4)=COSF(DATA(I2+6))	A1	524
DATA(I2+5)=SINF(DATA(I2+6))	A1	525
14 DATA(I2+6)=0	A1	526
1001 FORMAT(2(1A6,2F8.3))	A1	527
1002 FORMAT(1I1)	A1	528
2001 FORMAT(12H0ANGLE INPUT/1H0,9X,14HTHETA CHI//)	A1	529
2002 FORMAT(1H (2(1A6,2F10.3,5X)))	A1	530
RETURN	A1	531
END (0,1,0,0,0)	A1	532
	A1	533

C	SUBROUTINE ANGLE(2), FOR 3-D ORIENTING EQUIPMENT	A2	534
C	WRITTEN 9-9-60	A2	535
	SUBROUTINE ANGLE(IDENT,DATA,IANG,NANGL,DATE,NANGL1)	A2	536
	DIMENSION IDENT(12),DATA(1)	A2	537
	K1#0	A2	538
	I3#IANG+7*NANGL-1	A2	539
	IF(NANGL1)30,31,30	A2	540
31	READINPUTTAPE10,1002,NOP	A2	541
30	DO!0I2#IANG,I3,14	A2	542
	CONTINUE	A2	543
	CONTINUE	A2	544
	I4#I2+13	A2	545
	READINPUTTAPE10,1001,(DATA(I1),DATA(I1+1),DATA(I1+2),DATA(I1%3)	A2	546
	I,I1#I2,I4,7)	A2	547
	DO13I5#I2,I4,7	A2	548
	IF(DATA(I5)) I3,I2,I3	A2	549
13	K1#K1+1	A2	550
	IF(NANGL-K1-2) I1,I0,I0	A2	551
12	SENSELIGHT1	A2	552
	NANGL#K1	A2	553
	GOTO11	A2	554
10	CONTINUE	A2	555
11	I3#IANG+7*NANGL-1	A2	556
	IF(NOP)17,16,15	A2	557
15	NOP#-1	A2	558
	CALLIDOUT(IDENT,DATE,6)	A2	559
	WRITEOUTPUTTAPE6,2001	A2	560
17	WRITEOUTPUTTAPE6,2002,(DATA(I1),DATA(I1+1),DATA(I1+2), DATA(I1+3),	A2	561
	I1I#IANG,I3,7)	A2	562
	IF(SENSELIGHT1)20,16	A2	563
20	SENSELIGHT1	A2	564
	END FILE 6	A2	565
16	DO14I2#IANG,I3,7	A2	566
	DATA(I2+1)#DATA(I2+1)*0.174532925E-1	A2	567
	DATA(I2+2)#DATA(I2+2)*0.174532925E-1	A2	568
	DATA(I2+3)#DATA(I2+3)*0.174532925E-1	A2	569
	T1#SINF(DATA(I2+1))	A2	570
	T2#COSF(DATA(I2+1))	A2	571
	T3#VINF(DATA(I2+3))	A2	572
	T4#COSF(DATA(I2+3))	A2	573
	DATA(I2+6)#SINF(DATA(I2+2))*T1	A2	574
	DATA(I2+3)#COSF(DATA(I2+2))*T1	A2	575
	DATA(I2+4)#DATA(I2+3)*T4	A2	576

