BNWL-22

AEC
RESEARCH
and
DEVELOPMENT
REPORT

INDEX
A COMPUTER PROGRAM
FOR INDEXING X-RAY DIFFRACTION POWDER PATTERNS

JANUARY 1965

BATTELLE-NORTHWEST
RICHLAND, WASHINGTON

PACIFIC NORTHWEST LABORATORY operated by BATTELLE MEMORIAL INSTITUTE

LEGAL NOTICE

PACIFIC NORTHWEST LABORATORY

RICHLAND, WASHINGTON

Operated By

BATTELLE MEMORIAL INSTITUTE

For The

U. S. ATOMIC ENERGY COMMISSION Under Contract No. AT(45-1)-1830

PRINTED BY/FOR THE U. S. ATOMIC ENERGY COMMISSION

AEC-RLOO RICHLAND, WASH.

BNWL-22

UC-25, Metals, Ceramics
and Materials
(TID-4500, 39th Ed.)

INDEX
A COMPUTER PROGRAM
FOR INDEXING X-RAY DIFFRACTION POWDER PATTERNS

By

J. B. Goebel
Applied Mathematics Department

A. S. Wilson
Chemistry Department

January 1965

PACIFIC NORTHWEST LABORATORY

RICHLAND, WASHINGTON

Printed by/for the U. S. Atomic Energy Commission

Printed in USA. Price $3.00. Available from the
Clearinghouse for Federal Scientific and Technical Information,
National Bureau of Standards,
U. S. Department of Commerce,
Springfield, Virginia

ABSTRACT

Input data for this program are the 2θ reflections where θ is the Bragg angle as measured by a diffractometer from powdered crystalline samples. Additional data that may be supplied at the user's discretion are the X-ray wavelengths, a possible error value, empirical gram formula weight and observed density. The program indexes the set of reflections on the basis of the cubic, hexagonal, tetragonal, and orthorhombic systems in that order. All lattice constants are calculated by an iterative least squares procedure. The printed output includes the crystal system, the lattice constants and their standard deviations, the wavelengths used, observed and calculated $\sin^2\theta$ values, and the difference between these latter two. An error term, E, is also printed as well as the number of formula units per unit cell in some cases. The value of E, the size of the lattice constants and their standard deviations, indicate to the crystallographer the validity of the indexing. The Miller indices are printed in the form $h^2 + k^2 + \ell^2$ in the cubic case, $h^2 + hk + k^2$ and ℓ^2 in the hexagonal case, $h^2 + k^2$ and ℓ^2 in the tetragonal case, and h^2, k^2, and ℓ^2 in the orthorhombic case.

A typical indexing problem requires about 3 min on a 7090 IBM computer.

TABLE OF CONTENTS

INDEX
A COMPUTER PROGRAM
FOR INDEXING X-RAY DIFFRACTION POWDER PATTERNS

INTRODUCTION

The determination of the crystal class of substances using their X-ray diffraction powder patterns has involved a considerable amount of effort since the advent of W. L. Bragg's early experiments. The calculations needed for indexing an X-ray diffraction powder pattern can be time consuming and tedious even for crystals of the cubic class if several patterns are to be examined. With crystals that belong to the hexagonal, tetragonal, and orthorhombic classes, a considerable amount of time can be used to index their powder patterns. The use of indexing aids, such as the Hull-Davey charts, Bunn's chart, and Bjurstrom's ruler makes the work of indexing a powder pattern easier. Even so, the computational effort required of the crystallographer is considerable and for orthorhombic cases successful indexings are infrequent. For these reasons a logical system has been developed to yield rapidly the indexing of a given powder pattern by using an electronic computer. The speed of the indexing of the computer is such that several patterns depending upon their complexity can be indexed in a few minutes time.

The data required by the computer program are the reflections that are reported as 2θ, where θ is the Bragg angle; the sequential number above which the reflections are resolved into their α_1 and α_2 components; and the X-ray wavelengths if radiation other than CuKα radiation is used. The program generates the set of Miller indices for each reflection, 2θ, observed and calculated $\sin^2\theta$ values and their differences as well as the standard deviations of $\sin^2\theta$ and θ. The lattice constants and their standard deviations are also produced by the program.

PROGRAM INDEX

Program INDEX was written for and compiled on an IBM 7090 computer with 32K memory. The computer operates under the control of the FORTRAN Monitor system. The program consists of four subroutines: INPUT, CUBIC, HEXTET, and ORTHO. The latter three subroutines contain their own output statements.

SUBROUTINE INPUT

Subroutine INPUT reads in and counts the angles that are in the form of 2θ. A maximum of 150 angles can be accommodated, but a simple change in one statement can alter the program so that it will accept a larger number of angles. INPUT converts 2θ to $\sin^2\theta$ values that are stored in COMMON so that they will be available to all other subroutines. The program is devised to handle reflections measured using filtered CuKα radiation. If other radiation is used, the proper wavelengths must be supplied as input data. At the lower angles where α_1 and α_2 lines merge, the $\sin^2\theta$ values are multiplied by the ratio: λ_1/λ_{avg}, where λ_1 is the Kα_1 wavelength in Angstroms and λ_{avg} is the unresolved wavelength. This operation normalizes the lower reflections to the α_1 reflections that are used in those regions where α_1 and α_2 are resolved.

After INPUT, INDEX calls in order the subroutines CUBIC, HEXTET, and ORTHO. In general any set of reflections will yield solutions based on all four crystal systems, because there exist in the program no statements that determine the validity of the indexing. The validity of the indexing is judged by the crystallographer using information generated and printed by the subroutines. Each subroutine is available as a separate program so that if the crystal class of the substance of interest is known, the crystallographer can employ only the relevant program for indexing the pattern.

SUBROUTINE CUBIC

Subroutine CUBIC indexes the powder pattern on a cubic basis. This subroutine will find a solution for any set of reflections and the results will be printed. After printing the results from CUBIC, the program proceeds to HEXTET.

It is well known that the lattice constant a and reflections of any cubic crystal must satisfy the Bragg relation $\sin^2\theta = \frac{\lambda^2}{4a^2}(h^2 + k^2 + \ell^2)$.[1]

Thus the idea of the subroutine is to find a solution to a set of equations of the form:

$$\sin^2\theta_i = \underline{a}\, n_i\ . \tag{1}$$

It would seem plausible, first, to define a by dividing $\sin^2\theta_i$ by n_1, n_2, etc. , in succession until a suitable a is found. However, the fractional error in $\sin^2\theta$ from the first few angles is so great that this procedure can give incorrect solutions for crystals of large lattice constants. The procedure finally adopted begins with the last half of the set of angles, i. e. , $\{2\theta_i\}$ where i = [N/2], . . . ,N, and finds a solution. Having found the matching n_i for $i = \frac{N}{2}, \ldots, N$ the method of least squares is then used on the set of equations (1) to determine an $\hat{a}$ such that $\sin^2\theta_i = \hat{a}\, n_i$. With this $\hat{a}$, the entire solution is calculated by defining

$$n_i = \left(\frac{\sin^2\theta_i}{\hat{a}} + \frac{1}{2}\right)^{*}\ , \quad i = 1, \ldots, N \tag{2}$$

where the parentheses indicate the greatest integer less than or equal to the number within parentheses.

The detailed procedure used in finding the solution is as follows: Let k = [N/2]. Define the first trial a by $\underline{a} = \frac{\sin^2\theta_k}{n_k}$. Then for a given error E a check is made to see if a set of integers n_i exists in the list such that

$$\left| \frac{\sin^2\theta_i}{\underline{a}} - n_i \right| < E \text{ for } i = k, \ldots, N. \tag{3}$$

---

* Since the 7090 truncates fractional numbers by dropping all digits to the right of the decimal, the 1/2 is needed so that n_i will be truncated to the correct integer.

If there is not such a set of integers (n_i) $\underline{a}$ is redefined by $\underline{a} = \sin^2\theta_k/n_{k+1}$ and a solution is sought. If a solution is not found by trying all possible values of $\underline{a}$, then the error E is doubled and the procedure starts again. If a solution is found, the error E is replaced by E/2 and the procedure is repeated. The program is so written that once a solution is found, the error E will be reduced to a value that is 0.01 greater than that value of E where no solution exists. Thus, the solution is accepted and printed when a change in the error from E to E - .01 would result in no solution. Once this "minimum" error is determined, the program uses the method of least squares to determine a number "ahat" defined by $\hat{a} = \lambda^2/4\,a_0^2$ where a_0 is the lattice constant. Next, a set of "calculated $\sin^2\theta_i$" defined as $\sin^2\theta_i = \hat{a}\,n_i$ for i = 1, . . . , N, is determined. The program then prints the error, the lattice constant, the list of observed 2θ's, n_i, $\sin^2\theta_i$, calculated $\sin^2\theta_i$, and the difference between these latter two quantities. The lattice constant is calculated as $\lambda/(2\sqrt{\hat{a}})$ where λ is the X-ray wavelength.

If the above error limitation of 0.01 is increased to 0.1, at times an invalid indexing will result. This is most likely to occur when the list of data consists of only a few reflections. It should be noted that such a procedure always produces a solution, whether it is legitimate or not.

The program is devised to handle reflections measured by the use of filtered CuKα radiation. At the smaller angles where the α_1 and α_2 lines merge, the $\sin^2\theta$ values are multinlied by 0.99833 to normalize these reflections to the α_1 reflection that is used in those regions where α_1 and α_2 are resolved.

INTERPRETING THE RESULTS OF CUBIC

Since this subroutine is designed so that any set of reflections is indexed on a cubic basis, it is of paramount interest to know the criteria that determine whether the indexing is valid or not. Several of these criteria exist. The first is the size of the error, E. If E is ≥ 0.5 then an overlapping of the integers (n_i) is required to find the solution. Using CUBIC trials of known cubic crystals gave errors less than 0.20, whereas trials of noncubic crystals gave errors greater than 0.25. It thus seems reasonable to expect that the error E for a valid indexing would generally be less than 0.25 and certainly less than 0.5.

A second criterion for deciding the validity of the indexing is the appearance of forbidden integers such as seven or fifteen in the assigned integer list for the first half of the data. Whereas forbidden integers (those that cannot be expressed in the form $h^2 + k^2 + \ell^2$, where h, k, and ℓ are integers) should not occur in the last half of the data, they may occur in the first half of the data. The reason for this is that the entire indexing is based upon the last half of the set of reflections. Since the reflections are less accurate for small values of θ, the program bypasses the first half of the data, indexes the last half, and calculates the lattice constant $\hat{a}$. In this indexing of the higher angles only valid h, k, and ℓ values are used. However once $\underline{a}$ is calculated, this value is used in equation 1 to calculate the $h^2 + k^2 + \ell^2$ for the first half of the data. During this process the program may calculate one of the forbidden integers, indicating an invalid indexing.

Another criterion is the magnitude of the differences between the observed and calculated $\sin^2\theta_i$ values. CUBIC occasionally gives significantly larger differences for a noncubic crystal than for a cubic one. For a valid indexing the differences will be, on the average, less than 0.0005; whereas, an invalid indexing may give differences greater than 0.001.

The use of the above criteria for deciding whether an indexing is invalid or not depends upon the implied assumption that only a single crystalline phase is being examined. If some reflections exist that are caused by impurities then, in general, the use of the above criteria will result in the conclusion that an invalid indexing was achieved. Impurity reflections can be identified or removed by varying the procedure for the preparation of the sample. Moreover, if some reflections are suspected of impurity origins, then it is best on the first trial to eliminate these reflections from the list of input data. In this case, a valid indexing is achieved. If the impurity reflections occur within the first half of the input data list of a valid indexing, these reflections may be identified by: (1) an abnormally large difference between observed and calculated $\sin^2\theta$'s, and (2) a forbidden integer assigned to the preceding or following reflection.

In a very special case the program may locate an impurity reflection or an error in the last half of the input data list. This will be rare, however, because the program is designed to assign all reflections in the last half of the input data list to the cubic system and an invalid assignment may be necessary to achieve this objective, if extraneous reflections are present. Thus, even a single extraneous reflection may not be detected by the program.[2]

SUBROUTINE HEXTET

Subroutine HEXTET is designed to generate an indexing on a hexagonal basis and then on a tetragonal basis. Any set of reflections will in general yield solutions in both crystal systems. As with CUBIC the crystallographer judges the validity of the indexing from the information generated by HEXTET. From HEXTET the program proceeds to ORTHO.

HEXTET uses the basic equation

$$\sin^2\theta = XS + YL$$

where X and Y are related to the lattice constants, a_0 and c_0 and the X-ray wavelength, λ. For the hexagonal system $X = \lambda^2/3a_0^2$ and $Y = \lambda^2/4c_0^2$. For the tetragonal system, $X = \lambda^2/4a_0^2$ and $Y = \lambda^2/4c_0^2$. S and L are related to the Miller indices, h, k, and ℓ as follows: $S = (h^2 + hk + k^2)$ for hexagonal systems and $S = (h^2 + k^2)$ for tetragonal systems. $L = (\ell^2)$ for both the hexagonal and tetragonal system. The possible S ($S<1000$) and L ($\ell<32^2$) values are generated by HEXTET. Thus, these equations remain to be solved:

$$\sin^2\theta_i = XS_i + YL_i, \quad i = 1, \ldots, N. \qquad (4)$$

The solving procedure is straightforward. From the S and L lists are chosen values of S_1, S_2, L_1, and L_2; X and Y are determined by the simultaneous solution of the first two equations. With these values of X and Y, values of S_3 and L_3 from the S and L lists are used to calculate $\sin^2\theta_3$. Calculated $\sin^2\theta_3$ is compared with observed $\sin^2\theta_3$. If calculated $\sin^2\theta_3$ equals observed $\sin^2\theta_3$ within ± E, the assigned error, if S_3 and L_3 are different from S_2 and L_2, and if calculated $\sin^2\theta_3$ has the best agreement with observed $\sin^2\theta_3$ when compared with other calculated $\sin^2\theta_3$ values; then, S_3 and L_3 values used for calculating $\sin^2\theta_3$ are accepted. The fourth reflection is similarly indexed. Using now these four equations a new X and Y are determined by a least squares fit routine. The new X and Y are then used to index $\sin^2\theta_5$. The procedure is repeated until all reflections are indexed. However, if X and Y calculated from the first two reflections cannot be used to index all reflections, then new S and L values are used to determine other pairs of X and Y until all reflections can be indexed. The selection of S and L values is such that the largest values of X and Y are tried first. The values of X and Y cannot become vanishingly small because it is obvious that if this were so any indexing could be achieved. Thus, a lower limit is placed on X and Y by fixing the upper limits of S and L that can be tried. The present program fixes S and L at the fifth value in the ordered list of these values. When an indexing is found within E and the limited S and L values, E is reduced by one-half and another attempt is made to index the reflections with the smaller value. If no indexing is found within E an attempt is made with twice the original E, and if no success is obtained the program takes the next case. The original E is equal to the smallest difference between successive $\sin^2\theta$ values. When E has decreased to the smallest value within which an indexing can be achieved, the first five solutions within that E are found and printed. Solutions are first sought for the hexagonal case and then the tetragonal case. In general, any set of reflections will generate solutions explained by both the hexagonal and tetragonal cases.

When the program indexes the last reflection and calculates the estimates of X and Y, these values are then fixed and the program does the complete indexing over again with these fixed constants. The reason for this is that a set of indices may have been accepted for a given line, but subsequent least squares approximations may alter X and Y so much that the accepted indices are no longer the best possible indices.

If no change is made in the lattice constants then the solution is accepted and printed. If they do change then the program once again fixes a and b at the new values and reindexes. The program will go through five iterations before proceeding to the next case. Usually the X and Y are determined on the first or second iteration.

It should be noted that once a minimum error has been determined, the program will print at most five more solutions for this minimum error. This is because subsequent solutions, and quite likely these five, are usually multiples of the first solution, obtained for example, by doubling the length of one axis of the unit cell. In the hexagonal case for instance, new solutions can be obtained from the previous solutions by multiplying one axis by $\sqrt{3}$.

