A System of Fortran IV Computer Programs for Crystal Structure Computations
The National Bureau of Standards was established by an act of Congress March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau consists of the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Institute for Computer Sciences and Technology, and the Office for Information Programs.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of a Center for Radiation Research, an Office of Measurement Services and the following divisions:


THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement, standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; and develops, produces, and distributes standard reference materials. The Institute consists of the Office of Standard Reference Materials and the following divisions:


THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations leading to the development of technological standards (including mandatory safety standards), codes and methods of test; and provides technical advice and services to Government agencies upon request. The Institute consists of a Center for Building Technology and the following divisions and offices:


THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in improving cost effectiveness in the conduct of their programs through the selection, acquisition, and effective utilization of automatic data processing equipment; and serves as the principal focus within the executive branch for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Institute consists of the following divisions:

Computer Services — Systems and Software — Computer Systems Engineering — Information Technology.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal Government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System; provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:


---

1 Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.
2 Part of the Center for Radiation Research.
3 Located at Boulder, Colorado 80302.
4 Part of the Center for Building Technology.
A System of Fortran IV Computer Programs for Crystal Structure Computations

Larry W. Finger

Geophysical Laboratory
Carnegie Institution of Washington
Washington, D.C. 20008

and

E. Prince

Institute for Materials Research
National Bureau of Standards
Washington, D.C. 20234

U.S. DEPARTMENT OF COMMERCE, Frederick B. Dent, Secretary
NATIONAL BUREAU OF STANDARDS, Richard W. Roberts, Director

Issued February 1975
ABSTRACT

This report gives detailed descriptions and instructions for use of a system of programs for crystallographic calculations, including least-squares refinement with generalized systems of constraints, calculation of bond distances and angles with errors, Fourier synthesis, plotting of contours in Fourier maps, and preparation of structure factor tables for publication.

Key Words:
Computer programs; contour plotting; constrained refinement; crystallographic calculations; Fourier section; Fourier synthesis; least squares

This is also Paper Number 1651 from the Geophysical Laboratory, Carnegie Institution of Washington, Washington, D. C.
TABLE OF CONTENTS

RFINE4: A program for least squares refinement including general constraints.  
General Description  1  
File and Record Structure  1  
Description of Routines  2  
Common Blocks and Variables  11  
Dependent Parameter Input  14  
Details of Data Input  17  

BONDAN: A program for computing bond distances, angles, and thermal ellipsoids, with errors.  27  

FOURIER  29  
LISTFC  32  
CNTPLT  34  
ARBSECT  37  

References Cited  40  

APPENDIX A - Overlay scheme for RFINE  41  

APPENDIX B - BADTEA - alternate input for bond distances and angles, and thermal ellipsoids  42  

APPENDIX C - Listings of programs.  56  

These programs, although tested on many problems and several computing systems, may not give correct results in all circumstances. Therefore, neither the authors nor their institutions will guarantee the correctness of the program in all respects, but the authors will greatly appreciate being informed of discrepancies in the code or desirable options not currently included.
RFINE4: A program for least squares refinement including general constraints

General Description

This program, called RFINE4, performs the calculations necessary to refine crystal structures using the techniques of full-matrix least-squares refinement. In addition, it can compute bond distances and angles or thermal ellipsoids so that these functions may be monitored throughout the various stages of a refinement. File control capabilities have been designed so that parameters may be saved at the end of a given computer run or, alternatively, virtually any parameter may be changed at almost any time.

A scheme of derivative modification and parameter restoration which eliminates the need for coding special subroutines to treat constrained parameters has been included in this program. This method will handle thermal and positional restrictions for atoms in positions of special symmetry and, in conjunction with the occupancy refinement capability, a multiposition, partially ordered structure may be constrained to a particular chemical composition. Provision has been made for incorporating parameters which are not members of the set used in conventional least-squares refinement. Any set of linear constraints may be specified, and also groups of atoms may be treated as rigid bodies both with respect to position and orientation (shape constraints) and with respect to thermal motion.

In its present form, the program allows a maximum of 60 atoms, 20 chemical species, 10 scale factors, 1320 parameter dependencies and 200 refined parameters for a structure in any space group. With the maximum number of parameters and no use of overlays the program requires approximately 65000 words of core storage. However more restricted problems can be operated in smaller core sizes, and the program can be further reduced in size by means of overlays. A suggested overlay structure is given in Appendix A. Some operating systems allow for dynamic allocation of core. Such provisions can be used to advantage.

File and Record Structure

In the following discussion a record is the unit of information formally defined as a logical record, i.e. the amount of information transferred in a single non-formatted read or write statement. This description is independent of any physical device which may be associated with the file and the program has been coded so as to operate fairly efficiently with any type of auxiliary storage currently available. Furthermore, a data set shall mean the collection of records needed to describe a structure. Any reference to file or data labels should not be confused with file control information for a particular computer system.
Master File. The file referred to as ISCl and assigned to logical unit 10 is the main file of the program. This file consists of a data set containing all the information required to resume least-squares refinement. This data set has two label records, one or more structure factor records, a constraint relations record, a variance-covariance record, and a record containing variances of previously refined parameters (useful in large problems where the refinement must be "blocked"). The label records contain a 12-character alphanumeric identification tag, a cycle number and all the atom parameters, symmetry data and scattering factor information needed for this structure. The reflection data consist of the Miller indices, the observed structure factor on a relative scale, the standard deviation of \( F_{\text{obs}} \), the value of \( \sin^2 \theta / \lambda^2 \) for the reflection, an integer specifying which scale factor applies to this reflection, \( \beta \)--the extinction parameter, a flag to mark reflections which had an intensity less than the minimum observable, the diffractometer angles \( \chi, \phi \) and \( \omega + \theta \) in radians, and the value of each of the scattering curves for this reflection. These quantities are packed into a buffer, starting with word 2, until another reflection would require the buffer to be larger than 1023 words. An end of reflection records marker, which is non-zero only in the last block, is contained in word 1. The information for a single reflection is never split between two records and all records, including the last which may be only partially filled, are the same length. The end of the reflection list is marked by a Miller index \( h \) having a value of 3000. If the number of scattering curves is \( N \), the number of reflections per record \( n \) may be found from the integral part of \( 1022/(N+12) \). The length of the record is then 1023 minus 1022 modulo \( n \). The variance-covariance record contains the variance-covariance matrix for the previous cycle of refinement in packed, upper triangular format.

Intermediate File. This file, named ISC2 and assigned to logical unit 29, contains the structure factor records for the current structure. A distinct file allows the scattering factor information to be changed without reading the structure factors from cards.

Auxiliary Files. A file named ISC3 and assigned to logical unit 11 is used as an alternative structure factor input. A file named ISC4 and assigned to logical unit 12 is used to pass information to Fourier synthesis, structure factor tabulation and statistical analysis programs. A file named ISC5 and assigned to logical unit 28 is used for temporary storage of constraint relationships.

System Files. This program uses a file named IN, logical unit 5, as a card read file, a file named IOUT, logical unit 6, as the print file, and a file named IPUN, logical unit 1, as the punch file. These assignments may be changed for a particular system by a block data subprogram.

Description of Routines

Main Program. The main program serves as a driver for the least-squares section. It reads one card which contains a 12-character label.
If a new data set is being constructed, this label will be inserted into that file. Control is then transferred to subroutine RFINE. If the updated information is written back to file ISCI, the subroutine exits normally. In that event, the main program prints a message stating that the file has been updated.

Subroutine RFINE. This routine is the primary one for the structure factor and least-squares calculation and is responsible for controlling the functions computed and the parameters used.

When this routine is first entered, initialization is accomplished and the main cycle loop is entered. The first step is to read the title and control cards for the cycle. If this is the first cycle of the current run, the new parameter indicators from the control card are interrogated and if all are non-zero, signifying that all parameters are to be loaded from cards, the data set on file ISCI is presumed not to exist; however, if any of the indicators are zero, the first two records of that data set are loaded into memory. Next, any new parameters required are read from cards. As new parameters are read, the values are printed.

If the unit cell parameters are to be changed, that card is read and the direct space metric tensor is prepared. This is a 3 x 3 symmetric matrix whose elements are the scalar products of the unit cell vectors. This matrix is then inverted using subroutine SYMINV described below. The inverse matrix is then the reciprocal space metric tensor and the determinant of the original matrix is the square of the unit cell volume. Note: If the unit cell parameters are to be changed, the structure factors must also be changed if the effects of the cell data are to be applied to the values of sin θ/λ.

New symmetry cards, if required, are now read. Each card lists an equivalent position in very nearly the same form as Vol. I of the International Tables with only one major difference; positions related by a center of inversion or lattice centering must not be included since they are internally generated. Using this scheme, all space groups may be generated by using a maximum of 24 cards. Each symmetry data card is transformed into matrices R and T satisfying the equation

\[ X' = RX + T \]

with \( X' \) equal to the vector of transformed coordinates, \( X \) is the position vector with components \( xyz \), \( R \) is a 3 x 3 rotational matrix and \( T \) is a vector describing the translation associated with this symmetry operation.

At this point in the program, new scattering factor coefficients are read, if desired. We must emphasize two points of caution concerning the changing of the scattering factors. First, since the blocking of structure factors depends upon the number of scattering factors used, if that number is changed, new structure factors must be read. Violation of this rule will lead to unpredictable, but probably catastrophic, results. The second rule requires that the scattering factors be changed
The new atom parameters, if required, are now read from cards. The scale factors are read first, followed by the extinction parameters and the atom cards. The atom parameters needed are an identification label of 6 characters, an equipoint fraction which is the multiplicity of the position occupied by this atom divided by the multiplicity of an atom in the general position of this space group, the total occupancy of the site (usually one), the occupancy of species A (needed if more than one species present in this site), the identification of scattering factor(s) to be applied to this atom, fractional coordinates for the atom, and temperature factor information, as well as third and fourth cumulant parameters, consistent with the temperature factor type card described below. The forms of the cumulant expressions used in this program are:

\[
F = \sum_{j} \exp \left[ i \sum_{k,l,m,n} h_{k} \beta_{k} h_{l} \beta_{l} \right]
\]

\[
= \sum_{j} \exp \left[ i \sum_{k,l,m,n} h_{k} \beta_{k} h_{l} \beta_{l} \right] \sum_{k,l,m,n} j \beta_{j} k h_{m} h_{n} \beta_{m} \beta_{n} \]

atoms

\[ x_{j}^{k} \] is the \( k \text{th} \) fractional coordinate of atom \( j \), \( \beta \) is the array of anisotropic thermal coefficients, and \( c \) and \( d \) are the arrays of third and fourth cumulant coefficients. After the atom parameter cards have been read, the special parameter cards are read. The first card gives the total number of special parameters (up to 60). Then one card for each special parameter contains its name, three integers specifying its type, and its initial value.

At this point in the program, the parameter refinement selection information is read. A parameter of the model is varied in the least-squares refinement if its associated indicator is one; parameters with zero indicators are changed only if they are linked to refined parameters by dependency cards. If new parameter selection information is read, constraint data are also read. If no dependency relationships exist for this structure, a blank card must be included in the data deck. The method used to treat parameter constraints is from Finger (1969) and essentially consists of the application of the chain rule to the derivatives of the structure factor with respect to the parameters of the model. The user must determine the dependent parameters of the model using the rules of Levy (1956) for thermal coefficients or the equations of Finger (1969) for chemical constraints and then prepare the data cards. The program currently will not allow the constraining of the total occupancy for a site with only one species present; however, this may be accomplished by including a null scattering curve which is made the second species present. The amount of the first species may then be refined and constrained.
At this point, all structural data except the observed structure factors have been read. If these are to be read, a data card containing a FORMAT specification is read. Two optional cards may be read, one to specify the wavelength if it is needed for the computation of extinction corrections and the other to specify an "ignorance factor", ρ, to be used in computing-weights according to the formula

\[ w = 1/[σ^2 + (ρF_{obs})^2]. \]

Subsequently, the subroutine INPUT is called to read and process a reflection card. The data are then added to a buffer which, if full, is output on file ISC2. If the old structure factors are desired and this is the first cycle of a given computer run, the buffer is filled from file ISC1, the scattering factors are recomputed if necessary and the buffer is written on file ISC2. If this is not the first cycle of a given run, the old structure factors are read from file ISC2. After the data for a reflection have been obtained, the calculated structure factor and partial derivatives are computed in subroutine SFAC, the results tested to determine if this reflection should be used in the normal equations matrix, and the contributions of this reflection to the residuals are calculated. Next, the structure factors are printed, if desired, with the rejected data marked with a single asterisk in the right margin. In addition, the reflections having intensities less than the minimum observable value are marked with a single asterisk next to the value of F_{obs} and a double asterisk in the right margin.