	DATA(I2+5)=T3*T2	A2	577
	DATA(I2+1)=DATA(I2+4)+DATA(I2+5)	A2	578
	DATA(I2+4)=DATA(I2+4)-DATA(I2+5)	A2	579
	DATA(I2+5)=T4*T2	A2	580
	DATA(I2+3)=DATA(I2+3)*T3	A2	581
	DATA(I2+2)=DATA(I2+3)-DATA(I2+5)	A2	582
	DATA(I2+5)=DATA(I2+3)+DATA(I2+5)	A2	583
	DATA(I2+3)=DATA(I2+6)	A2	584
14	CONTINUE	A2	585
1001	FORMAT(2(1A6,3F8.3))	A2	586
1002	FORMAT(1I1)	A2	587
2001	FORMAT(1H016X,11HANGLE INPUT/1H010X,5HTHETA,8X,3HCHI,8X,3HPHI//)	A2	588
2002	FORMAT(1H (2(A6,3F11.3,6X)))	A2	589
	RETURN	A2	590
	END (0,1,0,0,0)	A2	591
C	SUBROUTINE FACE(IDENT,N,DATA,NCOF,DATE), 3 DIMENSIONAL PROBLEM	F1	592
C	BEFORE ENTRY, IDENT, NCOF ARE SET	F1	593
C	UPON EXIT, SUBROUTINE SUPPLIES N AND COEFFICIENTS WHICH	F1	594
C	DESCRIBE PLANES FORMING FACES OF CRYSTAL	F1	595
C	NOTATION IDENT = IDENTIFICATION OF CRYSTAL	F1	596
C	N = NUMBER OF FACES	F1	597
C	DATA = BLOCK OF GENERAL STORAGE	F1	598
C	NCOF = ELEMENT OF DATA CONTAINING FIRST	F1	599
C	COEFFICIENT DESCRIBING PLANES, OR FIRST	F1	600
C	AVAILABLE CELL OF STORAGE UPON ENTRY	F1	601
C	NOP = 0 IMPLIES DO NOT REPORT INPUT	F1	602
C	= 1 IMPLIES REPORT INPUT	F1	603
C	DATE = BCD DATE	F1	604
	SUBROUTINEFACE(IDENT,N,DATA,NCOF,DATE)	F1	605
	DIMENSION IDENT(12),DATA(1)	F1	606
	READINPUTTAPE10,1001,N,NOP	F1	607
	NEND=4*N+NCOF-1	F1	608
	READINPUTTAPE10,1002,(DATA(I1),I1=NCOF,NEND)	F1	609
	IF(NOP)2,2,3	F1	610
3	NBEG=NCOF	F1	611
	K3=0	F1	612
7	CALLIDOUT(IDENT,DATE,9)	F1	613
	WRITEOUTPUTTAPE9,2001,N	F1	614
	DO4K1=NBEG,NEND,4	F1	615
	K2=K1+3	F1	616

	K3=K3+1		F1	617
	WRITEOUTPUTTAPE9,2002,K3,(DATA(I1),I1=K1,K2)		F1	618
	IF(50-K3)5,5,4		F1	619
5	NBEG=K1		F1	620
	GOTO7		F1	621
4	CONTINUE		F1	622
1001	FORMAT(2I5)		F1	623
1002	FORMAT(4E10.8)		F1	624
2001	FORMAT(1H+,70X,17HNUMBER OF FACES =I5/75H0COEFFICIENTS A, B, C, D		F1	625
	1WHERE AX + BY + CZ = D DESCRIBES FACE OF CRYSTAL/1H0,2X,1HN,7X,1F1		F1	626
	2HA,9X,1HB,9X,1HC,9X,1HD//)		F1	627
2002	FORMAT(1H I3,2X,4F10.5)		F1	628
2	RETURN		F1	629
	END(0,1,0,0,0)		F1	630
*	SUBROUTINE SIZE		S	631
	REM TO DETERMINE AMOUNT OF STORAGE AVAILABLE FOR DATA		S	632
	REM MAKE DECREMENT OF SIZE + 5 =PART OF UPPER CORE NOT AVAILABLE		S	633
	REM ENTER WITH TSX SIZE,4		S	634
	REM HTR NS		S	635
	REM EXIT WITH TRA 2,4		S	636
	REM NS WILL CONTAIN AMOUNT OF AVAILABLE STORAGE IN DECREMENT.		S	637
	REM THIS SUBROUTINE MUST BE THE LAST SUBROUTINE LOADED.		S	638
	FUL		S	639
	ORG 0 PROGRAM CARD		S	640
	MZE 0,0,4 9L		S	641
	PZE 0 9R		S	642
	PZE END+1 8L		S	643
	PZE 0 8R		S	644
	BCD 1SIZE 7L		S	645
	PZE 0 7R		S	646
	ORG 0		S	647
	REL		S	648
SIZE	SXD END,4		S	649
	CLA 1,4		S	650
	STA SIZE+6		S	651
	LXA END,4		S	652
	TIX *+1,4,END		S	653
	TIX *+1,4,205		S	654
	SXD 0,4 NS		S	655
	LXD END,4		S	656
	TRA 2,4		S	657
END	HTR -1		S	658
	END 0		S	659

8
•019855072•101666761•237233795•408282679
•591717321•762766205•898333239•980144928
•050614268•111190517•156853323•181341892
•181341892•156853323•111190517•050614268

M 660
U1 661
U2 662
R1 663
R2 664

DISTRIBUTION

1. D. J. Wehe
- 2-4. W. R. Busing
- 5-7. H. A. Levy
- 8-9. Central Research Library
10. Document Reference Section
- 11-25. Division of Technical Information Extension
26. Research and Development Division
27. Laboratory Records - Record Copy
- 28-200. Laboratory Records