The printed output includes a column headed "difference" where each difference, $D_i = S_i - C_i$. That is, D_i is the difference between observed and calculated $\sin^2\theta_i$. These are used in calculating standard deviations. The standard deviation of $\sin^2\theta$ is defined by

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} D_i^2}{N-2}} \qquad (5)$$

where N is the number of angles indexed. Then, noting that if $Y = \sin^2\theta$ then $dY = \sin 2\theta d\theta$, the standard deviation of θ is defined by

$$\sigma_\theta = \sqrt{\frac{\sum_{i=1}^{N} \frac{D_i^2}{\sin^2 2\theta_i}}{N-2}} \qquad (6)$$

The standard deviations for X and Y were obtained by using the theorems of W. E. Deming,[3]. The procedure is as follows: If A is the matrix to be inverted in solving the normal equations of the least squares method and if $C = A^{-1}$ and if σ^2 is the variance of the data, then $\sigma^2_X = c_{11}\sigma^2$ and $\sigma^2_Y = c_{22}\sigma^2$ where c_{11} and c_{22} are the diagonal elements of A^{-1}. Having found $\sigma_{\hat{a}}$ and $\sigma_{\hat{b}}$ and noting that $X = \frac{\lambda^2}{3a_0^2}$ implies $\frac{da_0}{a_0} = -1/2\frac{dX}{X}$ and similarly for Y and c_0, we find σ_{a_0}, and σ_{c_0} by letting $\sigma_{a_0} = \frac{a_0\sigma_X}{2X}$ and $\sigma_{c_0} = \frac{c_0\sigma_Y}{2Y}$.

INTERPRETING THE RESULTS OF HEXTET

It seems reasonable to conclude that the correct indexing is that one that has the smallest lattice constants and gives the best agreement between observed and calculated $\sin^2\theta$ values. Since this agreement could be made vanishingly small by allowing the lattice constants to increase without limit, an upper limit as explained above has been fixed. It is best to know the expected agreement for the diffraction apparatus being used. This is done by using a known material to measure the expected standard deviation between observed and calculated $\sin^2\theta$ values. When the value of the expected standard deviation is known, it can be used as a guide in selecting the most likely correct indexing of an unknown sample.

In addition, if the empirical formula weight and the crystal density are supplied as input data, the program will calculate the number of formula units per unit cell and print these in the output under the heading "number." The size of "number" and its deviation from integral values are an aid to selecting the valid indexing.

Subroutine HEXTET uses a sorting subroutine not described in this report. This is SHARE program WDSORT, Distribution Number 1249, available through SHARE to users of large IBM computers.

SUBROUTINE ORTHO

Subroutine ORTHO will generate an indexing on an orthorhombic basis. As with CUBIC and HEXTET any set of reflections will in general yield an orthorhombic based solution. The validity is once again decided by the crystallographer. From ORTHO the program proceeds to the next crystal.

ORTHO uses the $\sin^2\theta$ values into which the input subroutine converted the 2θ values. These are then used in the basic equation

$$\sin^2\theta = XH + YK + ZL \tag{7}$$

where $H = h^2$ $K = k^2$, $L = \ell^2$, $X = \lambda^2/4a_0^2$, $Y = \lambda^2/4b_0^2$, and $Z = \lambda^2/4c_0^2$. Thus the equations to be solved are

$$\sin^2\theta_i = XH_i + YK_i + ZL_i; \quad i = 1, \ldots, N. \tag{8}$$

One of the most important aspects of this program seems to be the manner in which the first three equations are solved. The solving procedure is relatively straightforward. A set of tentative values of H_i, K_i, and L_i are selected for i = 1, 2, and 3. These are substituted into the first three equations which are then solved for X, Y, and Z. The speed of the entire program depends upon an elaborate counting procedure designed to select the <u>most probable</u> values of H, K, and L first. Once X, Y, and Z are found, these are substituted into equation 4 and tentative values of H_4, K_4, and L_4 are used to calculate $\sin^2\theta_4$. If these values agree within ± E, the assigned error, and if H_4, K_4, and L_4 differ from H_3, K_3, and L_3 and if the calculated $\sin^2\theta_4$ has the best agreement with observed $\sin^2\theta_4$ when compared with other calculated $\sin^2\theta_4$, then these tentative values of H_4, K_4, and L_4 are accepted. Using, now, these first four equations, a least squares fit routine calculates new estimate of X, Y, and Z. These are then used to try to index the fifth reflection. The procedure is repeated until all reflections are indexed. However, if X, Y, and Z calculated from the first three reflections cannot be used to index all reflections, then new values of H, K, and L are used to determine other triples of X, Y, and Z until all reflections are indexed. The selection of H, K, and L values is such that the larger values of X, Y, and Z are tried first. The values of X, Y, and Z cannot become vanishingly small because it is obvious that if this were so any indexing could be achieved. Thus a lower limit is placed on X, Y, and Z by fixing the upper limits of H, K, and L, which can be tried When an indexing is found within E and the limited H, K, and L values, E is reduced by one-half and another attempt is made to index the reflections with the smaller value. If no indexing is found within E an attempt is made with twice the original E, and if no success is obtained the program takes the next case. The original E is equal to the smallest difference between successive $\sin^2\theta$ values. When E has decreased to the smallest value within which an indexing can be achieved, the first twenty solutions within that E are found and printed.

The principle limitations of the program are those that arise from the limited list of Miller indices that are used in determining the provisional X, Y, and Z values. Only combinations of the Miller indices 0, 1, and 2 are used. The list with these indices contains 1254 usable combinations.* Usable combinations were selected on the basis that, since the $\sin^2\theta$ values are strictly increasing, some index in row i must be greater than the corresponding index in row j, for $j < i$. Once a combination was selected on the above basis, those permutations of this combination that would represent an interchange of the crystallographic axes were deleted. If it should occur that the first three reflections are dependent, e. g. , one reflection is the (001) line and one is the (002) line, then a higher value reflection is sought in order to have an independent set from which to determine provisional X, Y, and Z values. It could be that this independent set would require a Miller index larger than 2. In this case this pattern would not be properly indexed.

Included near the end of the program is a calculation to determine the number of formula units per unit cell. This calculation requires the observed density in grams per cubic centimeter and empirical gram formula weight. The result of the calculation is called "number" in the output.

---

* See Appendix A

INTERPRETING THE RESULTS OF ORTHO

All of the remarks about interpreting the results of HEXTET apply to interpreting the results of ORTHO and will not be repeated here. The formulas[4] for calculating the standard deviations are also similar to those of HEXTET (see equations 5 and 6).

THE DATA DECK

The data deck consists of three types of cards: a name card, a "cut-number" card, and "two-theta" cards.

The first card in the deck contains any information the user wishes. Usually it will contain the name of the crystalline compound, the date the data were collected, or other identifying information. The contents of this first card will be printed as the heading over the solution. Thus in the first card Columns 1-72 contain any 72 alphanumeric characters, and Columns 73-80 are blank.

The second card contains an integer in Columns 1-5, this integer being right adjusted. This integer is referred to as the "cut-number" and designates the last unresolved reflection in the data. All subsequent angles will be treated as $K_{\alpha 1}$ lines. For example, if the data consists of 36 reflections (the first 19 of which are unresolved and the rest are $K_{\alpha 1}$ reflections), then the cut-number is 19 and this number should be in Columns 4 and 5 of this second card.

In addition to the cut-number, this card may contain the X-ray wavelengths, a predicted error term, the formula weight, and the density of the compound. The $K_{\alpha 1}$ and $K_{\alpha 2}$ wavelengths are entered in Columns 6-15 and 16-25, respectively, under F10.5 formats. If the X-ray target is copper, the wavelengths may be omitted.

If the user is familiar enough with his equipment to know what the maximum error will be between observed and calculated $\sin^2 \theta_i$, this number may be supplied to the program by putting the number on this second card in Columns 26-35. If this field is left blank, the program sets this test error at 0.0005.

Two other pieces of information may be supplied on the second card. The empirical gram formula weight can be put in Columns 46-55 and the observed density in grams per cubic centimeter can be put in Columns 56-65. If this information is supplied, the program will calculate the number of formula units per unit cell, and print this in the output under the title of "number."

Thus, the second card will be:

Columns 1-5	An integer called the "cut-number", right adjusted
Columns 6-15	The $K_{\alpha 1}$ wavelength using an F10.5 format
Columns 16-25	The $K_{\alpha 2}$ wavelength using an F10.5 format
Columns 26-35	Maximum error between observed and calculated $\sin^2\theta_i$ using F10.5 format
Columns 36 45	These columns are left blank. If this data deck is to be run on program HEXTET then this field can be used as a tetragonal block to prevent a tetragonal indexing when the crystal is known to be hexagonal. (5)
Columns 46-55	The empirical gram formula weight of the compound
Columns 56-65	The observed density in grams per cubic centimeter
Columns 66-80	Blank

As an example, the following line could be the information on card two where b means blank.

bbb19bbbbbbbbbbbbbbbbbbbb0.0002bbbbbbbbbbbbbb101.1bbbbb2.106bbbbbbbbbbb

It should be noted that both of these first two cards may be entirely blank and an indexing will still be produced.

The third and following cards contain the two theta angles in degrees and decimals under a 7F10.2 format. This means seven angles to a card: the first angle in Columns 1-10, the second angle in Columns 11-20, etc., with the seventh angle in Columns 61-70.

The last card in the data deck must be blank. (See Appendix B for a complete listing of a data deck.)

As many of these data decks may be combined as is desired. The blank card at the end of each deck will ensure the processing of that crystal before the next data deck is read into memory.

For example let us suppose that a power pattern has been read and 25 reflections recorded in increasing order. If the crystal was alpha uranium, the first 5 reflections unresolved, the last 20 reflections being $K_{\alpha 1}$, and the target being copper, the data deck might look as follows:

First Card	Alpha Uranium. 10 May 1964
Second Card	5, where this integer 5 appears in the fifth column, .0002 in Columns 26-35, (this is a test error), 238.07 in Columns 46-55, (this is the formula weight), and 18.7 in Columns 56-65 is the density.
Third, 4th, 5th and 6th Cards	The angles in increasing order, under a 7F10.2 format.
Last Card	Blank card.

Acknowledgements

The authors gratefully acknowledge the many valuable suggestions offered by J. E. Schlosser during the preparation of this program.

REFERENCES

1. C. S. Barrett. Structure of Metals, McGraw-Hill, New York, 1952. p. 76.

2. J. B. Goebel and A. S. Wilson. A 709/7090 Program for Indexing X-Ray Diffraction Powder Patterns: I. The Cubic Case, HW-74393, General Electric Company, Richland, Washington, July 12, 1962.

3. W. E. Deming. Statistical Adjustment of Data, John Wiley and Sons, New York, 1946. p. 176.

4. J. B. Goebel and A. S. Wilson. A 709/7090 Program for Indexing X-Ray Diffraction Powder Patterns: III. The Orthorhombic Case, HW-89062, General Electric Company, Richland, Washington, June 2, 1964.

5. J. B. Goebel and A. S. Wilson. A 709/7090 Program for Indexing X-Ray Diffraction Powder Patterns: II. The Hexagonal-Tetragonal Case, HW-77714, General Electric Company, Richland, Washington, May 23, 1963.

APPENDIX A

The equation for the Bragg angles of an orthorhombic crystal is

$$\sin^2\theta = \frac{\lambda^2}{4}\left(\frac{h^2}{a^2} + \frac{k^2}{b^2} + \frac{\ell^2}{c^2}\right)$$

Since λ, a, b, and c are constants with h, k, and ℓ being different for different θ's, we can write this equation as

$$S_i = XH_i + YK_i + ZL_i$$

where $S_i = \sin^2\theta_i$, $X = \lambda^2/4a^2$, $Y = \lambda^2/4b^2$, $Z = \lambda^2/4c^2$, $H_i = h^2$ for the i^{th} reflection, etc. So H_i, K_i, and L_i are squares of integers and may be selected from the list 0, 1, 4, 9, 16, 25, etc. For the first three reflections the equations to be solved for X, Y, and Z are then

$$\begin{aligned} S_1 &= XH_1 + YK_1 + ZL_1 \\ S_2 &= XH_2 + YK_2 + ZL_2 \\ S_3 &= XH_3 + YK_3 + ZL_3 \end{aligned}$$

Since the S_i are the observed $\sin^2\theta_i$, the unknowns are the H_i, K_i, and L_i. If we always write the S_i in increasing order then we can make the following assertion: in row i some one of the numbers H_i, K_i, and L_i must be larger than the respective H_j, K_j, and L_j in row j for all $j < i$. This is because $S_i > S_j$ and X, Y, and Z are positive constants.

In matrix form our equations are

$$\begin{pmatrix} S_1 \\ S_2 \\ S_3 \end{pmatrix} = \begin{pmatrix} H_1 & K_1 & L_1 \\ H_2 & K_2 & L_2 \\ H_3 & K_3 & L_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} .$$

The solution then involves choosing a 3 x 3 matrix

$$M = \begin{pmatrix} H_1 & K_1 & L_1 \\ H_2 & K_2 & L_2 \\ H_3 & K_3 & L_3 \end{pmatrix}$$

whose entries are squares of integers (indeed squares of Miller indices), and then inverting this matrix.

After consultation with several crystallographers, it was decided that the vast majority of orthorhombic crystals could be indexed using the Miller indices 0, 1, and 2 in the first three lines. Therefore the matrix M can be restricted to being a 3 x 3 matrix with entries 0, 1, and 4. As we have noted before, at least one entry on each line must be larger than the corresponding entry in any previous line. Thus we do not consider matrices of the form

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 4 \\ 0 & 1 & 1 \end{pmatrix}$$

because it implies S_3 is smaller than S_2 and S_1 contrary to the ordering. In addition, matrices whose first row consists entirely of zeros are rejected.

Another criterion for rejecting matrices is the interchangeability of crystal axes in the orthorhombic system. If two columns of the matrix M are interchanged this corresponds to interchanging the respective lattice constants. Thus the matrix

$$M = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

may yield lattice constants a = 2, b = 4, and c = 6 while the matrix

$$M = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

obtained by a permutation of the last two columns would give a = 2, b = 6, and c = 4. As the assigning of the axis is completely arbitrary either matrix yields the same indexing. Thus, once a matrix M is selected, all matrices formed by permuting the columns of M are rejected. Since there are six ways of permuting the three columns we reject five of these. This leaves only about one-sixth of the original matrices as acceptable matrices.

After all of the above acceptability criteria were applied to the set of 3 x 3 matrices, there were 1254 acceptable matrices left. These were stored as octal numbers in a compact form as an integral part of a subroutine called subroutine FETCH. When called, the subroutine selected one of the matrices, unpacked it from its octal form, and presented it to the program as a usable 3 x 3 matrix with integers as entries.

