OR FFE,
A FORTRAN CRYSTALLOGRAPHIC FUNCTION AND ERROR PROGRAM

W. R. Busing
K. O. Martin
H. A. Levy

NOTICE
This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report. The information is not to be abstracted, reprinted or otherwise given public dissemination without the approval of the ORNL patent branch, Legal and Information Control Department.
LEGAL NOTICE

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

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

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "serving on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.
ORNL-TM-306

Contract No. W-7405-eng-26

OR FFE,
A FORTRAN CRYSTALLOGRAPHIC FUNCTION AND ERROR PROGRAM

by

W. R. Busing, K. O. Martin, and H. A. Levy

Chemistry Division and Mathematics Division

MARCH 1964

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>1</td>
</tr>
<tr>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>Mathematical Method</td>
<td>3</td>
</tr>
<tr>
<td>Constraints on the Parameters</td>
<td>6</td>
</tr>
<tr>
<td>Atom Designation and Symmetry Transformations</td>
<td>7</td>
</tr>
<tr>
<td>Data Input</td>
<td>9</td>
</tr>
<tr>
<td>Instruction Input</td>
<td>14</td>
</tr>
<tr>
<td>Tapes Required</td>
<td>25</td>
</tr>
<tr>
<td>Error Indicators</td>
<td>26</td>
</tr>
<tr>
<td>Specifications for Subroutines to Compute</td>
<td></td>
</tr>
<tr>
<td>New Types of Functions</td>
<td></td>
</tr>
<tr>
<td>Arithmetic Subprograms Available</td>
<td>29</td>
</tr>
<tr>
<td>Discussion of Examples</td>
<td>33</td>
</tr>
<tr>
<td>Card Decks Provided</td>
<td>35</td>
</tr>
<tr>
<td>Glossary of Symbols</td>
<td>40</td>
</tr>
<tr>
<td>Flow Charts</td>
<td>44</td>
</tr>
<tr>
<td>Symbolic Program Listing</td>
<td>49</td>
</tr>
<tr>
<td>Subroutines for Examples</td>
<td>77</td>
</tr>
<tr>
<td>Data for Examples</td>
<td></td>
</tr>
<tr>
<td>Program Output for Examples</td>
<td></td>
</tr>
<tr>
<td>Distribution</td>
<td></td>
</tr>
</tbody>
</table>
ABSTRACT

This report describes a computer program which calculates interatomic distances, bond angles, principal axes of thermal motion, and other functions of the unit cell parameters, the atomic coordinates, and the temperature factor coefficients of a crystal structure. The standard errors of these functions are computed from the variance-covariance matrix of the parameters if this is known. The program may be used independently, or the input may be taken from a magnetic tape prepared by the crystallographic least-squares program OR FLS.

The program is written entirely in Fortran, and detailed instructions for its use are included. The necessary card decks may be obtained from the authors.
INTRODUCTION

Given the unit cell parameters of a crystal together with the atomic coordinates and/or the anisotropic temperature factor coefficients, this program computes various functions of these parameters, such as the distance between two atoms, an angle defined by three atoms, the principal axes of the anisotropic temperature factor, etc. If the errors of the input parameters are available to the program in the form of a variance-covariance matrix obtained, for example, from the inverse matrix of a least squares refinement, the program will also compute the standard errors of the functions with and without the contribution of the cell parameter errors. This program may be used independently, with all input data read from cards, or it may be used in conjunction with the least squares refinement program OR FLS, in which case much of the input is taken from a magnetic tape produced by that program.

The types of functions which can be evaluated are defined by subroutines, of which fifteen are included in the present program. The detailed specification of each function or group of functions to be computed is made by means of an instruction card which will be described below. If functions are desired other than those already included, the user need only write a Fortran II subroutine for the function, compile it, and add it to the deck. To facilitate such programming, several subroutines for the manipulation of matrices and vectors are included in the program, and descriptions of these
will be given below together with detailed specifications for writing subroutines for new functions.

This program is a revision of the earlier OR XFE (W. R. Busing and H. A. Levy, "A Crystallographic Function and Error Program for the IBM 704," ORNL-CF 59-12-3 (1959)). It is now written entirely in Fortran II and comments have been added to facilitate any modification which may be desired. Card decks are available from the authors on request.

MATHEMATICAL METHOD

The functions which the program computes are of the form

\[ f = f(p_1, p_2, p_3 \ldots, a_1, a_2 \ldots, a_6) \]

where the \( p \)'s are the atomic parameters and the \( a \)'s are the unit cell parameters. Each kind of function is computed by a special subroutine for this purpose, and the mathematical expressions used are best obtained from the Fortran language listings of these subroutines designated FUN1, FUN2, etc.

The standard error of \( f \) is given by

\[ e = (e'^2 + e''^2)^{1/2} \]

where

\[
    e'^2 = \sum_{j=1}^{n} \sum_{i=1}^{n} C_{ij} \left( \frac{\partial f}{\partial p_i} \right) \left( \frac{\partial f}{\partial p_j} \right) V_{ij}
\]

and

\[
    e''^2 = \sum_{j=1}^{6} \sum_{i=1}^{6} C_{ij} \left( \frac{\partial f}{\partial a_i} \right) \left( \frac{\partial f}{\partial a_j} \right) U_{ij}.
\]
Here $V_{ij}$ is an element of the variance-covariance matrix which describes the errors of the atomic parameters, and $U_{ij}$ is an element of a similar matrix for the unit cell parameters. $C_{ij} = 1$ if $i = j$; otherwise $C_{ij} = 2$.

When errors are to be computed the user has the option of obtaining $V$ from a tape produced by OR FLS. In this case the program stores

$$V_{ij} = [\Sigma w(OBS - CALC)^2/(m-n)] b_{ij}$$

where $b_{ij}$ is an element of the inverse matrix of the normal equations, and the constant in brackets is the weighted sum of the squares of the residuals divided by the number of degrees of freedom. The other option available to the user is to read $V_{ij}$ directly from cards.

The unit cell errors may be put in in two ways. If no covariances are known then the user may supply the six standard errors $\sigma(a_i)$. The program then sets

$$U_{ii} = \sigma^2(a_i)$$

$$U_{ij} = 0, \ i \neq j.$$  

Alternatively the 21 independent values of $U_{ij}$ ($i \leq j$) can be read from cards.

The necessary derivatives are computed numerically by adding an increment $\Delta p_i$ to $p_i$ and re-entering the subroutine for $f$. The derivative is then

$$\frac{\partial f}{\partial p_i} \approx \frac{f(p_1, p_2, \ldots, p_i + \Delta p_i, \ldots, a_1, \ldots) - f(p_1, p_2, \ldots, p_i, \ldots, a_1, \ldots)}{\Delta p_i}.$$  

The increment used is $\Delta p_i = (0.01) V_{ii}^{1/2}$. The derivatives
with respect to the cell parameters are obtained similarly.

The program is arranged so that it computes derivatives only with respect to the parameters actually involved in f. The information as to which these are is provided by the subroutines PRE1, PRE2, etc., which are supplied for each type of function. Derivative computations are also omitted with respect to fixed parameters which are not represented in the variance-covariance matrix. If this situation arises because certain atoms are in special positions, then the errors are computed correctly. However, if parameters have been fixed for other reasons, it is possible that important terms are omitted from the calculated errors.

Note that for the derivative computation to be correct all symmetry transformations must be made after $\Delta p_1$ has been added to $p_1$. Similarly all constraints on the parameters must be set after incrementing. (See the section entitled "Constraints on the Parameters"). In other words, the function $f$ is always computed directly from the fundamental parameters, the errors of which are described by the matrices $V$ and $U$.

The output of the program includes a description of the function, the value $f$ of the function, the standard error $e$ of the function, and the standard error $e'$ not including the effect of unit cell errors.
CONSTRAINTS ON THE PARAMETERS

In order to insure that this program correctly computes the errors of various functions, it is necessary for the user to consider whether the symmetry of the crystal introduces constraints either on the cell parameters, $a_1$, or on the atomic parameters, $p_1$. A simple case occurs when the symmetry forces a parameter to have a fixed value, and this situation is correctly treated by specifying that the error of this parameter is zero. A more complicated procedure is required when the symmetry imposes a relationship between two or more parameters. In this case one of the interrelated parameters is chosen as independent and a Fortran II subroutine is written to set the values of the dependent parameters in terms of it. The main program will enter this subroutine immediately before each entry to the function-computing subroutine so that the derivatives needed for the error calculation are correctly computed. In the input data the errors associated with the dependent parameters should be set to zero. (When the atomic parameters are taken from a least-squares refinement these errors are automatically zero, since the dependent parameters would not have been varied.)

Specifications are given here for subroutines SETA and SETP which establish these constraints, and examples are included in the sample calculations. The statements required are summarized as follows:
SUBROUTINE SETA(A)
DIMENSION A(6)
Include statements to set the values of the A(I)'s which are taken as dependent in terms of those chosen as independent.
RETURN
END
SUBROUTINE SETP(P)
DIMENSION P(300)
Include statements to set the values of the P(I)'s which are taken as dependent in terms of those chosen as independent.
RETURN
END

These routines should be compiled and substituted in the binary deck for the dummy routines with the same names.

ATOM DESIGNATION AND SYMMETRY TRANSFORMATIONS

In the course of preparing an instruction card to specify a function to be computed, it will be necessary to define the one or more atoms involved in this function. This is done by means of an atom designation consisting of the two integers a and 100c + s. Here a = 1, 2, 3, .... is the number of the atom in the parameter list, s = 0, 1, 2, .... is the number of the symmetry transformation to be applied, and c = 0, 1, .... 7 defines the unit cell translations as described below.
The program obtains the coordinates of an atom in the following way:

1. The integer a is used to compute the location of the coordinates in the parameter list and \( x, y, z \) are picked up. If \( a = 0 \) the program sets \( x = y = z = 0 \).

2. These coordinates are then transformed to \( x', y', z' \) according to the symmetry information punched on symmetry cards. If \( s = 0 \) no transformation is made.

3. The cell translations are then made according to the following table:

<table>
<thead>
<tr>
<th>( c )</th>
<th>( z' )</th>
<th>( y' )</th>
<th>( x' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( z' )</td>
<td>( y' )</td>
<td>( x' )</td>
</tr>
<tr>
<td>1</td>
<td>( z' )</td>
<td>( y' )</td>
<td>( x' - 1 )</td>
</tr>
<tr>
<td>2</td>
<td>( z' )</td>
<td>( y' - 1 )</td>
<td>( x' )</td>
</tr>
<tr>
<td>3</td>
<td>( z' )</td>
<td>( y' - 1 )</td>
<td>( x' - 1 )</td>
</tr>
<tr>
<td>4</td>
<td>( z' - 1 )</td>
<td>( y' )</td>
<td>( x' )</td>
</tr>
<tr>
<td>5</td>
<td>( z' - 1 )</td>
<td>( y' )</td>
<td>( x' - 1 )</td>
</tr>
<tr>
<td>6</td>
<td>( z' - 1 )</td>
<td>( y' - 1 )</td>
<td>( x' )</td>
</tr>
<tr>
<td>7</td>
<td>( z' - 1 )</td>
<td>( y' - 1 )</td>
<td>( x' - 1 )</td>
</tr>
</tbody>
</table>

(As a memory aid, note that the 1's in this table correspond to the binary representation of \( c \)).

For example, atom (3,208) is atom 3 in the parameter list transformed to symmetry position 8 of cell 2, while atom (5,0) is atom 5 just as it appears in the list of parameters. In a subsequent section we will refer to atom 3 in asymmetric unit 208 or atom 5 in the basic asymmetric unit (unit 0).
Note that transformations such as \( x' = -x \) and \( x' = 1-x \) are not in general equivalent for purposes of this program.

The program obtains the matrix of the anisotropic temperature factor coefficients of an atom in a similar way except that the elements of this matrix are transformed as the corresponding products of the coordinates. This procedure is valid for an atom in either a special or general position provided that the symmetry cards are written for general positions. (See H. A. Levy, Acta Cryst. (1956), 9, 679). The translational part of the symmetry transformation is irrelevant in this case as is the cell translation \( c \). If \( a = 0 \), a null matrix will be generated.

**DATA INPUT**

1. **Title card.**
   
   **Columns** 1-72
   
   Title consisting of any desired Hollerith information. This will be printed unchanged on the output.

2. **Control card.**
   
   **Columns** 1-3
   
   **IPM**, the atomic parameter error indicator. The variance-covariance matrix and parameter selection information will
   
   (IPM = 0) not be used.
   
   (IPM = 1) be used to compute errors.

   **Columns** 4-6

   **NP**, the input indicator. If \( NP = 0 \), the structure parameters will be read from the private tape written by OR FLS. (If the variance-covariance matrix and parameter selection information are to be used, they will also be read from this tape when \( NP = 0 \)). If \( NP > 0 \), data will be taken entirely from cards, and \( NP \) is equal to the number of structure parameters to be read from the cards.

   \[ 0 \leq NP \leq 300. \]
Columns
7-9
IAM, the cell parameter error indicator.
The cell parameter errors are
(IAM = 0) not to be used.
(IAM = 1) to be read in the form of standard errors.
(IAM = 2) to be read in the form of a variance-covariance matrix.

10-12
NS, the number of symmetry cards to be read.
0 ≤ NS ≤ 48.

13-15
NV, the order of the variance-covariance matrix, PM, if it is to be read from cards. If the matrix is to be read from the OR FLS tape, NV will also be read from this tape and this field is irrelevant. If the variance-covariance matrix is not to be used, set NV = 0.
0 ≤ NV ≤ 200.

The following parameter arrangement integers must be supplied if the structure parameters are read from cards. They are supplied automatically by the program if the parameters are read from the private output tape of OR FLS. For more details see 3 below.

Columns
16-18
JXP, the period of the position parameters in the parameter list. If no position parameters are included in the list, set JXP and JX = 0.

19-21
JX, the position of the first x coordinate in the parameter list.

22-24
JBP, the period of the temperature factor coefficients in the parameter list. If no temperature factor coefficients are included in the list, set JBP and JB = 0.

25-27
JB, the position of the first temperature factor coefficient in the parameter list.

3. Atomic parameter cards. FORMAT(8F9.4)
If NP = 0, the atomic parameters will be read from the private output tape of OR FLS, and the atomic parameter cards are omitted from the card input. Otherwise NP parameters must be punched eight per card. The arrangement of these parameters is subject to the following restrictions:
a) For each atom the position parameters \( x, y, z \) must be in sequence.

b) Position parameters for successive atoms must appear periodically in the order of the atom designation.

c) If anisotropic temperature factor coefficients are included, they must be in the sequence \( \beta_{11}, \beta_{22}, \beta_{33}, \beta_{12}, \beta_{13}, \beta_{23} \), for each atom. These \( \beta \)'s are defined by the following expression for the temperature factor:

\[
\exp\left[-(h^2\beta_{11}+k^2\beta_{22}+l^2\beta_{33}+2hk\beta_{12}+2hl\beta_{13}+2kl\beta_{23})\right].
\]

d) Temperature factor coefficients of successive atoms must appear periodically in the order of the atom designation.

Thus the position parameters of atom I start with \( P(J) \) where \( J = JXP*(I-1) + JX \), and the temperature factor coefficients of atom I start with \( P(K) \) where \( K = JBP*(I-1) + JB \).

4. Parameter selection cards. FORMAT(72I1)

The purpose of these cards is to identify those parameters which are associated with the elements of the variance-covariance matrix PM. They are exactly the same as the parameter selection cards used with OR FLS.

These cards are not to be supplied unless the control card integers IPM and NP specify that the variance-covariance matrix and parameter selection information will be used and will be read from cards (IPM = 1 and NP > 0).

Each column of a parameter selection card corresponds to one parameter. The first card is associated with parameters 1 to 72, the second with parameters 73 to 144, etc. If a parameter is represented in the matrix a one is punched in the corresponding column; otherwise a zero or blank is punched. The total number of one punches must be equal to NV, the order of the matrix. The first parameter in the list for which a one punch is found is assumed to correspond to the first row and column of the matrix, the second with the second, etc.

5. Variance-covariance matrix, PM, for structure parameters. FORMAT(6E12.6)

These cards are to be supplied only if the variance-covariance matrix is to be used and is to be read from cards (IPM = 1 and NP > 0).
First card
Columns 1-12
A scale factor by which the matrix will be multiplied. This could be unity or $\Sigma w(F_O - F_C)^2/(m-n)$, for example.

Subsequent cards
The $NV(NV+1)/2$ unique elements of the matrix $PM$ punched six per card in the order $PM(1,1)$, $PM(1,2)$, ..., $PM(1,NV)$, $PM(2,2)$, $PM(2,3)$, ..., $PM(2,NV)$, $PM(3,3)$, ..., $PM(NV,NV)$, where $NV$ is the order of the matrix. The parameters with which this matrix is associated are defined by the parameter selection cards.

6. Cell parameter card. FORMAT(6F9.4)
These six direct unit cell parameters must always be supplied.

Columns
1-9 $A(1) = a$
10-18 $A(2) = b$
19-27 $A(3) = c$
28-36 $A(4) = \cos \alpha$
37-45 $A(5) = \cos \beta$
46-54 $A(6) = \cos \gamma$

7. Cell parameter error cards.
If the effect of the cell parameter errors is not to be considered (IAM = 0), these cards are omitted. If it is to be included, the cell parameter errors must be supplied in one of two forms as specified by the control card integer IAM.

a) Standard error form (IAM = 1). FORMAT(6F9.4)
If this input is used the covariances between the cell parameters are assumed to be zero.

Columns
1-9 $\sigma(a)$
10-18 $\sigma(b)$
19-27 $\sigma(c)$
28-36 $\sigma(\cos \alpha)$
37-45 $\sigma(\cos \beta)$
46-54 $\sigma(\cos \gamma)$
b) Variance-covariance matrix form (IAM = 2). If information on the covariances between cell parameters is available, the 21 unique elements of this $6 \times 6$ matrix should be punched on three cards.