If the reflection is to be included in the matrices, subroutine MATRIX is entered; then the next observation is processed. As soon as the end of the reflection list is encountered, the residuals for this cycle are printed. Next, the normal equations matrices are converted to correlation matrices to prevent overflow or underflow of the determinant during the matrix inversion, which is accomplished by subroutine SYMINV, adapted from Busing and Levy (1962). If the matrix is non-singular, the inverse is converted to a variance-covariance matrix and the parameter shifts are computed. Subroutine RESET is entered to compute the new values of any dependent parameters. The ratios of the parameter shifts to the standard deviations are tested to find the largest absolute value and the magnitudes of the ratios are averaged. At this point in the program, the correlation matrix, if requested, is printed. The twenty off-diagonal elements of the correlation matrix with largest absolute values are always printed. Then the optional bond functions are computed by routine BODANI and the thermal functions by routine ELVIBI.

Finally, a run continuation card is read. If column 1 is non-zero, another cycle is to be performed. However, if this column is blank or zero, the program prepares to terminate. If the new values for the parameters are not to be saved, the program exits. On the other hand, if these parameters are to be kept, a new label record is written on ISC1, the structure factor records are copied from ISC2 to ISC1 and the variance-covariance record is output. The current values of all parameters may be punched on cards. This routine then returns control to the main program.
**Subroutine BODAN1.** This subroutine uses the symmetry operators supplied on cards combined with the action of centers of symmetry, if present, and the effects of multiple lattice points and cell translations to generate all atomic locations within a spherical shell centered on each atom in the asymmetric unit. The search for the contents of this shell is done efficiently by enclosing the central atom with a unit cell shaped parallelopiped tangent to the sphere. For each location inside this box, a table of generated positions is checked to determine if this location is unique and if it is, the bond distance is computed. If this is within the desired range, the distance and coordinates are printed and this location is added to the table of generated positions.

If bond angles are also desired, each atom in the asymmetric unit is used as the central atom with the atom positions generated in the bond distance calculation considered. The peripheral atoms are identified only by their alphanumeric tag and a number which points to an entry in the bond distance output where the coordinates of the generated atom are listed. Caution—if the maximum bond distance is large, very many angles will be generated.

**Subroutine ELVIB1.** For each anisotropic atom, the equivalent isotropic temperature factor of Hamilton (1959), the root-mean-square (rms) amplitudes of vibration (in Å) of the principal axes and the orientations of these axes with respect to the direct cell axes are computed, or if the temperature factor matrix of an atom is nonpositive-definite, this is reported. This routine will also compute the thermal corrections to bonds using the formulae of Busing and Levy (1964). To assist the interpretation of the orientations of thermal ellipsoids relative to the bonding, the rms amplitude of the ellipsoid in the direction of the bond is computed and printed. BODAN1 is used to generate the necessary atomic locations.

**Subroutine INPUT.** This subroutine reads a reflection data card, computes \( \sin^2 \theta / \lambda^2 \), evaluates the scattering factors for this observation and computes \( \sigma_F \) for the observation. Maximum and minimum values of \( F_{obs} \) and \( \sin^2 \theta / \lambda^2 \), are determined in this routine.

**Subroutine MATRIX.** This subroutine calculates the derivatives of the residual from the derivatives of the calculated structure factor and accumulates the sums required in the normal equations matrices.

**Subroutine MODIFY.** This routine applies the equations

\[
\frac{\partial A}{\partial p_n} = \Sigma \frac{\partial A}{\partial p_m} \frac{\partial p_m}{\partial p_n}
\]

and

\[
\frac{\partial B}{\partial p_n} = \Sigma \frac{\partial B}{\partial p_m} \frac{\partial p_m}{\partial p_n}
\]

to the calculated derivatives. \( A \) and \( B \) are the real and imaginary portions of the calculated structure factor, \( p_n \) is the independent parameter, \( p_m \)
represents a dependent parameter, the summation is over all the dependencies, the unprimed derivative denotes the one calculated without regard for dependencies and the primed derivative is needed to form the least-squares matrices.

**Subroutine RCALC.** This subroutine accumulates the terms needed to calculate the discrepancy factors which are computed for all reflections and separately for those data used in the refinement. A weighted residual, \( R_w \), which is computed using the formula

\[
R_w^2 = \frac{\Sigma w (|F_o| - |F_c|)^2}{\Sigma w |F_o|^2}
\]

for refinement on \( F \) or

\[
R_w^2 = \frac{\Sigma w (|F_o|^2 - |F_c|^2)^2}{\Sigma w |F_o|^4}
\]

for refinement on \( F^2 \), is determined for each class of data. In addition, the conventional discrepancy index

\[
R = \frac{\Sigma |F_o| - |F_c|}{\Sigma |F_o|}
\]

is computed for all reflections in each class. In addition, the ranges of variation of \( F \) and \( \sin^2 \theta/\lambda^2 \) are each divided into eight subranges and the sums needed for \( R \) are accumulated. (Note: the maximum and minimum values are dynamically determined during any cycle in which new structure factors are read and the distribution of residuals for such a cycle will be meaningless.) If the weighting scheme is correct, the expression \((1/s) \Sigma w (F_o - F_c)^2\), where the sum is taken over a subset of the full data set containing \( s \) reflections, should be independent of the composition of the subset. If we divide this expression by the constant \((1/t) \Sigma w F_o^2\), where \( t \) is the total number of reflections in the data set and the sum is taken over the entire set, and take the square root of the quotient, the result should also be independent of the composition of the subset. This number, which can be considered the average contribution to weighted \( R \) for reflections in the subset (Note that the expression is equal to weighted \( R \) if \( s = t \)) is printed in the output for each subset.

**Subroutine RESET.** This subroutine is entered once for special parameters and once for each atom to compute the updated values of all dependent parameters. Linear relationships and rigid-body parameters are computed directly. Other non-linear systems of constraints may be handled by the user through subroutine SPVAL, supplied as a dummy.

**Subroutine SFAC.** This subroutine computes the structure factor and partial derivatives with the symmetry operators applied to reciprocal space; i.e., the Miller indices are transformed rather than the atomic coordinates. If \( H' \) represents the transpose of the vector of Miller indices and \( X' \),
represents the \( j \)th symmetry transformed coordinate for the \( n \)th atom, then the argument of the trigonometric functions needed to calculate the contribution of this atom to the structure factor may be written as \( H^T X'_n \).

Utilizing the expression for \( X' \) from page 3,

\[
H^T X' = H^T (R X + T) = H^T R X + H^T T.
\]

The final form is the method used in this subroutine.

As the routine is entered for each reflection, the matrix products \( H^T R_j = H^T_j \) and \( H^T T_j \) are computed and saved. Then the structure factors and partial derivatives are accumulated. If 3rd and 4th cumulant terms are absent, the structure factor

\[
F = \sum_n a f_n \sum_j T_{nj} P_{nj}
\]

with

\[
P_{nj} = \exp [2\pi i (H^T X_n + H^T T_j)]
\]

and

\[
T_{nj} = \exp (-H^T_j \beta_n H'_j)
\]

for anisotropic atoms or

\[
T_{nj} = \exp (-B_n \sin^2 \theta / \lambda^2)
\]

for isotropic atoms, the \( n \) summation is over the atoms in the asymmetric unit, the \( j \) summation is over the symmetry operators, \( a \) is an occupancy term, \( f \) is the value of the scattering curve for the \( n \)th atom, \( i \) is \( \sqrt{-1} \) and \( \beta \) is the matrix of anisotropic thermal coefficients. When anomalous dispersion is included, the scattering factor becomes

\[
f_n = f_0^n + f'_n + i f''_n
\]

with \( f_0^n \) the wave length independent part, \( f'_n \) the real part of the anomalous dispersion and \( f''_n \) the imaginary part. Including this expression in the structure factor equation, applying Euler's rule and writing \( F \) in the form

\[
F = A + i B,
\]

we find that

\[
A = \sum_n \left[ a_n \left( f_0^n + f'_n \right) \sum_j T_{nj} \cos 2\pi (H^T_j X_n + H^T T_j) \right. \\
- \left. a_n f''_n \sum_j T_{nj} \sin 2\pi (H^T_j X_n + H^T T_j) \right]
\]
\[
B = \sum \left[ a_n \left( f_n + f'_n \right) \sin 2\pi(H^T_j X_n + H^T_j) + a_n f''_n \sum \cos 2\pi \left( H^T_j X_n + H^T_j \right) \right].
\]

The required partial derivatives are found by differentiating each of the above equations with respect to the components of \( X_n \) and \( \beta_j \). After all sums have been accumulated, the subroutine MODIFY is called to perform any chain rule operations required. If this is to function properly, all derivatives must be calculated although the refinement indicators would suggest that certain derivatives are not needed.

Subroutine SFAC also includes an entry point named EXCALC which calculates the extinction factor of Zachariasen (1967) as extended by Coppens and Hamilton (1970), and revised by Nelmes and Thornley (1973). Using this scheme, the structure factor is corrected for extinction by the relationship

\[
F^*_c = \frac{F_c y^{1/2}}{(1 + \beta(\theta) F^2_c r^*)^{-1/2}}
\]

\[
y = \frac{\lambda^2}{V^2 A \sin 2\theta} \frac{\mathrm{d}A}{\mu}
\]

for x-rays

\[
\beta(\theta) = \frac{-2 e^4 2}{m c^2 v^2 A \sin 2\theta (1 + \cos^2 2\theta)} \frac{\mathrm{d}A}{\mu}
\]

for neutrons

where \( \lambda \) is the wavelength, \( V \) is the cell volume, \( A \) is the transmission factor, \( \mu \) is the linear absorption coefficient, and \( e, m \) and \( c \) have their usual meanings. The program allows three different forms for the extinction: a) isotropic with \( r^* \) refined, b) Type I anisotropic with \( r^* = (D^2 Z D)^{-1/2} \) with \( z_{ij} \) refined and c) Type II anisotropic with \( r^* = (N W N)^{-1/2} \) with \( W_{ij} \) refined. If \( p \) is a unit vector parallel to the incident beam and \( s \) is a unit vector parallel to the diffracted beam, \( D \) is a unit vector parallel to \( p \times s \) and \( \bar{N} = \bar{p} \times \bar{D} \). The reference coordinate system for \( Z \) and \( W \) is attached to the crystal and has its \( x \)-axis parallel to the incident beam and its \( z \)-axis vertical when all diffractometer angles are 0.

Subroutine SYMINV. The method used by this matrix inversion routine has been described by Busing and Levy (1962) and was coded by Mr. James Mundstock of the Numerical Analysis Center of the University of Minnesota.

Subroutine REJECT. This routine may be used to exclude reflections from the normal equations matrices for other than the standard reasons. If a reflection is to be rejected, the variable NREJ in Common Block /F/ should be set to one.
**Subroutine WEIGHT.** This routine may be used to compute a non-standard weighting scheme. Upon entry, the information contained in the standard deviation field of the structure factor card will be in the variable WT in Common Block /F/. This variable must be set to the desired weight information when the routine is exited.

**Subroutine RDEGN.** This subroutine reads the dependent parameter information from cards, and rearranges the array of relationships in ascending order first of independent atom and parameter designators and then of dependent atom and parameter designators. Special parameters are arbitrarily assigned atom number 0.

**Subroutine DERIVS.** This subroutine is entered once at the beginning of each cycle. It computes the current values of derivatives of conventional parameters with respect to rigid body parameters. (Other non-linear derivatives may be supplied by the user through subroutine SPDERI). If any independent parameters are themselves actually dependent on other parameters (such as when the values of the elements of the rigid-body motion tensors are restricted by symmetry) the derivatives with respect to the true independent parameters are computed. Then the independent parameter designators are replaced by their row indices in the normal equations matrix, and the array is packed to eliminate any redundant relations.

**Subroutine PAREXT.** This subroutine reads the special parameter cards, and prints the names and initial values of the parameters.

**Subroutine TRNFRM.** This subroutine is entered once at the beginning of a run to compute the transformation matrices required to convert parameters from the crystal system to a standard orthonormal coordinate system and back again.

**Subroutine FILLIN.** This subroutine stores special parameter values in the proper positions in rigid-body parameter arrays.

**Function ERRCAL.** This function subroutine determines the standard deviation of a dependent parameter according to the approximate formula

\[ \sigma_k = \left( \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial P_i}{\partial P_j} \frac{\partial P_k}{\partial P_j} \sigma_{ij} \right)^{1/2}, \]

where \( n \) is the number of refined parameters and \( \sigma_{ij} \) is the covariance of the \( i \)-th refined parameter with the \( j \)-th refined parameter.