Following is a listing of subroutine FETCH.

```
*       FAP
        COUNT    200
*       SUBROUTINE  FETCH AND GET
        LBL      GET,5
*
*       CALLING PROCEDURE
*
*       G=GET(N)    RETURNS NTH MATRIX IN ACCUMULATOR IN OCTAL.
*
*       CALL FETCH (N,L)  RETURNS NTH MATRIX AS FORTRAN INTEGERS
*                         IN 9 LOCATIONS STARTING WITH L AND WORKING
*                         DOWNWARD.  L MUST BE DIMENSIONED TO AT LEAST 9.
*
*       SENSE LIGHT 1 MUST BE OFF ON ENTRY TO THIS SUBROUTINE.
*
*       IF N IS TOO LARGE, A ZERO MATRIX WILL BE RETURNED.
*
        ENTRY    GET
        ENTRY    FETCH
FETCH   SLN      1
        SXA      AXT1,1                       STORE INDEX
        SXA      AXT2,2                         REGISTERS 1 AND 2
        CLA      2,4                          STORE ADDRESS OF ARRAY FOR
        STA      STD                          FETCH ENTRY
GET     SXA      RETURN,1
        CLA*     1,4                          BRING IN NUMBER OF DESIRED
        CAS      NUMBER                        MATRIX AND COMPARE WITH NUMBER
        TRA      TOOBIG                        OF MATRICES AVAILABLE.
        NOP
        PDC      0,1                          LOAD INDEX 1 WITH
        CAL      MATRIX-1,1                   PROPER INDEX AND GET NTH MATRIX.
        TZE      SLT                          IF MATRIX IS ZERO, LEAVE IT ZERO.
        SUB      =0111111111                  SUBTRACT 1-S.
SLT     SLT      1                            CHECK SL 1. IF OFF
        TRA      RETURN                       RETURN VIA GET EXIT.
        AXT      0,2                          LOAD ZERO INTO XR 2.
        LGR      9                            SHIFT RIGHT TO INITIALIZE.
LGL     LGL      3                            SHIFT NEXT WORD INTO DECREMENT.
        ANA      =07777777                    MASK OFF PREVIOUS NUMBER.
STD     STD      **,2                         STORE IN ARRAY.
        TXI      *+1,2,1                      INCREMENT INDEX AND
        TXL      LGL,2,8                      TEST FOR END.
AXT2    AXT      **,2                         RESTORE INDEX
AXT1    AXT      **,1                           REGISTERS AND
        TRA      3,4                              RETURN.
RETURN  AXT      **,1                         RESTORE INDEX FOR
        TRA      2,4                             GET RETURN.
```

```
TOOBIG PXD    0,0                              CLEAR AC AND                                  00470
       TRA    SLT                               RETURN TO UNPACK LOGIC.                      00480
MATRIX BSS    0                                START OF MATRIX LIST.                         00500
       OCT    211121112,121112122,211112122,112211122,121221112        1  00000009
       OCT    121112222,112212122,121212122,211212122,221112122        2  00000109
       OCT    112122222,211122222,221112222,221212122,212122222        3  00000209
       OCT    121112113,112121113,211121113,121112123,211112123        4  00000309
       OCT    112121123,211121123,112211123,121211123,112122113        5  00000409
       OCT    121122113,211122113,221112113,112221113,121221113        6  00000509
       OCT    212121113,121112223,112121223,211121223,112122123        7  00000609
       OCT    121122123,211122123,112212123,121212123,211212123        8  00000709
       OCT    221112123,112221123,121221123,211221123,212121123        9  00000809
       OCT    122211123,112222113,121222113,212122113,221122113       10  00000909
       OCT    122221113,112122223,121122223,211122223,221112223       11  00001009
       OCT    112221223,121221223,212121223,112222123,121222123       12  00001109
       OCT    211222123,212122123,221122123,122212123,221212123       13  00001209
       OCT    122221123,212221123,122222113,221222113,112222223       14  00001309
       OCT    121222223,212122223,221122223,122221223,122222123       15  00001409
       OCT    212222123,221222123,122222223,221222223,112113122       16  00001509
       OCT    121113122,211113122,112113221,121113221,112113121       17  00001609
       OCT    211113121,112213121,211213121,212113121,112113222       18  00001709
       OCT    121113222,112123222,121123222,211123222,122113222       19  00001809
       OCT    221113222,112213122,121213122,211213122,212113122       20  00001909
       OCT    221113122,212213122,221213122,112123221,121123221       21  00002009
       OCT    211123221,122113221,122123221,212123221,212213121       22  00002109
       OCT    122123222,212123222,221123222,113121122,113211122       23  00002209
       OCT    113121221,113212121,113211121,113122222,113221222       24  00002309
       OCT    113121222,213121222,113212122,113221122,213221122       25  00002409
       OCT    213121122,113122221,213122221,213121221,213122222       26  00002509
       OCT    123221222,121112133,211112133,112211133,121112233       27  00002609
       OCT    211112233,112211233,112122133,211122133,112212133       28  00002709
       OCT    121212133,211212133,221112133,122211133,112122233       29  00002809
       OCT    211122233,112212233,121212233,211212233,221112233       30  00002909
       OCT    122211233,112222133,211222133,212122133,122212133       31  00003009
       OCT    221212133,112222233,211222233,212122233,122212233       32  00003109
       OCT    221212233,122222133,212222133,122222233,212222233       33  00003209
       OCT    112113123,121113123,211113123,112113223,121113223       34  00003309
       OCT    112213123,121213123,211213123,122113123,212113123       35  00003409
       OCT    221113123,112123223,121123223,211123223,122113223       36  00003509
       OCT    221113223,122213123,212213123,221213123,222113123       37  00003609
       OCT    122123223,212123223,221123223,222113223,222213123       38  00003709
       OCT    222123223,112131113,121131113,211131113,112131123       39  00003809
       OCT    121131123,211131123,112131213,121131213,211131213       40  00003909
       OCT    112132113,121132113,211132113,112231113,121231113       41  00004009
       OCT    211231113,122131113,212131113,221131113,112131223       42  00004109
       OCT    121131223,211131223,112132123,121132123,211132123       43  00004209
       OCT    112231123,121231123,211231123,122131123,212131123       44  00004309
```

```
OCT       121231213,211231213,122131213,212131213,221131213     46   00004509
OCT       112232113,121232113,211232113,122132113,212132113     47   00004609
OCT       221132113,122231113,212231113,221231113,222131113     48   00004709
OCT       112132223,121132223,211132223,112231223,121231223     49   00004809
OCT       211231223,122131223,212131223,221131223,112232123     50   00004909
OCT       121232123,211232123,122132123,212132123,221132123     51   00005009
OCT       122231123,212231123,221231123,222131123,112232213     52   00005109
OCT       121232213,211232213,122132213,212132213,221132213     53   00005209
OCT       122231213,212231213,221231213,222131213,122232113     54   00005309
OCT       212232113,221232113,222132113,222231113,112232223     55   00005409
OCT       121232223,211232223,122132223,212132223,221132223     56   00005509
OCT       122231223,212231223,221231223,222131223,122232123     57   00005609
OCT       212232123,221232123,222132123,222231123,122232213     58   00005709
OCT       212232213,221232213,222132213,222231213,222232113     59   00005809
OCT       122232223,212232223,221232223,222132223,222231223     60   00005909
OCT       222232123,222232213,222232223,113121123,113211123     61   00006009
OCT       113121223,113122123,113212123,113221123,213121123     62   00006109
OCT       113122223,113221223,213121223,113222123,213122123     63   00006209
OCT       213221123,113222223,213122223,123221223,213222123     64   00006309
OCT       123222223,131112113,131211113,131112123,131211123     65   00006409
OCT       131112213,131211213,131122113,131212113,231112113     66   00006509
OCT       131221113,132211113,131112223,131211223,131122123     67   00006609
OCT       131212123,231112123,131221123,132211123,131122213     68   00006709
OCT       131212213,231112213,131221213,132211213,131222113     69   00006809
OCT       231122113,132212113,231212113,132221113,131122223     70   00006909
OCT       131212223,231112223,131221223,132211223,131222123     71   00007009
OCT       231122123,132212123,231212123,132221123,131222213     72   00007109
OCT       231122213,132212213,231212213,132221213,132222113     73   00007209
OCT       231222113,131222223,231122223,132212223,231212223     74   00007309
OCT       132221223,132222123,231222123,132222213,231222213     75   00007409
OCT       132222223,231222223,112133211,112133222,211133222     76   00007509
OCT       122133222,212133222,112133212,121133212,211133212     77   00007609
OCT       122133212,221133212,122133211,113213121,113213122     78   00007709
OCT       113123221,113123222,213123221,213123222,131113211     79   00007809
OCT       131113122,131113212,131113221,131123211,132113211     80   00007909
OCT       131113222,131213122,231113122,131123212,132113212     81   00008009
OCT       231113212,131123221,131213221,132113221,132123211     82   00008109
OCT       131123222,131213222,132113222,231113222,231213122     83   00008209
OCT       132123212,231123212,132123221,132213221,132123222     84   00008309
OCT       231123222,132213222,231213222,133212222,133211222     85   00008409
OCT       133221212,133211212,121112333,112122333,211122333     86   00008509
OCT       221112333,112222333,212122333,122222333,112113133     87   00008609
OCT       121113133,211113133,112113233,121113233,211113233     88   00008709
OCT       112123133,121123133,211123133,112213133,121213133     89   00008809
OCT       211213133,122113133,212113133,221113133,112123233     90   00008909
OCT       121123233,211123233,112213233,121213233,211213233     91   00009009
OCT       122113233,212113233,221113233,112223133,121223133     92   00009109
OCT       211223133,122123133,212123133,221123133,122213133     93   00009209
```

```
OCT      212213133,221213133,222113133,112223233,121223233      94   00009309
OCT      211223233,122123233,212123233,221123233,122213233      95   00009409
OCT      212213233,221213233,222113233,122223133,212223133      96   00009509
OCT      221223133,222123133,222213133,122223233,212223233      97   00009609
OCT      221223233,222123233,222213233,222223133,222223233      98   00009709
OCT      112311133,211311133,112311233,211311233,112312133      99   00009809
OCT      121312133,211312133,122311133,212311133,112312233     100   00009909
OCT      121312233,211312233,122311233,212311233,112322133     101   00010009
OCT      211322133,122312133,212312133,221312133,222311133     102   00010109
OCT      112322233,211322233,122312233,212312233,221312233     103   00010209
OCT      222311233,122322133,212322133,222312133,122322233     104   00010309
OCT      212322233,222312233,222322133,222322233,113121133     105   00010409
OCT      113211133,113121233,113211233,113122133,113212133     106   00010509
OCT      113221133,213121133,123211133,113122233,113212233     107   00010609
OCT      113221233,213121233,123211233,113222133,213122133     108   00010709
OCT      123212133,123221133,213221133,113222233,213122233     109   00010809
OCT      123212233,123221233,213221233,123222133,213222133     110   00010909
OCT      123222233,213222233,311112133,311112233,311122133     111   00011009
OCT      311212133,321112133,311122233,311212233,321112233     112   00011109
OCT      311222133,312122133,321212133,311222233,312122233     113   00011209
OCT      321212233,312222133,312222233,112133213,121133213     114   00011309
OCT      211133213,112133223,121133223,211133223,122133213     115   00011409
OCT      212133213,221133213,122133223,212133223,221133223     116   00011509
OCT      222133213,222133223,113213123,113123223,213123223     117   00011609
OCT      131113123,131113213,131113223,131213123,132113123     118   00011709
OCT      231113123,131123213,132113213,231113213,131123223     119   00011809
OCT      131213223,132113223,231113223,132213123,231213123     120   00011909
OCT      232113123,132123213,231123213,232113213,132123223     121   00012009
OCT      231123223,132213223,231213223,232113223,232213123     122   00012109
OCT      232123213,232123223,232213223,112331113,121331113     123   00012209
OCT      112331123,121331123,211331123,112332113,121332113     124   00012309
OCT      122331113,221331113,112331223,121331223,112332123     125   00012409
OCT      121332123,211332123,122331123,212331123,221331123     126   00012509
OCT      122332113,221332113,222331113,112332223,121332223     127   00012609
OCT      122331223,221331223,122332123,212332123,221332123     128   00012709
OCT      222331123,222332113,122332223,221332223,222331223     129   00012809
OCT      222332123,222332223,113131123,113131213,113131223     130   00012909
OCT      113132123,113231123,213131123,113132213,113231213     131   00013009
OCT      123131213,113132223,113231223,123131223,213131223     132   00013109
OCT      113232123,213132123,213231123,113232213,123132213     133   00013209
OCT      123231213,113232223,123132223,213132223,123231223     134   00013309
OCT      213231223,213232123,123232213,123232223,213232223     135   00013409
OCT      131132113,131231113,131132123,131231123,131132213     136   00013509
OCT      131231213,131232113,231132113,132231113,131132223     137   00013609
OCT      131231223,131232123,231132123,132231123,131232213     138   00013709
OCT      231132213,132231213,132232113,231232113,131232223     139   00013809
OCT      231132223,132231223,132232123,231232123,132232213     140   00013909
OCT      231232213,132232223,231232223,311131113,311131123     141   00014009
```

```
OCT      311131213,311132113,311231113,312131113,321131113    142  00014109
OCT      311131223,311132123,311231123,312131123,321131123    143  00014209
OCT      311132213,311231213,312131213,321131213,311232113    144  00014309
OCT      312132113,321132113,312231113,321231113,322131113    145  00014409
OCT      311132223,311231223,312131223,321131223,311232123    146  00014509
OCT      312132123,321132123,312231123,321231123,322131123    147  00014609
OCT      311232213,312132213,321132213,312231213,321231213    148  00014709
OCT      322131213,312232113,321232113,322132113,322231113    149  00014809
OCT      311232223,312132223,321132223,312231223,321231223    150  00014909
OCT      322131223,312232123,321232123,322132123,322231123    151  00015009
OCT      312232213,321232213,322132213,322231213,322232113    152  00015109
OCT      312232223,321232223,322132223,322231223,322232123    153  00015209
OCT      322232213,322232223,133211213,133211223,133212213    154  00015309
OCT      133221213,133212223,133221223,133222213,133222223    155  00015409
OCT      331112113,331112123,331122113,331112223,331122123    156  00015509
OCT      331212123,331222113,331122223,331222123,331222223    157  00015609
OCT      113133211,113133212,113133221,123133211,113133222    158  00015709
OCT      123133212,123133221,213133221,123133222,213133222    159  00015809
OCT      311133212,311133222,321133212,312133222,133213221    160  00015909
OCT      133213222,331113122,331113222,331213122,331123222    161  00016009
OCT      112113333,121113333,112123333,121123333,211123333    162  00016109
OCT      122113333,221113333,112223333,121223333,122123333    163  00016209
OCT      212123333,221123333,222113333,122223333,221223333    164  00016309
OCT      222123333,222223333,113121333,113122333,113221333    165  00016409
OCT      213121333,113222333,213122333,123221333,123222333    166  00016509
OCT      112133233,211133233,122133233,212133233,222133233    167  00016609
OCT      112313133,121313133,211313133,112313233,121313233    168  00016709
OCT      211313233,112323133,121323133,211323133,122313133    169  00016809
OCT      212313133,221313133,112323233,121323233,211323233    170  00016909
OCT      122313233,212313233,221313233,122323133,212323133    171  00017009
OCT      221323133,222313133,122323233,212323233,221323233    172  00017109
OCT      222313233,222323133,222323233,113123133,113213133    173  00017209
OCT      113123233,113213233,113223133,213123133,123213133    174  00017309
OCT      113223233,213123233,123213233,123223133,213223133    175  00017409
OCT      123223233,213223233,131113133,131113233,131123133    176  00017509
OCT      131213133,132113133,231113133,131123233,131213233    177  00017609
OCT      132113233,231113233,131223133,132123133,231123133    178  00017709
OCT      132213133,231213133,232113133,131223233,132123233    179  00017809
OCT      231123233,132213233,231213233,232113233,132223133    180  00017909
OCT      231223133,232123133,232213133,132223233,231223233    181  00018009
OCT      232123233,232213233,232223133,232223233,311113133    182  00018109
OCT      311113233,311123133,311213133,312113133,321113133    183  00018209
OCT      311123233,311213233,312113233,321113233,311223133    184  00018309
OCT      312123133,321123133,312213133,321213133,322113133    185  00018409
OCT      311223233,312123233,321123233,312213233,321213233    186  00018509
OCT      322113233,312223133,321223133,322123133,322213133    187  00018609
OCT      312223233,321223233,322123233,322213233,322223133    188  00018709
OCT      322223233,113311133,113311233,113312133,113321133    189  00018809
```

```
OCT    123311133,213311133,113312233,113321233,123311233   190   00018909
OCT    213311233,113322133,123312133,213312133,123321133   191   00019009
OCT    213321133,223311133,113322233,123312233,213312233   192   00019109
OCT    123321233,213321233,223311233,123322133,213322133   193   00019209
OCT    223312133,223321133,123322233,213322233,223312233   194   00019309
OCT    223321233,223322133,223322233,311312133,311312233   195   00019409
OCT    311322133,321312133,311322233,321312233,312322133   196   00019509
OCT    312322233,133211233,133212233,133222233,313121133   197   00019609
OCT    313121233,313122133,313221133,313122233,313221233   198   00019709
OCT    313222133,313222233,113133213,113133223,123133213   199   00019809
OCT    123133223,213133223,131133213,131133223,132133213   200   00019909
OCT    231133213,132133223,231133223,232133213,232133223   201   00020009
OCT    311133213,311133223,312133213,321133213,312133223   202   00020109
OCT    321133223,322133213,322133223,133213223,331113123   203   00020209
OCT    331113223,331213123,332113123,331123223,332113223   204   00020309
OCT    332213123,332123223,113331123,113331223,113332123   205   00020409
OCT    213331123,113332223,123331223,213332123,123332223   206   00020509
OCT    131331113,131331123,131331213,131332113,132331113   207   00020609
OCT    231331113,131331223,131332123,132331123,231331123   208   00020709
OCT    131332213,132331213,231331213,132332113,231332113   209   00020809
OCT    232331113,131332223,132331223,231331223,132332123   210   00020909
OCT    231332123,232331123,132332213,231332213,232331213   211   00021009
OCT    232332113,132332223,231332223,232331223,232332123   212   00021109
OCT    232332213,232332223,133231213,133231223,133232213   213   00021209
OCT    133232223,313131123,313131223,313132123,313231123   214   00021309
OCT    313132223,313231223,313232123,313232223,331132113   215   00021409
OCT    331132123,331132213,331232113,331132223,331232123   216   00021509
OCT    331232213,331232223,313133221,313133222,112133333   217   00021609
OCT    211133333,112233333,211233333,122133333,212133333   218   00021709
OCT    122233333,212233333,222133333,222233333,113123333   219   00021809
OCT    113223333,213123333,123223333,131113333,131123333   220   00021909
OCT    131213333,132113333,231113333,131223333,132123333   221   00022009
OCT    231123333,132213333,231213333,232113333,132223333   222   00022109
OCT    231223333,232123333,232213333,232223333,133211333   223   00022209
OCT    133212333,133222333,113133233,123133233,213133233   224   00022309
OCT    223133233,311133233,312133233,322133233,113313133   225   00022409
OCT    113313233,113323133,123313133,213313133,113323233   226   00022509
OCT    123313233,213313233,123323133,213323133,223313133   227   00022609
OCT    123323233,213323233,223313233,223323133,223323233   228   00022709
OCT    131313133,131313233,131323133,132313133,231313133   229   00022809
OCT    131323233,132313233,231313233,132323133,231323133   230   00022909
OCT    232313133,132323233,231323233,232313233,232323133   231   00023009
OCT    232323233,311313133,311313233,311323133,312313133   232   00023109
OCT    321313133,311323233,312313233,321313233,312323133   233   00023209
OCT    321323133,322313133,312323233,321323233,322313233   234   00023309
OCT    322323133,322323233,133213233,133223233,313123133   235   00023409
OCT    313123233,313223133,313223233,331113133,331113233   236   00023509
OCT    331123133,331213133,332113133,331123233,331213233   237   00023609
```

```
         OCT      332113233,331223133,332123133,332213133,331223233   238  00023709
         OCT      332123233,332213233,332223133,332223233,133311233   239  00023809
         OCT      133312233,133322233,313321133,313321233,313322133   240  00023909
         OCT      313322233,313133223,331133213,331133223,332133213   241  00024009
         OCT      332133223,133331213,133331223,133332213,133332223   242  00024109
         OCT      331332113,331332123,331332223,113133333,113233333   243  00024209
         OCT      123133333,213133333,123233333,213233333,223133333   244  00024309
         OCT      223233333,311133333,311233333,312133333,312233333   245  00024409
         OCT      322133333,322233333,133213333,133223333,331113333   246  00024509
         OCT      331123333,332113333,331223333,332123333,332223333   247  00024609
         OCT      313133233,323133233,133313233,133323233,313323133   248  00024709
         OCT      313323233,331313133,331313233,331323133,332313133   249  00024809
         OCT      331323233,332313233,332323133,332323233,133233333   250  00024909
         OCT      313133333,313233333,323133333,323233333             251  00025009

NUMBER   PZE      0,0,*-MATRIX      NUMBER OF MATRICES IN LIST              09000
         END                                                                09500
```