FORMAT(8F9.4)

Card 1: AM(a,a), AM(a,b),..., AM(a, cos $\gamma$), AM(b,b), AM(b,c).
Card 2: AM(b, cos $\alpha$),..., AM(b, cos $\gamma$), AM(c,c),..., AM(cos $\alpha$, cos $\alpha$).
Card 3: AM(cos $\alpha$, cos $\beta$),..., AM(cos $\gamma$, cos $\gamma$).

8. Symmetry cards. FORMAT(F11.6,2I2,F11.6,2I2,F11.6,2I2)

The number of symmetry cards to be read is given by NS on the control card. There must be a card for each symmetry transformation which will be called for by an instruction card (see the section "Atom Designation and Symmetry Transformations"). If the functions to be computed require no symmetry transformations, then no symmetry cards need be supplied.

These symmetry cards have the same format as those for OR FLS.

Columns
1-11 Translational part of $x_j$ or blank.
12-13 1, 2, 3, -1, -2, -3, or blank for $x$, $y$, $z$, $-x$, $-y$, $-z$, or blank, respectively, as used in the expression for the transformed $x_j$.
14-15 1, 2, 3, -1, -2, -3, or blank for $x$, $y$, $z$, $-x$, $-y$, $-z$, or blank, respectively, as used in the expression for the transformed $x_j$. (Columns 12 and 13 are exactly equivalent to 14 and 15. Also, note that an expression such as $x_j = 2x$ must be treated as $x_j = x + x$.)
16-26 Translational part of $y_j$ or blank.
27-30 Integers representing plus or minus $x$, $y$, or $z$ in the expression for the transformed $y_j$ as described above.
31-41 Translational part of $z_j$ or blank.
42-45 Integers representing plus or minus $x$, $y$, or $z$ in the expression for the transformed $z_j$. 
9. Instruction cards. FORMAT(24I3)
   See the next section, "Instruction Input," for the specifications for this input.

10. Instruction termination card. FORMAT(24I3)
    Columns
    1-3       0 (zero) as a sentinel for the end of the instruction deck.

INSTRUCTION INPUT

Each function to be computed by the program is specified by a sequence of integers, IN, which are read from one or more instruction cards. The first integer in this sequence, IN(1), defines the type of function to be computed, and the interpretation of the remaining instruction integers will in general be different for different types of functions. Details of the instruction integers for each type of function are given below.

Each instruction card is read with FORMAT(24I3). Of the 24 integers on this card only the first 23 are considered to be part of the instruction, IN. Usually one card will suffice to specify a function, but if a function requires more than 23 integers to define it, up to ten cards may be used with 23 integers on each card. Punching a one (or any non-zero integer) in field 24 of an instruction card indicates that the instruction is continued on the next card.
Function 1

One interatomic distance.

Columns

<table>
<thead>
<tr>
<th></th>
<th>1-3</th>
<th>4-6</th>
<th>7-9</th>
<th>10-12</th>
<th>13-15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>(a_i)</td>
<td>(100c_1 + s_1)</td>
<td>(a_2)</td>
<td>(100c_2 + s_2)</td>
</tr>
</tbody>
</table>

atom designation 1.

Function 101

All distances (less than \(d(\text{max})\)) between atoms in two asymmetric units.

Columns

<table>
<thead>
<tr>
<th></th>
<th>1-3</th>
<th>4-6</th>
<th>7-9</th>
<th>10-12</th>
<th>13-15</th>
</tr>
</thead>
</table>
|    | 101 | \(a(\text{max})\), the number of atoms in the parameter list. | \(100c_1 + s_1\), designation of first asymmetric unit. | --- | \(100c_2 + s_2\), designation of second asymmetric unit.
|    |     | (This may be the same as the first.) |     |     |     |
|    | 16-18 | The integer 10d(\(\text{max}\)). If this is left blank, \(d(\text{max}) = 4.0\text{Å}.\) |     |     |     |

Function 201

All distances (less than \(d(\text{max})\)) between atoms in the basic asymmetric unit and atoms in all asymmetric units (i.e., all combinations of \(c\) and \(s\)).

In order to cause this instruction to compute all possible distances less than \(d(\text{max})\), the basic asymmetric unit should be chosen close to the origin, and the symmetry transformations specified should generate asymmetric units to fill out a unit cell on the positive sides of this basic unit. Even so, for certain symmetry situations or for small unit cells, some of the desired distances will be omitted if the atoms involved lie outside of the eight unit cells.
generated. The user may prefer to use instruction 101 and to select in advance the asymmetric units likely to be involved in these contacts.

Columns

1-3  201
4-6  a(max), the number of atoms in the parameter list.
7-15  ---
16-18 The integer 10d(max). If this is left blank then d(max) = 4.0Å.

Function 2

Angle defined by three atoms.

Columns

1-3  2
4-9  Atom designation 1.
10-15 Atom designation 2 (vertex).
16-21 Atom designation 3.

Function 3

Angle between normals to planes each defined by three atoms. The direction of the normals is that of (1,2)x(1,3) and (4,5)x(4,6) where 1,2 is the vector defined by atom designations 1 and 2, etc.

Columns

1-3  3
4-9  Atom designation 1
10-15 Atom designation 2  Plane 1
16-21 Atom designation 3
22-27 Atom designation 4
28-33 Atom designation 5  Plane 2
34-39 Atom designation 6
Function 4

Difference between two interatomic distances.

Columns

1-3 4
4-9 Atom designation 1
10-15 Atom designation 2
16-21 Atom designation 3
22-27 Atom designation 4

Function 5

Difference between two angles each defined by three atoms.

Columns

1-3 5
4-9 Atom designation 1
10-15 Atom designation 2 (vertex)
16-21 Atom designation 3
22-27 Atom designation 4
28-33 Atom designation 5 (vertex)
34-39 Atom designation 6

Function 6

The sum of several angles each defined by three atoms.

Columns

1-3 6
4-6 n, the number of angles to be summed.
7-12 Atom designation 1
13-18 Atom designation 2 (vertex)
19-24 Atom designation 3
Columns

25-30 Atom designation 4
31-36 Atom designation 5 (vertex) \{ Angle 2
37-42 Atom designation 6
etc.

Function 7

The RMS component of thermal displacement of one atom along principal axis \( r \).

Columns

1-3 7
4-9 Atom designation
10-12 \( r (= 1, 2, \text{ or } 3) \)

Function 107

Same as Function 7 computed for each of the three values of \( r \).

Columns

1-3 107
4-9 Atom designation

Function 207

Same as Function 7 computed for each atom of a given asymmetric unit, each with the three values of \( r \).

Columns

1-3 207
4-6 \( a\text{(max)}, \) the number of atoms in the parameter list.
7-9 Designation of the asymmetric unit.
Function 8

Angle between principal axis \( r \) of atom 1 and a vector defined by atoms 2 and 3.

Columns

<table>
<thead>
<tr>
<th>Columns</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>8</td>
</tr>
<tr>
<td>4-9</td>
<td>Atom designation 1</td>
</tr>
<tr>
<td>10-12</td>
<td>( r (= 1, 2, \text{ or } 3) )</td>
</tr>
<tr>
<td>13-18</td>
<td>Atom designation 2</td>
</tr>
<tr>
<td>19-24</td>
<td>Atom designation 3</td>
</tr>
</tbody>
</table>

Vector

Function 108

Same as Function 8 computed for each of the three values of \( r \).

Columns

<table>
<thead>
<tr>
<th>Columns</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>108</td>
</tr>
<tr>
<td>4-9</td>
<td>Atom designation 1</td>
</tr>
<tr>
<td>10-12</td>
<td>---</td>
</tr>
<tr>
<td>13-18</td>
<td>Atom designation 2</td>
</tr>
<tr>
<td>19-24</td>
<td>Atom designation 3</td>
</tr>
</tbody>
</table>

Vector

Function 208

Same as Function 8 computed for each atom in a given molecule, each with the three values of \( r \).

Columns

<table>
<thead>
<tr>
<th>Columns</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>208</td>
</tr>
<tr>
<td>4-6</td>
<td>( a(\text{max}), \text{ the number of atoms in the parameter list.} )</td>
</tr>
<tr>
<td>7-9</td>
<td>Designation of the asymmetric unit.</td>
</tr>
<tr>
<td>10-12</td>
<td>---</td>
</tr>
<tr>
<td>13-18</td>
<td>Atom designation 2</td>
</tr>
<tr>
<td>19-24</td>
<td>Atom designation 3</td>
</tr>
</tbody>
</table>

Vector
Function 9

The RMS component of thermal displacement of atom 1 along principal axis \( r \), projected on a vector defined by atoms 2 and 3.

Columns

1-3 9
4-9 Atom designation 1
10-12 \( r (= 1, 2, \text{ or } 3) \)
13-18 Atom designation 2
19-24 Atom designation 3

Function 109

Same as Function 9 computed for the three values of \( r \).

Columns

1-3 109
4-9 Atom designation 1
10-12 ---
13-18 Atom designation 2
19-24 Atom designation 3

Function 209

Same as Function 9 computed for each atom in a given asymmetric unit, each with the three values of \( r \).

Columns

1-3 209
4-6 \( a(\text{max}) \), the number of atoms in the parameter list.
7-9 Designation of the asymmetric unit.
10-12 ---
Columns
13-18 Atom designation 2
19-24 Atom designation 3

**Function 10**

Angle between principal axis \( r \) of atom 1 and axis \( i \) of a cartesian coordinate system. The latter is defined by atoms 2, 3, 4, and 5 as follows: Axis 1 is in the direction of vector \((2, 3)\). Axis 2 is in the direction of \((\text{Axis } 1) \times (4, 5)\). Axis 3 is in the direction of \((\text{Axis } 1) \times (\text{Axis } 2)\).

Columns

<table>
<thead>
<tr>
<th>1-3</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-9</td>
<td>Atom designation 1</td>
</tr>
<tr>
<td>10-12</td>
<td>( r ) (= 1, 2, or 3)</td>
</tr>
<tr>
<td>13-15</td>
<td>( i ) (= 1, 2, or 3)</td>
</tr>
<tr>
<td>16-21</td>
<td>Atom designation 2</td>
</tr>
<tr>
<td>22-27</td>
<td>Atom designation 3</td>
</tr>
<tr>
<td>28-33</td>
<td>Atom designation 4</td>
</tr>
<tr>
<td>34-39</td>
<td>Atom designation 5</td>
</tr>
</tbody>
</table>

**Vectors defining axes**

**Function 110**

Same as Function 10 computed for the nine combinations of \( r \) and \( i \).

Columns

<table>
<thead>
<tr>
<th>1-3</th>
<th>110</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-9</td>
<td>Atom designation 1</td>
</tr>
<tr>
<td>10-15</td>
<td>---</td>
</tr>
</tbody>
</table>
Columns
16-21 Atom designation 2
22-27 Atom designation 3
28-33 Atom designation 4
34-39 Atom designation 5

Function 210
Same as Function 10 computed for all atoms in a given asymmetric unit, each with the nine combinations of r and i.

Columns
1-3 210
4-6 \( a(\text{max}) \), the number of atoms in the parameter list.
7-9 Designation of the asymmetric unit.
10-15 ---
16-21 Atom designation 2
22-27 Atom designation 3
28-33 Atom designation 4
34-39 Atom designation 5

Function 11
The RMS component of thermal displacement of atom 1 along principal axis r, projected on axis i of a cartesian coordinate system defined by atoms 2, 3, 4, and 5 as described under Function 10.

Columns
1-3 11
4-9 Atom designation 1
10-12 \( r (= 1, 2, \text{or } 3) \)
Columns

13-15 \( i (= 1, 2, \text{ or } 3) \)

16-21 Atom designation 2 \( \{ \text{Vector} \} \)

22-27 Atom designation 3 \( \{ \text{Vectors defining axes} \} \)

28-33 Atom designation 4 \( \{ \text{Vector} \} \)

34-39 Atom designation 5

**Function 111**

Same as Function 11 computed for the nine combinations of \( r \) and \( i \).

1-3 111

4-9 Atom designation 1

10-15 ---

16-21 Atom designation 2 \( \{ \text{Vector} \} \)

22-27 Atom designation 3 \( \{ \text{Vectors defining axes} \} \)

28-33 Atom designation 4 \( \{ \text{Vector} \} \)

34-39 Atom designation 5

**Function 211**

Same as Function 11 computed for all atoms in a given asymmetric unit, each with the nine combinations of \( r \) and \( i \).

Columns

1-3 211

4-6 \( a(\text{max}) \), the number of atoms in the parameter list.

7-9 Designation of asymmetric unit.

10-15 ---
Columns
16-21 Atom designation 2 \{ Vector \} Vectors defining axes
22-27 Atom designation 3 \{ Vector \}
28-33 Atom designation 4 \{ Vector \}
34-39 Atom designation 5

Function 12
The RMS component of thermal displacement of atom 1 in the direction of a vector defined by atoms 2 and 3.

Columns
1-3 12
4-9 Atom designation 1
10-15 Atom designation 2 \{ Vector \}
16-21 Atom designation 3

Function 13
The RMS radial thermal displacement of an atom.

Columns
1-3 13
4-9 Atom designation

Function 14
Interatomic distance averaged over thermal motion.
Second atom is assumed to ride on the first. The function is \[ R = R_0 + (\bar{r}_i^2 - \bar{s}_i^2 - \bar{r}_i' + \bar{s}_i')/2R_0 \] where \( R_0 \) is the uncorrected interatomic distance, \( \bar{r}_i^2 \) is the mean square radial thermal displacement of atom i, and \( \bar{s}_i^2 \) is the mean square component of displacement of atom i in the direction defined by the interatomic vector.
Columns
1-3  14
4-9  Atom designation 1
10-15 Atom designation 2

Function 15

Interatomic distance averaged over thermal motion.
Atoms assumed to move independently. The function is 
\[ R = R_o + \left( \bar{r}_{\frac{1}{2}} - \bar{\xi}_{\frac{1}{2}} + \bar{r}_{\bar{r}} - \bar{\xi}_{\bar{r}} \right) / 2R_o \]
where the symbols are defined as in Function 14 above.

Columns
1-3  15
4-9  Atom designation 1
10-15 Atom designation 2

TAPES REQUIRED

Listed here are the monitor control cards required for operation at the Oak Ridge Central Data Processing Facility. At other installations the necessary tapes should be specified appropriately.

*TAPE(3, Reel Number, SAVE)

This tape is required only if NP = 0, indicating that information is to be taken from the output of OR FLS. It will be rewound before being read and may be file protected if desired.

*TAPE(9, OUTPUT)

This is the monitor output tape to be listed.

*TAPE(10, INPUT)

This is the monitor input tape prepared from cards.
ERROR INDICATORS

If an error is detected in the course of computing a function, an indicator is put out in place of the result, and the program proceeds to compute the next function. The form of the indicator is "***NG" where NG is an integer, the meaning of which is tabulated here.

<table>
<thead>
<tr>
<th>NG</th>
<th>Subprogram</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ATOM, BETA</td>
<td>The symmetry position specified is out of range of the symmetry cards supplied.</td>
</tr>
<tr>
<td>4</td>
<td>BETA</td>
<td>The temperature factors are not in anisotropic form. Either ITF = 1 on OR FLS tape or JBP &lt; 6 when parameters are read from cards.</td>
</tr>
<tr>
<td>5</td>
<td>ATOM, BETA</td>
<td>The atom specified is out of range of the parameter list.</td>
</tr>
<tr>
<td>6</td>
<td>STOBB</td>
<td>A cell parameter A(1), A(2), or A(3) &lt; 0.</td>
</tr>
<tr>
<td>7</td>
<td>EIGVAL</td>
<td>The eigenvalues are complex.</td>
</tr>
<tr>
<td>8</td>
<td>EIGVEC</td>
<td>Only null eigenvectors are found.</td>
</tr>
<tr>
<td>9</td>
<td>COSVV</td>
<td>One of the vectors is null.</td>
</tr>
<tr>
<td>10</td>
<td>FUNXI</td>
<td>The vector used to specify the direction is null.</td>
</tr>
<tr>
<td>11</td>
<td>FUNI</td>
<td>The function specified is out of range. The program requires 0 &lt; XMDF(IN(1),100) ≤ 15.</td>
</tr>
</tbody>
</table>

SPECIFICATIONS FOR SUBROUTINES TO COMPUTE NEW TYPES OF FUNCTIONS

To program a new type of function the user selects a function designation integer, \(i\). Function designations 1–15 have already been assigned in the program as written so that normally new designations would be 16, 17....
Subprograms HED, PRE, FUN, and OUT must each be modified as follows:

1. Change the second IF statement to permit the new function designation.
2. Add a branch to the computed GO TO.
3. Add a CALL statement for the new routine followed by a GO TO 160.

The user then writes four Fortran II subroutines named HED, PRE, FUN, and OUT as described below. All these subprograms are compiled and added to the binary deck removing the original routines where necessary.

The programmer should refer to the symbolic listing of functions 1-15 for examples of the following routines.

HED

The user may omit this subroutine if desired. Its purpose is to put out an over-all heading which defines the type of function. It is entered each time the function designation IN(1) is changed.

1. The subroutine has no arguments.
2. No COMMON or DIMENSION statements are needed.
3. Put out the heading on tape 9.
4. The first Hollerith character should be zero to cause double spacing.
5. More than one line may be put out if desired.

PRE

This subroutine need be included only if standard errors are to be computed. Its purpose is to instruct the program as
to which parameters are involved in the function to be computed. The program thus avoids computing unnecessary derivatives.

1. The subroutine has no arguments.
2. Include the standard COMMON and DIMENSION statements used throughout the program.
3. Enter the subroutines SETKX and/or SETKB once for each atom involved in the function. See the section "Arithmetic Subprograms Available" for details of these routines.