**Subroutine EULER.** This subroutine computes an orthogonal matrix which is necessary to transform parameters from a general orthonormal coordinate system to the standard orthonormal coordinate system defined by \( x \) parallel to \( a \) and \( z \) parallel to \( c^* \). The relative orientations are defined by Eulerian angles \( \phi, \chi, \) and \( \omega \). Here \( \omega \) is the angle through which the special coordinate system must be rotated clockwise, as viewed down the positive \( z \) axis of the standard system, about the \( z \) axis of
the standard system to bring the z axis of the special system into the
x-z plane of the standard system; \( \chi \) is a clockwise rotation about the y
axis of the standard system (viewed down the positive y axis) to bring
the z axes of the two systems into coincidence; \( \phi \) is a clockwise rotation
about the common z axis to bring the x and y axes into coincidence.
Three additional entry points, DTDPHI, DTDCHI, and DTDOMG compute the
derivatives of the elements of the transformation with respect to \( \phi \), \( \chi \),
and \( \omega \) respectively.

**Subroutine MULMM.** This subroutine forms the product of two 3 x 3
matrices. An alternate entry, MULMV, multiplies a three component
column vector by a 3 x 3 matrix.

**Subroutine TFM.** This subroutine has two entry points. TFM2 trans-
forms a second cumulant tensor in an orthonormal system to a set of aniso-
tropic temperature factors in the crystal system. TFM3 does the same for
a third cumulant tensor.

**Subroutine TRNSPS.** This subroutine generates the transpose of a
3 x 3 matrix.

**Function IPAK.** This function packs 4 integer variables into a
single computer word. The version given is appropriate only to the UNIVAC
1108. Other versions must be used for other machines. If the integer
word length of the machine is shorter than 32 bits other adjustments
must be made.

**Functions SKTLS, D2KDT, D2KDL, and D2KDS.** These functions compute
the second cumulants for an atom and their derivatives with respect to the
rigid body motion parameters \( T \), \( L \), and \( S \).

**Functions TKTLS, D3KDL, and D3KDS.** compute the third cumulants
and their derivatives with respect to \( L \) and \( S \).

**Subroutine CORECT.** This subroutine applies libration corrections to
the atomic positions in a rigid body.

**Function DETERM and subroutine SOLVE** are used by CORECT. Subroutine
VXV forms the vector product of two vectors. **Functions AIJ, BIJK, CIJKL,
DIJKLM, EIJ, FJI, AXA, AXAXB, PIJKLM, EL4.** are a set of functions used
in the rigid body relations.

**Common Blocks and Variables**

1. Common Block /LSMAT/

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B(200)</td>
<td>Normal equations vector</td>
</tr>
<tr>
<td>IEXT</td>
<td>Extinction type indicator</td>
</tr>
<tr>
<td>IPARA (1000)</td>
<td>Atom Parameter refinement indicators</td>
</tr>
<tr>
<td>MRANK</td>
<td>Order of the normal equations matrices</td>
</tr>
<tr>
<td>KCYL</td>
<td>Cycle number for the data set</td>
</tr>
<tr>
<td>LABEL (3)</td>
<td>12 character label for the data set</td>
</tr>
</tbody>
</table>
JSCRF (10) Refinement switches for scale factors
NRSF Number of scale factors refined
SIGMA Standard deviation of a parameter
MDEGN Twice the number of constraint relationships
A(22740) Constraints array and normal equations matrix or inverse. The dimensions are a maximum. The actual number for a given problem is given by the expression MDEGN+[(MRANK*(MRANK+1))/2]. If, as on the UNIVAC 1108, the operating system allows dynamic core allocation, this number can be determined at run time.

2. Common Block /A/: 

BETA(6,60) Temperature factor coefficients
G(3,3) Real space metric tensor
ISOT(60) Temperature factor type indicators
NATOM Number of atoms in the asymmetric unit
TAG(2,60) Alphanumeric atom tags or labels
TITLE(18) Alphanumeric title information
XYZ(3,60) Atomic positions
PI Value of π
CR(10,60) Third cumulant coefficients
DR(15,60) Fourth cumulant coefficients

3. Common Block /C/: 

CELLP(3,4) Coordinates of lattice points
ICENT Centric indicator
JCELL Number of lattice points
NSYM Number of symmetry operators
RMAT(3,3,24) Rotational symmetry operators
TRANS(3,24) Translational symmetry operators

4. Common Block /D/: 

ACALC Real part of calculated structure factor
AIND(60) Individual atom contributions to ACALC
ANOM(2,20) Anomalous dispersion corrections
BCALC Imaginary part of calculated structure factor
BIND(60) Individual atom contributions to BCALC
DADB(6,60) Derivatives of ACALC with respect to temperature factors
DADO(60) Derivatives of ACALC with respect to occupancy
DADX(3,60) Derivatives of ACALC with respect to position
DBDB(6,60) Derivatives of BCALC with respect to temperature factor
DBDO(60) Derivatives of BCALC with respect to occupancy
DBDX(3,60) Derivatives of BCALC with respect to position
H(3)  Miller indices in real representation
IH(3)  Miller indices in integer representation
ISCAT(2,60)  Table used to select atomic scattering factor
ISCL  Number of scale factor for this reflection
MFSEL  Function selector
OCCA(60)  Occupancy of species A
OCCUP(60)  Total occupancy of site
SCALE(10)  Scale factors
SCAT(20)  Values of scattering factor for current reflection
SITE(60)  Equipoint fraction for atom site
DADC(10,60)  Derivatives of ACALC with respect to 3rd cumulant
DBDC(10,60)  Derivatives of BCALC with respect to 3rd cumulant
DADD(15,60)  Derivatives of ACALC with respect to 4th cumulant
DBDD(15,60)  Derivatives of BCALC with respect to 4th cumulant
AMINUS  ACALC for opposite absolute orientation
BMINUS  BCALC for opposite absolute orientation
DRDE(6)  Derivatives of r* (extinction factor) with respect to coefficients of tensor
NDEGN  Number of constraint relationships

5. Common Block /E/:

DMAX  Maximum bond distance
DMIN  Minimum bond distance
RECELL(6)  Reciprocal cell lengths and cosines
EXANG(3)  \( \phi, \chi \) and \( \Theta + \omega \) (in radians) for current reflection

6. Common Block /F/:

DELTA  Weighted residual for current reflection
FCALC  Calculated structure factor
FOBS  Observed structure factor
FOMAX  Maximum observed structure factor
FOMIN  Minimum observed structure factor
IREJ  Rejection mode indicator
MREF  Refinement mode indicator
MREJ  Minimum observable indicator
NREJ  Used to indicate whether or not a reflection is rejected
NUMBER(36)  Counters for subdivisions of R factors
RDEN(36)  Denominators for R factors
RHO  Current value of \( \sin^2 \theta / \lambda^2 \)
RHOMN  Minimum value of RHO
RHOMX  Maximum value of RHO
RNUM(36)  Numerators for R factors
WT  Weight for current reflection  
EXBETA  Factor \( \beta(\theta) \) for extinction correction  
EXX  Factor \( \beta(\theta)F^2r^* \) used in extinction  
EXY  \((1 + EXX)^{-1/2}\)  
FISGN  Absolute orientation parameter  
IFSGN  Orientation determination indicator  
EXTIN(6)  Extinction parameter matrix  
KEXTRF(60)  Refinement switches for extinction parameters  
DELMAX  Maximum permitted value of \( \Delta F/\sigma_F \)  

7. Common Block /G/:  
BUFF(1023)  Array used primarily for blocking structure factor records  
FCURVE(9,20)  Scattering curve coefficients  
FRMT(18)  Variable format specification for reading structure factors  
IWT  Weight calculation type indicator  
NSCAT  Number of scattering factors  
BETFAC  Used in extinction correction  
WVLNTH  Wavelength of radiation  
WTFAC  "Ignorance factor" in computation of weights  

8. Common Block /TAPE/:  
IN  Logical unit number of the card read unit (5)  
INN  Logical unit number of the structure factor input unit (5 or 11)  
IOUT  Logical unit number of the print file (6)  
ISC1  Logical unit number of the least-squares file (10)  
ISC2  Logical unit number of the intermediate file (29)  
ISC3  Logical unit number of the auxiliary file (11)  
ISC4  Logical unit number of the Fourier file (12)  
ISC5  Logical unit number for saving constraints list (28)  
IPUN  Logical unit number for punch file (1)  

Dependent Parameter Input

In certain structures, the parameters of the model may not be independently variable. If they are merely constrained to have a fixed value, there are no difficulties as the refinement selection variable for this parameter is set to zero. However, if a parameter of the model is required to be equal to a linear combination of variable quantities, the situation is greatly complicated. In the least-squares refinement of such a structure, the contribution of the dependent parameter to the derivative of the structure factor with respect to the independent variable must be computed if the shifts are to be properly evaluated.
Similarly, the correct value of the dependent parameter must be restored at the end of the cycle if the structure factors are to be properly computed in the next cycle. This program utilizes a new algorithm to perform the derivative modification and parameter restoration. The method of preparing the input data cards will be shown with the aid of examples.

Example 1: Atom 6 is an anisotropic atom on a 3-fold axis parallel to \( \mathbf{c} \). Employing the rules of Levy (1956), the following relationships are required for this atom:

\[
\begin{align*}
\beta_{13} &= \beta_{23} = 0 \\
\beta_{11} &= \beta_{22} = 2 \beta_{12}
\end{align*}
\]

The first equations relate parameters to a constant and thus may be ignored. However, this is not the case for the second, and since \( \beta_{11} \) must be the independent variable, the following equations are applied as constraints to the least-squares solutions:

\[
\begin{align*}
\beta_{22} &= \beta_{11} \\
\beta_{12} &= \frac{1}{2} \beta_{11}
\end{align*}
\]

The resulting Parameter Constraint Cards (See page 24) for this situation would be:

<table>
<thead>
<tr>
<th>Dependent</th>
<th>Independent</th>
<th>Constant</th>
<th>Comments (not on card)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6</td>
<td>6 5</td>
<td>( \beta_{22} = \beta_{11} ) for atom 6</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>6 5</td>
<td>( \beta_{12} = 0.5 \beta_{11} )</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>End of list</td>
</tr>
</tbody>
</table>

Example 2: Grunerite has four independent sites (atoms 1 to 4) for Fe and Mg, but the total amount of each is known from the chemical analysis. Therefore, we want to refine the values of the Mg-occupancies for the first three atoms and constrain the fourth to agree with the bulk chemistry. In the notation of Finger (1969), the constraining equation is

\[
\sum_{m} b_{m} a_{mn} = C_{n}
\]

with \( b_{m} \) the multiplicity of the \( m \)th site, \( a_{mn} \) the fractional occupancy for the \( n \)th species in the \( m \)th site, and \( C_{n} \) the total number of atoms of species \( n \) per unit cell. In grunerite, sites 1 through 4 have multiplicities 4, 4, 2, and 4 respectively and there are two formula units per unit cell. If we constrain the formula to be \( \text{Fe}_{6.2} \text{Mg}_{0.8} \text{Si}_{8} \text{O}_{22} \text{(OH)}_{2} \) and substitute into the above equation, we find that
Solving for $a_{4\text{Mg}}$,

$$a_{4\text{Mg}} = 0.4 - a_{1\text{Mg}} - a_{2\text{Mg}} - \frac{1}{2} a_{3\text{Mg}}.$$  

The Parameter Dependency Cards needed to constrain this chemistry would be:

<table>
<thead>
<tr>
<th>Dependent</th>
<th>Independent</th>
<th>Constant</th>
<th>Comments (not on card)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 1</td>
<td>0 0</td>
<td>0.40</td>
<td>Initialization</td>
</tr>
<tr>
<td>4 1</td>
<td>1 1</td>
<td>-1.0</td>
<td>Site 1 dependence</td>
</tr>
<tr>
<td>4 1</td>
<td>2 1</td>
<td>-1.0</td>
<td>Site 2</td>
</tr>
<tr>
<td>4 1</td>
<td>3 1</td>
<td>-0.5</td>
<td>Site 3</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>End of list</td>
</tr>
</tbody>
</table>

Alternatively, the latest version allows up to 5 independent parameters to affect one dependent one on the same card, so the first four cards above may be replaced by the card as follows:

$$4 \; 1 \; 0 \; 0 \; 0.40 \; 1 \; 1 \; -1.0 \; 2 \; 1 \; -1.0 \; 3 \; 1 \; -0.5$$

Note that when occupancies are constrained, the dependent value must first be initialized with a card containing zeros in the independent parameter fields.