APPENDIX B

TEST EXAMPLES

The examples shown here were run as a test of the program. First is shown the exact input decks for three crystals. The output shown is reduced somewhat from the actual program format so that one page of output fits one page of this report.

SAMPLE INPUT DECKS

```
              Test Example 1       UO2        Cubic
  5
28.30      55.75      75.80      94.12      112.90      115.38      125.97

        CA(OH)2  A=3.582   B=4.902

                                                74.10        2.343
18.15      28.79      34.18      47.25      50.92       54.45       62.69

MG2SI04 SWANSON AND TATGE NBS CIRCULAR 539 V1  A=4.76 B=10.20
C=5.99
                                              140.73     3.11
17.3390  22.9008  23.8350  25.5230  29.7548  32.3141  35.6978
36.5245  38.3006  38.8509  39.7073  40.0385  41.7628  44.5051
46.6590  48.4281  50.3419  50.9135  52.2906  54.9331  56.1746
56.8560  57.9911  58.6791  60.4114
```

ERROR IS 0.0687

THE LATTICE CONSTANT IS 5.46893 ANGSTROMS.

THE LEAST SQUARES FIT GIVES AN AHAT OF 0.01933647

I	TWO TH	N	SINE SQRD TH	SINE SQRD TH HAT	DIFFERENCE
1	28.30	3.	0.05966	0.05951	0.00015
2	55.75	11.	0.21823	0.21820	0.00003
3	75.80	19.	0.37672	0.37689	-0.00018
4	94.12	27.	0.53503	0.53558	-0.00056
5	112.90	35.	0.69340	0.69428	-0.00087
6	115.38	36.	0.71431	0.71411	0.00020
7	125.97	40.	0.79368	0.79346	0.00022
8	134.98	43.	0.85343	0.85297	0.00046
9	138.25	44.	0.87303	0.87280	0.00022

Test Example 2 $Ca(OH)_2$ Hexagonal

This sample was one of three prepared to test the hexagonal subroutine. The three samples MgO_2, SnO, and $Ca(OH)_2$ were obtained as commercially available analytical grade reagents. These powders were ground and passed through a 200 mesh sieve. The samples were mounted on a North American Phillips wide range goniometer. The scanning speed was kept low at $1/4° \ min^{-1}$ to obtain good measurements of 2θ. The standard deviation of 2θ is ca. 0.50°. Suitable 2θ values for this program have also been obtained with a cylindrical powder camera in which film is used. Care must be exercised to insure a sample of small diameter so that measured reflections from the forward areas have minimum errors.

The entire output from the $Ca(OH)_2$ indexing is not present. Only the valid indexing is shown. The first indexing was hexagonal with E = 0.013092. The value of E was decreased to 0.006546 and another indexing obtained. The value of E continued to decrease until it reached 0.000818. Below this value no indexing was obtained; thus, the error was fixed at 0.000818, and the first indexing plus five more were printed. The additional five indexings were related to the first indexing by multiplying one or both lattice constants by some integer (see sample below.)

CA(OH)$_2$ A=3. 582 B=4. 902

THE FOLLOWING INDEXING IS ON THE BASIS OF THE HEXAGONAL SYSTEM WITH THE ERROR LESS THAN 0. 000818

THE UNRESOLVED WAVELENGTH USED WAS 1. 54180 AND THE K-ALPHA-ONE WAVELENGTH USED WAS 1. 54051
THE DENSITY IS 2. 343, THE FORMULA WEIGHT IS 74. 100, AND THE NUMBER IS 1. 037.

THE FIRST -0 LINES ARE UNRESOLVED. THE REST ARE K-ALPHA-ONE LINES.
AN ESTIMATE OF THE STANDARD DEVIATION OF SINE SQUARED THETA IS 0. 0002
AN ESTIMATE OF THE STANDARD DEVIATION OF THETA IS 0. 0193
AN ESTIMATE OF THE STANDARD DEVIATION OF AZERO IS 0. 1029
AN ESTIMATE OF THE STANDARD DEVIATION OF CZERO IS 0. 3937
AHAT IS 0. 061524 BHAT IS 0. 024799

THE LATTICE CONSTANT AZERO IS 3. 58575 ANGSTROMS.

THE LATTICE CONSTANT CZERO IS 4. 89119 ANGSTROMS.

I	TWO THETA	HH+KK+HK	LL	OBSERVED SINE SQUARED THETA	CALCULATED SINE SQUARED THETA	DIFFERENCE
1	18. 150	0	1	0. 02488	0. 02480	0. 00008
2	28. 790	1	0	0. 06180	0. 06152	0. 00028
3	34. 180	1	1	0. 08636	0. 08632	0. 00004
4	47. 250	1	4	0. 16060	0. 16072	-0. 00012
5	50. 920	3	0	0. 18480	0. 18457	0. 00022
6	54. 450	3	1	0. 20929	0. 20937	-0. 00008
7	62. 690	4	1	0. 27060	0. 27090	-0. 00030
8	64. 400	3	4	0. 28396	0. 28377	0. 00019

Test Example 3 $Mg2S_iO$ Orthorhombic

The data for this example were taken from "Standard X-Ray Diffraction Powder Patterns," National Bureau of Standards Circular No. 539.

The example showed how the error was divided by two until the minimum error of 0. 0009 was reached. Since no test error was provided, the program used the automatic "E-TEST" of . 0005. The error could not be divided by two again without falling below this minimum; therefore, the first 20 solutions were printed with **E** = . 0009. The first such solution is the correct indexing as indicated by the agreement between the known lattice constants and those found by the programs. This solution is shown below. The integer "number" of this indexing is 3. 87. It is noted that the other 19 "solutions" give values of "number" larger than 4. Large values of "number" are more likely to indicate invalid indexings as indicated by this example.

MG2S104 SWANSON AND TATGE NBS CIRCULAR 539 V1 A=4.76 B=10.20 C=5.99
THE ERROR IS 0.000917
THE LATTICE CONSTANT AZERO IS 4.75243 ANGSTROMS.
THE LATTICE CONSTANT BZERO IS 5.98528 ANGSTROMS.
THE LATTICE CONSTANT CZERO IS 10.21303 ANGSTROMS.

THE UNRESOLVED WAVELENGTH USED WAS 1.54180 AND THE K-ALPHA-ONE WAVELENGTH USED WAS 1.54051.
THE DENSITY IS 3.110 THE FORMULA WEIGHT IS 140.730, AND THE NUMBER IS 3.87.

THE FIRST -0 LINES ARE UNRESOLVED. THE REST ARE K-ALPHA-ONE LINES.
AN ESTIMATE OF THE STANDARD DEVIATION OF THETA IS 0.02746
AN ESTIMATE OF THE STANDARD DEVIATION OF SINE SQUARED THETA IS 0.00024
AN ESTIMATE OF THE STANDARD DEVIATION OF AZERO IS 0.00180
AN ESTIMATE OF THE STANDARD DEVIATION OF BZERO IS 0.00261
AN ESTIMATE OF THE STANDARD DEVIATION OF CZERO IS 0.00345

I	TWO THETA	HH	KK	LL	OBSERVED SINE SQUARED THETA	CALCULATED SINE SQUARED THETA	DIFFERENCE
1	17.33900	0	1	1	0.02272	0.02225	0.00047
2	22.90080	0	1	4	0.03941	0.03931	0.00010
3	23.83500	1	1	0	0.04264	0.04283	-0.00019
4	25.52300	1	1	1	0.04879	0.04852	0.00028
5	29.75480	1	1	4	0.06592	0.06558	0.00034
6	32.31410	1	0	9	0.07743	0.07746	-0.00003
7	35.69780	1	1	9	0 09395	0.09402	-0.00008
8	36.52450	1	4	1	0.09820	0.09820	-0.00000
9	38.30060	0	1	16	0.10762	0.10757	0.00005
10	38.85090	4	0	1	0.11061	0.11076	-0.00015
11	39.70730	1	4	4	0.11534	0.11527	0.00007
12	40.03850	1	0	16	0.11719	0.11728	-0.00008
13	41.76280	4	1	1	0.12705	0.12732	-0.00028
14	44.50510	1	4	9	0.14341	0.14371	-0.00030
15	46.65900	0	4	16	0.15683	0.15725	-0.00042
16	48.42810	1	0	25	0.16822	0.16847	-0.00025
17	50.34190	1	9	1	0.18090	0.18101	-0.00011
18	50.91350	1	1	25	0.18475	0.18503	-0.00028
19	52.29060	4	4	4	0.19417	0.19407	0.00010
20	54.93310	4	1	16	0.21273	0.21264	0.00009
21	56.17460	0	1	36	0.22167	0.22133	0.00034
22	56.85600	1	9	9	0.22663	0.22651	0.00011
23	57.99110	1	4	25	0.23497	0.23471	0.00026
24	58.67910	0	9	16	0.24008	0.24006	0.00002
25	60.41140	9	1	0	0.25312	0.25298	0.00014