**FUNi**

The purpose of this subroutine is to compute the desired function when given the various input data found in common storage.

1. The subroutine has no arguments.
2. Include the standard COMMON and DIMENSION statements used throughout the program.
3. The significance of the instruction integers IN(K) except for IN(1) is at the discretion of the programmer.
4. The necessary atomic coordinates or temperature factor coefficients are called for by means of subroutines ATOM or BETA.
5. If the metric tensors AA or BB are to be used, the subroutines STOAA and/or STOBB must be entered. See the section "Arithmetic Subprograms Available" for a description of these routines.
6. Store the function at FX in COMMON and RETURN.

**OUTi**

The purpose of this subroutine is to put out the detailed description of the individual function computed. Its inclusion is optional but certainly desirable.

1. The subroutine has no arguments.
2. Include the standard COMMON and DIMENSION statements used throughout Segment 2.
3. The description which probably includes the relevant instruction integers, IN, should be put out on tape 9.
4. The first character of the first line of this description should be a Hollerith zero to cause double spacing.
5. More than one line may be put out if desired.

**ARITHMETIC SUBPROGRAMS AVAILABLE**

1. CALL SETKX (IN(K))
   
   Instructs the program to calculate derivatives with respect to the coordinates of the atom designated in IN(K). This subroutine is used by the preliminary subroutines, PREi.
2. CALL SETKB (IN(K))
   
   Instructs the program to calculate derivatives with respect to the anisotropic temperature factor coefficients of the atom designated in IN(K). This subroutine is used by the preliminary subroutines, PREi.
3. CALL STOAA

Stores the 3 x 3 metric tensor $g$ (where $g_{ij} = a_i \cdot a_j$) at AA in common storage.

4. CALL STOBB

Stores the 3 x 3 reciprocal metric tensor $g^{-1}$ (where $(g^{-1})_{ij} = b_i \cdot b_j$) at BB in common storage.

5. CALL ATOM(IN(K),Z)

DIMENSION Z(3)

Stores at Z the coordinates of the atom specified by the instruction integers IN(K) and IN(K+1). These coordinates refer to the triclinic axes. They have been transformed according to the symmetry specified.

6. CALL BETA(IN(K),Z)

DIMENSION Z(3,3)

Stores at Z the 3 x 3 matrix of the temperature factor coefficients of the atom specified by instruction integers IN(K) and IN(K+1). The matrix has been transformed according to the symmetry specified.

7. CALL MM(X, Y, Z)

DIMENSION X(3,3), Y(3,3), Z(3,3)

Performs the matrix multiplication $XY = Z$. The location of Z must be different from X and Y.

8. CALL MV(X, Y, Z)

DIMENSION X(3,3), Y(3), Z(3)

Performs the matrix-vector multiplication $XY = Z$. The location of Z must be different from Y.
9. CALL VM (X, Y, Z)
   
   DIMENSION X(3), Y(3,3), Z(3)

   Performs the vector-matrix multiplication \( \mathbf{X}^{T} \mathbf{Y} = \mathbf{Z}^{T} \).
   
   The locations of \( \mathbf{Z} \) must be different from \( \mathbf{X} \).

10. Function VV(X, Y)

    DIMENSION X(3), Y(3)

    Performs the vector-vector multiplication \( \mathbf{X}^{T} \mathbf{Y} = \mathbf{Z} \)
    (a scalar).

11. Function VMV (W, X, Y)

    DIMENSION W(3), X(3,3), Y(3)

    Performs the vector-matrix-vector multiplication \( \mathbf{W}^{T} \mathbf{X} \mathbf{Y} = \mathbf{Z} \)
    (a scalar).

12. CALL DIFV (X, Y, Z)

    DIMENSION X(3), Y(3), Z(3)

    Performs the vector subtraction \( \mathbf{X} - \mathbf{Y} = \mathbf{Z} \). \( \mathbf{Z} \) may have
    the same location as \( \mathbf{X} \) or \( \mathbf{Y} \).

13. CALL SUMV (X, Y, Z)

    DIMENSION X(3), Y(3), Z(3)

    Performs the vector summation \( \mathbf{X} + \mathbf{Y} = \mathbf{Z} \). \( \mathbf{Z} \) may have
    the same location as \( \mathbf{X} \) or \( \mathbf{Y} \).

14. Function COSVV (X, Y)

    DIMENSION X(3), Y(3)

    Computes the cosine of the angle defined by vectors \( \mathbf{X} \) and \( \mathbf{Y} \). These vectors are assumed to refer to the triclinic
    coordinate system, and it is also assumed that the metric
tensor has been stored at AA in common storage.
15. Function \textsc{ARCCOS}(X)

Computes \( \theta \), the arc \( \cos \) of \( X \) in degrees. \( 0 \leq \theta \leq 180 \).

16. CALL \textsc{NORM}(X, Y, Z)

\textsc{DIMENSION} X(3), Y(3), Z(3)

Stores at \( Z \) a vector (not a unit vector) perpendicular to both \( X \) and \( Y \). The sense of \( Z \) is that of the vector product \( X \times Y \). All vectors are referred to the triclinic coordinate system, and it is assumed that the reciprocal metric tensor has been stored at BB in common storage.

17. CALL \textsc{AXES}(U, V, X)

\textsc{DIMENSION} U(3), V(3), X(3,3)

Stores three mutually perpendicular vectors (not unit vectors) at \( X(I,1), X(I,2), \) and \( X(I,3) \). \( X(I,1) = U(I) \).

\( X(I,2) \) is normal to \( U \) and \( V \), i.e., in the direction of \( U \times V \).

\( X(I,3) \) is normal to \( X(I,1) \) and \( X(I,2) \) so as to yield a right-handed coordinate system. All these vectors are referred to the triclinic coordinate system, and it is assumed that the reciprocal metric tensor has been stored at BB in common storage.

18. CALL \textsc{EIGVAL}(W, Y)

\textsc{DIMENSION} W(3,3), Y(3)

Stores the three eigenvalues \( Y \) of the matrix \( W \).

19. CALL EIGVEC (W, Y, Z)

DIMENSION W(3,3), Z(3)

Given an eigenvalue Y of the matrix W, this subroutine stores the corresponding eigenvector Z.

Solving any pair of the three available equations would yield a vector in the desired direction. Because of special symmetry situations, however, the pairs chosen could be redundant so that a null vector would be obtained. The subroutine therefore repeats the calculation with each of the three possible pairs, and the vector of largest magnitude is taken as the correct result. If all three pairs yield null vectors, the subroutine sets an error indicator, because this implies that the direction of the eigenvector is indeterminate.

20. Function TRACE (X)

DIMENSION X(3,3)

Computes the trace of the matrix X.

DISCUSSION OF EXAMPLES

Data for two test problems are included with the card deck for OR FFE. Both are extensions of the hypothetical problem based on alpha quartz which was included with the program OR FLS. Example 1 uses card input only, and tests the present program independently. Example 2 takes part of its input from a tape produced by OR FLS, and tests the two programs together. To obtain the required magnetic tape from OR FLS it is necessary to modify the test data card FLS 1414 by substituting 1 for 0 in column 18.
It should be emphasized again that this problem is a hypothetical one. Although the structure considered is qualitatively that of alpha quartz, the observations used in the OR FLS test are synthetic, and the parameters which result are not related to the true structure of this material.

Silicon atom (2,300) lies on a single twofold rotation axis and has oxygen atoms (1,2), (1,3), (1,300), and (1,305) as four nearest neighbors in an approximately tetrahedral configuration. Example 1 computes the independent distances and angles of this group. The dihedral angle computed is a measure of the distortion of the tetrahedron. The differences computed show how the program can be used to determine whether chemically equivalent but crystallographically distinct distances or angles have significant deviations from each other.

Example 2 includes computations relating to the anisotropic temperature factor coefficients. First the root-mean-square displacements in the principal axis directions are obtained. Then the angles which principal axes of the O atom (1,2) make with the Si-O bond (2,300) (1,2) are computed. Next a cartesian coordinate system is defined in terms of the Si-O vectors (2,300) (1,2) and (2,300) (1,3) so that the principal axes of O can be related to this. Note that the axis with I = 1 is just the Si-O bond direction used above so that the angles with I = 1 repeat those computed earlier.

Because the Si atom is located on a twofold axis, one of its principal axes is constrained to lie in that direction.
By choosing a coordinate system based on Si atoms in different cells, cartesian axis 1 is made to fall on the twofold axis, axis 2 lies in the x,y plane and axis 3 is parallel to z. Principal axes 1 and 3 are both seen to make angles of 90° with cartesian axis 1, as expected. Principal axis 2 should therefore make angles of 0° (or 180°), 90°, and 90° with cartesian axes 1, 2, and 3, respectively. Unfortunately, the results are only approximately correct in the present test problem. The parameters from the hypothetical OR FLS test problem represent atoms which are nearly isotropic, and it is believed that this situation permits round-off errors to cause the observed discrepancies.

Finally three other types of functions are computed to complete the test.

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Symbolic card numbers</th>
<th>Column binary card numbers*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calling program</td>
<td>1 - 246</td>
<td>2051 - 2103**</td>
</tr>
<tr>
<td>SUB8</td>
<td>247 - 300</td>
<td>2104 - 2109</td>
</tr>
<tr>
<td>SUB10</td>
<td>301 - 321</td>
<td>2110 - 2112</td>
</tr>
<tr>
<td>SUB11</td>
<td>322 - 347</td>
<td>2113 - 2115</td>
</tr>
<tr>
<td>SUB13</td>
<td>348 - 491</td>
<td>2116 - 2133</td>
</tr>
<tr>
<td>SUB19</td>
<td>492 - 508</td>
<td>2134 - 2136</td>
</tr>
<tr>
<td>SUB21</td>
<td>509 - 581</td>
<td>2137 - 2145</td>
</tr>
<tr>
<td>HEDI</td>
<td>582 - 637</td>
<td>2146 - 2155</td>
</tr>
<tr>
<td>Subprogram</td>
<td>Symbolic card numbers</td>
<td>Column binary card numbers*</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>PREI</td>
<td>638 - 678</td>
<td>2156 - 2162</td>
</tr>
<tr>
<td>FUNI</td>
<td>679 - 746</td>
<td>2163 - 2172</td>
</tr>
<tr>
<td>OUTI</td>
<td>747 - 786</td>
<td>2173 - 2179</td>
</tr>
<tr>
<td>SETKX</td>
<td>787 - 803</td>
<td>2180 - 2182</td>
</tr>
<tr>
<td>SETKB</td>
<td>804 - 820</td>
<td>2183 - 2186</td>
</tr>
<tr>
<td>STOA A</td>
<td>821 - 840</td>
<td>2187 - 2190</td>
</tr>
<tr>
<td>STOBB</td>
<td>841 - 874</td>
<td>2191 - 2197</td>
</tr>
<tr>
<td>ATOM</td>
<td>875 - 927</td>
<td>2198 - 2209</td>
</tr>
<tr>
<td>BETA</td>
<td>928 - 987</td>
<td>2210 - 2224</td>
</tr>
<tr>
<td>MM</td>
<td>988 - 999</td>
<td>2225 - 2228</td>
</tr>
<tr>
<td>MV</td>
<td>1000 - 1010</td>
<td>2229 - 2232</td>
</tr>
<tr>
<td>VM</td>
<td>1011 - 1021</td>
<td>2233 - 2236</td>
</tr>
<tr>
<td>VV</td>
<td>1022 - 1031</td>
<td>2237 - 2239</td>
</tr>
<tr>
<td>VMV</td>
<td>1032 - 1040</td>
<td>2240 - 2242</td>
</tr>
<tr>
<td>DIFV</td>
<td>1041 - 1049</td>
<td>2243 - 2245</td>
</tr>
<tr>
<td>SUMV</td>
<td>1050 - 1058</td>
<td>2246 - 2248</td>
</tr>
<tr>
<td>COSV V</td>
<td>1059 - 1075</td>
<td>2249 - 2253</td>
</tr>
<tr>
<td>ARCCOS</td>
<td>1076 - 1081</td>
<td>2254 - 2256</td>
</tr>
<tr>
<td>NORM</td>
<td>1082 - 1101</td>
<td>2257 - 2260</td>
</tr>
<tr>
<td>AXES</td>
<td>1102 - 1113</td>
<td>2261 - 2265</td>
</tr>
<tr>
<td>EIGVAL</td>
<td>1114 - 1160</td>
<td>2266 - 2277</td>
</tr>
<tr>
<td>EIGVEC</td>
<td>1161 - 1195</td>
<td>2278 - 2284</td>
</tr>
<tr>
<td>TRACE</td>
<td>1196 - 1204</td>
<td>2285 - 2287</td>
</tr>
<tr>
<td>FUNA</td>
<td>1205 - 1224</td>
<td>2288 - 2292</td>
</tr>
<tr>
<td>Subprogram</td>
<td>Symbolic card numbers</td>
<td>Column binary card numbers*</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>FUND</td>
<td>1225 - 1241</td>
<td>2293 - 2296</td>
</tr>
<tr>
<td>HED1</td>
<td>1242 - 1248</td>
<td>2297 - 2299</td>
</tr>
<tr>
<td>PRE1</td>
<td>1249 - 1261</td>
<td>2300 - 2302</td>
</tr>
<tr>
<td>FUN1</td>
<td>1262 - 1273</td>
<td>2303 - 2305</td>
</tr>
<tr>
<td>OUT1</td>
<td>1274 - 1286</td>
<td>2306 - 2308</td>
</tr>
<tr>
<td>HED2</td>
<td>1287 - 1293</td>
<td>2309 - 2311</td>
</tr>
<tr>
<td>PRE2</td>
<td>1294 - 1306</td>
<td>2312 - 2314</td>
</tr>
<tr>
<td>FUN2</td>
<td>1307 - 1318</td>
<td>2315 - 2317</td>
</tr>
<tr>
<td>OUT2</td>
<td>1319 - 1331</td>
<td>2318 - 2320</td>
</tr>
<tr>
<td>HED3</td>
<td>1332 - 1339</td>
<td>2321 - 2323</td>
</tr>
<tr>
<td>PRE3</td>
<td>1340 - 1352</td>
<td>2324 - 2326</td>
</tr>
<tr>
<td>FUN3</td>
<td>1353 - 1377</td>
<td>2327 - 2333</td>
</tr>
<tr>
<td>OUT3</td>
<td>1378 - 1391</td>
<td>2334 - 2337</td>
</tr>
<tr>
<td>HED4</td>
<td>1392 - 1398</td>
<td>2338 - 2340</td>
</tr>
<tr>
<td>PRE4</td>
<td>1399 - 1411</td>
<td>2341 - 2343</td>
</tr>
<tr>
<td>FUN4</td>
<td>1412 - 1423</td>
<td>2344 - 2346</td>
</tr>
<tr>
<td>OUT4</td>
<td>1424 - 1437</td>
<td>2347 - 2350</td>
</tr>
<tr>
<td>HED5</td>
<td>1438 - 1444</td>
<td>2351 - 2353</td>
</tr>
<tr>
<td>FUN5</td>
<td>1445 - 1456</td>
<td>2354 - 2356</td>
</tr>
<tr>
<td>HED6</td>
<td>1457 - 1463</td>
<td>2357 - 2359</td>
</tr>
<tr>
<td>PRE6</td>
<td>1464 - 1477</td>
<td>2360 - 2362</td>
</tr>
<tr>
<td>FUN6</td>
<td>1478 - 1492</td>
<td>2363 - 2365</td>
</tr>
<tr>
<td>OUT6</td>
<td>1493 - 1507</td>
<td>2366 - 2369</td>
</tr>
<tr>
<td>FUNB</td>
<td>1508 - 1527</td>
<td>2370 - 2374</td>
</tr>
<tr>
<td>Subprogram</td>
<td>Symbolic card numbers</td>
<td>Column binary card numbers*</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>FUNC</td>
<td>1528 - 1548</td>
<td>2375 - 2379</td>
</tr>
<tr>
<td>FUNX</td>
<td>1549 - 1573</td>
<td>2380 - 2386</td>
</tr>
<tr>
<td>HED7</td>
<td>1574 - 1581</td>
<td>2387 - 2390</td>
</tr>
<tr>
<td>PRE7</td>
<td>1582 - 1593</td>
<td>2391 - 2393</td>
</tr>
<tr>
<td>FUN7</td>
<td>1594 - 1606</td>
<td>2394 - 2396</td>
</tr>
<tr>
<td>OUT7</td>
<td>1607 - 1619</td>
<td>2397 - 2399</td>
</tr>
<tr>
<td>HED8</td>
<td>1620 - 1627</td>
<td>2400 - 2403</td>
</tr>
<tr>
<td>PRE8</td>
<td>1628 - 1641</td>
<td>2404 - 2406</td>
</tr>
<tr>
<td>FUN8</td>
<td>1642 - 1654</td>
<td>2407 - 2409</td>
</tr>
<tr>
<td>OUT8</td>
<td>1655 - 1668</td>
<td>2410 - 2413</td>
</tr>
<tr>
<td>HED9</td>
<td>1669 - 1677</td>
<td>2414 - 2417</td>
</tr>
<tr>
<td>FUN9</td>
<td>1678 - 1690</td>
<td>2418 - 2420</td>
</tr>
<tr>
<td>HED10</td>
<td>1691 - 1699</td>
<td>2421 - 2424</td>
</tr>
<tr>
<td>PRE10</td>
<td>1700 - 1713</td>
<td>2425 - 2427</td>
</tr>
<tr>
<td>FUN10</td>
<td>1714 - 1726</td>
<td>2428 - 2430</td>
</tr>
<tr>
<td>OUT10</td>
<td>1727 - 1741</td>
<td>2431 - 2434</td>
</tr>
<tr>
<td>HED11</td>
<td>1742 - 1751</td>
<td>2435 - 2438</td>
</tr>
<tr>
<td>FUN11</td>
<td>1752 - 1764</td>
<td>2439 - 2441</td>
</tr>
<tr>
<td>FUNR</td>
<td>1765 - 1780</td>
<td>2442 - 2445</td>
</tr>
<tr>
<td>FUNCR</td>
<td>1781 - 1799</td>
<td>2446 - 2451</td>
</tr>
<tr>
<td>FUNXI</td>
<td>1800 - 1824</td>
<td>2452 - 2457</td>
</tr>
<tr>
<td>HED12</td>
<td>1825 - 1833</td>
<td>2458 - 2461</td>
</tr>
<tr>
<td>PRE12</td>
<td>1834 - 1847</td>
<td>2462 - 2464</td>
</tr>
<tr>
<td>FUN12</td>
<td>1848 - 1860</td>
<td>2465 - 2467</td>
</tr>
<tr>
<td>Subprogram</td>
<td>Symbolic card numbers</td>
<td>Column binary card numbers*</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>OUT12</td>
<td>1861 - 1874</td>
<td>2468 - 2471</td>
</tr>
<tr>
<td>HED13</td>
<td>1875 - 1881</td>
<td>2472 - 2474</td>
</tr>
<tr>
<td>FUN13</td>
<td>1882 - 1894</td>
<td>2475 - 2477</td>
</tr>
<tr>
<td>OUT13</td>
<td>1895 - 1907</td>
<td>2478 - 2480</td>
</tr>
<tr>
<td>HED14</td>
<td>1908 - 1915</td>
<td>2481 - 2483</td>
</tr>
<tr>
<td>PRE14</td>
<td>1916 - 1929</td>
<td>2484 - 2486</td>
</tr>
<tr>
<td>FUN14</td>
<td>1930 - 1943</td>
<td>2487 - 2489</td>
</tr>
<tr>
<td>HED15</td>
<td>1944 - 1951</td>
<td>2490 - 2492</td>
</tr>
<tr>
<td>FUN15</td>
<td>1952 - 1965</td>
<td>2493 - 2495</td>
</tr>
<tr>
<td>ASIN</td>
<td></td>
<td>2496 - 2499***</td>
</tr>
<tr>
<td>SETA (Dummy)</td>
<td>1966 - 1970</td>
<td>2500 - 2501</td>
</tr>
<tr>
<td>SETP (Dummy)</td>
<td>1971 - 1975</td>
<td>2502 - 2503</td>
</tr>
<tr>
<td>SETA (Examples 1 and 2)</td>
<td>1976 - 1982</td>
<td>2504 - 2506</td>
</tr>
<tr>
<td>SETP (Example 1)</td>
<td>1983 - 1989</td>
<td>2507 - 2509</td>
</tr>
<tr>
<td>SETP (Example 2)</td>
<td>1990 - 1998</td>
<td>2510 - 2512</td>
</tr>
<tr>
<td>Transfer card and data for Example 1</td>
<td>1999 - 2022</td>
<td></td>
</tr>
<tr>
<td>Transfer card and data for Example 2</td>
<td>2023 - 2050</td>
<td></td>
</tr>
</tbody>
</table>