**Example 3:** Like many chemical groups, phosphate groups often have thermal vibration that can be represented by the motions of a rigid body, which can be expressed as the elements of the $T$, $L$, and $S$ tensors introduced by Schomaker and Trueblood (1969). Expressions for the second and third cumulants of the probability distribution functions for individual atoms (Johnson, 1969) as functions of the elements of $T$ and $L$ are given by Prince and Finger (1973), and the corresponding expressions including $S$ can be readily derived (Prince, unpublished but given as a FORTRAN expression in this program). If atom 3 is an oxygen atom of a phosphate group and special parameters 1 through 24 are 3 coordinates of the origin to which the motions are referred, 6 elements of $T$, 6 elements of $L$, and 9 elements of $S$, the constraint cards will be set up as follows:

First, because of the so called "trace of $S$ singularity", the three diagonal elements of $S$ are not independent, and $S_{33}$ can be set equal to $-S_{11} - S_{22}$ by means of the card 0 18 0 16 -1. 0 17 -1.

The anisotropic temperature factors may depend on all of the rigid body elements, so there must be six cards of the form

$$
\begin{align*}
3 & \; 5 \; 0 \; 4 \; 0. \; 0-24 \; 0. \\
3 & \; 6 \; 0 \; 4 \; 0. \; 0-24 \; 0. \\
3 & \; 7 \; 0 \; 4 \; 0. \; 0-24 \; 0. \\
3 & \; 8 \; 0 \; 4 \; 0. \; 0-24 \; 0. \\
3 & \; 9 \; 0 \; 4 \; 0. \; 0-24 \; 0. \\
3 & \; 10 \; 0 \; 4 \; 0. \; 0-24 \; 0. 
\end{align*}
$$
Here the minus sign before the 24 indicates that the dependent parameter depends on special parameters 4 through 24 inclusive. The coefficients are given as zeros because, being non-linear, they change from cycle to cycle and are therefore computed in each cycle according to the current values of the refined parameters. Third cumulants are independent of $T$, but may be dependent on any element of $L$ or $S$. Therefore a typical constraint card for a third cumulant would have the form

$$3 \ 12 \ 0 \ 10 \ -0. \ 0-24 \ 0.$$  

The program uses the information on the constraint cards to allocate space in any array and then, in non-linear cases, fills in the space at the beginning of each cycle. Therefore if a given relationship is not listed it will never be used whether or not its value is actually zero. Because of symmetry some relationships, particularly those involving $T$, may be identically zero. It will save space and almost certainly time to omit these from the list.

### Details of Data Input

1. **File Name Card:** Columns 1-12 contain an alphanumeric label to be written into a new data set. This label will be printed when the file is read.

2. **Title Card:** Any desired Hollerith information in columns 1-72. This information appears at the top of most pages of the output.

3. **Control Card:**

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | MFSEL    | Function selector  
|        |          | 0 – Structure factors only  
|        |          | 1 – Least-squares refinement  
| 2      | MSF      | Structure factor output selector  
|        |          | 0 – No structure factor output  
|        |          | 1 – Printed structure factors  
|        |          | 2 – Printed and written (for a Fourier program) on file ISC4  
|        |          | 3 – Written on file ISC4  
| 3      | MCORR    | Correlation matrix*  
|        |          | 0 – Not printed  
|        |          | 1 – Printed  
| 4      | MBODAN   | Bond distance, angle selector  
|        |          | 0 – None computed  
|        |          | 1 – Distances only  
|        |          | 2 – Distances and angles computed  
| 5      | MELVIB   | Ellipsoids of vibration  
|        |          | 0 – None computed  

*The values of the 20 largest off-diagonal correlation coefficients will be printed for every least-squares cycle.*
3. Control Card (Cont.):

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1 - Ellipsoids computed for anisotropic atoms</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2 - Ellipsoids and thermal corrections to bonds computed</td>
</tr>
<tr>
<td>6</td>
<td>MREF</td>
<td>Refinement type indicator</td>
</tr>
<tr>
<td>7</td>
<td>IFILE</td>
<td>Rewrite data set indicator</td>
</tr>
<tr>
<td>8</td>
<td>NEW(1)</td>
<td>If non-zero, read new cell parameters*</td>
</tr>
<tr>
<td>9</td>
<td>NEW(2)</td>
<td>If non-zero, read new bond distance limits</td>
</tr>
<tr>
<td>10</td>
<td>NEW(3)</td>
<td>If non-zero, read new symmetry operators</td>
</tr>
<tr>
<td>11</td>
<td>NEW(4)</td>
<td>If non-zero, read new scattering factors†</td>
</tr>
<tr>
<td>12</td>
<td>NEW(5)</td>
<td>If non-zero, read new scale factors and atom parameters. If equal to one, all atoms will be read. If equal to two, only selected atoms will be read. If equal to 3 only scale and extinction will be read. If equal to 4 only special parameters will be read.</td>
</tr>
<tr>
<td>13</td>
<td>NEW(6)</td>
<td>If non-zero, read temperature factor type card</td>
</tr>
<tr>
<td>14</td>
<td>NEW(7)</td>
<td>If non-zero, read parameter selection card(s) and parameter dependency card(s)</td>
</tr>
<tr>
<td>15</td>
<td>NEW(8)</td>
<td>If non-zero, read new structure factors. Values 1, 2, and 3 correspond to various options for the order of the list, as explained below.</td>
</tr>
</tbody>
</table>

For all zero values in columns 8-15, the old values are read from file ISC1 or the values in memory are used. The above integers are required for all cycles.

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-20</td>
<td>DMIN</td>
<td>Minimum bond distance in Angstroms. Needed only if NEW(2) ≠ 0.</td>
</tr>
<tr>
<td>21-25</td>
<td>DMAX</td>
<td>Maximum bond distance in Angstroms. Needed only if NEW(2) ≠ 0.</td>
</tr>
</tbody>
</table>

*Note that if the new cell parameters will affect $\sin \theta / \lambda$, and therefore the atomic scattering factors for various reflections, structure factors must be read in.

†Warning: New scattering factor curves must be read only in the first cycle of a given computer run or when new structure factors are being read in, or they will never be used in computation.
### 3. Control Card (Cont.):

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>ICENT</td>
<td>Centric indicator, needed only if NEW(3) ≠ 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Centric</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Acentric</td>
</tr>
<tr>
<td>27</td>
<td>ICELL</td>
<td>Cell type indicator, needed only if NEW(3) ≠ 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Primitive cell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - A-centered cell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - B-centered cell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - C-centered cell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 - I-centered cell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 - F-centered cell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6 - R-centered cell</td>
</tr>
<tr>
<td>28</td>
<td>MANOD</td>
<td>Anomalous dispersion indicator, needed only if NEW(4) ≠ 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - No corrections are to be used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Include anomalous dispersion corrections</td>
</tr>
<tr>
<td>29-30</td>
<td>NSCAT</td>
<td>Number of scattering curves. This is needed only if NEW(4) ≠ 0. (0 &lt; NSCAT &lt; 21)</td>
</tr>
<tr>
<td>31-32</td>
<td>NATOM</td>
<td>Number of atoms in asymmetric unit. This is needed only if NEW(5) ≠ 0. (0 &lt; NATOM &lt; 61)</td>
</tr>
<tr>
<td>33-34</td>
<td>NSF</td>
<td>Number of scale factors. This is needed only if NEW(5) ≠ 0. (0 &lt; NSF &lt; 11)</td>
</tr>
<tr>
<td>35</td>
<td>NEWEXT</td>
<td>Extinction change indicator (needed only if NEW(5) ≠ 0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - New extinction parameters not read</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Extinction parameters read from cards</td>
</tr>
<tr>
<td>36</td>
<td>IWT</td>
<td>Weight type, needed only if NEW(8) ≠ 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Standard deviation of F on cards</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Standard deviation of F^2 on cards</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - Weight on cards</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - Sigma for F is to be computed in WEIGHT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 - Compute weight in subroutine WEIGHT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 - Standard deviation of F on cards will be altered by &quot;ignorance factor&quot; by</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[ \sigma = (\sigma_F^2 + (pF)^2)^{1/2} ]</td>
</tr>
<tr>
<td>37</td>
<td>INN</td>
<td>Structure factor source indicator, needed if NEW(8) ≠ 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Structure factors on cards</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - New structure factors from file ISC3</td>
</tr>
<tr>
<td>38</td>
<td>IREJ</td>
<td>Rejection parameter, needed only if NEW(8) ≠ 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0,5 - No reflections rejected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,6 - Reject if I less than the minimum obs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,7 - Reject if I less than the minimum obs. or (</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3,8 - Reject if (</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In both of these cases weights will be altered by [ w' = w(1-3x^4+2x^5) ], where [ x =</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4,9 - Use special routine REJECT</td>
</tr>
</tbody>
</table>
3. Control Card (Cont.):

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>39-43</td>
<td>DELMAX</td>
<td>Maximum allowable difference between observed and calculated structure factors on an absolute scale. This is needed only if NEW(8) ≠ 0 and IREJ is 2 or 3, or if IREJ is 7 or 8.</td>
</tr>
<tr>
<td>44-53</td>
<td>JSCRF</td>
<td>Scale factor refinement switches (needed only if NEW(7) ≠ 0)</td>
</tr>
<tr>
<td>54-59</td>
<td>KEXTRF</td>
<td>Extinction factor refinement switches (needed only if NEW(7) ≠ 0)</td>
</tr>
<tr>
<td>60</td>
<td>JXPAR</td>
<td>Extra parameter indicator&lt;br&gt;0 - No extra parameters&lt;br&gt;1 - Extra parameters will be read (needed only if NEW(5) ≠ 0)</td>
</tr>
<tr>
<td>61</td>
<td>IFSGN</td>
<td>Orientation determination parameter&lt;br&gt;0 - Do not determine the orientation&lt;br&gt;1 - Change to orientation with lower weighted r (needed only if NEW(7) ≠ 0)</td>
</tr>
</tbody>
</table>

NOTE: Values of IREJ from 5 to 9 allow the rejection mode to be changed even though NEW(8) = 0.

4. Cell Card: Needed only if NEW(1) ≠ 0.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-8</td>
<td>A(1)</td>
<td>a axis length in angstroms</td>
</tr>
<tr>
<td>9-16</td>
<td>A(2)</td>
<td>b axis length</td>
</tr>
<tr>
<td>17-24</td>
<td>A(3)</td>
<td>c axis length</td>
</tr>
<tr>
<td>25-32</td>
<td>A(4)</td>
<td>Alpha in degrees</td>
</tr>
<tr>
<td>33-40</td>
<td>A(5)</td>
<td>Beta in degrees</td>
</tr>
<tr>
<td>41-48</td>
<td>A(6)</td>
<td>Gamma in degrees</td>
</tr>
</tbody>
</table>

5. Symmetry Cards: Needed only if NEW(3) ≠ 0.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>JTNS(1)</td>
<td>Translational part of x'</td>
</tr>
<tr>
<td>5-6</td>
<td>IRMAT(1,1)</td>
<td>First positional part of x'</td>
</tr>
<tr>
<td>8-9</td>
<td>IRMAT(2,1)</td>
<td>Second positional part of x'</td>
</tr>
<tr>
<td>11-13</td>
<td>JTNS(2)</td>
<td>Translational part of y'</td>
</tr>
<tr>
<td>15-16</td>
<td>IRMAT(1,2)</td>
<td>First positional part of y'</td>
</tr>
<tr>
<td>18-19</td>
<td>IRMAT(2,2)</td>
<td>Second positional part of y'</td>
</tr>
<tr>
<td>21-23</td>
<td>JTNS(3)</td>
<td>Translational part of z'</td>
</tr>
<tr>
<td>25-26</td>
<td>IRMAT(1,3)</td>
<td>First positional part of z'</td>
</tr>
<tr>
<td>28-29</td>
<td>IRMAT(2,3)</td>
<td>Second positional part of z'</td>
</tr>
<tr>
<td>34</td>
<td>IEF</td>
<td>End flag, non-zero on last card only&lt;br&gt;0 - Another symmetry card follows&lt;br&gt;1 - This is the last card</td>
</tr>
</tbody>
</table>
The translational operators are of the following form: 
\[ \begin{align*} 
\text{bbb} & \quad 1/2 \quad 1/3 \\
\text{1/4} & \quad 3/4 \quad 1/6 \quad 5/6. 
\end{align*} \]
The positional operators are of the form \( bW = W \) or \(-W\) with \( W \) equal to \( X \), \( Y \), or \( Z \) and \( b \) signifies a blank. The symmetry position \( X \) \( Y \) \( Z \) must be included.