APPENDIX C

PROGRAM LISTING

```
C      PROGRAM INDEX
*      LIST                                                               00040
*      LABEL                                                              00060
       CALL HELPER
       DIMENSION TWO TH(150),S(150),S2SQ(150),NAME(12),EL(60),ES(600)
       COMMON EL,  TWO TH,NAME,S2SQ,NCUT,XLAMDA,YLAMDA,NI,ETEST,TETBLK ,
      *S,ES,FORMWT,DENSTY
     1 CONTINUE
C      GENERATE THE L LIST OF ALL LL                                    HEX00310
       DO 5 I=1,30
     5 EL(I)=(I-1)*(I-1)
    10 CONTINUE
       CALL INPUT
    20 CONTINUE
       CALL CUBIC
    30 CONTINUE
       CALL HEXTET
       CONTINUE
       CALL ORTHO
       GO TO 10
       END
       SUBROUTINE INPUT
*      LIST
*      LABEL
       DIMENSION TWO TH(150),S(150),S2SQ(150),NAME(12),EL(60),ES(600)
       COMMON EL,  TWO TH,NAME,S2SQ,NCUT,XLAMDA,YLAMDA,NI,ETEST,TETBLK ,
      *S,ES,FORMWT,DENSTY
    50 FORMAT(12A6)
    51 FORMAT(1X,10F10.5)
    60 FORMAT(I5,2F10.5)
    70 FORMAT(7F10.3)
   550 FORMAT(I5,6F10.5)                                                HEX00380
       READ INPUT TAPE 2,50,NAME
       READINPUTTAPE2,550,NCUT,XLAMDA,YLAMDA,ETEST,TETBLK,FORMWT,DENSTY
       IF(ETEST) 562,561,562                                            HEX00490
   561 ETEST=0.0005                                                     HEX00500
   562 CONTINUE                                                         HEX00510
C                                                                       HEX00520
C                                                                       HEX00530
C                                                                       HEX00540
C      ETEST IS THE PROBABLE EXPERIMENTAL ERROR IN THE DATA.  IF UNKNOWN,HEX00550
C      LEAVE ETEST BLANK AND PROGRAM WILL SET ETEST=.0005.     IF UNCER- HEX00560
C      TAIN, GUESS LOW.  E.G. IF ERROR IS ABOUT .00025, SET ETEST=.0002. HEX00570
C      NO SOLUTION WILL BE PRINTED FOR E LESS THAN ETEST.               HEX00580
C                                                                       HEX00590
C                                                                       HEX00600
C                                                                       HEX00610
C      TETBLK WILL PREVENT THE PROGRAM FROM PRODUCING A TETRAGONAL      HEX00620
```

```
C     INDEXING UNLESS NO HEXAGONAL INDEXING HAS BEEN FOUND.  IN THE
C     LATTER CASE THE TETRAGONAL CASE WILL BE TRIED IN SPITE OF THE
C     BLOCK.  IF A TETRAGONAL INDEXING IS NOT DESIRED, PUT A POSITIVE
C     DECIMAL NUMBER, E.G. .5 IN COLUMNS 36 TO 45 OF THE CUT NUMBER      HEX00640
C     CARD. OTHERWISE LEAVE THAT SPACE BLANK.                            HEX00650
C                                                                        HEX00660
C                                                                        HEX00670
C                                                                        HEX00680
C     XLAMDA IS THE UNRESOLVED WAVELENGTH AND YLAMDA IS THE WAVELENGTH   HEX00690
C     OF THE K-ALPHA-1 RADIATION.                                        HEX00700
C                                                                        HEX00710
C                                                                        HEX00720
      IF(XLAMDA) 80,80,90
   80 XLAMDA=1.5418
      YLAMDA=1.54051
      RATIO=YLAMDA/XLAMDA                                                  00566
   90 CONTINUE
C                                                                        HEX00770
C                                                                        HEX00780
C     IF NO WAVELENGTHS ARE READ IN THE PROGRAM USES THOSE OF COPPER.    HEX00790
C                                                                        HEX00800
      DO 100 I=1,22000,7
      J=I+6
      READ INPUT TAPE 2,70,(TWO TH(K),K=I,J)
      IF(ABSF(TWOTH(I))+ABSF(TWOTH(I+1))+ABSF(TWOTH(I+2))+ABSF(TWOTH(I+3
     X))+ABSF(TWOTH(I+4))+ABSF(TWOTH(I+5))+ABSF(TWOTH(J))) 100,150,100
  100 CONTINUE
  150 DO 200 J=1,7
      N=I-J
      IF(TWO TH(N)) 250,200,250
  200 CONTINUE
  250 NI=XMINOF(N,150)
      IF(NCUT) 300,300,280
  280 DO 290 I=1,NCUT
      S(I)=(RATIO**2)*SINF(TWO TH(I)*0.008726646)**2
      S2SQ(I)=SINF(TWO TH(I)*0.017453292)**2
  290 CONTINUE
  300 NKUT=NCUT+1
      DO 310 I=NKUT,NI
      S(I)=SINF(TWO TH(I)*0.008726646)**2
      S2SQ(I)=SINF(TWO TH(I)*0.017453292)**2
  310 CONTINUE
      MARK=8
      RETURN
      END
```

```
      SUBROUTINE CUBIC
*     LIST
      DIMENSION YHAT(150),R(150),TWO TH(150)
      DIMENSION C(150),A(999),B(150,2),S(150)
      DIMENSION NAME(12),OMIT(175),BE(175),S2SQ(150),EL(60),ES(600)
      COMMON EL,  TWO TH,NAME,S2SQ,NCUT,XLAMDA,YLAMDA,NI,ETEST,TETBLK ,
     *S,ES,FORMWT,DENSTY
   30 FORMAT(I5,F15.2,F20.5,3F25.5)
   35 FORMAT( 1X,46H THE NUMBER OF FORMULA UNITS PER UNIT CELL IS  F10.2
     *)
   40 FORMAT(77H THE CRYSTAL HAS BEEN INDEXED IN THE CUBIC SYSTEM WITH A
     * LATTICE CONSTANT OF F7.4,11H ANGSTROMS./66H AN ESTIMATE OF THE ST
     *ANDARD DEVIATION OF THE LATTICE CONSTANT IS F8.6///)
   50 FORMAT(52X8HOBSERVED16X10HCALCULATED/4X1HI6X9HTWO THETA8X12HCALCUL
     *ATED N 7X44HSINE SQUARED THETA       SINE SQUARED THETA   15X10HDI
     *FFERENCE)
   60 FORMAT(77H THE CRYSTAL HAS BEEN INDEXED IN THE CUBIC SYSTEM WITH A
     * LATTICE CONSTANT OF F7.4,35H ANGSTROMS,BUT IN VIEW OF THE LARGE /
     *41H ERROR THIS INDEXING IS PROBABLY INVALID.    /66H AN ESTIMATE O
     *F THE STANDARD DEVIATION OF THE LATTICE CONSTANT IS F8.6///)
   70 FORMAT(10H0 ERROR IS    F9.4//)
   80 6ORMAT(12A6)
   90 FORMAT(1H1//24X12A6///)
      +O 100 I=1,125
      OMIT(I)=8*I-1
  100 CONTINUE
      DO 110 I=1,31
      OMIT(I+125)=4*(8*I-1)
  110 CONTINUE
      DO 120 I=1,7
      OMIT(I+156)=16*(8*I-1)
  120 CONTINUE
      DO 130 I=1,2
      OMIT(I+163)=64*(8*I-1)
  130 CONTINUE
      DO 180 I=1,164
      IF(OMIT(I)-OMIT(I+1))180,180,140
  140 KB=OMIT(I)
      OMIT(I)=OMIT(I+1)
      OMIT(I+1)=KB
      JB=I
  150 IF(JB-1) 180,180,160
  160 IF(OMIT(JB)-OMIT(JB-1))170,180,180
  170 KB=OMIT(JB)
      OMIT(JB)=OMIT(JB-1)
      OMIT(JB-1)=KB
      JB=JB-1
      GO TO 150
```

```
180 CONTINUE
    9=0
    DO 220 J=1,165
    N=I+1
    40 210 I=N,999
    Z=FLOATF(I)
    IF(Z-OMIT(J))190,200,190
190 A(I)=I
    GO TO 210
200 A(I)=0.0000
    GO TO 220
210 CONTINUE
220 CONTINUE
    E=0.2
    U=0.0
    V=0.0
    YLE=E
    NB=500
    1SSIGN 490 TO LGO
    ASSIGN 510 TO L
    ASSIGN 450 TO M
    MSW=0
    KAY=NI/2
330 K=(NI/2)-1
340 K=K+1
    N=A(K)
    I=(NI/2)-1
    J=(NI/2)-1
    IF(N) 350,340,350
350 IF(N-NB) 360,440,440
360 X=S(KAY)/FLOATF(N)
370 I=I+1
    IF(I-NI) 390,390,380
380 GO TO LGO,(490,600)
390 C(I)=S(I)/X
400 J=J+1
    IF(J-999) 410,340,410
410 IF(C(I)-A(J)-E) 420,420,400
420 IF(C(I)-A(J)+E) 340,430,430
430 B(I,1)=S(I)
    B(I,2)=A(J)
    GO TO 370
440 GO TO M,(450,540)
450 IF(MSW+1) 460,470,460
460 IF(MSW) 480,470,480
470 YLE=E
    E=E*2.
    MSW=-1
    GO TO 330
```

```
480 ASSIGN 540 TO M                                          01090
    GO TO 530                                                01100
490 IF(MSW) 500,590,500                                      01110
500 GO TO L,(510,560)                                        01120
510 IF(MSW-1) 520,590,520                                    01130
520 ASSIGN 560 TO L                                          01140
    MSW=1                                                    01150
530 U=MIN1F(E,YLE)                                           01160
    V=MAX1F(E,YLE)                                           01170
    E=(U+V)/2.                                               01180
    YLE=E                                                    01185
    ASSIGN 540 TO M                                          01190
    ASSIGN 560 TO L                                          01200
    GO TO 330                                                01210
540 U=E                                                      01220
    YLE=E                                                    01225
    E=(E+V)/2.                                               01230
    YLE=E                                                    01235
    IF(V-U-.05) 550,550,330                                  01240
550 E=V                                                      01250
    ASSIGN 600 TO LGO                                        01260
    GO TO 330                                                01270
560 V=E                                                      01280
570 IF(V-U-.05) 600,600,580                                  01290
580 E=(E+U)/2.                                               01300
    GO TO 330                                                01310
590 YLE=E                                                    01320
    E=E/2.                                                   01330
    MSW=1                                                    01340
    GO TO 330                                                01350
600 SUMSQ=0.0                                                01360
    DO 610 I=KAY,NI                                          01370
610 SUMSQ=SUMSQ+(B(I,2)*B(I,2))                              01380
    DNUM=0.0                                                 01390
    DO 620 I=KAY,NI                                          01400
620 DNUM=DNUM+(B(I,1)*B(I,2))                                01410
    AHAT=DNUM/SUMSQ                                          01420
    SUM=0.0
    SUMSQ=0.
    DO 630 I=1,NI                                            01430
    B(I,1)=S(I)                                              01440
    B(I,2)=XINTF((S(I)/AHAT)+.5)                             01450
    YHAT(I)=AHAT*B(I,2)                                      01460
    R(I)=B(I,1)-YHAT(I)                                      01470
    SUM=SUM+R(I)**2
    SUMSQ=SUMSQ+B(I,2)**2
630 CONTINUE                                                 01480
    DEGFRD=NI-1
    SIGMAA=SQRTF(SUM/(DEGFRD*SUMSQ))
    XLC=YLAMDA/(2.*SQRTF(AHAT))
```

```
      WRITE OUTPUT TAPE 3,90,NAME
      WRITE OUTPUT TAPE 3,70,E
      ZNMBER= A0ERO**3.0 *DENSTY* 0.6023/FORMWT
      WRITE OUTPUT TAPE 3,35,ZNMBER
      9F(E-.3) 632,632,634
  632 WRITE OUTPUT TAPE 3,40,XLC,SIGMAA
      GO TO 636
  634 WRITE OUTPUT TAPE 3,60,XLC,SIGMAA
  636 WRITE OUTPUT TAPE 3,50
      DO 660 I=1,NI
      IF(XMODF(I+6,40))650,640,650
  640 WRITE OUTPUT TAPE 3,90,NAME
      WRITE OUTPUT TAPE 3,50
  650 WRITE OUTPUT TAPE 3,30, (I,TWO TH(I),B(I,2),S(I),YHAT(I),R(I))
  660 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HEXTET
C     PROGRAM TO DETERMINE CRYSTAL INDEXES, HEXAGONAL CASE              HEX00040
      DIMENSION TWO TH(150),S(150),S2SQ(150),NAME(12),EL(60),ES(600) ,
     *O(150),D(150),A(150,2),KA(150,2),ASTOR(25),CSTOR(25),FACTOR(25),
     *C(150)
      COMMON EL,  TWO TH,NAME,S2SQ,NCUT,XLAMDA,YLAMDA,NI,ETEST,TETBLK ,
     *S,ES,FORMWT,DENSTY
  500 FORMAT(44X9H OBSERVED15X10HCALCULATED/                            HEX00080
     *2X1HI4X9HTWO THETA4X8HHH+KK+HK5X2HLL5X18HSINE SQUARED THETA       HEX00090
     X5X18HSINE SQUARED THETA8X10HDIFFERENCE//)                         HEX00100
  505 FORMAT(44X9H OBSERVED15X10HCALCULATED/                            HEX00110
     *2X1HI4X9HTWO THETA4X5HHH+KK   8X2HLL5X18HSINE SQUARED THETA       HEX00120
     X X18HSINE SQUARED THETA8X10HDIFFERENCE//)                         HEX00130
  510 FORMAT(1XI3,F11.3,2I10,F20.5,F23.5,F20.5)                         HEX00090
  511 FORMAT(90H0THE FOLLOWING INDEXING IS ON THE BASIS OF THE TETRAGCNAHEX00150
     *L SYSTEM WITH THE ERROR LESS THAN F8.6)                           HEX00160
  512 FORMAT(89H0THE FOLLOWING INDEXING IS ON THE BASIS OF THE HEXAGONALHEX00170
     * SYSTEM WITH THE ERROR LESS THAN F8.6)                            HEX00180
  515 6ORMAT(64H AN ESTIMATE OF THE STANDARD DEVIATION OF SINE SQUARED THEX00190
     *HETA IS F10.4/51H AN ESTIMATE OF THE STANDARD DEVIATION OF THETA IHEX00200
     *S F10.4/51H AN ESTIMATE OF THE STANDARD DEVIATION OF AZERO IS     HEX00210
     *F10.4/51H AN ESTIMATE OF THE STANDARD DEVIATION OF CZERO IS       HEX00220
     *F10.4/9H AHAT IS F10.6,2X9H BHAT IS F10.6/31H0THE LATTICE CONSTANTHEX00230
     * AZERO IS F10.5,11H ANGSTROMS./31H0THE LATTICE CONSTANT CZERO IS  HEX00240
     *F10.5,11H ANGSTROMS.//)                                           HEX00250
  520 FORMAT(36H0THE UNRESOLVED WAVELENGTH USED WAS F8.5,41H AND THE K-AHEX00260
     *LPHA-ONE WAVELENGTH USED WAS F8.5/16H THE DENSITY IS F6.3,1H,,22H
     *THE FORMULA WEIGHT IS F8.3,21H, AND THE NUMBER IS    F6.3, 1H.)
  525 FORMAT(11H0THE FIRST I3,53H LINES ARE UNRESOLVED. THE REST ARE K-AHEX00340
     *LPHA-1 LINES. )                                                   HEX00350
  537 FORMAT(92H0THE PROGRAM IS UNABLE TO INDEX THE CRYSTAL ON A HEXAGONHEX00280
     *AL BASIS WITHIN THE MAXIMUM ERROR OF F10.5)                       HEX00290
  538 FORMAT(93H0THE PROGRAM IS UNABLE TO INDEX THE CRYSTAL ON A TETRAGOHEX00300
     *NAL BASIS WITHIN THE MAXIMUM ERROR OF F10.5)                      HEX00310
  539 FORMAT(64H0THE PROGRAM IS UNABLE TO INDEX THE CRYSTAL WITHIN THE EHEX00320
     *RROR OF F8.6)                                                     HEX00330
  558 FORMAT(1H1/24X12A6//)                                             HEX00400
      NSTORE=0                                                          HEX00470
      KKKK=0                                                            HEX00820
C     GET SET FOR THE HEXAGONAL CASE.                                   HEX01070
      R=1.0                                                             HEX01080
      KT=1                                                              HEX01090
C     GENERATE THE S LIST OF ALL HH+KK+HK                               HEX01100
  584 I=0                                                               HEX01110
      NAY=1                                                             HEX01120
      KAY=1                                                             HEX01130
      DO 592 M=1,32                                                     HEX01140
      DO 591 K=1,M                                                      HEX01150
```

```
      I=I+1
      ES(I)=(M-1)*(M-1)+(K-1)*(K-1)+KT*(M-1)*(K-1)
  591 CONTINUE
  592 CONTINUE
      DO 593 K=1,25
      ASTOR(K)=0.
      CSTOR(K)=0.
  593 +ONTINUE
      ASSIGN 727 TO KSKIP
C     ORDER THE S LIST
      CALL SORTAC(ES,528)
      LP=0
      DO 3 I=2,277
  594 IF(ES(I-1)-ES(I)) 3,1,800
    1 LP=LP+1
      M=529-LP
      DO 2 J=I,M
      ES(J-1)=ES(J)
    2 CONTINUE
      GO TO 594
    3 CONTINUE
COMMANDS 594 TO 3 DELETE DUPLICATE INTEGERS.
      NES=277
      NEL=30
      B=1.0
      DO 4  I=2,NI
      D=ABSF(S(I)-S(I-1))
      B=MIN1F(B,D)
    4 CONTINUE
C      B IS THE MINIMUM DIFFERENCE BETWEEN S(I) AND S(I+1).
    5 E=.49*B
COMMANDS   TO HERE FIX THE MAXIMUM ERROR E
      YLE=4.*E
      KAT=0
C     KAT IS INCREASED BY 1 WHEN WE FAIL TO INDEX THE CRYSTAL.
      NOD=0
      ASSIGN 8 TO IGO
C     THE VALUE OF IGO DETERMINES WHETHER THE ERROR IS BEING DECREASED
C     OR IS FIXED.
    6 KAP=0
C     KAP IS INCREASED BY 1 EACH ITERATION.
      IF(KAT) 7,7,12
C     KAT=1 MEANS THE MINIMUM ERROR HAS BEEN DETERMINED AND THE
C     PROGRAM IS NOW FINDING ALL SOLUTIONS FOR THAT FIXED ERROR.
    7 GO TO IGO,(8,9)
    8 E=YLE/2.0
      IF(E-ETEST) 250,250,9
    9 MI=1
```

```
10 MI=MI+1                                                      HEX01630
   IF(MI-5) 11,11,250                                           HEX01640
12 NOD=NOD+1                                                    HEX01650
   IF(NOD-5) 35,35,250                                          HEX01660
11 NUT=1                                                        HEX01670
   MUT=0                                                        HEX01680
   ASSIGN 22 TO KUP                                             HEX01690
   ASSIGN 28 TO LUP                                             HEX01700
   KQ=MI-1                                                      HEX01710
15 KU=MI                                                        HEX01720
20 KV=0                                                         HEX01730
22 KV=KV+1                                                      HEX01740
   IF(KV-KQ) 26,26,70                                           HEX01750
26 KX=0                                                         HEX01760
28 KX=KX+1                                                      HEX01770
   IF(KX-KQ) 34,34,32                                           HEX01780
32 GO TO KUP,(22,70)                                            HEX01790
34 KY=0                                                         HEX01800
35 KY=KY+1                                                      HEX01810
   IF(KY-KQ) 40,40,36                                           HEX01820
36 GO TO LUP,(28,70)                                            HEX01830
40 GO TO (42,44,46,48,50,52,54,56,58,60,62,64,66,68,69),NUT     HEX01840
42 IE=KU                                                        HEX01850
   JE=KV                                                        HEX01860
   NE=KX                                                        HEX01870
   KE=KY                                                        HEX01880
   GO TO 90                                                     HEX01890
44 IE=KV                                                        HEX01900
   JE=KU                                                        HEX01910
   NE=KX                                                        HEX01920
   KE=KY                                                        HEX01930
   GO TO 90                                                     HEX01940
46 IE=KX                                                        HEX01950
   JE=KV                                                        HEX01960
   NE=KU                                                        HEX01970
   KE=KY                                                        HEX01980
   GO TO 90                                                     HEX01990
48 IE=KV                                                        HEX02000
   JE=KX                                                        HEX02010