*The column binary cards provided were compiled on an IBM 7090. Most of the subprograms must be recompiled if they are to be used on an IBM 704.

**Card No. 2051 is a control card for the Oak Ridge monitor system. It may not be appropriate at other installations.

***The ASIN routine is included because it is not available from the Oak Ridge library tape. It may not be needed at other installations. Library routines SQRT, COS, and EXIT (which terminates the job) should also be inserted here if they are not supplied by the monitor system.
Users who received a temporary card deck during 1963 will probably find that the only change has been to add identifying numbers to the final version. The most recent modifications to the program were made in the cards now numbered FFE 072, 073, 162, 939, 940, and 1140-1143. If these cards are verified and corrected where necessary, then the temporary deck will conform to this report.

GLOSSARY OF SYMBOLS

The following list defines the symbols used in the main parts of the program. Some of the independent arithmetic subroutines repeat certain symbols with different meanings, and these are not included in this glossary.

Arrays are represented by symbols followed by their minimum allowable dimensions. All other symbols refer to single-storage variables. Those marked with an asterisk (*) are sometimes read as input data.

*A(6)
Direct cell parameters a, b, c, cos α, cos β, and cos γ.

AA(3,3)
Metric tensor g where \( g_{i,j} = a_i \cdot a_j \).

AM(21)
Upper triangular part of the variance-covariance matrix associated with the six cell parameters. \( AM(1) = U_{11} \), \( AM(2) = U_{12} \), ..., \( AM(6) = U_{16} \), \( AM(7) = U_{22} \), ..., etc.

BB(3,3)
Reciprocal metric tensor \( g^{-1} \) where \( (g^{-1})_{i,j} = b_i \cdot b_j \).

C
Factor \( C_{i,j} \) used in computing variances in SUBL3. \( C_{i,j} = 1 \) if \( i = j \); otherwise \( C_{i,j} = 2 \).

DA(6)
Increments \( \Delta a_i = (0.01) U_{11}^{1/2} \).

DFDA(6)
Derivatives \( \frac{\partial f}{\partial a_i} \).

DFDP(NV)
Derivatives \( \frac{\partial f}{\partial p_i} \).

DMAX
Maximum distance which will be printed when using instructions 101 or 201.

DP(NV)
Increments \( \Delta p_i = (0.01) V_{11}^{1/2} \).

E
Standard error, e, of the function including contributions from cell parameter errors and atomic parameter errors.
**El** Standard error, $e'$, of the function including only contributions from least-squares parameter errors.

**F** Computed value of function $f$ to be put out.

**FX** Computed value of function $f$ as stored by the subroutines FUNi.

**I**AM** Cell parameter error indicator. If IAM = 0, cell parameter errors are not to be used. If IAM = 1, they are to be read as standard errors. If IAM = 2, they are to be read as a variance-covariance matrix.

**IH** Hundreds digit of the instruction number, IN(1).

**IN(231)** Array of integers used to define the function to be computed. IN(1) specifies the kind of function.

**INSAVE** IN(1) saved from the previous instruction integers. The function heading is put out if IN(1) ≠ INSAVE.

**IPM** Atomic parameter error indicator. If IPM = 0, these errors will not be used in computing the standard error of the function. If IPM = 1, they will be used.

**IS(2,3,NS)** Integers representing the non-translational part of the symmetry information. For example, if the Jth symmetry transformation is $y-x$, $-x$, $1/3 + z$, this information would be stored as follows:

$$
\begin{align*}
TS(1,J) &= 0.0 \\
IS(1,1,J) &= 2 \text{ (for y)} \\
IS(2,1,J) &= -1 \text{ (for -x)} \\
TS(2,J) &= 0.0 \\
IS(1,2,J) &= -1 \text{ (for -x)} \\
IS(2,2,J) &= 0 \\
TS(3,J) &= 0.333333 \\
IS(1,3,J) &= 3 \text{ (for z)} \\
IS(2,3,J) &= 0
\end{align*}
$$

**ITF** Temperature factor indicator. ITF = 0 if parameters were read from cards. Otherwise ITF is read from the private output tape of OR FLS and is 1 or 2 for isotropic or anisotropic temperature factors, respectively.
*JB  Position of the first temperature factor coefficient in the parameter list.

*JBP  Period of the temperature factor coefficients in the parameter list.

*JX  Position of the first x-coordinate in the parameter list.

*JXP  Period of the coordinates in the parameter list.

*KII(NP)  Parameter selection integers.  \( KII(I) = 1 \) if \( P(I) \) corresponds to a row-column of the variance-covariance matrix, \( PM \). Otherwise \( KII(I) = 0 \).

KI2(NP)  Integers set by subroutine PREi to specify the parameters involved in the function being computed. \( KI2(I) \) is set to 1 if \( P(I) \) is involved in the function. Otherwise \( KI2(I) = 0 \).

KK  Index used so that \( PM(KK) \) is a diagonal element of the variance-covariance matrix.

KKD  Increment to be added to \( KK \) to step from one diagonal element of \( PM \) to the next.

LNZ  Integer saved to record that \( DFDP(LNZ) \) is the last non-zero derivative in the array.

NA  Number of atoms in the parameter list as specified by the instruction integers for certain multivalued functions.

NG  Error indicator set to a non-zero integer when an error is found in the course of calculating a function or its derivative. See the section "Error Indicators" for a list of these errors.

NM  \( NV(NV+1)/2 \), the number of elements in the upper triangular part of the variance-covariance matrix, \( PM \).

*NP  Number of parameters in the parameter list.

NQ  Number of scale factors in a parameter list which has been read from the private output tape of OR FLS.
*NS  Number of symmetry cards read.

NSP  NS+1, the number of symmetry positions including the basic asymmetric unit.

*NV  Order of the variance-covariance matrix for the atomic parameters.

*P(NP) List of atomic parameters read either from cards or from the private output tape of OR FLS.

*PM(NM) Upper triangular part of variance-covariance matrix associated with the atomic parameters PM(1) = V11, PM(2) = V12, ..., PM(NV) = V1NV, PM(NV+1) = V22, ..., etc.

ROW(6) Auxiliary array used in putting out the matrix AM.

SAVEA Temporary storage for a cell parameter which is being incremented during the derivative calculation.

SAVEP Temporary storage for an atomic parameter which is being incremented during the derivative calculation.

*SCALE Constant used to scale the variance-covariance matrix if it is read from cards.

*TITLE(12) Alphanumeric title read at the start of the problem and transcribed to the output.

*TS(3,NS) Translational part of symmetry information. (See IS for an example.)

VARA Variance of f based on the cell parameter errors only.

VARP Variance of f based on the atomic parameter errors only.
Read input and make preliminary calculations → Read instruction array, IN → CALL SUB19 for single-valued function → CALL EXIT to end job → CALL HEDI to put out heading

IN(1) ≠ 0 → IN(1) - INSAME = 0

< 0 → IN(1) - 100

> 0 → INSAME = IN(1)

CALL SUB21 for multiple-valued function

Figure 1. Schematic flow diagram of the main program showing the subprograms called by it.

NG = 0 to clear error indicator → CALL FUNI to evaluate function → F = FX → CALL SUB13 to compute error if required and put out results → RETURN

Figure 2. SUB19 which controls the calculation and output for a single-valued function.
Figure 3. SUB13 which calculates the error if required and puts out the results.
46

CALL SUB11 to compute functions of 3 principal axes and 3 cartesian axes

Loop thru all kinds of atoms and CALL SUB11 to compute functions of 3 principal axes and 3 cartesian axes

CALL SUB10 to compute functions of 3 principal axes

Loop thru all kinds of atoms and CALL SUB10 to compute functions of 3 principal axes

CALL SUB8 to compute interatomic distances between atoms in two asymmetric units

Loop thru cell and symmetry positions and CALL SUB8 to compute distances

Figure 4. SUB21 which controls the calculations for multiple-values functions.
Figure 5. SUB8 which calculates all distances less than DMAX between atoms of two asymmetric units.

Figure 6. SUB10 which controls the calculation of functions of each of the three principal axes of thermal motion.
Figure 7. SUB11 which controls the calculation of functions of each of the three principal axes and of three axes of a specified cartesian coordinate system.

Figure 8. FUNI which transfers control to the appropriate function-calculating subroutine. HEDI, PREI, and OUTI are almost analogous to FUNI.
SYMBOLIC PROGRAM LISTING

*TYPE (FORTRAN)
C ** OR FFE. FORTRAN CRYSTALLOGRAPHIC FUNCTION AND ERROR PROGRAM **F** FFE 001
C CALLING PROGRAM **F** FFE 002

COMMON NG,IPM,NP,IAM,NS,NV,JXP,JX,JB,NM,SCALE,ITF,NG,INSAVE **F** FFE 003
COMMON DMA,MA,FX,LNZ,KK,KD,SAVEA,SAVEP,VARA,VARP,E1,E1H,NSP **F** FFE 004
COMMON TITLE,IPM,DP,DFDP,AM,DA,DFDA,IN,T,S,IS,A,AA,AB,P,ROW,K11,K12 **F** FFE 005
DIMENSION TITLE(12),PM(200),DP(200),DFDP(200),AM(21),DA(6) **F** FFE 006
DIMENSION DFDA(16),IN(231),TS(3,48),IS(2,3,48),A(6),AA(3,3),BB(3,3) **F** FFE 007
DIMENSION P(300),ROW(6),K11(300),K12(300) **F** FFE 008

C READ AND PUT OUT TITLE AND CONTROL CARD **F** FFE 009
READ INPUT TAPE 1D,00102,(TITLE(1),1#1,12) **F** FFE 0010
00102 FORMAT (12A6) **F** FFE 0011
WRITE OUTPUT TAPE 9,00104,(TITLE(1),1#1,12) **F** FFE 0012
00104 FORMAT (11H12A6) **F** FFE 0013
READ INPUT TAPE 1D,00202,IPM,NP,IAM,NS,NV,JXP,JX,JB,NM **F** FFE 0014
00202 FORMAT (24I1) **F** FFE 0015
IF (NP) 00297,00294,00299 **F** FFE 0016
00294 WRITE OUTPUT TAPE 9,00295 **F** FFE 0017
00295 FORMAT (39HINPUT DATA TO BE READ FROM OR FLS TAPE) **F** FFE 0018
GO TO 00301 **F** FFE 0019
00297 WRITE OUTPUT TAPE 9,00298,NP **F** FFE 0020
00298 FORMAT (33HINPUT DATA TO BE READ FROM CARDS/ **F** FFE 0021
133HNUMBER OF STRUCTURE PARAMETERS IS14) **F** FFE 0022
00301 WRITE OUTPUT TAPE 9,00302 **F** FFE 0023
00302 FORMAT (63HCOVARIANCE-COVARIANCE MATRIX AND PARAMETER SELECTION INFFFE 0024
FORMATION WILL) **F** FFE 0025
IF (IPM) 00308,00305,00308 **F** FFE 0026
00305 WRITE OUTPUT TAPE 9,00306 **F** FFE 0027
00306 FORMAT (12HNOT TO BE USED) **F** FFE 0028
GO TO 00310 **F** FFE 0029
00308 WRITE OUTPUT TAPE 9,00309 **F** FFE 0030
00309 FORMAT (8HBE USED) **F** FFE 0031
00310 WRITE OUTPUT TAPE 9,00311,N5 **F** FFE 0032
00311 FORMAT (23HNUMBER OF SYMMETRY CARDS IS13) **F** FFE 0033
WRITE OUTPUT TAPE 9,00314 **F** FFE 0034
00314 FORMAT (26HCOVARIANCE-COVARIANCE MATRIX ERRORS ARE) **F** FFE 0035
IF (IAM-1) 00316,00319,00322 **F** FFE 0036
00316 WRITE OUTPUT TAPE 9,00317 **F** FFE 0037
00317 FORMAT (15HNOT TO BE USED) **F** FFE 0038
GO TO 00401 **F** FFE 0039
00319 WRITE OUTPUT TAPE 9,00320 **F** FFE 0040
00320 FORMAT (42HTO BE READ IN THE FORM OF STANDARD ERRORS) **F** FFE 0041
GO TO 00401 **F** FFE 0042
00322 WRITE OUTPUT TAPE 9,00323 **F** FFE 0043
00323 FORMAT (63HBE USED) **F** FFE 0044
GO TO 00401 **F** FFE 0045
00401 IF (NP) 00401,00401,00401 **F** FFE 0046
C READ PARAMETERS AND COVARIANCE-COVARIANCE MATRIX FROM CARDS **F** FFE 0047
00401 WRITE OUTPUT TAPE 9,00502,NV **F** FFE 0048
00502 FORMAT (44HNUMBER OF THE COVARIANCE-COVARIANCE MATRIX FOR **F** FFE 0049
7H THE STRUCTURE PARAMETERS IS13) **F** FFE 0050
IF (JXP) 00509,00509,00509 **F** FFE 0051
00401 WRITE OUTPUT TAPE 9,00502,NV **F** FFE 0052
00502 FORMAT (44HNUMBER OF THE COVARIANCE-COVARIANCE MATRIX FOR **F** FFE 0053
7H THE STRUCTURE PARAMETERS IS13) **F** FFE 0054
IF (JXP) 00509,00509,00509 **F** FFE 0055
00401 WRITE OUTPUT TAPE 9,00502,NV **F** FFE 0056
00502 FORMAT (44HNUMBER OF THE COVARIANCE-COVARIANCE MATRIX FOR **F** FFE 0057
7H THE STRUCTURE PARAMETERS IS13) **F** FFE 0058
IF (JXP) 00509,00509,00509 **F** FFE 0059
C
WRITE OUTPUT TAPE 9,0,7506,JXP,JX
FORMAT 14(6H,2F8.2)
READ INPUT TAPE 10,0,9061,FIP,FJ
READ INPUT TAPE 10,0,9062,FIP,FJ
READ INPUT TAPE 10,0,9063,FIP,FJ
READ INPUT TAPE 10,0,9064,FIP,FJ
READ INPUT TAPE 10,0,9065,FIP,FJ
READ INPUT TAPE 10,0,9066,FIP,FJ
READ INPUT TAPE 10,0,9067,FIP,FJ
READ INPUT TAPE 10,0,9068,FIP,FJ
READ INPUT TAPE 10,0,9069,FIP,FJ
READ INPUT TAPE 10,0,9070,FIP,FJ
READ INPUT TAPE 10,0,9071,FIP,FJ
READ INPUT TAPE 10,0,9072,FIP,FJ
READ INPUT TAPE 10,0,9073,FIP,FJ
READ INPUT TAPE 10,0,9074,FIP,FJ
READ INPUT TAPE 10,0,9075,FIP,FJ
READ INPUT TAPE 10,0,9076,FIP,FJ
READ INPUT TAPE 10,0,9077,FIP,FJ
READ INPUT TAPE 10,0,9078,FIP,FJ
READ INPUT TAPE 10,0,9079,FIP,FJ
READ INPUT TAPE 10,0,9080,FIP,FJ
READ INPUT TAPE 10,0,9081,FIP,FJ
READ INPUT TAPE 10,0,9082,FIP,FJ
READ INPUT TAPE 10,0,9083,FIP,FJ
READ INPUT TAPE 10,0,9084,FIP,FJ
READ INPUT TAPE 10,0,9085,FIP,FJ
READ INPUT TAPE 10,0,9086,FIP,FJ
READ INPUT TAPE 10,0,9087,FIP,FJ
READ INPUT TAPE 10,0,9088,FIP,FJ
READ INPUT TAPE 10,0,9089,FIP,FJ
READ INPUT TAPE 10,0,9090,FIP,FJ
READ INPUT TAPE 10,0,9091,FIP,FJ
READ INPUT TAPE 10,0,9092,FIP,FJ
READ INPUT TAPE 10,0,9093,FIP,FJ
READ INPUT TAPE 10,0,9094,FIP,FJ
READ INPUT TAPE 10,0,9095,FIP,FJ
READ INPUT TAPE 10,0,9096,FIP,FJ
READ INPUT TAPE 10,0,9097,FIP,FJ
READ INPUT TAPE 10,0,9098,FIP,FJ
READ INPUT TAPE 10,0,9099,FIP,FJ
READ INPUT TAPE 10,0,9100,FIP,FJ
READ INPUT TAPE 10,0,9101,FIP,FJ
READ INPUT TAPE 10,0,9102,FIP,FJ
READ INPUT TAPE 10,0,9103,FIP,FJ
READ INPUT TAPE 10,0,9104,FIP,FJ
READ INPUT TAPE 10,0,9105,FIP,FJ
READ INPUT TAPE 10,0,9106,FIP,FJ
READ INPUT TAPE 10,0,9107,FIP,FJ
READ INPUT TAPE 10,0,9108,FIP,FJ
READ INPUT TAPE 10,0,9109,FIP,FJ
READ INPUT TAPE 10,0,9110,FIP,FJ
READ INPUT TAPE 10,0,9111,FIP,FJ
READ INPUT TAPE 10,0,9112,FIP,FJ
READ INPUT TAPE 10,0,9113,FIP,FJ
READ INPUT TAPE 10,0,9114,FIP,FJ
READ INPUT TAPE 10,0,9115,FIP,FJ
READ INPUT TAPE 10,0,9116,FIP,FJ
READ INPUT TAPE 10,0,9117,FIP,FJ
READ INPUT TAPE 10,0,9118,FIP,FJ
READ INPUT TAPE 10,0,9119,FIP,FJ
READ INPUT TAPE 10,0,9120,FIP,FJ
READ INPUT TAPE 10,0,9121,FIP,FJ
READ INPUT TAPE 10,0,9122,FIP,FJ
READ INPUT TAPE 10,0,9123,FIP,FJ
READ INPUT TAPE 10,0,9124,FIP,FJ
READ INPUT TAPE 10,0,9125,FIP,FJ
READ INPUT TAPE 10,0,9126,FIP,FJ
READ INPUT TAPE 10,0,9127,FIP,FJ
READ INPUT TAPE 10,0,9128,FIP,FJ
READ INPUT TAPE 10,0,9129,FIP,FJ
READ INPUT TAPE 10,0,9130,FIP,FJ
READ INPUT TAPE 10,0,9131,FIP,FJ
READ INPUT TAPE 10,0,9132,FIP,FJ
READ INPUT TAPE 10,0,9133,FIP,FJ
READ INPUT TAPE 10,0,9134,FIP,FJ
READ INPUT TAPE 10,0,9135,FIP,FJ
READ INPUT TAPE 10,0,9136,FIP,FJ
C  READ STANDARD ERRORS OF CELL PARAMETERS
DO 02402  #1,21