6. Scattering Factor Cards: Needed only if NEW(4) \( \neq 0 \).

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-8</td>
<td>( a_1 )</td>
</tr>
<tr>
<td>9-16</td>
<td>( b_1 )</td>
</tr>
<tr>
<td>17-24</td>
<td>( a_2 )</td>
</tr>
<tr>
<td>25-32</td>
<td>( b_2 )</td>
</tr>
<tr>
<td>33-40</td>
<td>( a_3 )</td>
</tr>
<tr>
<td>41-48</td>
<td>( b_3 )</td>
</tr>
<tr>
<td>49-56</td>
<td>( a_4 )</td>
</tr>
<tr>
<td>57-64</td>
<td>( b_4 )</td>
</tr>
<tr>
<td>65-72</td>
<td>( c )</td>
</tr>
<tr>
<td>73-80</td>
<td></td>
</tr>
</tbody>
</table>

Coefficients for scattering curves of this form have been tabulated by Cromer and Waber (1965) for relativistic Dirac-Slater wave functions, by Cromer and Mann (1968) for Hartree-Fock wave functions and by Doyle and Turner (1968) for relativistic Hartree-Fock wave functions. If anomalous dispersion coefficients are being used (\( \text{MANOD} = 1 \)), the above card is followed by one with the real part of the anomalous dispersion in col. 1-8 and the imaginary part in col. 9-16. Coefficients for the hydrogen atom are given by Forsyth and Wells (1959).

NOTE: The scattering curves should be changed only on the first cycle of a given computer run or when NEW(8) = 1. New structure factors must be read (NEW(8) = 1) if the number of scattering curves is to be changed.

7. Scale Factor Cards: Needed only if NEW(5) \( \neq 0 \).

The NSF scale factors are punched with 8 columns per value and 9 values per card and should be the values needed to place the calculated structure factors on the same scale as the observed.

8. Extinction Cards: Needed only if NEW(5) \( \neq 0 \) and NEWEXT \( \neq 0 \). Format (6E12.6)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-12</td>
<td>( r^* ) for isotropic extinction or ( r^* ) for anisotropic</td>
</tr>
<tr>
<td>13-24</td>
<td>( r^*22 )</td>
</tr>
<tr>
<td>25-36</td>
<td>( r^*33 )</td>
</tr>
<tr>
<td>37-48</td>
<td>( r^*12 )</td>
</tr>
<tr>
<td>49-60</td>
<td>( r^*13 )</td>
</tr>
<tr>
<td>61-72</td>
<td>( r^*23 )</td>
</tr>
</tbody>
</table>
9. Atom Parameter Cards: Needed only if NEW(5) ≠ 0.

If NEW(5) = 1, the entire atom list is read with two cards per atom. If NEW(5) = 2, the Atom Number is read with Format (I2) followed by the two cards for that atom. Card reading is terminated by a blank Atom Number card.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card</td>
<td>1-6</td>
<td>TAG</td>
</tr>
<tr>
<td></td>
<td>7-12</td>
<td>SITE</td>
</tr>
<tr>
<td></td>
<td>13-18</td>
<td>OCCUP</td>
</tr>
<tr>
<td></td>
<td>19-24</td>
<td>OCCA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Card</td>
<td>1-2</td>
<td>ISCAT(1)</td>
</tr>
<tr>
<td></td>
<td>3-4</td>
<td>ISCAT(2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9-15</td>
<td>XYZ(1)</td>
</tr>
<tr>
<td></td>
<td>16-22</td>
<td>XYZ(2)</td>
</tr>
<tr>
<td></td>
<td>23-29</td>
<td>XYZ(3)</td>
</tr>
<tr>
<td></td>
<td>30-36</td>
<td>BETA(1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BETA(2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BETA(3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BETA(4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BETA(5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BETA(6)</td>
</tr>
<tr>
<td></td>
<td>73</td>
<td>IEF</td>
</tr>
<tr>
<td>Card</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1-8</td>
<td>C111</td>
</tr>
<tr>
<td></td>
<td>9-16</td>
<td>C222</td>
</tr>
<tr>
<td></td>
<td>17-24</td>
<td>C333</td>
</tr>
<tr>
<td></td>
<td>25-32</td>
<td>C112</td>
</tr>
<tr>
<td></td>
<td>33-40</td>
<td>C122</td>
</tr>
<tr>
<td></td>
<td>41-48</td>
<td>C113</td>
</tr>
<tr>
<td></td>
<td>49-56</td>
<td>C133</td>
</tr>
<tr>
<td></td>
<td>57-64</td>
<td>C223</td>
</tr>
<tr>
<td></td>
<td>65-72</td>
<td>C233</td>
</tr>
<tr>
<td></td>
<td>73-78</td>
<td>C123</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>IEF</td>
</tr>
</tbody>
</table>

Card 4-5 - Needed if IEF ≠ 0.

Punch d_{1111}, d_{2222}, d_{3333}, d_{1112}, d_{1113}, d_{1122}, d_{1133}, d_{1222}, d_{1232}, d_{1223}, d_{1233}, d_{1333}, d_{2223}, d_{2233}, d_{2333} in order with 8 columns per value.

10. Special parameter cards: Needed only if NEW(5) ≠ 0 and JXPAR ≠ 0.

Card 1: Columns 3-4 contain the number of special parameters, NPAR.
If NPAR>0 the succeeding cards contain the names, initial values, and identifiers according to the following form.

Columns 1-8  Name of parameter, up to 8 alphanumeric characters.
Columns 9-10 Type of parameter, 0-linear function with coefficient given on constraint card.
   1-rigid body system
   2-defined by user in subroutines SPVAL and SPDERI
Columns 11-12 Integer indicating which rigid-body system, if more than one, this parameter refers to.
Column 13-14 Parameter designator. If the type is 1 the code is as follows
   1-X coordinate of origin (in crystal axes system)
   2-Y coordinate of origin
   3-Z coordinate of origin
   4-\( \omega \) angle for transformation to standard, orthonormal coordinate system
   5-\( \chi \) angle for transformation
   6-\( \phi \) angle for transformation
   7-X coordinate of an atom in the special system
   8-Y coordinate of an atom
   9-Z coordinate of an atom
   10-T
   11-T
   12-T
   13-T
   14-T
   15-T
   16-L
   17-L
   18-L
   19-L
   20-L
   21-L
   22-S
   23-S
   24-S
   25-S
   26-S
   27-S
   28-S
   29-S
   30-S
   31-S
   32-S
Columns 15-24 initial value of this parameter.

11. Extinction and Temperature Factor Type Card: Needed only if NEW(5) or NEW(6) \ne 0.

Column 1 contains the extinction factor code:
   0 - Isotropic, do not change
   1 - Anisotropic Type I (mosaic spread dominates)
2 - Anisotropic Type II (domain size dominates)
3 - Now Isotropic, change to Type I
4 - Now Isotropic, change to Type II
5 - Now Type I, change to Isotropic
6 - Now Type II, change to Isotropic
7 - Now Type I, change to Type II
8 - Now Type II, change to Type I

The temperature factor type code is punched with one column per atom beginning in Col. 2 with the following code:

.0 - Now isotropic, do not change
1 - Now anisotropic, do not change
2 - Now isotropic, change to anisotropic
3 - Now anisotropic, change to isotropic
4 - Now 3rd cumulant, do not change
5 - Now anisotropic, change to 3rd cumulant
6 - Now 3rd cumulant, change to anisotropic
7 - Now 4th cumulant, do not change
8 - Now 3rd cumulant, change to 4th
9 - Now 4th cumulant, change to 3rd

12. Parameter Selection Cards. Needed only if NEW(7) ≠ 0. If JXPAR ≠ 0 the first card contains refinement selection for the special parameters in the first NPAR columns. A zero in a column means that the corresponding parameter will not be refined and a one means that it will. After this cards with the refinement indicators for the conventional parameters punched in columns 1-80. If the number of atom parameters is such that more than 80 values are required, the parameter selection information should be continued in column 1 of the next card. The order of parameters is:

a) occupancy - If single occupancy, the total occupancy is refined. If multiple occupancy, the occupancy of species A is refined.
b) x coordinate
c) y coordinate
d) z coordinate
e) thermal parameters, one for isotropic atoms, six for anisotropic atoms. The thermal coefficients occur in the same order as on the Atom Parameter Card 2.
f) 3rd and 4th cumulant parameters (if needed).

13. Parameter Constraint Cards(s): Needed only if NEW(7) ≠ 0.

a) Columns

<table>
<thead>
<tr>
<th>Column(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>Integer specifying the dependent atom number.</td>
</tr>
<tr>
<td>5-6</td>
<td>Dependent parameter number. This number is 1 for occupancy, 2-4 for x through z respectively, 5 for B or β_{11}, and 6, 7, 8, 9, 10 for β_{22}, β_{33}, β_{12}, β_{13} and β_{23} respectively. Values of 11 to 20 treat 3rd cumulant parameters and 21-35 are for the 4th cumulants.</td>
</tr>
</tbody>
</table>
13. Parameter Constraint Cards(s) (Cont.):

a) **Columns**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-9</td>
<td>Atom number of first independent parameter. Special parameters are arbitrarily indicated as atom number 0.</td>
</tr>
<tr>
<td>10-12</td>
<td>Parameter number of first independent parameter. Atom 0 parameter 0 designates an initializing constant in a linear constraint function.</td>
</tr>
<tr>
<td>13-20</td>
<td>Coefficient of first relationship.</td>
</tr>
<tr>
<td>21-23</td>
<td>Atom number of second independent parameter</td>
</tr>
<tr>
<td>24-26</td>
<td>Second independent parameter number</td>
</tr>
<tr>
<td>27-34</td>
<td>Second coefficient</td>
</tr>
<tr>
<td>35-37</td>
<td>Third independent atom number</td>
</tr>
<tr>
<td>38-40</td>
<td>Third independent parameter number.</td>
</tr>
<tr>
<td>41-48</td>
<td>Third coefficient</td>
</tr>
<tr>
<td>49-51</td>
<td>Fourth independent atom number</td>
</tr>
<tr>
<td>52-54</td>
<td>Fourth independent parameter number</td>
</tr>
<tr>
<td>55-62</td>
<td>Fourth coefficient</td>
</tr>
<tr>
<td>63-65</td>
<td>Fifth independent atom number</td>
</tr>
<tr>
<td>66-68</td>
<td>Fifth independent parameter number</td>
</tr>
<tr>
<td>69-76</td>
<td>Fifth coefficient.</td>
</tr>
</tbody>
</table>

If the independent atom number is zero, designating a special parameter, and the constraints are non linear, the second and fourth independent parameter numbers may be negative to denote the end of an inclusive list beginning with the first (third) independent parameter number.

b) One (1) blank card or a card with zero in column 3.

14. Structure Factor Cards: Needed only if NEW(8) ≠ 0.

a) **Format Card:** This card contains the format of the input structure factor data. The number and order of variables on a card (or tape record) depends on the value of NEW(8), as explained in detail below. All records must contain at least four integer fields, for the indices and an observed/unobserved indicator, and two floating point fields for Fobs and σF or other weight information. They may contain other information, such as a scale factor group identifier, the factor β(0) used in extinction corrections, and the setting angles, needed for anisotropic extinction corrections.

b) Only if IWT=5 include a card with the "ignorance factor" punched as a real number in the first 8 columns.

c) Only if NEW(8)=3 include the wavelength in Å in the first 8 columns. For neutron data only an approximation to β(0) will be computed according to the formula β(0) = [(2λ^2/(V^2 sin^2θ))] x 10^8, where V is the unit cell volume in Å^3. The refined quantity then will be the product Tr*, with T assumed to be approximately constant.
d) Data Cards in a form consistent with Format Card: If NEW(8)=1 the list is in this order.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IH(1)</td>
<td>h for reflection</td>
</tr>
<tr>
<td>IH(2)</td>
<td>k</td>
</tr>
<tr>
<td>IH(3)</td>
<td>l</td>
</tr>
<tr>
<td>FOBS</td>
<td>Observed structure factor</td>
</tr>
<tr>
<td>SIGMA</td>
<td>Weight information consistent with IWT on Control Card</td>
</tr>
<tr>
<td>EXBETA</td>
<td>Factor $\beta(\theta)$ for extinction</td>
</tr>
<tr>
<td>I</td>
<td>Number of scale factor for this reflection. If zero, the scale factor is the same as the preceding reflection. Scale factor 1 is assumed for the first reflection.</td>
</tr>
<tr>
<td>MREJ</td>
<td>Minimum Observable Parameter</td>
</tr>
<tr>
<td></td>
<td>1 - Intensity above minimum observable.</td>
</tr>
<tr>
<td></td>
<td>2 - Intensity below minimum observable.</td>
</tr>
<tr>
<td>IEF</td>
<td>End-of-list marker, zero on all data cards</td>
</tr>
<tr>
<td>TTH</td>
<td>$2\theta$ These angles are needed only if anisotropic</td>
</tr>
<tr>
<td>OMEG</td>
<td>$\omega$ extinction is to be calculated and must</td>
</tr>
<tr>
<td>CHI</td>
<td>$\chi$ conform to the diffractometer conventions</td>
</tr>
<tr>
<td>PHI</td>
<td>$\phi$ given by Busing and Levy (1967, p. 458).</td>
</tr>
</tbody>
</table>

If NEW(8) = 2 the input list is IH(1), IH(2), IH(3), MREJ, I, FOBS, EXBETA, SIGMA, with the same meanings as above. The end of the data must be marked by an end-of-file.