   NE=KY                                                        HEX02020
   KE=KU                                                        HEX02030
   GO TO 90                                                     HEX02040
50 IE=KU                                                        HEX02050
   JE=KV                                                        HEX02060
   NE=KX                                                        HEX02070
   KE=KY                                                        HEX02080
   GO TO 90                                                     HEX02090
52 IE=KX                                                        HEX02100
```

```
      JE=KU
      NE=KY
      KE=KV
      GO TO 90
   54 IE=KU
      JE=KX
      NE=KV
      KE=KY
      GO TO 90
   56 IE=KX
      JE=KY
      NE=KU
      KE=KV
      GO TO 90
   58 IE=KX
      JE=KU
      NE=KV
      KE=KY
      GO TO 90
   60 IE=KU
      JE=KX
      NE=KY
      KE=KV
      GO TO 90
   62 IE=KU
      JE=KV
      NE=KX
      KE=KY
      GO TO 90
   64 IE=KU
      JE=KV
      NE=KY
      KE=KX
      GO TO 90
   66 IE=KU
      JE=KY
      NE=KV
      KE=KX
      GO TO 90
   68 IE=KY
      JE=KU
      NE=KV
      KE=KX
      GO TO 90
   69 IE=KU
      JE=KV
      NE=KX
      KE=KY
```

```
      GO TO 90                                                          HEX02590
   70 NUT=NUT+1                                                         HEX02600
      IF(NUT-14) 72,72,10                                               HEX02610
   72 MUT=MUT+1                                                         HEX02620
      GO TO (15,15,15,74,26,26,26,26,26,76,34,34,34,69),MUT             HEX02630
   74 ASSIGN 70 TO KUP                                                  HEX02640
      GO TO 26                                                          HEX02650
   76  ASSIGN 70 TO LUP                                                 HEX02660
      GO TO 34                                                          HEX02670
   90 T=ES(IE)*EL(KE)-ES(NE)*EL(JE)                                     HEX02680
C                                                                       HEX02690
COMMANDS 9  THROUGH 76 COUNT THROUGH ALL 2X2 MATRICES WITH ENTRIES LESS HEX02700
C     THAN MI+2                                                         HEX02710
C                                                                       HEX02720
C     T IS THE COEFFICIENT DETERMINANT                                  HEX02730
C     IF T=0. WE HAVE DEPENDENCE                                        HEX02740
      KAN=KAY+NAY                                                       HEX02750
C     X AND Y ARE DETERMINED FROM LINES KAY AND KAN.                    HEX02760
  100 IF(T) 110,101,110                                                 HEX02770
COMMANDS 101 THROUGH 109 CHECK FOR DEPENDENCE                           HEX02780
  101 IF(ES(IE)) 103,102,103                                            HEX02790
  102 IF(EL(JE)) 104,35,104                                             HEX02800
  103 SA=ES(IE)/ES(NE)                                                  HEX02810
      GO TO 105                                                         HEX02820
  104 SA=EL(JE)/EL(KE)                                                  HEX02830
  105 IF(SA-(S(KAY)/S(KAN))-0.001) 106,35,35                            HEX02840
  106 IF(SA-(S(KAY)/S(KAN))+0.001)35,35,107                             HEX02850
  107 NAY=NAY+1                                                         HEX02860
      IF(KAY+NAY-NI) 5,5,310                                            HEX02870
  110 X=(S(KAY)*EL(KE)-S(KAN)*EL(JE))/T                                 HEX02880
      IF(X-.001) 35,35,115                                              HEX02890
  115 Y=(S(KAN)*ES(IE)-S(KAY)*ES(NE))/T                                 HEX02900
      IF(Y-.001) 35,35,120                                              HEX02910
  120 KP=0                                                              HEX02920
  124 NND=0                                                             HEX02930
C     NND=0 UNTIL WE FIND AN ACCEPTABLE SET OF INDICES AT COMMANDS 244  HEX02940
C     AND FOLLOWING.                                                    HEX02950
      OLD DE=1.1                                                        HEX02960
  125 KP=KP+1                                                           HEX02970
C     KP IS THE NUMBER OF THE REFLECTION CURRENTLY BEING INDEXED        HEX02980
      YLE=E                                                             HEX02990
      IF(KP-4) 130,130,126                                              HEX03000
  126 KP=4                                                              HEX03010
      GO TO 600                                                         HEX03020
  130 NR=0                                                              HEX03030
  140 NR=NR+1                                                           HEX03040
      IF(NR-NES) 141,141,238                                            HEX03050
  141 IF(X*ES(NR)-S(KP)-E)150,150,238                                   HEX03060
```

```
  150 NS=0
  160 NS=NS+1
      IF(NS-NEL) 205,205,140
  205 C(KP)=X*ES(NR)+Y*EL(NS)
C(KP) IS THE CALCULATED SINE SQUARED THETA TO COMPARE WITH THE
C     OBSERVED SINE SQUARED THETA.
  210 IF(C(KP)-S(KP)-E) 220,220,140
  220 IF(C(KP)-S(KP)+E) 160,233,233
  233 DE=ABSF(S(KP)-C(KP))
      IF(OLD DE-DE) 160,160,234
  234 IF(KP-1) 244,244,246
  236 A(KP,1)=B1
      A(KP,2)=B2
C     WE ACCEPT THE TWO INDICES ES(NR) AND EL(NS) SO GO INCREASE KP
C     BY 1 AND TRY TO INDEX THE NEXT REFLECTION.
      GO TO 124
  238 IF(NND) 35,35,236
  244 B1=ES(NR)
      B2=EL(NS)
      OLD DE=DE
      NND=1
      GO TO 160
COMMAND 244 STORES THE CURRENT BEST SELECTION OF INDICES FOR LINE KP
  246 IF(A(KP-1,1)-ES(NR)) 244,247,244
  247 IF(A(KP-1,2)-EL(NS)) 244,248,244
COMMANDS 246 AND 247 AVOID DUPLICATE SETS OF INDICES.
  248 GO TO 160
  250 KAT=KAT+1
      IF(KAT-1) 260,260,265
  260 E=2.*E
      ASSIGN 721 TO KSKIP
COMMANDS 250 AND 260 ACKNOWLEDGE THE FIRST UNSUCCESSFUL E AND SET THE
C     PROGRAM TO FIND AND PRINT ALL SOLUTIONS OBTAINABLE FOR THE LAST
C     SUCCESSFUL E.
      ASSIGN 9 TO IGO
      GO TO 9
  265 IF(KT) 267,267,702
C     KT=0 IS THE TETRAGONAL CASE, KT=1 IS THE HEXAGONAL CASE.
  267 IF(KAT-1) 260,260,268
  268 IF(KKKK) 790,790,795
  310 XKT=KT
      IF(XKT-0.5) 312,312,311
  311 WRITE OUTPUT TAPE 3,558,NAME
      WRITE OUTPUT TAPE 3,537,E
      GO TO 702
C
COMPLETE FAILURE FOR HEXAGONAL CASE.  GO TO TETRAGONAL CASE.
C
  312 WRITE OUTPUT TAPE 3,558,NAME
```

```
      WRITE OUTPUT TAPE 3,538,E                                         HEX03560
      GO TO 795
C                                                                       HEX03580
C            COMPLETE FAILURE FOR BOTH CASES.                           HEX03590
C                                                                       HEX03600
C                                                                       HEX03610
COMMANDS 600 TO 628 PREPARE FOR LEAST SQUARES CALCULATION OF            HEX03620
C     AHAT AND BHAT.                                                    HEX03630
  600 ESIX=S(1)*A(1,1)+S(2)*A(2,1)+S(3)*A(3,1)                          HEX03640
      ESIY=S(1)*A(1,2)+S(2)*A(2,2)+S(3)*A(3,2)                          HEX03650
      YX=A(1,1)*A(1,2)+A(2,1)*A(2,2)+A(3,1)*A(3,2)                      HEX03660
      YI=A(1,2)**2+A(2,2)**2+A(3,2)**2                                  HEX03670
      XI=A(1,1)**2+A(2,1)**2+A(3,1)**2                                  HEX03680
  605 ESIX=ESIX+S(KP)*A(KP,1)                                           HEX03690
      ESIY=ESIY+S(KP)*A(KP,2)                                           HEX03700
      YX=YX+A(KP,1)*A(KP,2)                                             HEX03710
      YI=YI+A(KP,2)**2                                                  HEX03720
      XI=XI+A(KP,1)**2                                                  HEX03730
  610 AHAT=(ESIX*YI-ESIY*YX)/(XI*YI-YX**2)                              HEX03740
      BHAT=(ESIY-YX*AHAT)/YI                                            HEX03750
C     AHAT AND BHAT ARE THE LEAST SQUARES ESTIMATES OF A AND C IN THE   HEX03760
C     EQUATION S(I)=A*ES(I)+C*EL(I)                                     HEX03770
  628 IF(KP-NI) 629,706,706                                             HEX03780
C                                                                       HEX03790
COMMAND 628 ASKS IF WE ARE DONE.  IF NOT, COMMANDS 629 THROUGH 690 INDEXHEX03800
C     THE REST OF THE REFLECTIONS.                                      HEX03810
C                                                                       HEX03820
  629 KP=KP+1                                                           HEX03830
  657 ND=0                                                              HEX03840
      OLD DE=1.1                                                        HEX03850
  660 IR=0                                                              HEX03860
  665 IR=IR+1                                                           HEX03870
COMMAND 657 SAYS THIS IS THE FIRST ATTEMPT TO INDEX THIS REFLECTION     HEX03880
  666 IF(IR-NES) 668,668,667                                            HEX03890
  667 IF(ND) 800,685,690                                                HEX03900
COMMAND 667 ASKS IF WE FAILED TO INDEX THIS REFLECTION YES,YES,NO       HEX03910
  668 IS=0                                                              HEX03920
  669 IS=IS+1                                                           HEX03930
      IF(IS-NEL)670,670,665                                             HEX03940
  670 C(KP)=AHAT*ES(IR)+BHAT*EL(IS)                                     HEX03950
  671 IF(C(KP)-S(KP)-E) 675,675,672                                     HEX03960
  672 IF(EL(IS)) 800,673,665                                            HEX03970
  673 IF(ND) 800,685,690                                                HEX03980
  675 IF(C(KP)-S(KP)+E)669,680,680                                      HEX03990
  680 DE=ABSF(S(KP)-C(KP))                                              HEX04000
      IF(OLD DE-DE) 669,669,687                                         HEX04010
C                                                                       HEX04020
C     WE ASK IF THE NEW SET OF INDICES IS BETTER THAN THE PREVIOUS SET. HEX04030
C                                                                       HEX04040
  685 IF(KAP) 800,35,35                                                 HEX04050
```

```
C      IS THIS THE FIRST ITERATION                                        HEX04060
  686 B1=ES(IR)                                                           HEX04070
      B2=EL(IS)                                                           HEX04080
      OLD DE=DE                                                           HEX04090
      ND=1                                                                HEX04100
      GO TO 669                                                           HEX04110
COMMAND 686 STORES PRESENT BEST INDICES AS B1 AND B2.                     HEX04120
  687 IF(A(KP-1,1)-ES(IR)) 686,688,686                                    HEX04130
  688 IF(A(KP-1,2)-EL(IS)) 686,689,686                                    HEX04140
COMMANDS 687 AND 688 CHECK FOR DUPLICATE SETS OF INDICES.                 HEX04150
  689 GO TO 669                                                           HEX04160
  690 A(KP,1)=B1                                                          HEX04170
      A(KP,2)=B2                                                          HEX04180
COMMAND 690 ACCEPTS B1 AND B2 AS THE BEST SET OF INDICES FOR LINE KP      HEX04190
      IF(KAP) 605,605,691                                                 HEX04200
C      IS THIS THE FIRST ITERATION YES,YES,NO                             HEX04210
  691 IF(KP-NI) 629,692,692                                               HEX04220
  692 ESIX=0.                                                             HEX04230
      ESIY=0.                                                             HEX04240
      YX=0.                                                               HEX04250
      YI=0.                                                               HEX04260
      XI=0.                                                               HEX04270
      DO 693 L=1,NI                                                       HEX04280
      ESIX=ESIX+S(L)*A(L,1)                                               HEX04290
      ESIY=ESIY+S(L)*A(L,2)                                               HEX04300
      YX=YX+A(L,1)*A(L,2)                                                 HEX04310
      YI=YI+A(L,2)**2                                                     HEX04320
      XI=XI+A(L,1)**2                                                     HEX04330
  693 CONTINUE                                                            HEX04340
      X=AHAT                                                              HEX04350
      Y=BHAT                                                              HEX04360
      GO TO 610                                                           HEX04370
COMMANDS 692 TO 693 PERFORM ALL LEAST SQUARES FITS EXCEPT THE FIRST       HEX04380
  701 IF(KKKK) 584,584,795
C                                                                         HEX04400
COMMAND 701 SKIPS THE TETRAGONAL CASE IF A HEXAGONAL SOLUTION HAS BEEN    HEX04410
C      FOUND WITH E LESS THAN .0000.                                      HEX04420
  702 KT=0                                                                HEX04430
      NSTOR=0                                                             HEX04440
      ASSIGN 727 TO KSKIP                                                 HEX04450
      R=0.8660255                                                         HEX04460
      IF(TETBLK) 584,584,701                                              HEX04470
COMMAND 702 BRANCHES THE PROGRAM INTO THE TETRAGONAL CASE.                HEX04480
C                                                                         HEX04490
  703 IF(KAP-5) 704,704,709                                               HEX04500
C      IS THIS THE SIXTH ITERATION NO,NO,YES                              HEX04510
  704 U=ABSF(X-AHAT)                                                      HEX04520
      V=ABSF(Y-BHAT)                                                      HEX04530
```

```
      IF(U+V-.002) 709,709,705                                          HEX04540
  705 X=AHAT                                                            HEX04550
      Y=BHAT                                                            HEX04560
      GO TO 629                                                         HEX04570
  706 KAP=KAP+1                                                         HEX04580
C     THE PROGRAM ADVANCES TO 706 UPON SUCCESSFULLY INDEXING THE CRYSTALHEX04590
      KP=0                                                              HEX04600
      IF(KAP-1) 629,629,703                                             HEX04610
  709 DO 710 K=1,NI                                                     HEX04620
      O(K)=AHAT*A(K,1)+BHAT*A(K,2)                                      HEX04630
  710 CONTINUE                                                          HEX04640
C     O(K) IS THE CALCULATED SINE SQUARED THETA.                        HEX04650
  720 DO 730 K=1,NI                                                     HEX04660
      D(K)=S(K)-O(K)                                                    HEX04670
  730 CONTINUE                                                          HEX04680
      SIGMA=0.                                                          HEX04690
      SIGTH=0.                                                          HEX04700
      DO 731 I=1,NI                                                     HEX04710
      SIGMA=SIGMA+D(I)**2                                               HEX04720
      SIGTH=SIGTH+((D(I)**2)/S2SQ(I))                                   HEX04730
  731 CONTINUE                                                          HEX04740
      SIGMA=SIGMA/FLOATF(NI-2)                                          HEX04750
      SIGMA=SQRTF(SIGMA)                                                HEX04760
      SIGTH=SIGTH/FLOATF(NI-2)                                          HEX04770
      SIGTH=SQRTF(SIGTH)*57.2957795                                     HEX04780
      SIGHAT=YI*(SIGTH**2)/(XI*YI-YX**2)                                HEX04790
      SIGHBT=XI*SIGHAT/YI                                               HEX04800
      SIGHAT=SQRTF(SIGHAT)                                              HEX04810
      SIGHBT=SQRTF(SIGHBT)                                              HEX04820
COMMANDS FOLLOWING 731 CALCULATE THE VARIOUS STANDARD DEVIATIONS.       HEX04830
      AZERO=YLAMDA  *R /SQRTF(3.0*AHAT)                                 HEX04840
      CZERO=YLAMDA /(2.*SQRTF(BHAT))                                    HEX04850
      GO TO KSKIP,(721,727)                                             HEX04860
  721 ASTORE = AZERO                                                    HEX04880
      CSTORE = CZERO                                                    HEX04890
      DO 723 I=1,25                                                     HEX04900
      DO 722 K=1,20                                                     HEX04910
      FACTOR(K)=SQRTF(ES(K))                                            HEX04920
      IF(ABSF(ASTORE-FACTOR(K)*ASTOR(I))-0.001) 1723,1723,722           HEX04930
  722 CONTINUE                                                          HEX04940
  723 CONTINUE                                                          HEX04950
 1723 CONTINUE                                                          HEX04960
      DO 725 I=1,25                                                     HEX04970
      DO 724 K=1,20                                                     HEX04980
      FACTOR(K)=SQRTF(EL(K))                                            HEX04990