C  READ VARIANCE-COVARIANCE MATRIX FOR CELL PARAMETERS
READ INPUT TAPE 10,02202,AM(I),AM(J),AM(K),AM(L),
WRITE OUTPUT TAPE 9,02503,AM(I),AM(J),AM(K),AM(L),

C  FORMAT (4,H9,STANDARD ERRORS, RESPECTIVELY, OF THE ABOVE CELL
        PARAMETER S/1H06F11,4)
DO 02602  #1,21

C  GO TO 03201

C  COMPUTE CELL PARAMETER INCREMENTS USED TO OBTAIN DERIVATIVES
DO 03003  #1,6

C  READ AND PUT OUT SYMMETRY TRANSFORMATIONS
READ INPUT TAPE 10,03602,TS(I,J),TS(K,J),TS(K,I),J#1,3
WRITE OUTPUT TAPE 9,03604,TS(I,J),TS(K,J),TS(K,I),J#1,NS

C  FORMAT (2,H10,SYMMETRY INFORMATION/19HD
TRANSFORMED X
4B1
TRANSFORMED Y
TRANSFORMED Z/1H1
WRITE OUTPUT TAPE 9,03608,TS(I,J),TS(K,J),TS(K,I),J#1,3

C  COMPUTE PARAMETER INCREMENTS USED TO OBTAIN DERIVATIVES
DO 03903  #1,NV
READ ONE SET OF INSTRUCTIONS
DO 04501 IN#1+10

READ INPUT TAPE 10,0024,11,J#1,K#1,X#1
IF(IN(J)104501~04601~04501
K#1

IF(IN(J)104501~04601~04501

CALL EXIT

IF(IN(J)104501~04601~04501

CALL HENT(IN(J))

INSAVE#IN(J)

CALL SUB19

GO TO 04701

C

SUBROUTINE SUBR

END(0,1,0,0,7)

TYPE(FORTRAN)

C

COMMON NG,NF,NP,IPAM,NAM,NSM,NV,JKP,JP,JBP,JBM,NSC,1TP,NO,INSAVE

COMMON DMAX,NA,F,FX,FXZ,XX,DDA,DDA,IN,J,I5,J,AA,AB,AB,AB,AMX,K1I,K12

COMMON TITLE,PM,DP,DFDP,AM,DA,ADAF,IN,J,J5,J,AA,AB,AB,AB,AMX,K1I,K12

COMMON TITLE,PM,DP,DFDP,AM,DA,ADAF,IN,J,J5,J,AA,AB,AB,AB,AMX,K1I,K12

COMMON TITLE,PM,DP,DFDP,AM,DA,ADAF,IN,J,J5,J,AA,AB,AB,AB,AMX,K1I,K12

COMMON TITLE,PM,DP,DFDP,AM,DA,ADAF,IN,J,J5,J,AA,AB,AB,AB,AMX,K1I,K12

COMMON TITLE,PM,DP,DFDP,AM,DA,ADAF,IN,J,J5,J,AA,AB,AB,AB,AMX,K1I,K12

DIMENSION TITLE,PM,DP,DFDP,AM,DA,ADAF,IN,J,J5,J,AA,AB,AB,AB,AMX,K1I,K12

DIMENSION TITLE,PM,DP,DFDP,AM,DA,ADAF,IN,J,J5,J,AA,AB,AB,AB,AMX,K1I,K12

DIMENSION TITLE,PM,DP,DFDP,AM,DA,ADAF,IN,J,J5,J,AA,AB,AB,AB,AMX,K1I,K12

C

SET MAXIMUM DISTANCE ACCEPTED
IF(IN(J1)1077,107,

I07 DMAX#1.0

GOTO 0011

I11 DMAX#1.0

I13 NA#1

I17 DO2151#1,NA

I21#1

C

START LOOP FOR FIRST ATOM
DO2151#1

IN(J1)

C

START LOOP FOR SECOND ATOM
DO213J#1,NA
C TEST TO AVOID DUPLICATION
IF(IN(3)-IN(5)) I201,125,201
175 IF(J-1) I201,215,215     
201 IN(4)#J
NG0
C COMPUTE AND TEST DISTANCE
CALL F1N1(IN(1))
1F(FX-DMAX) I209,209,213
C COMPUTE ERROR AND PUT OUT RESULTS
FXFX
CALLSUB13
213 CONTINUE
C END LOOP FOR SECOND ATOM
215 CONTINUE
C END LOOP FOR FIRST ATOM
217 RETURN
END(1,0,0,0)
*TYPE(FORTRAN)
C MULTIVALUED FUNCTIONS 7, 8, AND 9
SUBROUTINE SUB1D
C COMMON NG,NP,IAJ,NS,NJ,KK,KKJ,SAVEA,SAVEE,VARA,VARP,ELF,ELHF,NSP
COMMON NF,PM,NF,IAJ,NS,NJ,KK,KKJ,SAVEA,SAVEE,VARA,VARP,ELF,ELHF,NSP
COMMON DIM(12),PM(231),DN(201),DFP(231),AM(21),DA(6)
DIMENSION DFD(13),RDM(61),K11(300),K12(300)
C START LOOP THRU PRINCIPAL AXES
DIM(1)#1,3
IN(4)#1
C COMPUTE AND PUT OUT FUNCTION AND ERROR
100 CALLSUB19
RETURN
END(1,0,0,0)
*TYPE(FORTRAN)
C MULTIVALUED FUNCTIONS 10 AND 11
SUBROUTINE SUB11
C COMMON NG,NP,IAJ,NS,NJ,KK,KKJ,SAVEA,SAVEE,VARA,VARP,ELF,ELHF,NSP
COMMON DM(12),NA,FX,LFNZ,KK,KKJ,SAVEA,SAVEE,VARA,VARP,ELF,ELHF,NSP
COMMON TITL,FPM,DFD,AM,DA,DFD,IN,TS,15,AA,ABPR,ROW,K11,K12
DIMENSION TITL(12),PM(231),DN(201),DFD(231),AM(21),DA(6)
DIMENSION DFD(13),RDM(61),K11(300),K12(300)
C START LOOP THRU PRINCIPAL AXES
DIM(11)#1,3
C START LOOP THRU REFERENCE AXES
DIM(11)#1,3
IN(5)#J
C COMPUTE AND PUT OUT FUNCTION AND ERROR
110 CALLSUB19
RETURN
*TYPE(FORTRAN)
C ERROR CALCULATION AND OUTPUT
SUBROUTINE EVARI
COMMON NG,IP,NP,IA,MS,XP,JXP,JBP,JB,NM,SCALE,TF,NQ,INSAVE
COMMON DMA,MN,FX,FXN,Z,KK,KDD,SAVEA,SAVEF,VARF,VARP,VARF,F,E,X,JH,NP
COMMON TITLE,PM,DP,DFP,AMP,DA,DFDA,IN,TF,5,15,AA,AP,P,R,KI,K12
DIMENSION T(12),PM(200),DP(200),DFD(200),AM(21),DA(6)
DIMENSION DFDA(6),IN(231),TS(3,48),TS(2,48),A16,A1(3,3),BA(3,3)
C PUT OUT FUNCTION DESCRIPTION
CALL OUTINF(1)
IF(ING1107,1,13,107
C PUT OUT ERROR INDICATOR IF NOT ZERO
107 WRITE OUTPUT TAPE 9,109,NG
109 FORMAT (1H51X3H**13)
GO TO 723
113 VAR#P#D VAR#P#D IF(ING1119,319,119
C COMPUTE DERIVATIVES WITH RESPECT TO CELL PARAMETERS
119 DO 211 I#1,6
 IF(DFDA(I))#201,123,201
123 DFDA(I)#0,0
GO TO 211
201 SAVEA#A(I)
A(I)#A(I)+DA(I)
CALL FUNK(INI1))
A(I)#SAVEA
DFDA(I)#(FX-F1/DA(I)
211 CONTINUE
C COMPUTE VARIANCE BASED ON CELL PARAMETERS
K#1
L#6
DO 311 I#1,6
 IF(DFDA(I))#225,221,225
221 K#K+L
GO TO 311
225 C#1,0
DO 309 J#1,6
(IF(DFDA(J))#305,307,305
306 VARA#VARA+C#DFDA(I)#DFDA(J)#AM(K)
307 K#K+1
309 C#C+1
311 L#L-1
313 IF(1#1#315,615,315
C SELECT DERIVATIVES TO BE COMPUTED
315 DO 319 I#1,6
319 K#K#10
CALL P2F1(IN111)

C COMPUTE DERIVATIVES WITH RESPECT TO STRUCTURE PARAMETERS
J#N
DO 513 1#I,NV
403 J#J+1
  IF(KI1(J))407,403,407
407 IF(KI2(J))413,409,413
410 DFP(I)#N+J
  GO TO 513
413 IF(DP(I))501,409,501
501 SAVEP#P(J)
  P(J)#P(J)+DP(I)
  CALL FUN1(IN111)
  P(J)#SAVEP
  DFP(I)#(FX-F)/DP(I)
  LNZ#I
513 CONTINUE
C COMPUTE VARIANCE BASED ON STRUCTURE PARAMETERS
K#1
KKD#NV
DO 613 1#1,LMZ
  IF(DFP(I))523,612,523
523 K#K
C#1,N
  DO 611 J#1,LMZ
  IF(DFP(I))607,609,607
607  VARP#VARP+C*DFDPI1*DFDPI1*PMK1
609  K#K+1
611  C#2,N
612  KK#KK+KKD
613  KKD#KKD-1
614 IF(IN111)617,623,617
C PUT OUT ERROR INDICATOR IF NOT ZERO
617 WRITE OUTPUT TAPE 9,610,6,NG
619 FORMAT (1H48X,F9.4,6X,3H**13)
  GO TO 723
C COMPUTE STANDARD ERRORS AND PUT OUT RESULTS
623 E#SORTF(VARP)
  E#SORTF(VARP+VARA)
  IF(1PM)703,705,703
703 IF(IAPM)707,713,707
705 IF(IAPM)713,719,713
717 WRITE OUTPUT TAPE 9,709,F,E,E1
720 FORMAT (1H48X,F9.4,5H+OR-F7.4,2H (F7.4,1H))
    GO TO 723
713 WRITE OUTPUT TAPE 9,715,F,E
715 FORMAT (1H48X,F9.4,5H+OR-F7.4)
    GO TO 723
710 WRITE OUTPUT TAPE 9,721,F
721 FORMAT (1H 48X,F9.4)
723 RETURN
END(0,1,0,0,0)
*TYPEF(FORTRAN)
C FUNCTION AND ERROR CALCULATION
SUBROUTINE SUP19
COMMOM NG,IPM,NP,IA,NS,IV,X,J,XB,JB,IV,N,S,SCALE,ITF,NG,INSAVE
COMMOM DMAX,NA,F,FX,LNZ,KK,KKAP,SAPA,SAPF,VARA,VAR,VARP,E1,F,II,NSP
COMMON TITLE,P,M,DP,DMDP,AM,DA,DMDA,IN,TS,IS,A,AA,BB,P,ROX,KI,K12
DIMENSION DMDA(6),IN(231),TS(3,48),IS(2,3,48),A16,AA(3,3),BB(3,3)
DIMENSION PI(300),ROW(6),K11(300),K12(300)
C COMPUTE MULTIVALUED FUNCTIONS
* SUBROUTINE KUR21
COMMOM NG,IPM,NP,IA,NS,IV,X,J,XB,JB,IV,N,S,SCALE,ITF,NG,INSAVE
COMMOM DMAX,NA,F,FX,LNZ,KK,KKAP,SAPA,SAPF,VARA,VAR,VARP,E1,F,II,NSP
COMMON TITLE,P,M,DP,DMDP,AM,DA,DMDA,IN,TS,IS,A,AA,BB,P,ROX,KI,K12
DIMENSION DMDA(6),IN(231),TS(3,48),IS(2,3,48),A16,AA(3,3),BB(3,3)
DIMENSION PI(300),ROW(6),K11(300),K12(300)
C UNPACK INSTRUCTION NUMBER
IN#IN(1)/107
IN#IN(1)/106
C TRANSFER TO APPROPRIATE SECTION
IF(IN#IN(1)111111111111,110
111 IF(IN#IN(1)91113,113,305
113 IF(IN#IN(1)6115,115,215
115 IF(IN#IN(1)11117,117,317
117 IF(IN#IN(1)11,19,123
C COMPUTE ALL DISTANCES BETWEEN TWO UNITS
CALLSUBL
GOTO319
C COMPUTE DISTANCES INVOLVING BASIC UNIT AND ALL OTHERS
NA#IN(2)
NSP#NS+1
IN#IN
DC2IN#IN
O0211J#1,NSP
IN#IN(2)#NA
IN(5)#IN(11-1)+1
C CALLSUBL
GOTO319
215 IF(IN#IN(1121,217,221
C COMPUTE FUNCTIONS INVOLVING THREE PRINCIPAL AXES
CALLSUBL
GOTO319
C COMPUTE FUNCTIONS INVOLVING THREE PRINCIPAL AXES
FOR ALL ATOMS

C 211 NA#IN(2)
    DD3011#1,NA
    IN(2)#1
311 CALLSUB1B
    GOTO319
315 IF(IH-11307,307,311)
C 307 COMPUTE FOR ALL PRINCIPAL AXES AND ALL REFERENCE AXES
    CALLSUB1B
    GOTO319
C 311 NA#IN(2)
    DD3171#1,NA
    IN(2)#1
317 CALLSUB11
319 RETURN
END0(1+0,0,0)
*TYPE* (FORTRAN)
C SELECT THE HED SUBROUTINE TO BE ENTERED
    SUBROUTINE HED1(I1)
        K#XMGDF(I1,100)
        I(K)160,160,4
4 IF(K-1518,8,160
8 GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160)
10 CALL HED1
    GO TO 160
20 CALL HED2
    GO TO 160
30 CALL HED3
    GO TO 160
40 CALL HED4
    GO TO 160
50 CALL HED5
    GO TO 160
60 CALL HED6
    GO TO 160
70 CALL HED7
    GO TO 160
80 CALL HED8
    GO TO 160
90 CALL HED9
    GO TO 160
100 CALL HED10
    GO TO 160
110 CALL HED11
    GO TO 160
120 CALL HED12
    GO TO 160
139 CALL HED13
GO TO 160