If NEW(8) = 3 the list is IH(1), IH(2), IH(3), MREJ, FOBS, SIGMA. EXBETA is computed, and again the end of data must be marked by an end-of-file.

e) End-of-list card - A card with a non-zero value in the field of variable IEF or an end-of-file card appropriate to the system.

15. Run Continuation and Parameter Punch Card: Needed for all cycles.

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ICONT</td>
<td>0 - No more cycles. The data set on file ISCI will be updated if IFILE from the Control Card is zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Read a new title card and start another cycle</td>
</tr>
<tr>
<td>2</td>
<td>JPUN</td>
<td>0 - Do not punch any atom parameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Punch atom cards in form required by 'X-RAY67'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - Punch atom cards in form required by this program</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - Punch atom cards in both forms</td>
</tr>
<tr>
<td>3-4</td>
<td>ISCI</td>
<td>If this field is non-zero its value will be substituted as the unit number for the binary master file, making it possible to try new courses of refinement without destroying previous stages.</td>
</tr>
</tbody>
</table>
BONDAN: A program for computing bond distances, angles, and thermal ellipsoids, with errors

This program uses the master binary file created by RFINE to compute bond distances and angles, and also thermal ellipsoids, with associated standard deviations. If the rigid body model is used, the atomic positions of the rigid body will be corrected for libration before calculation of distances and angles. The program will also accept input from cards by means of the program BADTEA. The complete description of BADTEA is included in Appendix B. The details of data input are as follows:

1. Input format indicator card

Col. 1-2
1-input from cards via BADTEA.
2-input from binary file created by RFINE.

If the input format indicator is 1, see appendix B for further input. If the input format indicator is 2, the card input continues as follows.

2. Title Card: Any desired hollerith information in columns 1-72.

3. Search Card: Format(I2,3A4)

If column 2 is zero or blank, a search of the data tape will be made to find a data set with a label the same as columns 3-14. The error calculation will be performed on that data set. If column 2 is non-zero, the first data set on the tape will be used in the error calculation.

4. Parameter Card: Format(2I1,3F8.0)

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MBODAN</td>
<td>Bond distance, angle selector</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - None computed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Distances only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - Distances and angles computed</td>
</tr>
<tr>
<td>2</td>
<td>MELVIB</td>
<td>Thermal ellipsoid selector</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - No ellipsoids computed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Thermal ellipsoids computed for anisotropic atoms</td>
</tr>
<tr>
<td>3-10</td>
<td>DLIMIT(1)</td>
<td>Minimum bond distance output</td>
</tr>
<tr>
<td>11-18</td>
<td>DLIMIT(2)</td>
<td>Maximum bond distance used in angles output</td>
</tr>
<tr>
<td>19-26</td>
<td>DLIMIT(3)</td>
<td>Maximum bond distance output</td>
</tr>
</tbody>
</table>

5. Cell variance-covariance matrix: Format(6E12.6)

This matrix is the matrix output by the lattice constant program of C. W. Burnham.
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card 1</td>
<td>1-12</td>
</tr>
<tr>
<td></td>
<td>13-24</td>
</tr>
<tr>
<td></td>
<td>25-36</td>
</tr>
<tr>
<td></td>
<td>37-48</td>
</tr>
<tr>
<td></td>
<td>49-60</td>
</tr>
<tr>
<td></td>
<td>61-72</td>
</tr>
<tr>
<td>Card 2</td>
<td>1-12</td>
</tr>
<tr>
<td></td>
<td>13-24</td>
</tr>
<tr>
<td></td>
<td>25-36</td>
</tr>
<tr>
<td></td>
<td>37-48</td>
</tr>
<tr>
<td></td>
<td>49-60</td>
</tr>
<tr>
<td></td>
<td>61-72</td>
</tr>
<tr>
<td>Card 3</td>
<td>1-12</td>
</tr>
<tr>
<td></td>
<td>13-24</td>
</tr>
<tr>
<td></td>
<td>25-36</td>
</tr>
<tr>
<td></td>
<td>37-48</td>
</tr>
<tr>
<td></td>
<td>49-60</td>
</tr>
<tr>
<td></td>
<td>61-72</td>
</tr>
<tr>
<td>Card 4</td>
<td>1-12</td>
</tr>
<tr>
<td></td>
<td>13-24</td>
</tr>
<tr>
<td></td>
<td>25-36</td>
</tr>
</tbody>
</table>

If the errors in the cell constants have a small effect on the distances, the last 4 cards may be replaced by blanks.
FOURIER

This program uses the binary structure factor file created by RFINE (ISC4-unit 12) as input for a Fourier synthesis. The map is written on unit 11 for input to the plotting program CNTPLT or the arbitrary section program ARBSECT. The details of data input are as follows:

1. Title Card: Any desired alphanumeric information in columns 1-72. This information heads all map sections.

2. Parameter Card: Format (6I1, 2X, 9I4, 3F8.04)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ITYPE</td>
<td>Map type indicator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Fourier map</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Difference Fourier</td>
</tr>
<tr>
<td>2</td>
<td>ISECT</td>
<td>Section parameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - (x, y, z) sections</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - (x) across, (y) down and (z) sections</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - (y) across, (x) down and (z) sections</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - (z) across, (y) down and (x) sections</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 - (z) across, (y) down and (x) sections</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 - (z) across, (y) down and (x) sections</td>
</tr>
<tr>
<td>3</td>
<td>INCENT</td>
<td>Centric indicator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Centric</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Acentric</td>
</tr>
<tr>
<td>4</td>
<td>KREJ</td>
<td>0 - Ignore rejection flag and include all reflections.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Do not include any reflections with the rejection flag set.</td>
</tr>
<tr>
<td>5</td>
<td>IPLT</td>
<td>If non-zero, map sections will be written to unit S2 for plotting or calculation of an arbitrary section.</td>
</tr>
<tr>
<td>6</td>
<td>LSPC</td>
<td>If zero, the lines of the map will be double spaced. A non-zero value indicates single spacing.</td>
</tr>
<tr>
<td>9-12</td>
<td>IX</td>
<td>Number of divisions along horizontal axis (&gt;0)</td>
</tr>
<tr>
<td>13-16</td>
<td>IXI</td>
<td>Initial point</td>
</tr>
<tr>
<td>17-20</td>
<td>IXF</td>
<td>Final point</td>
</tr>
<tr>
<td>21-24</td>
<td>IY</td>
<td>Number of division along vertical axis (&gt;0)</td>
</tr>
<tr>
<td>25-28</td>
<td>TYI</td>
<td>Initial line</td>
</tr>
<tr>
<td>29-32</td>
<td>TYF</td>
<td>Final line</td>
</tr>
<tr>
<td>33-36</td>
<td>IZ</td>
<td>Number of divisions along section axis (&gt;0)</td>
</tr>
<tr>
<td>37-40</td>
<td>IZI</td>
<td>Initial section</td>
</tr>
<tr>
<td>41-44</td>
<td>IZF</td>
<td>Final section</td>
</tr>
<tr>
<td>45-52</td>
<td>EXXAG</td>
<td>Map exaggeration desired</td>
</tr>
<tr>
<td>53-60</td>
<td>F000</td>
<td>Number of electrons in unit cell for Fourier.</td>
</tr>
</tbody>
</table>
2. Parameter Card (cont.):

<table>
<thead>
<tr>
<th>61-68</th>
<th>TMULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero for difference Fourier.</td>
<td></td>
</tr>
<tr>
<td>Multiplicity of a general reflection</td>
<td></td>
</tr>
</tbody>
</table>

3. Multiplicity Card: Format (12F6.0)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>Multiplicity of h00 or 0k0</td>
</tr>
<tr>
<td>7-12</td>
<td>Multiplicity of 001</td>
</tr>
<tr>
<td>13-18</td>
<td>Multiplicity of hk0</td>
</tr>
<tr>
<td>19-24</td>
<td>Multiplicity of h01</td>
</tr>
<tr>
<td>25-30</td>
<td>Multiplicity of Ok1</td>
</tr>
<tr>
<td>31-36</td>
<td>Multiplicity of hh0</td>
</tr>
<tr>
<td>37-42</td>
<td>Multiplicity of h0h</td>
</tr>
<tr>
<td>43-48</td>
<td>Multiplicity of 0kk</td>
</tr>
<tr>
<td>49-54</td>
<td>Multiplicity of hh</td>
</tr>
<tr>
<td>55-60</td>
<td>Multiplicity of hh1</td>
</tr>
<tr>
<td>61-66</td>
<td>Multiplicity of hkk or hkh</td>
</tr>
<tr>
<td>67-72</td>
<td>Multiplicity of hkl</td>
</tr>
</tbody>
</table>

C. Size of Program:

This program uses the array FBUF in common block ABLE as pooled storage. The size of this array, S, is governed by the map to be calculated and may be estimated using:

\[
S = 3N_{\text{ref}} + 2(h'_{\text{max}} + 2)(2k'_{\text{max}} + 1) \\
+ 2(h'_{\text{max}} + 1)(P_x + 1)
\]

where \(N_{\text{ref}}\) is the number of reflections included, \(h'_{\text{max}}\) is the maximum value of the index associated with the horizontal axis, \(k'_{\text{max}}\) is the maximum value of the index associated with the axis plotted down the page, and \(P_x\) is the number of points calculated in the horizontal direction. If the Fourier sections are to be saved for later plotting, S should be increased by \(P_xP_y\). If the computer operating system supports dynamic core allocation, the program may be converted to automatically request the proper memory size.

D. Output:

Each map point is converted to a four-digit integer before printing with zero values replaced by blanks. Each point will occupy an area 4 columns by 1 or 2 rows depending upon the spacing selected. For most printers, the area is 0.4 x 1/6 inches for single spacing or 0.4 x 1/3 inches for double spacing of the lines. If the map requires more than 30 points, the section will be printed in strips with 30 points per strip.
E. Additional Problems:

If additional maps are to be run using the same structure factor list, a complete set of cards beginning with the Title Card should be included. An end-of-file in the card stream terminates the program.
This program is an adaption of the XRAY67 program LIST FC (Stewart et al., 1967). It has been altered to stand alone and read the structure factor file (ISC4 - unit 12) created by RFINE. The output of the program has been changed so that the "phase in millicycles" has been replaced by sigma (Fobs). Input is as follows:

Card 1.
Col. 1-6 compound identification; Col. 7-78 any title.

Card 2.
Col. 14 (1)/(2)/(3) for h index varies (most)/(next most)/
(least) rapidly
15 Same for k index
16 Same for l index
(Note that the sum and product ofCols. 14, 15, and 16
must equal 6.)
17-20 Number of lines per list page.
21 (blank)/(A) for (DO)/(DO NOT) internally change the
value of the number of lines in order to make the
bottom of the last page as even as possible.
22-24 Number of list columns per page. Note that the pro-
duct of the number of lines and the number of
columns cannot exceed 2000. That is no more
than 3000 items for the number of reflections and
headings (including spaces) per page.
26 Number of blank print columns before the LISTFC
column.
28 Number of print columns for most rapidly changing
index.
30 Number of print columns for Fobs
32 Number of print columns for Fcalc
34 Number of print columns for Fp
36 Special flag for unobserved reflections (blank=*)
38 Special flag for severely extinct reflections
(blank=E)
40 Special flag for special reflection (JCODE=4)
The next eight fields are (blank)/(1) for (do not)/do
42 Print symbol for unobserved or severely extinct
reflections.
This symbol switch adds one print column
44 For centric structures, attach sign of A to Fcalc
46 Double space the lines
48 Restore each LISTFC page to the top of the printer
page
50 Print title at the top of each printer page
52 Punch a set of "FCARD" cards
56 Print a minus sign on Fobs for unobserved reflections
58 Write a separate copy of FC list on unit 7
62 Number of times to try to get headings at the top of
all columns
(1)/(2)/(3) for headings separated by (no blank lines)/(1 blank line above)/(1 above and 1 below)(blank=3)

65-72 factor Fs are to be multiplied by (blank=10)
73-80 factor σs are to be multiplied by (blank=10)

Note that all F and σ values are printed as integers, so that scale factor must be sufficient to preserve a sufficient number of significant figures.
CNTPLT

CNTPLT is a contour plotting program for Fourier maps and other similar functions. The routines contained in this program have been derived from multiple sources with the original version of subroutine ICONT written by S. H. Zisk and N. M. Brenner of M.I.T. Lincoln Lab. Subroutine INTPLT uses the rational polynomial spline interpolation of D. V. Ahujo, *I.B.M. Systems Journal*, p. 208-217, (1968). The other routines have been modified from code prepared by G. Ford, Purdue University.