      IF (ABSF(CSTORE-FACTOR(K)*CSTOR(I))-0.001) 780,780,724            HEX05000
  724 CONTINUE                                                          HEX05010
  725 CONTINUE                                                          HEX05020
```

```
      IF(KT) 800,1725,1740                                              HEX05030
 1725 CONTINUE                                                          HEX05040
      DO 1727 I=1,25                                                    HEX05050
      DO 1726 K=1,20                                                    HEX05060
      FACTOR(K)=K                                                       HEX05070
      IF(ABSF(ASTORE- FACTOR(K)*CSTOR(I))-.001) 1729, 1729,1726         HEX05080
 1726 CONTINUE                                                          HEX05090
 1727 CONTINUE                                                          HEX05100
      GO TO 1740                                                        HEX05110
 1729 CONTINUE                                                          HEX05120
      DO 1732 I=1,25                                                    HEX05130
      DO 1731 K=1,20                                                    HEX05140
      FACTOR(K)=K                                                       HEX05150
      IF(ABSF(CSTORE-FACTOR(K)*ASTOR (I))-.001) 780,780,1731            HEX05160
 1731 CONTINUE                                                          HEX05170
 1732 CONTINUE                                                          HEX05180
 1740 CONTINUE                                                          HEX05190
      NSTOR=NSTOR+1                                                     HEX05200
      ASTOR(NSTOR)=ASTORE                                               HEX05205
      CSTOR(NSTOR)=CSTORE                                               HEX05210
      IF(NSTOR-25) 726,726,702                                          HEX05220
  726 CONTINUE                                                          HEX05230
  727 CONTINUE                                                          HEX05240
C                                                                       HEX01040
C     ONCE THE MINIMUM ERROR HAS BEEN DETERMINED, THE COMMANDS FROM
C     721 TO 727 WILL ELIMINATE SOLUTIONS WHICH ARE MULTIPLES OF
C     PREVIOUSLY ACCEPTED SOLUTIONS FOR THAT ERROR.
C                                                                       HEX01060
      SIGAZO=AZERO*SIGHAT/(2.*AHAT)                                     HEX05250
      SIGCZO=CZERO*SIGHBT/(2.*BHAT)                                     HEX05260
      DO 732 I=1,NI                                                     HEX05270
      KA(I,1)=A(I,1)                                                    HEX05280
      KA(I,2)=A(I,2)                                                    HEX05290
  732 CONTINUE                                                          HEX05300
  733 WRITE OUTPUT TAPE 3,558,NAME                                      HEX05310
      IF(KT-1)734,735,735                                               HEX05320
C     WAS THIS CASE HEXAGONAL OR TETRAGONAL                             HEX05330
  734 WRITE OUTPUT TAPE 3,511,E                                         HEX05340
      ASSIGN 762 TO KK                                                  HEX05350
      ASSIGN 774 TO KKK                                                 HEX05360
      VOLUME=AZERO*CZERO*AZERO
      GO TO 740                                                         HEX05370
  735 WRITE OUTPUT TAPE 3,512,E                                         HEX05380
      ASSIGN 764 TO KK                                                  HEX05390
      ASSIGN 776 TO KKK                                                 HEX05400
      VOLUME=AZERO*AZERO*CZERO*0.866026
  740 CONTINUE                                                          HEX05410
  741 IF(FORMWT) 742,742,744
```

```
  742 ZNMBER=0.0
      GO TO 755
  744 CONTINUE
      ZNMBER=VOLUME*DENSTY*0.6023/FORMWT
  755 CONTINUE
      WRITE OUTPUT TAPE 3,520,XLAMDA,YLAMDA,DENSTY,FORMWT,ZNMBER
      WRITE OUTPUT TAPE 3,525,NCUT                                     HEX05430
      WRITE OUTPUT TAPE 3,515, SIGMA,SIGTH,SIGAZO,SIGCZO,AHAT,BHAT,AZEROHEX05440
     *,CZERO                                                           HEX05450
  760 GO TO KK,(762,764)                                               HEX05460
  762 WRITE OUTPUT TAPE 3,505                                          HEX05470
      GO TO 770                                                        HEX05480
  764 WRITE OUTPUT TAPE 3,500                                          HEX05490
  770 DO 779 I=1,NI                                                    HEX05500
      IF(XMODF(I+6,36))777,771,777                                     HEX05510
  771 WRITE OUTPUT TAPE 3,558,NAME                                     HEX05520
COMMANDS 770 TO 771 COUNT THE LINES TO A PAGE SO YOU DO                HEX05530
C     NOT WRITE OVER THE FOLD.                                         HEX05540
  772 GO TO KKK,(774,776)                                              HEX05550
  774 WRITE OUTPUT TAPE 3,505                                          HEX05560
      GO TO 777                                                        HEX05570
  776 WRITE OUTPUT TAPE 3,500                                          HEX05580
  777 WRITE OUTPUT TAPE 3,510,(I,TWO TH(I),KA(I,1),KA(I,2),S(I),O(I),D(IHEX05590
     *))                                                               HEX05600
  779 CONTINUE                                                         HEX05610
  780 CONTINUE                                                         HEX05620
      KKKK=KKKK+1                                                      HEX05630
      GO TO 6                                                          HEX05640
  790 WRITE OUTPUT TAPE 3,558,NAME                                     HEX05650
      WRITE OUTPUT TAPE 3,539,E                                        HEX05660
  795 RETURN
  800 CALL DUMP                                                        HEX05680
COMMAND 800 IS USED FOR AN ERROR INDICATOR -- IF THE PROGRAM GOES TO   HEX05690
COMMAND 800 SOMETHING IS AMISS                                         HEX05700
      END                                                              HEX05710
```

```
      SUBROUTINE ORTHO
C     PROGRAM TO DETERMINE CRYSTAL INDEXES, ORTHORHOMBIC CASE.                00070
      DIMENSION TWO TH(150),S(150),NAME(12),S2SQ(150),SQ(60),AH(150),
     *AK(150),AL(150),ESS(150),C(150),DE(150),KAH(150),KAK(150),KAL(150)
     *,K9(9),STOREA(30),STOREB(30),STOREC(30),ES(600)
      COMMON SQ,  TWO TH,NAME,S2SQ,NCUT,XLAMDA,YLAMDA,NI,ETEST,TETBLK ,
     *S,ES,FORMWT,DENSTY
   50 FORMAT (1X,20I5)                                                        00120
   51 FORMAT( 1X,10F10.5)                                                     00130
   10 FORMAT(40X8HOBSERVED 11X10HCALCULATED/2X1HI3X9HTWO THETA3X2HHH3X2H      00140
     *KK3X2HLL5X38HSINE SQUARED THETA  SINE SQUARED THETA5X10HDIFFERENCE      00150
     *//)                                                                     00160
   20 FORMAT (1H1/ 1X12A6/ 93H THE FOLLOWING INDEXING IS ON THE BASIS OF
     * THE ORTHORHOMBIC SYSTEM WITH THE ERROR LESS THAN  F8.6/
     $                                           31H THE LATTICE CONSTANT A
     *ZERO IS F10.5,11H ANGSTROMS./31H THE LATTICE CONSTANT BZERO IS F        00180
     *10.5,11H ANGSTROMS./31H THE LATTICE CONSTANT CZERO IS F10.5,11H AN      00190
     *GSTROMS. )
   25 FORMAT(57H0THE PROGRAM FINDS NO SOLUTION FOR VALUES OF E LESS THAN      00210
     - F8.6///)                                                               00220
   30 FORMAT(1XI3,F12.5,3I5,F17.5,F20.5,F18.5)
   40 FORMAT(36H0THE UNRESOLVED WAVELENGTH USED WAS F8.5,41H AND THE K-AHEX00260
     *LPHA-ONE WAVELENGTH USED WAS F8.5/16H THE DENSITY IS F6.3, 22H THE
     * FORMULA WEIGHT IS F8.3,21H, AND THE NUMBER IS   F6.2, 1H.)
   45 FORMAT(11H0THE FIRST I3,53H LINES ARE UNRESOLVED. THE REST ARE K-AHEX00340
     *LPHA-1 LINES. )
   65 FORMAT(52H AN EST+MATE OF THE STANDARD DEVIATION OF THETA IS            00240
     1F10.5/65H AN ESTIMATE OF THE STANDARD DEVIATION OF SINE SQUARED TH      00250
     2ETA IS  F10.5/ 51H AN ESTIMATE OF THE STANDARD DEVIATION OF AZERO       00260
     3IS F10.5/51H AN ESTIMATE OF THE STANDARD DEVIATION OF BZERO IS F10      00270
     4.5/51H AN ESTIMATE OF THE STANDARD DEVIATION OF CZERO IS F10.5//)       00280
   70 FORMAT(12A6)                                                            00290
   89 FORMAT(31H0ERROR. PROBABLY CUBIC CRYSTAL.//)                            00310
   90 FORMAT(7F10.2)                                                          00320
   99 FORMAT(1H1//24X12A6//)                                                  00330
      KKKK=0                                                                  00350
      I=40                                                                    00770
      J=41                                                                    00780
      K=42                                                                    00790
      L=43                                                                    00800
      M=44                                                                    00810
      N=45                                                                    00820
      LE=46                                                                   00830
      ME=47                                                                   00840
      NE=48                                                                   00850
      KIT=0                                                                   00870
      KAT=0                                                                   00880
      ASSIGN 308 TO IGO                                                       00890
```

```
      NOD=0
      MAY=NI
  230 +=1.0
      DO 240 I=2,NI
      B=MIN1F(B,ABSF(S(I)-S(I-1)))
  240 CONTINUE
      E=.99B
      EMAX=E/2.
      E=32.*EMAX
COMMAND 230 AND FOLLOWING COMMANDS FIX THE MAXIMUM ERROR
      KAN=1
      KS=1
C     KS=1 WILL START THE VALUE OF K AT 1 SO THE PROGRAM DOES NOT LOOK
C     FOR THE 0,0,0 LINE.
      JAN=KAN
      NAN=JAN+1
C     X,Y, AND Z WILL BE DETERMINED FROM REFLECTIONS KAN,JAN, AND NAN.
  301 JAN=JAN+1
  305 NAN=NAN+1
  306 IF(KIT) 307,307,314
C     IF KIT=1 WE SEEK ALL SOLUTIONS FOR A FIXED ERROR
  307 GO TO IGO,(308,309)
  308 E=E/2.0
      +SSIGN 480 TO KSKIP
      NSTORE=0
      IF(E-ETEST) 331,331,309
  309 DO 391 IT=1,1255
      CALL FETCH(IT,K9)
      DO 310 ICK=1,9
      SQ(ICK+39)=K9(ICK)**2
  310 CONTINUE
      GO TO 500
  314 NOD=NOD+1
      IF(NOD-20) 391,391,620
  321 KIT=KIT+1
      IF(KIT-1) 322,322,391
  322 ASSIGN 309 TO IGO
      ASSIGN 588 TO KSKIP
      GO TO 309
  331 IF(KIT) 332,332,321
  332 E=E*2.
      GO TO 321
C    THIS FIXES THE ERROR AND SEEKS ALL SOLUTIONS
  500 D=SQ(I)*(SQ(M)*SQ(NE)-SQ(N)*SQ(ME))-SQ(J)*(SQ(L)*SQ(NE)-SQ(N)*SQ(L
     XE))+SQ(K)*(SQ(L)*SQ(ME)-SQ(M)*SQ(LE))
C     D IS THE COEFFICIENT DETERMINANT
      IF(D) 526,501,526
  501 RANK=SQ(I)*SQ(L)+SQ(J)*SQ(M)+SQ(K)*SQ(N)
```

```
      IF(RANK) 391,391,502
  502 RANK=SQ(I)*SQ(LE)+SQ(J)*SQ(ME)+SQ(K)*SQ(NE)
      IF(RANK) 391,391,505
C
COMMANDS 501 AND 502 REJECT ROWS WITH ALL ENTRIES ZERO.
C
  505 U=S(KAN)*(SQ(M)*SQ(NE)-SQ(N)*SQ(ME))-S (JAN)*(SQ(J)*SQ(NE)-SQ(K)*S
     *Q(ME))+S(NAN)*(SQ(J)*SQ(N)-SQ(K)*SQ(M))
      IF(U) 391,506,391
  506 V=SQ(I)*(S (JAN)*SQ(NE)-S (NAN)*SQ(N))-S (KAN)*(SQ(L)*SQ(NE)-SQ(N)
     **SQ(LE))+SQ(K)*(S (NAN)*SQ(L)-S (JAN)*SQ(LE))
      IF(V) 391,507,391
  507 W=SQ(I)*(S (NAN)*SQ(M)-S (JAN)*SQ(ME))-SQ(J)*(S (NAN)*SQ(L)-S (JAN
     *)*SQ(LE))+S (KAN)*(SQ(L)*SQ(ME)-SQ(M)*SQ(LE))
      IF(W) 391,510,391
C
COMMANDS 505 THROUGH 507 REJECT INCONSISTENT EQUATIONS.
C
  508 NAN=NAN+1
      IF(NAN-5) 309,309,509
  509 JAN=JAN+1
      NAN=JAN+1
      IF(NAN-5) 309,309,608
  510 IF(SQ(I)) 511,518,511
  511 A22=SQ(M)-SQ(J)*SQ(L)/SQ(I)
      A32=SQ(ME)-SQ(LE)*SQ(J)/SQ(I)
      A23=SQ(N)-SQ(K)*SQ(L)/SQ(I)
      A33=SQ(NE)-SQ(K)*SQ(LE)/SQ(I)
      IF(A32**2+A33**2) 512,515,512
  512 IF(A22**2+A23**2) 513,516,513
  513 C=1.
      B=-A23/A22
      A=-(SQ(K)-B*SQ(J))/SQ(I)
      GO TO 514
  514 IF(ABSF(A*S(1)+B*S(2)+C*S(3))-.001) 508,508,391
  515 IF(ABSF(SQ(LE)/SQ(I)-S(3)/S(1))-.001) 508,508,391
  516 IF(ABSF(SQ(L)/SQ(I)-S(2)/S(1))-.001) 508,508,391
  518 IF(SQ(J)) 520,522,520
  520 KOLDI=I
      KOLDL=L
      KOLDLE=LE
      I=J
      L=M
      LE=ME
      J=KOLDI
      M=KOLDL
      ME=KOLDLE
      GO TO 511
  522 KOLDI=I
```

```
      KOLDL=L
      KOLDLE=LE
      I=K
      L=N
      LE=NE
      K=KOLDI
      N=KOLDL
      NE=KOLDLE
      GO TO 511
C
COMMANDS 510 TO 526 PERFORM AN ANALYSIS OF ALL DEPENDENT CASES.
C
  526 X=S(KAN)*(SQ(M)*SQ(NE)-SQ(N)*SQ(ME))-S (JAN)*(SQ(J)*SQ(NE)-SQ(K)*S
     *Q(ME))+S(NAN)*(SQ(J)*SQ(N)-SQ(K)*SQ(M))
      X=X/D
      9F(X-.001) 391,391,528
  528 Y=SQ(I)*(S (JAN)*SQ(NE)-S (NAN)*SQ(N))-S (KAN)*(SQ(L)*SQ(NE)-SQ(N)
     *MSQ(LE))+SQ(K)*(S (NAN)*SQ(L)-S (JAN)*SQ(LE))
      Y=Y/D
      9F(Y-=001) 391,391,530
  530 Z=SQ(I)*(S (5AN)*SQ(M)-S (JAN)*SQ(ME))-SQ(J)*(S (NAN)*SQ(L)-S (JAN
     *)*SQ(LE))+S (KAN)*(SQ(L)*SQ(ME)-SQ(M)*SQ(LE))
      Z=Z/D
      IF(Z-.001) 391,391,535
  535 JAY=0
  540 JAY=JAY+1
C     JAY IS THE NUMBER OF THE REFLECTION BEING INDEXED
      NR=0
      MD=0
      OLD DE =1.1
  541 NR=NR+1
      IF(NR-20) 542,542,549
  542 NS=0
  544 NS=NS+1
      IF(NS-20)546,546,541
  546 NT=0
  548 NT=NT+1
      IF(NT-20)550,550,544
  549 IF(MD) 391,391,570
C
C          MD NON-ZERO MEANS AT LEAST ONE ACCEPTABLE SET OF INDICES HAVE
C          BEEN FOUND FOR REFLECTION JAY.
  550 C(JAY)=X*SQ(NR)+Y*SQ(NS)+Z*SQ(NT)
C(JAY) IS THE CALCULATED SINE SQUARED THETA
      IF(C(JAY)-S(JAY)-E) 557,557,551
  551 IF(X*SQ(NR)+Y*SQ(NS)-S(JAY)-E) 544,552,552
  552 IF(X*SQ(NR)-S(JAY)-E) 541,553,553
  553 IF(MD) 391,391,570
C     WERE THERE SOLUTIONS FOR THIS X,Y, AND Z  NO,NO,YES
  557 9F(C(1AY)-S(JAY)+E) 548,558,558
```

```
COMMAND 557 ASKS IF THIS IS A SOLUTION NO,YES,YES.
  558 DEE=ABSF(C(JAY)-S(JAY))
      IF(OLD DE-DEE) 548,548,564
C     THIS COMMAND ASKS IF THIS SET OF INDICES IS BETTER THAN ALL
C     PREVIOUS SOLUTIONS NO. NO, YES.
  564 IF(JAY-1) 566,566,567
  566 OLD DE =DEE
      MD=1
      B1=SQ(NR)
      B2=SQ(NS)
      B3=SQ(NT)
C     THE BI ARE THE CURRENT BEST SOLUTION FOR THE S(JAY) REFLECTION.
      GO TO 548
  567 IF(AH(JAY-1) - SQ(NR)) 566,568,566
  568 IF(AK(JAY-1)-SQ(NS)) 566,569,566
  569 IF(AL(JAY-1)-SQ(NT)) 566,548,566
C
COMMANDS 567,568, AND 569 REJECT DUPLICATE SETS OF INDICES.
C
  570 AH(JAY)=B
      AK(JAY)=B2
      AL(JAY)=B3
      MD=0
      OLD DE=1.1
      C(JAY)=X*B1+Y*B2+Z*B3
      IF(JAY-4) 540,575,578
  391 CONTINUE
      IF(KKKK) 610,610,574
  574 RETURN
  575 AO=AH(1)*S(1)+AH(2)*S(2)+AH(3)*S(3)
      BO=AH(1)**2+AH(2)**2+AH(3)**2
      CO=AH(1)*AK(1)+AH(2)*AK(2)+AH(3)*AK(3)
      DO=AH(1)*AL(1)+AH(2)*AL(2)+AH(3)*AL(3)
      AA=AK(1)*S(1)+AK(2)*S(2)+AK(3)*S(3)
      +B=CO
      CC=AK(1)**2+AK(2)**2+AK(3)**2
      DD=AK(1)*AL(1)+AK(2)*AL(2)+AK(3)*AL(3)
      AT=AL(1)*S(1)+AL(2)*S(2)+AL(3)*S(3)
      BT=DO
      CT=DD
      +T=AL(1)**2+AL(2)**2+AL(3)**2
C
COMMANDS FOLLOWING 575 DO THE LEAST SQUARES FIT FOR THE FIRST 3 LINES.
C
  578 AO=AO+AH(JAY)*S(JAY)
       O=BO+AH(JAY)**2
      CO=CO+AH(JAY)*AK(JAY)
      4O=DO+AH(JAY)*AL(JAY)
```

```
      AA=AA+AK(JAY)*S(JAY)
      BB=CO
      CC=CC+AK(JAY)**2
      DD=DD+AK(JAY)*AL(JAY)
      AT=AT+AL(JAY)*S(JAY)
      +T=DO
      CT=DD
      DT=DT+AL(JAY)**2
      DEN=BO*(CC*DT-DD*CT)-CO*(BB*DT-DD*BT)+DO*(BB*CT-CC*BT)
      AZERO=(AO*(CC*DT-CT*DD)-CO*(AA*DT-AT*DD)+DO*(AA*CT-AT*CC))/DEN
      BZERO=(BO*(AA*DT-AT*DD)-AO*(BB*DT-BT*DD)+DO*(BB*AT-BT*AA))/DEN
      CZERO=(BO*(CC*AT-CT*AA)-CO*(BB*AT-BT*AA)+AO*(BB*CT-BT*CC))/DEN
  580 IF(JAY-NI) 584,584,586
  584 X=AZERO
      Y=BZERO
      Z=CZERO
      GO TO 540
  586 AZERO=(YLAMDA/2.)/SQRTF(AZERO)
      BZERO=(YLAMDA/2.)/SQRTF(BZERO)
      +ZERO=(YLAMDA/2.)/SQRTF(CZERO)
COMMANDS 575 THROUGH 586 CALCULATE AZERO,BZERO, AND CZERO.
      GO TO KSKIP,(588,480)
  588 EX=AZERO
      KEX=0
  400 DO 450 IC=1,20
  410 DO 440 IB=1,10
      ZB=IB
      IF(EX-ZB*STOREA(IC)- 0.001) 451,451,415
  415 IF(EX-ZB*STOREB(IC)- 0.001) 451,451,420
  420 IF(EX-ZB*STOREC(IC)- 0.001) 451,451,440
  440 CONTINUE
  450 3ONTINUE
      GO TO 470
  451 KEX=KEX+1
      IF(KEX-2) 452,454,391
  452 EX=BZERO
      GO TO 400
  454 EX=CZERO
      GO TO 400
  470 NSTORE=NSTORE+1
      STOREA(NSTORE)=AZERO
      STOREB(NSTORE)=BZERO
      STOREC(NSTORE)=CZERO
  480 CONTINUE
      Q=(YLAMDA/2.)**2
  590 DO 594 IC=1,NI
      ESS(IC)=        Q*(AH(IC)/AZERO**2+AK(IC)/BZERO**2+AL(IC)/CZERO
     ***2)
```

```
  594 CONTINUE
COMMAND 590 FINDS CALCULATED SINE SQUARED THETA
      SUMA=0.
      SUM=0.
      DO 596 IC=1,NI
      DE(IC)=S(IC)-ESS(9C)
      KAH(IC)=AH(IC)
      KAK(IC)=AK(IC)
      KAL(IC)=AL(IC)
      SUM=SUM+DE(IC)**2
      SUMA=SUMA+(DE(IC)**2)/S2SQ(IC)
  596 +ONTINUE
      SIGMA=SQRTF(SUM/FLOATF(NI-3))
      SIGTH=SQRTF(SUMA/FLOATF(NI-3))*57.2957795
      SIGMAX=SIGMA*SQRTF((CC*DT-CT*DD)/DEN)
      SIGMAY=SIGMA*SQRTF((BO*DT-BT*DO)/DEN)
      SIGMAZ=SIGMA*SQRTF((BO*CC-BB*CO)/DEN)
      SIGMAA=SIGMAX*AZERO/(2.*X)
      SIGMAB=SIGMAY*BZERO/(2.*Y)
      SIGMAC=SIGMAZ*CZERO/(2.*Z)
  600 WRITE OUTPUT TAPE 3,20,NAME,E,AZERO,BZERO,CZERO
      VOLUME=AZERO*BZERO*CZERO
      IF(FORMWT) 601,601,603
  603 ZNMBER=DENSTY*VOLUME*.6023/FORMWT
      GO TO 604
  601 ZNMBER=0.
  604 CONTINUE
      WRITE OUTPUT TAPE 3,40,XLAMDA,YLAMDA,DENSTY,FORMWT,ZNMBER
      WRITE OUTPUT TAPE 3,45,NCUT
      WRITE OUTPUT TAPE 3,65,SIGTH,SIGMA,SIGMAA,SIGMAB,SIGMAC
      WRITE OUTPUT TAPE 3,10
      DO 607 IC=1,NI
      IF(XMODF(IC+6,32)) 606,602,606
  602 WRITE OUTPUT TAPE 3,99,NAME
      WRITE OUTPUT TAPE 3,10
  606 WRITE OUTPUT TAPE 3,30,(IC,TWO TH(IC),KAH(IC),KAK(IC),KAL(IC),
     *S(IC),ESS(IC),DE(IC))
  607 CONTINUE
      KKKK=KKKK+1
      GO TO 306
  608 IF(KKKK) 609,609,620
  609 WRITE OUTPUT TAPE 3,99,NAME
      WRITE OUTPUT TAPE 3,89
      RETURN
  610 WRITE OUTPUT TAPE 3,99,NAME
      WRITE OUTPUT TAPE 3,25,E
  620 RETURN
       END
```