140 CALL HED14
GO TO 160

150 CALL HED15

150 RETURN

END(I,O,0,0)

*TYPE (FORTRAN)
C SELECT THE PRE SUBROUTINE TO BE ENTERED
SUBROUTINE PRE(I)
4 IF(I)160,160,6
6 IF(I=15)17,17,160
8 GO TO 110,20,30,40,30,60,70,80,80,100,100,120,17,140,140,11

10 CALL PRE1
GO TO 160

20 CALL PRE2
GO TO 160

30 CALL PRE3
GO TO 160

40 CALL PRE4
GO TO 160

50 CALL PRE5
GO TO 160

60 CALL PRE6
GO TO 160

70 CALL PRE7
GO TO 160

80 CALL PRE8
GO TO 160

100 CALL PRE10
GO TO 160

120 CALL PRE12
GO TO 160

140 CALL PRE14

160 RETURN

END(I,O,0,0)

*TYPE (FORTRAN)
C SELECT THE FUN SUBROUTINE TO BE ENTERED
SUBROUTINE FUNJ(I)

COMMON NG,IP,NS,AM,NV,JXP,JB,JB,YM,SCALE,[TF,NO,INSAVE

COMMON DMX,NB,FX,LN2,KB,SAF,VARA,VARP,FI,FH

COMMON T,DEC,DP,DFP,AM,DA,DFDAIN,TS,IS,AA,BS,P,KI,KI2

DIMENSION TITLE(12),DP(200),DF(200),AM(210),DA(6)

DIMENSION DIA(6),IN(231),TS(13),IS(23),AA(6),BB(3,3)

DIMENSION PL(3,3),PK(6),KI(300),KIL(300)

CALL SETA(I)
CALL SETP(P)
IF(I)5,6,5
5 IF(I=15)18,8,6
6 NG=11
GO TO 160

8 GO TO (10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160)

10 CALL FUN1
   GO TO 160

20 CALL FUN2
   GO TO 160

30 CALL FUN3
   GO TO 160

40 CALL FUN4
   GO TO 160

50 CALL FUN5
   GO TO 160

60 CALL FUN6
   GO TO 160

70 CALL FUN7
   GO TO 160

80 CALL FUN8
   GO TO 160

90 CALL FUN9
   GO TO 160

100 CALL FUN10
   GO TO 160

110 CALL FUN11
   GO TO 160

120 CALL FUN12
   GO TO 160

130 CALL FUN13
   GO TO 160

140 CALL FUN14
   GO TO 160

150 CALL FUN15

160 RETURN

END(0, 0, 0, 0)

*TYPE (FORTRAN)
C SELECT THE OUT SUBROUTINE TO BE ENTERED
SUBROUTINE OUT(I)
4 IF(I)160, 160+6
   6 IF(1-15)8, 8, 160
8 GO TO (10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160)
10 CALL OUT1
   GO TO 160

20 CALL OUT2
   GO TO 160

30 CALL OUT3
   GO TO 160

40 CALL OUT4
   GO TO 160

FFE 697
FFE 698
FFE 699
FFE 700
FFE 701
FFE 702
FFE 703
FFE 704
FFE 705
FFE 706
FFE 707
FFE 708
FFE 709
FFE 710
FFE 711
FFE 712
FFE 713
FFE 714
FFE 715
FFE 716
FFE 717
FFE 718
FFE 719
FFE 720
FFE 721
FFE 722
FFE 723
FFE 724
FFE 725
FFE 726
FFE 727
FFE 728
FFE 729
FFE 730
FFE 731
FFE 732
FFE 733
FFE 734
FFE 735
FFE 736
FFE 737
FFE 738
FFE 739
FFE 740
FFE 741
FFE 742
FFE 743
FFE 744
FFE 745
FFE 746
FFE 747
FFE 748
FFE 749
FFE 750
FFE 751
FFE 752
FFE 753
FFE 754
FFE 755
FFE 756
FFE 757
FFE 758
FFE 759
FFE 760
FFE 761
FFE 762
FFE 763
FFE 764
FFE 765
FFE 766
60 CALL OUT6
GO TO 160

70 CALL OUT7
GO TO 160

80 CALL OUT8
GO TO 160

100 CALL OUT10
GO TO 160

120 CALL OUT12
GO TO 160

130 CALL OUT13

140 RETURN
ENDIF(1=0,0)

*TYPE(FORTRAN)
C SET KEY WORDS FOR ATOM COORDINATES
SUBROUTINE SETCA(1)

C #INCLUDE THE INSTRUCTION INTEGER SPECIFYING THE ATOM NUMBER
COMMON NG,IPMNP,IPAMNS,NV,JXP,JXR,JBP,JS,NM,SCALE,ITF,NG,INSAVE
COMMON DMAX,NAX,FX,LNZ,KK,KKD,SAVEV,SAVEF,VARA,XVARF,VARP,E1,E2,E3,E4,E5,E6,E7,E8,E9,E10
COMMON TITL1,PM,DP,DPF,AM,DAM,DFDA,IN,T1,S1,AA,AP,ROW,K11,K12
DIMENSION TITL1(12),PM(2001),DP(2001),DPFD,AM(21),DA(6)

DIMENSION DFDA(6),IN(231),TS(3,4,8),IS(2,3,4,8),AA(6),AA(3,3),BB(3,3)
DIMENSION P(300),ROW(6),K11(300),K12(300)
IF(11111111)767

111 J#JX+JXP*(1-1)

212 J1(J1+1)
213 J1(J1+2)

110 RETURN
ENDIF(1=0,0)

*TYPE(FORTRAN)
C SET KEY WORDS FOR ATOM BETAS
SUBROUTINE SETKB(1)

C #INCLUDE THE INSTRUCTION INTEGER SPECIFYING THE ATOM NUMBER
COMMON NG,IPMNP,IPAMNS,NV,JXP,JXR,JBP,JS,NM,SCALE,ITF,NG,INSAVE
COMMON DMAX,NAX,FX,LNZ,KK,KKD,SAVEV,SAVEF,VARA,XVARF,VARP,E1,E2,E3,E4,E5,E6,E7,E8,E9,E10
COMMON TITL1,PM,DP,DPF,AM,DAM,DFDA,IN,T1,S1,AA,AP,ROW,K11,K12
DIMENSION TITL1(12),PM(2001),DP(2001),DPFD,AM(21),DA(6)

DIMENSION DFDA(6),IN(231),TS(3,4,8),IS(2,3,4,8),AA(6),AA(3,3),BB(3,3)
DIMENSION P(300),ROW(6),K11(300),K12(300)
IF(11111111)767

111 J#JH+JBP*(1-1)

DO 117 K11=1,6

117 J1(J1)

119 RETURN
ENDIF(1=0,0)

*TYPE(FORTRAN)
C STORE METRIC TENSOR
SUBROUTINE STOA

COMMON NG,IPMNP,IPAMNS,NV,JXP,JXR,JBP,JS,NM,SCALE,ITF,NG,INSAVE
COMMON DMAX,NAX,FX,LNZ,KK,KKD,SAVEV,SAVEF,VARA,XVARF,VARP,E1,E2,E3,E4,E5,E6,E7,E8,E9,E10
COMMON TITL1,PM,DP,DPF,AM,DAM,DFDA,IN,T1,S1,AA,AP,ROW,K11,K12
DIMENSION TITL1(12),PM(2001),DP(2001),DPFD,AM(21),DA(6)

DIMENSION DFDA(6),IN(231),TS(3,4,8),IS(2,3,4,8),AA(6),AA(3,3),BB(3,3)
DIMENSION P(300),ROW(6),K11(300),K12(300)

AA(1,1)AA(1,1)AA(1)
AA(2,1)AA(2,1)AA(2)
AA(3,1)AA(3,1)AA(3)
AA(1,2)AA(1,2)AA(1,2)
AA(1,3)AA(1,3)AA(1,3)
AA(2,2)AA(2,2)AA(2,2)
AA(3,2)AA(3,2)AA(3,2)
AA(1,4)AA(1,4)AA(1,4)
AA(2,4)AA(2,4)AA(2,4)
AA(3,4)AA(3,4)AA(3,4)


 dimension D(12,12)

d=0.0

 do i=1,12
      do j=1,12
           d(i,j)=d(i,j)+a(i,j)*b(i,j)
      enddo
 enddo

 return
 end
215  Y(J)#TS(J,KS) FF1 911
215  D0309K#1,2
215  IF(L)225,307,375
225  L#-
235  Y(K)#Y(K-I)-FLOAT(KC1)
235  G0TO377
305  Y(K)#Y(K)+X(L)
309  CONTINUE
311  KC#KC-4#KC4
311  KC2#KC3/2
311  KC1#KC3-2#KC2
311  Z(1)#Y(I)-FLOAT(KC1)
311  Z(2)#Y(I2)-FLOAT(KC2)
311  Z(3)#Y(I3)-FLOAT(KC4)
325  RETURN
325  END(*,1,0,0)
325  *TYPE(FORTRAN)
325  STORE TRANSFORMED ANISOTROPIC TEMP FACTOR MATRIX
C INS IS ATOM DESCRIPTION, Z IS TRANSFORMED MATRIX
SUBROUTINE BETA(INS,7)
COMMON NG1,PM,NP,IA,NV,JX,JB,NM,SCALE,ITF,NQ,INSAVE
COMMON DMAX,NA,FX,LNZ,KK,SAVEA,SAVEP,VARA,VARP,E,C,EL,NSP
COMMON TITLE,PM,DP,DFP,AM,DA,DFDA,N,TS,Y,S,AA,BB,P,RK11,K12
DIMENSION TITLE(12),PM(27),NP(21),NM(21),AM(21),DA(6)
DIMENSION DFDA(6),IN(123),TS(3,48),ST(3,48),ST2,ST3,ST4
DIMENSION PM(30),NN(4),K1(30),K2(30)
DIMENSION INS(2),Z(3,3),B(4),B2(9)
IF(ITF-2)100,110,110
100  IF(JB#-6)110,115,115
110  NG#4
G0TO423
115  KS#XMODF(INS(2),100)
115  IF(KS#121,119,119)
115  IF(KS#-N512,125,121)
121  NG#1
G0TO423
125  IF(INS(1)121,201,207)
201  D03031#1,6
201  D03031#1,6
201  D02191#1,6
207  B(I)#(P(J)
G0TO221
277  JKB#JBP#INS(1)-1)
277  IF(JBP-5)125,215,211
211  NG#5
G0TO429
215  D02191#1,6
215  B(I)#(P(J)
219  JKB#1)
221  B2(1)#B1(1)
221  B2(2)#B1(6)
221  B2(3)#B1(5)
221  B2(4)#B1(2)
221  B2(5)#B1(4)
221  B2(6)#B1(3)
D0421#1,3
D0419#1,3
IF(KS#121,313,319
313  M#1#J
313  B#B21(M)
G0TO415
319  B3#3#0
D0413#1,2
D0411#1,2
M#151#1,KS#151#1,KS
IF(M#140#1.1,403
403  33#B3#52(M)
GOTO 411
407 M\#M
B3\#B3-B2(M)
411 CONTINUE
413 CONTINUE
415 Z(I,J)\#H
Z(I,J)\#H3
419 CONTINUE
421 CONTINUE
423 RETURN
END(0,0,0,0,0)
*TYPE(FORTRAN)
C MULTPLY TWO MATRICES
C Z(3,3)\#X(3,3)\#Y(3,3)
SUBROUTINE MM(X,Y,Z)
DIMENSIONX(3,3),Y(3,3),Z(3,3)
DO117 I=1,3
DO117 K=1,3
Z(I,K)\#D
DO117 J=1,3
117 Z(I,K)\#Z(1,K)+X(I,J)*Y(J,K)
RETURN
END(0,0,0,0,0)
*TYPE(FORTRAN)
C MATRIX \ VECTOR
C Z(3)\#X(3,3)\#Y(3)
SUBROUTINE MV(X,Y,Z)
DIMENSIONX(3,3),Y(3,3),Z(3,1)
DO118 I=1,3
Z(I)\#C\#D
DO118 J=1,3
118 Z(I)\#Z(I)+X(I,J)*Y(J)
RETURN
END(0,0,0,0,0)
*TYPE(FORTRAN)
C TRANSPOSED VECTOR TIMES MATRICES
C Z(3)\#X(3,3)\#Y(3,3)
SUBROUTINE MVV(X,Y,Z)
DIMENSIONX(3,3),Y(3,3),Z(3,1)
DO111 I=1,3
Z(J)\#C\#D
DO111 J=1,3
115 Z(J)\#Z(J)+X(I,J)*Y(I,J)
RETURN
END(0,1,0,0,0)
*TYPE(FORTRAN)
C TRANSPOSED VECTOR \ VECTOR
C VV\#X(3,3)\#Y(3)
FUNCTION MMV(X,Y)
DIMENSIONX(3,3),Y(3,3)
VV\#D
DO111 I=1,3
111 VV\#VV\#X(I)\#Y(I)
RETURN
END(0,1,0,0,0)
*TYPE(FORTRAN)
C TRANSPOSED VECTOR \ MATRIX \ VECTOR
C VV\#X(3,3)\#Y(3)
FUNCTION MMVW(X,Y)
DIMENSIONW(3,3),X(3,3),Y(3,3),Z(3,3)
CALLMV(X,Y,Z)
VV\#W\#Z
RETURN
END(3,1,0,0,0)
*TYPE(FORTRAN)
C VECTOR \ VECTOR
C Z(3)\#X(3)\#Y(3)
SUBROUTINE MVVW(X,Y,Z)
DIMENSIONX(3,3),Y(3,3),Z(3,3)
DO111 I=1,3
C COMPUTE THE SUM OF TWO VECTORS
C
Z(3)*X(3)+Y(3)
DIMENSION X(3),Y(3),Z(3)
DO I=1,3
111 Z(I)*X(I)+Y(I)
RETURN

*TYPE(FORTRAN) FFE 1050
C
C FInd EIGENVALUES Y OF MATRIX
C
FUNCTION COSVV(X,Y)
SUBROUTINE EIGVAL(W,Y)

DIMENSION X(3),Y(3),L(3),XI(3),YI(3),21(3)
DIMENSION TITLE(3),P(3),DP(3),DF(3),AY(3),DA(3)
COMMON I 0 8

D O I I 1 9 I #1,3
RETURN

FUNCTION ARCCOS(X)
YGVQ
IF(DZIII5 I FFE
RETURN

SUBROUTINE INF POPRm( X ,Y ,Z )
RETURN

FUNCTION ARCCOS#90.0-(57.2957751)*SIGNF(ASINF(X1,X)
END(F9

SUBROUTINE AxXf( X ,V )
DIMENSION U(3),V(3),X(3)
DO I=1,3
113 X(I)*XI(I)
RETURN

*TYPE(FORTRAN)
C
STORE A VECTOR Z NORMAL TO VECTORS X AND Y
C
SUBROUTINE NORM(X,Y,Z)

DIMENSION TITLE(3),P(3),DP(3),DF(3),AY(3),DA(3)
COMMON I 0 8

DO I=1,3
115 Y(I)=X(I)*XI(I)
RETURN

*TYPE(FORTRAN)
C
STORE THREE MUTUALLY PERPENDICULAR
C
VECTORS X(1,1), X(1,2), AND X(1,3) GIVEN
C
SUBROUTINE AXYF(U,V,X)
DIMENSION U(3),V(3),X(3)
DO I=1,3
113 X(I)*XI(I)
RETURN
COMMON NG
DIMENSION W(3,3), Y(3), X(3), Z(6,6)
DO 119 J=1,3
DO 119 I=1,3
Z(I,J)=0.0
Z(J,I)=0.0
Z(I+J,3,J)=0.0
Z(I+J,3)=0.0
I=19
Z(I,J)=Z(I,J)
VIR=Z(3,1)*Z(2,1)+Z(1,1)*Z(1,2)-Z(1,1)*Z(2,1)+Z(2,1)*Z(3,1)
C#0/4.5
A(1)=P3
B(2)=P3*P3-P3*Q+R
B(5)=P5*2
A(3)=A/3.0
B(4)=B2*0.2
B(7)=A3#A3#A3
IF(B4#.247)<303, 215
Z#0.5
GOTO 303
225 NGS=7
GOTO 317
303 PHI3#1ASIN(SGRTF(1.0+.84/A27)))
IF(0)<307, 305, 307
375 B#0.5
307 C2=-SIGNF(1.0+SGRTF(-A3)))
X(1)#C*COSF(PHI3)
X(2)#C*COSF(PHI3+4.188707D05)
X(3)#C*COSF(PHI3+2.094395103)
IF(R)<311, 313, 313
311 HOLDNKL(1)
X(1)#X(3)
X(3)#HOLD
313 D0315=#1, 3
315 Y(I)#X(1)=P3
317 RETURN
END(0,1)+D+0.0
*TYPE(FORTRAN)
C COMPUTE EIGENVECTOR Z OF MATRIX
C GIVEN EIGENVALUE Y
C
COMMON NG
DIMENSION W(3,3), X(3), Y(3)
DO 123 J=1,3
DO 123 I=1,3
X(I,J)=0.0
X(J,I)=0.0
X(I+J,3,J)=0.0
X(I+J,3)=0.0
I=123
X(I+J,3,J)=X(I+J,3)
Y#Y
DO 223 I=1,3
X(I+J,1)=X(I+J,1)-Y1
X(I+J,1)=X(I+J,1)-Y1
X(I+J,1)=X(I+J,1)-Y1
X(I+J,1)=X(I+J,1)-Y1
S#S
DO 223 J=1,3
PJX(I,J)=X(I+J,2)-X(I+J,2)*X(I+J,1)
PJ(I)=PJ
223 S#S+PJ#PJ
C ANGLE SUBROUTINE USED BY FUN2, FUN5, FUN6
FUNCTION FUNA(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
CALL DIFV(X, V, J)
RETURN
END