The present version of these routines have been written to be as machine independent as possible. It is believed that only the call to the plotter function PLOTS (Sequence No. A5300) and the two functions MARK and UNMARK called by ICONT are machine specific. This is the main contribution of the present author (LWF).

Method

The main contouring subroutine, ICONT, takes the function values at the grid points of the map and computes the coordinates of the intersection of a specified contour line with the lines of the grid. Linear interpolation is used in this process. The second contouring subroutine, CPLLOT, coordinates the drawing of the contours and produces any desired contour labels. The third contouring routine, INTPLT, produces a smooth curve through the intersections located by ICONT. Rational cubic polynomials are used to fit a spline-like curve through neighboring points. Two such curves meet smoothly because the first and second derivatives are single valued along the composite curve. The plotter line is drawn as a series of straight line segments with the number of such segments specified by the user.

The main program includes various options which allow labeling of plot sections and/or grid points, drawing of the bounding parallelogram, and other similar functions.

Input

The values of the function to be plotted are read from a binary input file (unit 10) with each section occupying two records. The first contains the variables MA, MB, JZ, MZ where MA and MB are the number of grid points along x and y respectively. The second record contains the data for the map section at JZ/MZ. These data are assumed to be in real form and are packed in an array which is MAXMB. The output file produced by program FOURIER is in this form.

Input

1. Title Card: Format (20A4)
2. Control Card: Format (F5.0, I5,3F5.0,6I5,F5.0,2I5)
<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>TLEHGT</td>
<td>Plotting height of title in inches. If &lt;0, no title will be plotted</td>
</tr>
<tr>
<td>6-10</td>
<td>NLABL</td>
<td>Number of vertex labels to be plotted</td>
</tr>
<tr>
<td>11-15</td>
<td>A</td>
<td>Width of plot (in inches) parallel to x-axis of Fourier. If A&gt;0, an 11' plotter is assumed. If a 30&quot; machine is available, A should be less than zero. Length of plot (in inches) parallel to y-axis of Fourier which is parallel to plotter x-axis.</td>
</tr>
<tr>
<td>16-20</td>
<td>B</td>
<td>Angle between Fourier axes (in degrees)</td>
</tr>
<tr>
<td>21-25</td>
<td>ALP</td>
<td>Number of grid points parallel to Fourier x-axis</td>
</tr>
<tr>
<td>26-30</td>
<td>NA</td>
<td>Number of grid points parallel to Fourier y-axis</td>
</tr>
<tr>
<td>31-35</td>
<td>NB</td>
<td>Number of grid points parallel to Fourier y-axis</td>
</tr>
<tr>
<td>36-40</td>
<td>NP</td>
<td>Number of interpolation points per spline. The default value is 4.</td>
</tr>
<tr>
<td>41-45</td>
<td>NCONT</td>
<td>Number of sets of contour levels (NCONT&lt;10)</td>
</tr>
<tr>
<td>46-50</td>
<td>NLINE</td>
<td>Number of line vertices specified (NLINE&lt;20)</td>
</tr>
<tr>
<td>51-55</td>
<td>NCROSS</td>
<td>Number of crosses to be placed on plot (NCROSS&lt;20)</td>
</tr>
<tr>
<td>56-60</td>
<td>SCROSS</td>
<td>Size of cross (in. inches)</td>
</tr>
<tr>
<td>61-65</td>
<td>NSECT</td>
<td>Number of sets of sections specified</td>
</tr>
<tr>
<td>66-70</td>
<td>NZ</td>
<td>Grid spacing of section levels.</td>
</tr>
</tbody>
</table>

3. Contour Card(s): Format (3F5.0,I5)
There should be NCONT of these cards.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>SBEG</td>
<td>Height of initial contour</td>
</tr>
<tr>
<td>6-10</td>
<td>CSTEP</td>
<td>Step between contours</td>
</tr>
<tr>
<td>11-15</td>
<td>SLIM</td>
<td>Height of final contour</td>
</tr>
<tr>
<td>16-20</td>
<td>NSIG</td>
<td>Number of digits after decimal point in contour label. If NSIG&lt;0, contours will not be labelled.</td>
</tr>
</tbody>
</table>

4. Line Vertex Card(s): Format (16F5.0)

The positions of the line vertices in map grid coordinates should be punched. A line will be drawn from point i to point i+1 if x_{i+1}>0. No line will be drawn if x_{i+1}<0. Omit these cards if NLINE<0.
Note: The first grid point is (1,1), not (0,0). i.e. add (1,1) to scaled coordinates.

5. Cross Card(s): Format (16F5.0)
The map grid coordinates of the crosses should be punched sequentially. Omit these cards if NCROSS<0.

6. Section Selection Cards: Format (3I5)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>INITZ</td>
<td>Sections from INITZ to IFINZ in steps of ISTEPZ</td>
</tr>
<tr>
<td>6-10</td>
<td>ISTEPZ</td>
<td></td>
</tr>
<tr>
<td>11-15</td>
<td>IFINZ</td>
<td></td>
</tr>
</tbody>
</table>

There should be NSECT of these cards.

7. Vertex Label Card(s): Format (5F5.0,A6)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>WX</td>
<td>x-grid coordinate</td>
</tr>
<tr>
<td>6-10</td>
<td>WY</td>
<td>y-grid coordinate</td>
</tr>
<tr>
<td>11-15</td>
<td>DX</td>
<td>Plotter x-axis offset</td>
</tr>
<tr>
<td>16-20</td>
<td>DY</td>
<td>Plotter y-axis offset</td>
</tr>
<tr>
<td>21-25</td>
<td>HGT</td>
<td>Label height in inches</td>
</tr>
<tr>
<td>26-31</td>
<td>LABL</td>
<td>Alphanumeric label</td>
</tr>
</tbody>
</table>

There should be NLABL of these cards.

Tape output for Calcomp plotter is on Unit 12.
ARBSECT

This program calculates an arbitrary section of a Fourier map from a complete set of sections calculated in regular intervals parallel to the crystallographic axes. The input sections may be computed with the regular Fourier program and are assumed to be in file No. 11. The plane of the output section is specified by the coordinates of the plane to be calculated.

Program ARBSECT has several options available. If only a fraction of the unit cell is used as input, it will calculate the electron density in the accessible portion only with the remainder replaced by asterisks in the printout. However, if the map for a complete unit-translation has been computed, the program will use the translation symmetry to calculate a complete section. No other symmetry operations are used. Other options include (a) describing the points used to define the plane in terms of map or fractional coordinates as desired, (b) saving the resulting section in file 12 for later plotting, (c) computing the entire section or selecting a portion of it, and (d) calculating and printing the geometry including axial lengths and interaxial angle, of the output section.

Method

The input coordinates of the three points defining the plane are initially converted to map coordinates if necessary. The coefficients of the plane in the equation \( a_1 x + a_2 y + a_3 z + a_4 = 0 \) may be found from the determinant equation

\[
\begin{vmatrix}
  x & y & z & 1 \\
  x_1 & y_1 & z_1 & 1 \\
  x_2 & y_2 & z_2 & 1 \\
  x_3 & y_3 & z_3 & 1 \\
\end{vmatrix} = 0
\]

where \( x_i, y_i, \) and \( z_i \) are the coordinates of the \( i \)'th point. This plane will have a line of intersection with each section of the original map. For the values of \( y \) and \( z \) corresponding to each line and section of the output map, the \( x \)-coordinate of this point in the plane is determined. A cubic function of the form \( ax^3 + bx^2 + cx + d \) is fitted to the four adjacent grid points on the line, and the value of this function at the intersection point is used for the output at that \( y \) and \( z \). This process is repeated until all output points have been determined.

Input

The input map sections are read from file 11 and the resulting map is output to file 12. These assignments may be changed by redefining variables ISC and ISC1 respectively.
1. Title Card - Format (18A4)

Columns 1-72 of this card may contain any desired information.

2. Control Card - Format (8I3)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>IO</td>
<td>If IO≠0, the output section will be written on Unit 12.</td>
</tr>
<tr>
<td>6</td>
<td>IZO</td>
<td>Section number of output</td>
</tr>
<tr>
<td>9</td>
<td>ISPC</td>
<td>If ISPC = 0 the output map will be single spaced; double spaced if ISPC=1.</td>
</tr>
<tr>
<td>12</td>
<td>IMRK</td>
<td>If IMRK=0, the y- and z- limits of the input sections will be used as the x-, and y-limits, respectively, of the output section. If IMRK ≠0, these values are read from this card.</td>
</tr>
<tr>
<td>13-15</td>
<td>IYI</td>
<td>Initial value of x in transformed section</td>
</tr>
<tr>
<td>16-18</td>
<td>IYF</td>
<td>Final value of x in transformed section</td>
</tr>
<tr>
<td>19-21</td>
<td>IZI</td>
<td>Initial value of y in transformed section</td>
</tr>
<tr>
<td>22-24</td>
<td>IZF</td>
<td>Final value of y in transformed section</td>
</tr>
</tbody>
</table>

3. Cell Card - Format (6F8.0)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-8</td>
<td>Length (in A) of cell axis across page in original section</td>
</tr>
<tr>
<td>9-16</td>
<td>Length of cell axis down page in original section</td>
</tr>
<tr>
<td>17-24</td>
<td>Length of cell axis out of page in original section</td>
</tr>
<tr>
<td>25-32</td>
<td>Angle (in degrees) between axis 1 and 2</td>
</tr>
<tr>
<td>33-40</td>
<td>Angle between axes 1 and 3</td>
</tr>
<tr>
<td>41-48</td>
<td>Angle between axes 2 and 3</td>
</tr>
</tbody>
</table>

The above length and axes are required only if the program is to compute and print the axial length and interaxial angle of the transformed section. If not, this card may be blank.

4. Plane Card - Format (9F8.0,I2)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-8</td>
<td>x- coordinate of first point</td>
</tr>
<tr>
<td>9-16</td>
<td>y- &quot; &quot; &quot; &quot; &quot;</td>
</tr>
<tr>
<td>17-24</td>
<td>z- &quot; &quot; &quot; second &quot;</td>
</tr>
<tr>
<td>25-32</td>
<td>x- &quot; &quot; &quot; second &quot;</td>
</tr>
<tr>
<td>33-40</td>
<td>y- &quot; &quot; &quot; &quot;</td>
</tr>
<tr>
<td>41-48</td>
<td>z- &quot; &quot; &quot; &quot;</td>
</tr>
<tr>
<td>49-56</td>
<td>x- &quot; &quot; third &quot;</td>
</tr>
<tr>
<td>57-64</td>
<td>y- &quot; &quot; &quot; &quot;</td>
</tr>
</tbody>
</table>
4. Plane Card - (Cont.)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>65-72</td>
<td>z-coordinate of third point</td>
</tr>
<tr>
<td>74</td>
<td>Coordinate type indicator. If zero, the coordinates are given in grid spacings. If non-zero, the points are given in fractional coordinates.</td>
</tr>
</tbody>
</table>

Additional arbitrary slices may be computed by supplying additional data decks beginning with the Title Card.