INTERNAL DISTRIBUTION

Copy Number	
1	L. L. Ames, Jr.
2	J. L. Bates
3	A. L. Bement
4	C. A. Bennett
5	T. K. Bierlein
6	L. L. Burger
7	J. B. Burnham
8	S. H. Bush
9	J. J. Cadwell
10	D. F. Carroll
11	T. D. Chikalla
12	E. W. Christopherson
13	W. I. Clark
14	T. T. Claudson
15	L. F. Coleman
16	R. Y. Dean
17	W. L. Delvin
18	L. R. Duncan
19-68	J. B. Goebel
69	R. L. Hales
70	W. A. Haney
71	K. M. Harmon
72	L. A. Hartcorn
73	F. E. Holt
74	Y. B. Katayama
75	W. E. Keder
76	H. E. Kissinger
77	M. C. Lambert
78	E. C. Martin
79	J. E. Minor
80	R. H. Moore
81	J. R. Morrey
82	J. M. Nielson
83	R. E. Olson
84	P. G. Palmer
85	J. L. Ryan
86	J. E. Schlosser
87	F. A. Scott
88	J. C. Sheppard
89	R. E. Skavdahl
90	J. M. Taylor
91	J. C. Tobin
92	R. H. Todd
93	O. J. Wick
94-118	A. S. Wilson
119	Technical Publications
120-124	Technical Information Files

EXTERNAL DISTRIBUTION (SPECIAL)

Number of Copies	
1	General Electric Company, Richland Attn: H. H Hopkins, Jr.
2	RLOO Attn: R. K. Sharp Technical Information Library

Ptd.	Standard Distribution
3	ABERDEEN PROVING GROUND
1	AEROJET-GENERAL CORPORATION
1	AEROJET-GENERAL NUCLEONICS
2	AERONAUTICAL SYSTEMS DIVISION
1	AEROPROJECTS INCORPORATED
1	AIR FORCE INSTITUTE OF TECHNOLOGY
	AIR FORCE INSTITUTE OF TECHNOLOGY (MAFB)
1	AIR FORCE RESEARCH AND TECHNOLOGY DIVISION
1	ALLIS-CHALMERS MANUFACTURING COMPANY
1	ALLIS-CHALMERS MANUFACTURING COMPANY, BETHESDA
1	ALLISON DIVISION–GMC
10	ARGONNE NATIONAL LABORATORY
1	ARMY ENGINEER RESEARCH AND DEVELOPMENT LABORATORIES
1	ARMY MATERIALS RESEARCH AGENCY
1	ARMY RESEARCH OFFICE, DURHAM
1	ARMY TANK-AUTOMOTIVE CENTER
1	ATOMIC ENERGY COMMISSION, BETHESDA
1	AEC SCIENTIFIC REPRESENTATIVE, BELGIUM
1	AEC SCIENTIFIC REPRESENTATIVE, FRANCE
1	AEC SCIENTIFIC REPRESENTATIVE, JAPAN
3	ATOMIC ENERGY COMMISSION, WASHINGTON
4	ATOMIC ENERGY OF CANADA LIMITED
2	ATOMIC ENERGY OF CANADA LIMITED, WHITESHELL
4	ATOMICS INTERNATIONAL
1	AVCO CORPORATION
4	BABCOCK AND WILCOX COMPANY
2	BATTELLE MEMORIAL INSTITUTE
2	BEERS (ROLAND F.), INC.
1	BERYLLIUM CORPORATION
1	BLUME (JOHN A.) AND ASSOCIATES
2	BROOKHAVEN NATIONAL LABORATORY
1	BUREAU OF MINES, ALBANY

Ptd.	Standard Distribution
1	BUREAU OF NAVAL WEAPONS
1	BUREAU OF SHIPS (CODE 1500)
1	CARBORUNDUM COMPANY
1	CHICAGO PATENT GROUP
1	COMBUSTION ENGINEERING, INC.
1	COMBUSTION ENGINEERING, INC. (NRD)
1	DEFENCE RESEARCH MEMBER
1	DEL ELECTRONICS CORPORATION
1	DENVER RESEARCH INSTITUTE
1	DOW CHEMICAL COMPANY, ROCKY FLATS
4	DU PONT COMPANY, AIKEN
1	DU PONT COMPANY, WILMINGTON
1	FRANKFORD ARSENAL
1	FRANKLIN INSTITUTE OF PENNSYLVANIA
1	FUNDAMENTAL METHODS ASSOCIATION
2	GENERAL ATOMIC DIVISION
2	GENERAL DYNAMICS/FORT WORTH
2	GENERAL ELECTRIC COMPANY, CINCINNATI
1	GENERAL ELECTRIC COMPANY, PLEASANTON
1	GENERAL ELECTRIC COMPANY, SAN JOSE
1	GOODYEAR ATOMIC CORPORATION
1	IIT RESEARCH INSTITUTE
1	INTERNATIONAL BUSINESS MACHINES CORPORATION
2	IOWA STATE UNIVERSITY
2	JET PROPULSION LABORATORY
1	JOHNS HOPKINS UNIVERSITY
3	KNOLLS ATOMIC POWER LABORATORY
1	LING TEMCO VOUGHT, INC.
1	LOCKHEED-GEORGIA COMPANY
1	LOCKHEED MISSILES AND SPACE COMPANY (NASA)
3	LOS ALAMOS SCIENTIFIC LABORATORY
1	M & C NUCLEAR, INC.
1	MALLINCKRODT CHEMICAL WORKS
1	MARE ISLAND NAVAL SHIPYARD
1	MARITIME ADMINISTRATION
1	MARTIN-MARIETTA CORPORATION
1	MINNESOTA MINING AND MANUFACTURING COMPANY (BRANDT)

Ptd.	Standard Distribution
1	MOUND LABORATORY
1	NASA LEWIS RESEARCH CENTER
1	NASA MANNED SPACECRAFT CENTER
2	NASA SCIENTIFIC AND TECHNICAL INFORMATION FACILITY
2	NATIONAL BUREAU OF STANDARDS
1	NATIONAL BUREAU OF STANDARDS (LIBRARY)
2	NATIONAL LEAD COMPANY OF OHIO
4	*NATIONAL REACTOR TESTING STATION (PPCO)
1	NAVAL POSTGRADUATE SCHOOL
3	NAVAL RESEARCH LABORATORY
1	NRA, INC.
1	NUCLEAR MATERIALS AND EQUIPMENT CORPORATION
1	NUCLEAR METALS, INC.
1	NUCLEAR UTILITY SERVICES, INC.
1	OFFICE OF ASSISTANT GENERAL COUNSEL FOR PATENTS (AEC)
2	OFFICE OF NAVAL RESEARCH
1	OFFICE OF NAVAL RESEARCH (CODE 422)
1	OHIO STATE UNIVERSITY
1	PETROLEUM CONSULTANTS
1	PHYSICS INTERNATIONAL, INC.
1	PICATINNY ARSENAL
1	POWER REACTOR DEVELOPMENT COMPANY
3	PRATT AND WHITNEY AIRCRAFT DIVISION
1	PURDUE UNIVERSITY
1	RADIOPTICS, INC.
1	RAND CORPORATION
1	REACTIVE METALS, INC.
1	REACTIVE METALS, INC., ASHTABULA
1	RENSSELAER POLYTECHNIC INSTITUTE

Ptd.	Standard Distribution
1	SAN FRANCISCO OPERATIONS OFFICE
2	SANDIA CORPORATION, ALBUQUERQUE
1	SANDIA CORPORATION, LIVERMORE
1	SOUTHWEST RESEARCH INSTITUTE
1	STANFORD UNIVERSITY (SLAC)
1	SYLVANIA ELECTRIC PRODUCTS, INC.
1	TENNESSEE VALLEY AUTHORITY
1	TRW SPACE TECHNOLOGY LABORATORIES (NASA)
1	UNION CARBIDE CORPORATION, CLEVELAND
2	UNION CARBIDE CORPORATION (ORGDP)
5	UNION CARBIDE CORPORATION (ORNL)
1	UNION CARBIDE CORPORATION (PADUCAH PLANT)
2	UNITED NUCLEAR CORPORATION (NDA)
1	U. S. GEOLOGICAL SURVEY, DENVER
1	U. S. GEOLOGICAL SURVEY, MENLO PARK
1	U. S. GEOLOGICAL SURVEY, WASHINGTON
1	U. S. PATENT OFFICE
2	UNIVERSITY OF CALIFORNIA, BERKELEY
2	UNIVERSITY OF CALIFORNIA, LIVERMORE
1	UNIVERSITY OF PUERTO RICO
1	WESTERN RESERVE UNIVERSITY (MAJOR)
4	WESTINGHOUSE BETTIS ATOMIC POWER LABORATORY
1	WESTINGHOUSE ELECTRIC CORPORATION
1	WESTINGHOUSE ELECTRIC CORPORATION (NASA)
285	DIVISION OF TECHNICAL INFORMATION EXTENSION
75	CLEARINGHOUSE FOR FEDERAL SCIENTIFIC AND TECHNICAL INFORMATION

*New listing or change in old listing.