FUNCTION FUNB(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUN9(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNA(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNB(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUN9(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNA(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNB(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUN9(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNA(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNB(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUN9(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNA(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNB(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUN9(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNA(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNB(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUN9(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNA(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUNB(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END

FUNCTION FUN9(J)
COMMON NG, IP, NP, IA, NA, N, JX, JY, JZ, CB, NM, SCALE, ITF, NO, INS
DIMENSION (J + 1)
RETURN
END
DIMENSION P(300),R(6),K11(300),K12(300)
CALL SETKX(IN(1))
CALL SETKX(IN(4))
RETURN
END(1,0,0,0)

* TYPE(FORTRAN)  FFE 1257
C
SUBROUTINE FUN1  FFE 1260
COMMON NG,IPM,NP,IA,M,NS,NV,JPX,JX,JB,JB,JB,NM,SCALE,ITF,NO,INSAVE  FFE 1265
COMMON DMAX,NA,F,F,X,LNZ,KK,KKD,SAVEA,SAYEP,VARA,VARP,EL,E1,E1,E2,E1,EH,INS  FFE 1266
COMMON TITLE,PM,DP,DFDP,AM,DA,DFDA,IN,T5,15,A,AA,BB,RO4,K11,K12  FFE 1267
DIMENSION TITLE(12),PM(200),DP(200),DFDP(200),AM(111),DA(6)  FFE 1268
DIMENSION DFDA(14,IN(231)),TS,15,A,AA,AB,AA,AA,AB,BB,K11,K12  FFE 1269
DIMENSION P(300),R(6),K11(300),K12(300)
FDFUN(1,IN(21))
RETURN
END(1,0,0,0)

* TYPE(FORTRAN)  FFE 1270
C
SUBROUTINE OUT  FFE 1276
COMMON NG,IPM,NP,IA,M,NS,NV,JPX,JX,JB,JB,JB,NM,SCALE,ITF,NO,INSAVE  FFE 1277
COMMON DMAX,NA,F,F,X,LNZ,KK,KKD,SAVEA,SAYEP,VARA,VARP,EL,E1,E2,E1,EH,INS  FFE 1278
COMMON TITLE,PM,DP,DFDP,AM,DA,DFDA,IN,T5,15,A,AA,AB,RO4,K11,K12  FFE 1279
DIMENSION TITLE(12),PM(200),DP(200),DFDP(200),AM(111),DA(6)  FFE 1280
DIMENSION DFDA(15,IN(231)),TS,15,A,AA,AB,AA,AA,AB,BB,K11,K12  FFE 1281
DIMENSION P(300),R(6),K11(300),K12(300)
WRITE(OUTPUTTAPE,107)IN(11),IN(12),IN(13)
RETURN
END(1,0,0,0)

* TYPE(FORTRAN)  FFE 1284
C
SUBROUTINE OUT  FFE 1287
COMMON NG,IPM,NP,IA,M,NS,NV,JPX,JX,JB,JB,JB,NM,SCALE,ITF,NO,INSAVE  FFE 1288
COMMON DMAX,NA,F,F,X,LNZ,KK,KKD,SAVEA,SAYEP,VARA,VARP,EL,E1,E2,E1,EH,INS  FFE 1289
COMMON TITLE,PM,DP,DFDP,AM,DA,DFDA,IN,T5,15,A,AA,AB,RO4,K11,K12  FFE 1290
DIMENSION TITLE(12),PM(200),DP(200),DFDP(200),AM(111),DA(6)  FFE 1291
DIMENSION DFDA(15,IN(231)),TS,15,A,AA,AB,AA,AA,AB,BB,K11,K12  FFE 1292
DIMENSION P(300),R(6),K11(300),K12(300)
WRITE(OUTPUTTAPE,107)IN(11),IN(12),IN(13)
RETURN
END(1,0,0,0)

* TYPE(FORTRAN)  FFE 1296
C
SUBROUTINE OUT  FFE 1297
COMMON NG,IPM,NP,IA,M,NS,NV,JPX,JX,JB,JB,JB,NM,SCALE,ITF,NO,INSAVE  FFE 1298
COMMON DMAX,NA,F,F,X,LNZ,KK,KKD,SAVEA,SAYEP,VARA,VARP,EL,E1,E2,E1,EH,INS  FFE 1299
COMMON TITLE,PM,DP,DFDP,AM,DA,DFDA,IN,T5,15,A,AA,AB,RO4,K11,K12  FFE 1300
DIMENSION TITLE(12),PM(200),DP(200),DFDP(200),AM(111),DA(6)  FFE 1301
DIMENSION DFDA(15,IN(231)),TS,15,A,AA,AB,AA,AA,AB,BB,K11,K12  FFE 1302
DIMENSION P(300),R(6),K11(300),K12(300)
WRITE(OUTPUTTAPE,107)IN(11),IN(12),IN(13)
RETURN
END(1,0,0,0)

* TYPE(FORTRAN)  FFE 1304
C
SUBROUTINE OUT  FFE 1305
COMMON NG,IPM,NP,IA,M,NS,NV,JPX,JX,JB,JB,JB,NM,SCALE,ITF,NO,INSAVE  FFE 1306
COMMON DMAX,NA,F,F,X,LNZ,KK,KKD,SAVEA,SAYEP,VARA,VARP,EL,E1,E2,E1,EH,INS  FFE 1307
COMMON TITLE,PM,DP,DFDP,AM,DA,DFDA,IN,T5,15,A,AA,AB,RO4,K11,K12  FFE 1308
DIMENSION TITLE(12),PM(200),DP(200),DFDP(200),AM(111),DA(6)  FFE 1309
DIMENSION DFDA(15,IN(231)),TS,15,A,AA,AB,AA,AA,AB,BB,K11,K12  FFE 1310
DIMENSION P(300),R(6),K11(300),K12(300)
WRITE(OUTPUTTAPE,107)IN(11),IN(12),IN(13)
RETURN
END(1,0,0,0)

* TYPE(FORTRAN)  FFE 1312
C
SUBROUTINE OUT  FFE 1313
COMMON NG,IPM,NP,IA,M,NS,NV,JPX,JX,JB,JB,JB,NM,SCALE,ITF,NO,INSAVE  FFE 1314
COMMON DMAX,NA,F,F,X,LNZ,KK,KKD,SAVEA,SAYEP,VARA,VARP,EL,E1,E2,E1,EH,INS  FFE 1315
COMMON TITLE,PM,DP,DFDP,AM,DA,DFDA,IN,T5,15,A,AA,AB,RO4,K11,K12  FFE 1316
DIMENSION TITLE(12),PM(200),DP(200),DFDP(200),AM(111),DA(6)  FFE 1317
DIMENSION DFDA(15,IN(231)),TS,15,A,AA,AB,AA,AA,AB,BB,K11,K12  FFE 1318
DIMENSION P(300),R(6),K11(300),K12(300)
WRITE(OUTPUTTAPE,107)IN(11),IN(12),IN(13)
RETURN
END(1,0,0,0)
*TYPE*(FORTRAN)

C "HEADING 2"

WRITE OUTPUT TAPE 9,107, (IN(1),J12,17)
RETURN
END(1,1,0,0)

*TYPE*(FORTRAN)

C "PRELIMINARY SUBROUTINE 3"

COMMON NG,IPM,NP,IA,INL,NS,VM,JS,VP,JB,NN,SCALE,1TF,NG,INSAVE
COM

COMMON DM,NA,FXN,NZ,DFV,AM,DA,DFSA,NN,TS,IS,AA,BB,DP,ROW,K11,K12

RETURN
END(1,1,0,0)

*TYPE*(FORTRAN)

C "DIHEDRAL ANGLE SUBROUTINE"

COMMON NG,IPM,NP,IA,INL,NS,VM,JS,VP,JB,NN,SCALE,1TF,NG,INSAVE
COM

RETURN
END(1,1,0,0)

*TYPE*(FORTRAN)

C "OUTPUT DESCRIPTION 3"

SUBROUTINE OUT3

COMMON NG,IPM,NP,IA,INL,NS,VM,JS,VP,JB,NN,SCALE,1TF,NG,INSAVE
COM

RETURN
END(1,1,0,0)
71

DIMENSION(W,3),X(1),X(2),X(3),V(3),V2(3)
CALLFUN(W,Y,Z)
1 IFNG1=25,111,125
111 CALLF16VFC(W,Y,V1)
1 IFNG1=25,115,125
115 CALLATOM(IN(5),X)
CALLATOM(IN(7),X)
1 IFNG1=25,121,125
121 CALLD16V(X1,V1,V2)
COS SVV1, V2)
126 RETURN
END(IN,1,0,0)
*TYPE(FORTRAN)
C

71
END(1,J,5,0,0)

*TYPE(FORTRAN) FEE 1677

C PRINCIPAL AXES PROJECTED ON VECTOR. SUBROUTINE FUN9 FEE 1678

COMMON NG, TMAX, NP, IAM, NS, NV, JXP, JX, JBP, JB, NM, SCALE, ITF, NO, INSAVE FEE 1679
COMMON DMAX, NA, F, FX, FLZ, KK, KD, SAVEA, SAVEP, VARA, VARP, E1, KH, INSP FEE 1680
COMMON TITLE, PM, DP, DFDP, AMP, DA, DFDA, IN, TS, 5, SA, TP, BB, ROW, KI, KI2 FEE 1681
DIMENSION TITL1(12), PM(200), DP(200), DFDP(200), AMP(21), DA(46) FEE 1682
DIMENSION DFDA(61, IN, 1231), TS(3, 5, 48), IS(12, 3, 48), A(6), AA(3, 3), BB(3, 3) FEE 1683
DIMENSION P(300), ROW(61), KI(300), KI2(300) FEE 1684
CALLFUNC(C, Z) FEE 1685
F*E*57 FEE 1686
RETURN FEE 1687
END(0, 1, 0, 0, 0) FEE 1688

*TYPE(FORTRAN) FEE 1689

C HEADER 1 FEE 1690

SUBROUTINE HED1 FEE 1691
RETURN FEE 1692
END(0, 1, 0, 0, 0) FEE 1693

WRITEOUTPUTPGE, 107 FEE 1694

107 FORMAT(80*HANG BETWEEN PRINCIPAL AXES R AND AXIS I OF CARTESIAN SYSTEM DEFINED BY TWO VECTORS/450A ATOM R I DEFINED BY TWO VECTORS/450A ATOM R I FEE 1695
PRINT NTH VECTORS) FEE 1696
RETURN FEE 1697
END(0, 1, 0, 0, 0) FEE 1698

*TYPE(FORTRAN) FEE 1699

C PRELIMINARY SUBROUTINE 10 SUBROUTINE PRE0 FEE 1700

COMMON NG, TMAX, NP, IAM, NS, NV, JXP, JX, JBP, JB, NM, SCALE, ITF, NO, INSAVE FEE 1701
COMMON DMAX, NA, F, FX, FLZ, KK, KD, SAVEA, SAVEP, VARA, VARP, E1, KH, INSP FEE 1702
COMMON TITLE, PM, DP, DFDP, AMP, DA, DFDA, IN, TS, 5, SA, TP, BB, ROW, KI, KI2 FEE 1703
DIMENSION TITL1(12), PM(200), DP(200), DFDP(200), AMP(21), DA(46) FEE 1704
DIMENSION DFDA(61, IN, 1231), TS(3, 5, 48), IS(12, 3, 48), A(6), AA(3, 3), BB(3, 3) FEE 1705
DIMENSION P(300), ROW(61), KI(300), KI2(300) FEE 1706
CALLKSET(IN(112)) FEE 1707
D0 109 IN(16), 12, 2 FEE 1708
109 CALLKSETX(IN(151)) FEE 1709
RETURN FEE 1710
END(0, 1, 0, 0, 0) FEE 1711

*TYPE(FORTRAN) FEE 1712

C ANGLE BETWEEN PRINCIPAL AND CARTESIAN AXES SUBROUTINE FUN10 FEE 1713

COMMON NG, TMAX, NP, IAM, NS, NV, JXP, JX, JBP, JB, NM, SCALE, ITF, NO, INSAVE FEE 1714
COMMON DMAX, NA, F, FX, FLZ, KK, KD, SAVEA, SAVEP, VARA, VARP, E1, KH, INSP FEE 1715
COMMON TITLE, PM, DP, DFDP, AMP, DA, DFDA, IN, TS, 5, SA, TP, BB, ROW, KI, KI2 FEE 1716
DIMENSION TITL1(12), PM(200), DP(200), DFDP(200), AMP(21), DA(46) FEE 1717
DIMENSION DFDA(61, IN, 1231), TS(3, 5, 48), IS(12, 3, 48), A(6), AA(3, 3), BB(3, 3) FEE 1718
DIMENSION P(300), ROW(61), KI(300), KI2(300) FEE 1719
CALLFUNC(C, Z) FEE 1720
RETURN FEE 1721
END(0, 1, 0, 0, 0) FEE 1722

*TYPE(FORTRAN) FEE 1723

C OUTPUT DESCRIPTION ID SUBROUTINE OUT0 FEE 1724

COMMON NG, TMAX, NP, IAM, NS, NV, JXP, JX, JBP, JB, NM, SCALE, ITF, NO, INSAVE FEE 1725
COMMON DMAX, NA, F, FX, FLZ, KK, KD, SAVEA, SAVEP, VARA, VARP, E1, KH, INSP FEE 1726
COMMON TITLE, PM, DP, DFDP, AMP, DA, DFDA, IN, TS, 5, SA, TP, BB, ROW, KI, KI2 FEE 1727
DIMENSION TITL1(12), PM(200), DP(200), DFDP(200), AMP(21), DA(46) FEE 1728
DIMENSION DFDA(61, IN, 1231), TS(3, 5, 48), IS(12, 3, 48), A(6), AA(3, 3), BB(3, 3) FEE 1729
DIMENSION P(300), ROW(61), KI(300), KI2(300) FEE 1730
WRITEOUTPUTPGE, 107, (IN(1)), (12, 14, 12, 14) FEE 1731
107 FORMAT(12*HANG BETWEEN PRINCIPAL AXES R AND AXIS I OF CARTESIAN SYSTEM DEFINED BY TWO VECTORS/450A ATOM R I DEFINED BY TWO VECTORS/450A ATOM R I FEE 1732
PRINT VECTORS) FEE 1733
RETURN FEE 1734
END(0, 1, 0, 0, 0) FEE 1735

*TYPE(FORTRAN) FEE 1736

C HEADER 11 SUBROUTINE HED11 FEE 1737
RETURN FEE 1738
END(0, 1, 0, 0, 0) FEE 1739

WRITEOUTPUTPGE, 107 FEE 1740

107 FORMAT(10*HANG COMPONENT OF THERMAL DISPLACEMENT ALONG PRINCIPAL AXIS) FEE 1741
IF(11, 12, 13, 20)
120 NG#10
GOTO 20
201 CALL MP(R, AA, BA)
CALL MP(AA, BB, AA, BA, A, BB)
X1=50, VMX(V, BAA, V1, 1, 59, 0, 0, 0, 0, 0, 18, Y)
207 RETURN
END(3, 1, 0, 0, 0)
*TYPE(FORTRAN)
C HEADING 12
SUBROUTINE HFD12
WRITE OUTPUT TAPES, 107
WRITE OUTPUT TAPES, 107
107 FORMAT (12HO)
COMPONENT OF THERMAL DISPLACEMENT IN DIRECTION DEF.
ENED BY TWO ATOMS, ANGSTROMS/4°K
ATOMIC FACTOR
RETURN
END(1, 0, 0, 0)
*TYPE(FORTRAN)
C PELIMINARY SUBROUTINE 12
SUBROUTINE HFD12
COMMON NG, IPM, NP, IAM, NS, NV, JXP, JX, JBP, JB, NM, SCALE, ITF, NG, SAVE
COMMON DMAK, NA, F, FX, LNZ, KK, KKD, SAVE, S PP, VARAP, E, E, IH, NSP
COMMON TITLE, PM, DP, DFD, AM, DA, DFD, IN, TS, IS, AA, BB, PP, ROW, K11, K12
DIMENSION DFD(6), IN(2), T5(3), 48, 3, 48, A(1), AA(3), BB(3)
DINDIMENSION P(300), ROW(4), K11, K12(400)
CALL SETK(IN(2))
DO 191 IN(4), 4, 19
191 CALLSET(IN(1))
RETURN
END(3, 1, 0, 0, 0)
*TYPE(FORTRAN)
C RMS DISPLACEMENT IN GIVEN DIRECTION
SUBROUTINE HFD12
COMMON NG, IPM, NP, IAM, NS, NV, JXP, JX, JBP, JB, NM, SCALE, ITF, NG, SAVE
COMMON DMAK, NA, F, FX, LNZ, KK, KKD, SAVE, S PP, VARAP, E, E, IH, NSP
COMMON TITLE, PM, DP, DFD, AM, DA, DFD, IN, TS, IS, AA, BB, PP, ROW, K11, K12
DIMENSION DFD(6), IN(2), T5(3), 48, 3, 48, A(1), AA(3), BB(3)
DIMENSION P(300), ROW(4), K11, K12(400)
CALL FUMI(IN(2), IN(4), JX)
RETURN
END(3, 1, 0, 0, 0)
*TYPE(FORTRAN)
C OUTPUT DESCRIPTION 12
SUBROUTINE HFD12
COMMON NG, IPM, NP, IAM, NS, NV, JXP, JX, JBP, JB, NM, SCALE, ITF, NG, SAVE
COMMON DMAK, NA, F, FX, LNZ, KK, KKD, SAVE, S PP, VARAP, E, E, IH, NSP
COMMON TITLE, PM, DP, DFD, AM, DA, DFD, IN, TS, IS, AA, BB, PP, ROW, K11, K12
DIMENSION DFD(6), IN(2), T5(3), 48, 3, 48, A(1), AA(3), BB(3)
DIMENSION P(300), ROW(4), K11, K12(400)
WRITE OUTPUT TAPES, 107
WRITE OUTPUT TAPES, 107
107 FORMAT (12HO)
(12, 1H, 13, 12H)
(12, 1H, 13, 12H)
(1H, 11)
RETURN
END(3, 1, 0, 0, 0)
*TYPE(FORTRAN)
C RMS RADIAL DISPLACEMENT
SUBROUTINE HFD12
COMMON NG, IPM, NP, IAM, NS, NV, JXP, JX, JBP, JB, NM, SCALE, ITF, NG, SAVE
COMMON DMAK, NA, F, FX, LNZ, KK, KKD, SAVE, S PP, VARAP, E, E, IH, NSP
RETURN
END(3, 1, 0, 0, 0)
*TYPE(FORTRAN)
C RMS RADIAL DISPLACEMENT
SUBROUTINE HFD12
COMMON NG, IPM, NP, IAM, NS, NV, JXP, JX, JBP, JB, NM, SCALE, ITF, NG, SAVE
COMMON DMAK, NA, F, FX, LNZ, KK, KKD, SAVE, S PP, VARAP, E, E, IH, NSP
RETURN
END(3, 1, 0, 0, 0)
SUBROUTINES FOR EXAMPLES

*TYPE( FORTRAN) C EXAMPLE 1 AND 2. HYPOTHETICAL PROBLEM BASED ON ALPHA QUARTZ. SUBROUTINE SETA(A) DIMENSION A(6) A(2) #A(1) RETURN END(0.10.0.0) *TYPE( FORTRAN) C EXAMPLE 1. HYPOTHETICAL PROBLEM BASED ON ALPHA QUARTZ. SUBROUTINE SETP(P) DIMENSION P(100) P(5) #P(4) RETURN END(0.10.0.0) *TYPE( FORTRAN) C EXAMPLE 2. HYPOTHETICAL PROBLEM BASED ON ALPHA QUARTZ. SUBROUTINE SETP(P) DIMENSION P(100) P(17) #P(16) P(20) #P(19) P(26) #P(23) RETURN END(0.10.0.0)

DATA FOR EXAMPLES

*DATA EXAMPLE 1. HYPOTHETICAL PROBLEM BASED ON ALPHA QUARTZ. USES CARD INPUT.

<table>
<thead>
<tr>
<th>I</th>
<th>1</th>
<th>6</th>
<th>1</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>4155727</td>
<td>2676554</td>
<td>1179998</td>
<td>5294765</td>
<td>5294765</td>
<td>3333333</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85384856-7</td>
<td>73775640-7</td>
<td>20955735-7</td>
<td>68440656-8</td>
<td>13675298-6</td>
<td>57040983-7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11775529-7</td>
<td>97987015-7</td>
<td>20795807-7</td>
<td>33345397-7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.913</td>
<td>n</td>
<td>5.404</td>
<td>n</td>
<td>n</td>
<td>-0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>n</td>
<td>010</td>
<td>n</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-2</td>
<td>-2</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-1</td>
<td>-1</td>
<td>33333333+3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2-1</td>
<td>33333333-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**DATA**

**EXAMPLE 2. HYPOTHETICAL PROBLEM BASED ON ALPHA QUARTZ. USES OR FLAT TAPE.**

```

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.913</td>
<td>0.005</td>
<td>-0.2</td>
<td>0.904</td>
<td>-0.5</td>
</tr>
<tr>
<td>2</td>
<td>-2</td>
<td>-0.3</td>
<td>-2</td>
<td>-0.03333333333</td>
<td>-2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

1   2300 1 2
1   2300 1300
2 1 2 2300 1 3
2 1300 2300 1 3
2 1300 2300 1 2
3 2300 1 2 1 3 2300 1300 1305
4 2300 1 2 2300 1300
5 1 2 2300 1 3 1300 2300 1305
```

**TYPE (BINARY)**

```

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
</table>
| 1 | 2300 1 2
| 1 | 2300 1300
| 2 | 1 2 2300 1 3
| 2 | 1300 2300 1 3
| 2 | 1300 2300 1 2
| 3 | 2300 1 2 1 3 2300 1300 1305
| 4 | 2300 1 2 2300 1300
| 5 | 1 2 2300 1 3 1300 2300 1305
```

---

```

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
</table>
| 1 | 2300 1 2
| 1 | 2300 1300
| 2 | 1 2 2300 1 3
| 2 | 1300 2300 1 3
| 2 | 1300 2300 1 2
| 3 | 2300 1 2 1 3 2300 1300 1305
| 4 | 2300 1 2 2300 1300
| 5 | 1 2 2300 1 3 1300 2300 1305
```
EXAMPLE 1. HYPOTHETICAL PROBLEM BASED ON ALPHA QUARTZ. USES CARD INPUT.

INPUT DATA TO BE READ FROM CARDS

NUMBER OF STRUCTURE PARAMETERS IS 6

VARIANCE-COVARIANCE MATRIX AND PARAMETER SELECTION INFORMATION WILL BE USED

NUMBER OF SYMMETRY CARDS IS 5

CELL PARAMETER ERRORS ARE TO BE READ IN THE FORM OF STANDARD ERRORS

ORDER OF THE VARIANCE-COVARIANCE MATRIX FOR THE STRUCTURE PARAMETERS IS 4

PERIOD OF THE POSITION PARAMETERS IN THE PARAMETER LIST IS 3

POSITION OF THE FIRST X COORDINATE IN THE PARAMETER LIST IS 1

INPUT DATA

<table>
<thead>
<tr>
<th>P(I)</th>
<th>KII(I)</th>
</tr>
</thead>
</table>
| 1    | 0.4156
| 2    | 0.2677
| 3    | 0.1180
| 4    | 0.5295
| 5    | 0.5295
| 6    | 0.3333

EXAMPLE 1. HYPOTHETICAL PROBLEM BASED ON ALPHA QUARTZ. USES CARD INPUT.

CELL PARAMETERS

4.9130 0. 5.4040 0. 0. -0.5000

STANDARD ERRORS, RESPECTIVELY, OF THE ABOVE CELL PARAMETERS

0.0050 0. 0.0100 0. 0. 0.

SYMMETRY INFORMATION

<table>
<thead>
<tr>
<th>TRANSFORMED X</th>
<th>TRANSFORMED Y</th>
<th>TRANSFORMED Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.</td>
<td>1-2</td>
<td>-0.</td>
</tr>
<tr>
<td>-0.</td>
<td>2-1</td>
<td>-0.</td>
</tr>
<tr>
<td>-0.</td>
<td>1-0</td>
<td>-0.</td>
</tr>
<tr>
<td>-0.</td>
<td>2-0</td>
<td>-0.</td>
</tr>
<tr>
<td>-0.</td>
<td>1-2</td>
<td>0.333333 3-0</td>
</tr>
<tr>
<td>-0.</td>
<td>2-1</td>
<td>0.333333 3-0</td>
</tr>
<tr>
<td>-0.</td>
<td>1-0</td>
<td>0.333333 3-0</td>
</tr>
<tr>
<td>-0.</td>
<td>2-0</td>
<td>0.666667 3-0</td>
</tr>
<tr>
<td>-0.</td>
<td>1-2</td>
<td>0.666667 3-0</td>
</tr>
</tbody>
</table>

EXAMPLE 1. HYPOTHETICAL PROBLEM BASED ON ALPHA QUARTZ. USES CARD INPUT.

INTERATOMIC DISTANCE IN ANGSTROMS

(2,300) (1, 2)

1.6012 + OR - 0.0019 (0.0012)

(2,300) (1,300)

1.6131 + OR - 0.0021 (0.0012)
BOND ANGLE IN DEGREES. CENTRAL ATOM IS VERTEX

\[
\begin{align*}
(1, 2) & (2,300) (1, 3) & 109.2048 \pm 0.1013 (0.0975) \\
(1,300) & (2,300) (1,305) & 110.1962 \pm 0.1689 (0.1443) \\
(1,300) & (2,300) (1, 2) & 109.9518 \pm 0.0875 (0.0755)
\end{align*}
\]

DIHEDRAL ANGLE BETWEEN PLANES EACH DEFINED BY THREE ATOMS

\[
\begin{align*}
(2,300) & (1, 2) (1, 3) & 89.1600 \pm 0.1128 (0.0478) \\
(2,300) & (1,300) (1,305)
\end{align*}
\]

DIFFERENCE BETWEEN TWO INTERATOMIC DISTANCES

\[
\begin{align*}
(2,300) & (1, 2) & -0.0119 \pm 0.026 (0.0022) \\
(2,300) & (1,300)
\end{align*}
\]

DIFFERENCE BETWEEN TWO BOND ANGLES

\[
\begin{align*}
(1, 2) & (2,300) (1, 3) & 0.9914 \pm 0.1787 (0.1681) \\
(1,300) & (2,300) (1,305)
\end{align*}
\]

EXAMPLE 2. HYPOTHETICAL PROBLEM BASED ON ALPHA QUARTZ. USES OR FLS TAPE.

INPUT DATA TO BE READ FROM OR FLS TAPE

VARIANCE-COVARIANCE MATRIX AND PARAMETER SELECTION INFORMATION WILL BE USED

NUMBER OF SYMMETRY CARDS IS 5

CELL PARAMETER ERRORS ARE TO BE READ IN THE FORM OF STANDARD ERRORS

NUMBER OF STRUCTURE PARAMETERS IS 24

PERIOD OF THE POSITION PARAMETERS IN THE PARAMETER LIST IS 11

POSITION OF THE FIRST X COORDINATE IN THE PARAMETER LIST IS 5

PERIOD OF THE TEMPERATURE FACTOR COEFFICIENTS IN THE PARAMETER LIST IS 11

POSITION OF THE FIRST TEMPERATURE FACTOR COEFFICIENT IN THE PARAMETER LIST IS 8

ORDER OF THE VARIANCE-COVARIANCE MATRIX FOR THE STRUCTURE PARAMETERS IS 15

INPUT DATA

\[
\begin{align*}
1 & P(1) \ K(1) \\
1 & 1.0004 \\
2 & 0.00 \\
3 & 1.0000 \\
4 & 1.0000 \\
5 & 0.4156 \\
6 & 0.2677 \\
7 & 0.1780 \\
8 & 0.0039 \\
9 & 0.0045 \\
10 & 0.0027
\end{align*}
\]
EXAMPLE 2. HYPOTHETICAL PROBLEM BASED ON ALPHA QUARTZ. USES OR FLS TAPE.

CELL PARAMETERS

4.9130  0.  5.4040  0.  0.  -0.5000

STANDARD ERRORS, RESPECTIVELY, OF THE ABOVE CELL PARAMETERS

0.0050  0.  0.0100  0.  0.  0.

SYMMETRY INFORMATION

TRANSFORMED X  TRANSFORMED Y  TRANSFORMED Z

-0.  1-2  -0.  -2-0  -0.  -3-0
-0.  2-1  -0.  -1-0  0.  33333  3-0
-0.  1-0  -0.  2-1  0.  33333  3-0
-0.  -2-0  -0.  1-2  0.666667  3-0
-0.  2-0  -0.  1-0  0.666667  3-0

EXAMPLE 2. HYPOTHETICAL PROBLEM BASED ON ALPHA QUARTZ. USES OR FLS TAPE.

INTERATOMIC DISTANCE IN ANGSTROMS

(2,300) (1, 2)  1.6012 +OR- 0.0019 (0.0012)

(2,300) (1,300)  1.6131 +OR- 0.0021 (0.0012)

BOND ANGLE IN DEGREES. CENTRAL ATOM IS VERTEX

(1, 2) (2,300) (1, 3)  109.2048 +OR- 0.1014 (0.0977)

(1,300) (2,300) (1,305)  110.1962 +OR- 0.1688 (0.1441)

(1,300) (2,300) (1, 2)  109.9518 +OR- 0.0875 (0.0756)

DIHEDRAL ANGLE BETWEEN PLANES EACH DEFINED BY THREE ATOMS

(2,300) (1, 2) (1, 3)  89.1600 +OR- 0.1128 (0.0477)

(2,300) (1,305) (1,305)

DIFFERENCE BETWEEN TWO INTERATOMIC DISTANCES

(2,300) (1, 2)  -0.0119 +OR- 0.0026 (0.0022)
### Difference Between Two Bond Angles

- \((1, 2) (2,300) (1, 3)\)  
- \((1,300) (2,300) (1,305)\)  

\[0.9914 \pm 0.1784 (0.1679)\]

### RMS Component of Thermal Displacement Along Principal Axis \(R\). Angstroms

<table>
<thead>
<tr>
<th>Atom</th>
<th>(R)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>((1, 0))</td>
<td>1</td>
<td>0.0600 \pm 0.0073 (0.0073)</td>
</tr>
<tr>
<td>((1, 0))</td>
<td>2</td>
<td>0.0619 \pm 0.0050 (0.0050)</td>
</tr>
<tr>
<td>((1, 0))</td>
<td>3</td>
<td>0.0668 \pm 0.0068 (0.0068)</td>
</tr>
<tr>
<td>((1, 0))</td>
<td>1</td>
<td>0.0743 \pm 0.0033 (0.0033)</td>
</tr>
<tr>
<td>((1, 0))</td>
<td>2</td>
<td>0.0785 \pm 0.0033 (0.0033)</td>
</tr>
<tr>
<td>((1, 0))</td>
<td>3</td>
<td>0.0798 \pm 0.0035 (0.0035)</td>
</tr>
</tbody>
</table>

### Angle Between Principal Axis \(R\) and Vector Defined By Two Atoms

<table>
<thead>
<tr>
<th>Atom</th>
<th>(R)</th>
<th>(\text{Vector} (1, 2))</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>((1, 2))</td>
<td>1</td>
<td>((2,300) (1, 2))</td>
<td>76.6802 \pm 0.4312 (56.4311)</td>
</tr>
<tr>
<td>((1, 2))</td>
<td>2</td>
<td>((2,300) (1, 2))</td>
<td>79.0386 \pm 0.5388 (57.5340)</td>
</tr>
<tr>
<td>((1, 2))</td>
<td>3</td>
<td>((2,300) (1, 2))</td>
<td>17.3908 \pm 39.1984 (39.1957)</td>
</tr>
</tbody>
</table>

### Angle Between Principal Axis \(R\) and Axis 1 of Cartesian System Defined By Two Vectors

<table>
<thead>
<tr>
<th>Atom</th>
<th>(R)</th>
<th>(I)</th>
<th>(\text{Defining Vectors} (1, 2))</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>((1, 2))</td>
<td>1</td>
<td>1</td>
<td>((2,300) (1, 2)) ((2,300) (1, 3))</td>
<td>76.6802 \pm 0.4312 (56.4311)</td>
</tr>
<tr>
<td>((1, 2))</td>
<td>2</td>
<td>1</td>
<td>((2,300) (1, 2)) ((2,300) (1, 3))</td>
<td>77.3633 \pm 51.2667 (51.2667)</td>
</tr>
<tr>
<td>((1, 2))</td>
<td>3</td>
<td>1</td>
<td>((2,300) (1, 2)) ((2,300) (1, 3))</td>
<td>161.4757 \pm 80.2633 (80.2633)</td>
</tr>
<tr>
<td>((1, 2))</td>
<td>2</td>
<td>2</td>
<td>((2,300) (1, 2)) ((2,300) (1, 3))</td>
<td>79.0386 \pm 57.5388 (57.5340)</td>
</tr>
<tr>
<td>((1, 2))</td>
<td>2</td>
<td>3</td>
<td>((2,300) (1, 2)) ((2,300) (1, 3))</td>
<td>164.9587 \pm 32.7963 (32.7957)</td>
</tr>
<tr>
<td>((1, 2))</td>
<td>3</td>
<td>3</td>
<td>((2,300) (1, 2)) ((2,300) (1, 3))</td>
<td>100.1726 \pm 48.7792 (48.7790)</td>
</tr>
<tr>
<td>Atom Vector</td>
<td>RMS Component of Thermal Displacement</td>
<td>RMS Radial Thermal Displacement</td>
<td>Interatomic Distance Averaged Over Thermal Motion</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------</td>
<td>-------------------------------</td>
<td>-----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>(1, 2)</td>
<td>((2, 300) (1, 2))</td>
<td>((2, 300) (1, 2))</td>
<td>((2, 300) (1, 2))</td>
<td></td>
</tr>
<tr>
<td>(1, 2)</td>
<td></td>
<td>0.0662 ± 0.0067 ((0.0067))</td>
<td>0.1090 ± 0.0036 ((0.0036))</td>
<td></td>
</tr>
<tr>
<td>(2, 300)</td>
<td></td>
<td>0.0791 ± 0.0022 ((0.0022))</td>
<td>0.1543 ± 0.0016 ((0.0016))</td>
<td></td>
</tr>
</tbody>
</table>

\(\text{RMS Component of Thermal Displacement in Direction Defined by Two Atoms, Angstroms}\)

\(\text{RMS Radial Thermal Displacement of Atom, Angstroms}\)

\(\text{Interatomic Distance Averaged Over Thermal Motion, Atoms Assumed to Move Independently}\)
DISTRIBUTION

1-3. W. R. Busing
4-6. K. O. Martin
7-9. H. A. Levy
10-11. Central Research Library
12. Document Reference Section
13-27. Division of Technical Information Extension
28. Research and Development Division
29. Laboratory Records - Record Copy
30-300. Laboratory Records