Form of Input Sections

The input Fourier sections are read from Unit 11 with unformatted reads. There must be two records per section as follows:

Record 1:

<table>
<thead>
<tr>
<th>Word</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>JXA</td>
<td>Number of grid points calculated parallel to x</td>
</tr>
<tr>
<td>2</td>
<td>JYA</td>
<td>Number of grid points calculated parallel to y</td>
</tr>
<tr>
<td>3</td>
<td>KZC</td>
<td>Section number</td>
</tr>
<tr>
<td>4</td>
<td>IZ</td>
<td>Number of sections per unit cell</td>
</tr>
<tr>
<td>5</td>
<td>IX</td>
<td>Number of intervals per unit cell parallel to x</td>
</tr>
<tr>
<td>6</td>
<td>IXI</td>
<td>Initial grid points parallel to x</td>
</tr>
<tr>
<td>7</td>
<td>IXF</td>
<td>Final grid point parallel to x</td>
</tr>
<tr>
<td>8</td>
<td>IY</td>
<td>Number of intervals per unit cell parallel to y</td>
</tr>
<tr>
<td>9</td>
<td>JYI</td>
<td>Initial grid point parallel to y</td>
</tr>
<tr>
<td>10</td>
<td>JYF</td>
<td>Final grid point parallel to y</td>
</tr>
<tr>
<td>11</td>
<td>JZI</td>
<td>Initial grid point parallel to z</td>
</tr>
<tr>
<td>12</td>
<td>JZF</td>
<td>Final grid point parallel to z</td>
</tr>
</tbody>
</table>

Record 2:

This record consists of the JXA*JYA values of the map in this section. These real values must be packed in an array with the x coordinate as the first subscript and y as the second subscript.
REFERENCES CITED


Appendix A
Overlay scheme for RFINE

When program RFINE is run under operating systems which charge according to the product of core allocation and CPU time, substantial savings can be achieved by using an overlay structure which causes only that portion of the program which is actually needed to be resident in core. Furthermore, there can be large savings if the core allocated for the dependency list and the normal equations matrix can be adjusted as appropriate to a particular problem. On UNIVAC machines operating under EXEC 8 this latter function is accomplished dynamically at run time by means of the subroutine CORSZ, which makes use of executive requests ER MCORE and ER LCORE. On other machines which don't have a similar function capability, this can be achieved by a run time recompilation of the short main program, with the dimension of the array A in common block LSMAT appropriately adjusted.

The program has been run successfully with a structure consisting of a main stem and three overlay segments as follows:

Main stem: main program, subroutines RFINE and CORSZ.
Overlay 1: Subroutines RESET, DERIVS, RDEGN, and PAREXT plus auxiliary functions called by them, including DIJKLM, DETERM, EL4, PIJLM, BLM, CIJKL, AIJ, VXV, SOLVE, EIJ, FJ, IPAK, PDEGN, D3KDS, D2KDS, P3KDL, P2KDL, D2KDT, TFM, SKTLS, CORECT, MULMM, TRNSPS, EULER, FILLIN, SPVAL, ERRCAL.
Overlay 2: Subroutines INPUT, SFAC, RCALC, MATRIX, MODIFY, and SYMINV.
Overlay 3: Subroutines TRNFRM, BODAN1, and ELVIB1.
Appendix B
BADTEA-Alternate input for bond distances and angles, and thermal ellipsoids

A. IDENTIFICATION

TITLE: Crystallographic error analysis

NAME: U M BADTEA

PROGRAMMER: Larry W. Finger

DATE: December 27, 1965 (revised November 1968)

B. PURPOSE

U M BADTEA (University of Minnesota program for computing Bond angles and Distances, and Thermal Ellipsoids with error Analysis) will calculate bond distances, bond angles, thermal ellipsoids and the errors associated with each computed value. The program was converted to Fortran IV with the assistance of C. T. Prewitt.

C. USAGE

1. Input deck order:

Title card
Cell card
Function card
Symmetry card(s)
*Standard Deviations of cell parameters
*Correlation matrix of cell parameters
Atom cards
*Temperature Factor Standard Deviations
*Temperature Factor Correlation matrix
*Atomic Position Standard Deviations
*Atomic Position Correlation Matrix

*These data may not be needed. The next section describes in detail the conditions requiring these input data. All input from Atom Cards to the end of the deck may come from a modified ORFFF tape.

2. Description of input deck:

a. Title Card

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 72</td>
<td>Identifier for problem. This should be any title which identifies the compound.</td>
</tr>
</tbody>
</table>
b. Cell Card

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 10</td>
<td>a axis length, in Å</td>
</tr>
<tr>
<td>11 - 20</td>
<td>b axis length</td>
</tr>
<tr>
<td>21 - 30</td>
<td>c axis length</td>
</tr>
<tr>
<td>31 - 40</td>
<td>α in degrees</td>
</tr>
<tr>
<td>41 - 50</td>
<td>β</td>
</tr>
<tr>
<td>51 - 60</td>
<td>γ</td>
</tr>
</tbody>
</table>

c. Function Card

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 4</td>
<td>Number of atoms in asymmetric unit (&lt;60)</td>
</tr>
</tbody>
</table>
| 5 - 8   | Centric Indicator  
|         | 0 - Structure is centric  
|         | 1 - Structure is non-centric |
| 9 - 12  | Bond Distance, Angle Parameter  
|         | 0 - No angles or distances calculated  
|         | 1 - Distances only  
|         | 2 - Angles and Distances |
| 13 - 16 | Ellipsoid Indicator  
|         | 0 - None calculated  
|         | 1 - Ellipsoids calculated for anisotropic atoms |
| 17 - 20 | Error Indicator  
|         | 0 - Errors not calculated  
|         | 1 - Errors calculated  
|         | 2 - Errors calculated with structure information read from modified ORFFE tape. |
| 24      | Lattice Type Code - This column should be punched with a P, A, B, C, F, I, or R depending on the lattice type. |
| 31 - 40 | Minimum bond distance to be printed |
| 41 - 50 | Maximum distance used in angle calculation. |
| 51 - 60 | Maximum bond distance to be printed. |

d. Symmetry Cards

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 3</td>
<td>Translational part of x'</td>
</tr>
<tr>
<td>5 - 6</td>
<td>First positional part of x'</td>
</tr>
<tr>
<td>8 - 9</td>
<td>Second positional part of x'</td>
</tr>
<tr>
<td>11 - 13</td>
<td>Translational part of y'</td>
</tr>
<tr>
<td>15 - 16</td>
<td>First positional part of y'</td>
</tr>
<tr>
<td>18 - 19</td>
<td>Second positional part of y'</td>
</tr>
<tr>
<td>21 - 23</td>
<td>Translational part of z'</td>
</tr>
<tr>
<td>25 - 26</td>
<td>First positional part of z'</td>
</tr>
<tr>
<td>28 - 29</td>
<td>Second positional part of z'</td>
</tr>
</tbody>
</table>
End of Symmetry Deck Indicator
Blank - more symmetry cards follow
1 - this is the last card

Note: The translational parts of the symmetry operators are of the form: bbb 1/2 1/3 2/3 1/4 3/4 1/6 or 5/6 where b signifies a blank
The positional parts are of the form: bw + w - w where w may be X, Y or Z. Symmetric positions which are related by a center of
inversion or a multiple lattice point must not be included since
these positions are internally generated. However, position x y z
must be included and must be first.

e. Standard Deviations of Cell Parameters FORMAT (6F10.8)

This card and the cards in section f. below are needed
only if errors are to be calculated.

Columns Description
1 - 10 Standard Deviation of a
11 - 20 Standard Deviation of b
21 - 30 Standard Deviation of c
31 - 40 Standard Deviation of α in degrees
41 - 50 Standard Deviation of β
51 - 60 Standard Deviation of γ

f. Cell Parameter Correlation Matrix FORMAT (6F8.5)

Columns Correlation Coefficient of
Card 1:
1 - 8 a with a (1.0)
Card 2:
1 - 8 a with b
9 - 16 b with b (1.0)
Card 3:
1 - 8 a with c
9 - 16 b with c
17 - 24 c with c (1.0)
Card 4:
1 - 8 a with α
9 - 16 b with α
17 - 24 c with α
25 - 32 α with α (1.0)
Card 5:
1 - 8 a with β
9 - 16 b with β
17 - 24 c with β
25 - 32 α with β
33 - 40 β with β (1.0)
Cell Parameter Correlation Matrix (Cont).

Columns | Correlation Coefficient of
---|---
Card 6:
1 - 8 | $a$ with $\gamma$
9 - 16 | $b$ with $\gamma$
17 - 24 | $c$ with $\gamma$
25 - 32 | $\alpha$ with $\gamma$
33 - 40 | $\beta$ with $\gamma$
41 - 48 | $\gamma$ with $\gamma$ (1.0)

**g. Atom Cards**

FORMAT (2A3, 7X, 3F7.5, I2/6F8.5)

There should be a set of 2 atom cards for each atom in the asymmetric unit.

**Position Card:**

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 6</td>
<td>Identification for atom</td>
</tr>
<tr>
<td>14 - 20</td>
<td>$x$ in fractional coordinates</td>
</tr>
<tr>
<td>21 - 27</td>
<td>$y$ in fractional coordinates</td>
</tr>
<tr>
<td>28 - 34</td>
<td>$z$ in fractional coordinates</td>
</tr>
<tr>
<td>36</td>
<td>Temperature factor type</td>
</tr>
<tr>
<td></td>
<td>0 - isotropic</td>
</tr>
<tr>
<td></td>
<td>1 - anisotropic</td>
</tr>
</tbody>
</table>

**Temperature factor card:**

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 8</td>
<td>$\beta_{11}$ or $B$</td>
</tr>
<tr>
<td>9 - 16</td>
<td>$\beta_{22}$</td>
</tr>
<tr>
<td>17 - 24</td>
<td>$\beta_{33}$</td>
</tr>
<tr>
<td>25 - 32</td>
<td>$\beta_{12}$</td>
</tr>
<tr>
<td>33 - 40</td>
<td>$\beta_{13}$</td>
</tr>
<tr>
<td>41 - 48</td>
<td>$\beta_{23}$</td>
</tr>
</tbody>
</table>

The anisotropic coefficients should be those for which the temperature factor expression is

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

**h. Temperature Factor Standard Deviations**

FORM (6F10.8)

These cards are needed only for anisotropic atoms when ellipsoids and errors are to be calculated.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 10</td>
<td>Standard Deviation of $\beta_{11}$</td>
</tr>
<tr>
<td>11 - 20</td>
<td>Standard Deviation of $\beta_{22}$</td>
</tr>
<tr>
<td>21 - 30</td>
<td>Standard Deviation of $\beta_{33}$</td>
</tr>
</tbody>
</table>
Temperature Factor Standard Deviations (Cont)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31 - 40</td>
<td>Standard Deviation of $\beta_{12}$</td>
</tr>
<tr>
<td>41 - 50</td>
<td>Standard Deviation of $\beta_{13}$</td>
</tr>
<tr>
<td>51 - 60</td>
<td>Standard Deviation of $\beta_{23}$</td>
</tr>
</tbody>
</table>

i. Temperature Factor Correlation Matrix  FORMAT (6 F8.5)

These cards are needed only for anisotropic atoms when ellipsoids and errors are to be calculated.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Correlation Coefficient of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card 1:</td>
<td>$\beta_{11}$ with $\beta_{11}$ (1.0)</td>
</tr>
<tr>
<td>Card 2:</td>
<td>$\beta_{11}$ with $\beta_{22}$, $\beta_{22}$ with $\beta_{22}$ (1.0)</td>
</tr>
<tr>
<td>Card 3:</td>
<td>$\beta_{11}$ with $\beta_{33}$, $\beta_{22}$ with $\beta_{33}$, $\beta_{33}$ with $\beta_{33}$ (1.0)</td>
</tr>
<tr>
<td>Card 4:</td>
<td>$\beta_{11}$ with $\beta_{12}$, $\beta_{22}$ with $\beta_{12}$, $\beta_{33}$ with $\beta_{12}$, $\beta_{12}$ with $\beta_{12}$ (1.0)</td>
</tr>
<tr>
<td>Card 5:</td>
<td>$\beta_{11}$ with $\beta_{13}$, $\beta_{22}$ with $\beta_{13}$, $\beta_{33}$ with $\beta_{13}$, $\beta_{13}$ with $\beta_{13}$ (1.0)</td>
</tr>
<tr>
<td>Card 6:</td>
<td>$\beta_{11}$ with $\beta_{23}$, $\beta_{22}$ with $\beta_{23}$, $\beta_{33}$ with $\beta_{23}$, $\beta_{13}$ with $\beta_{23}$, $\beta_{13}$ with $\beta_{23}$ (1.0)</td>
</tr>
</tbody>
</table>

Each anisotropic atom has a Standard Deviations card and a Correlation Matrix.

j. Standard Deviations of Atomic Positions  FORMAT (3F10.8)

These cards are needed only if bond distances or angles are to be calculated with errors.

46
Standard Deviations of Atomic Positions (Cont)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 10</td>
<td>Standard deviation of ( x )</td>
</tr>
<tr>
<td>11 - 20</td>
<td>Standard deviation of ( y )</td>
</tr>
<tr>
<td>21 - 30</td>
<td>Standard deviation of ( z )</td>
</tr>
</tbody>
</table>

There should be one card of this type for each atom in the input list.

k. Atomic Positions Correlation Matrix

These cards are needed only if bond distances or angles are to be calculated with errors. The lower triangle of this matrix is read by rows with each row starting on a new card. When a row contains more than ten elements, it should be continued on the next card.

<table>
<thead>
<tr>
<th>Card 1:</th>
<th>Correlation Coefficient of</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 8</td>
<td>( x ) of atom 1 with ( x ) of atom 1 (1.0)</td>
</tr>
<tr>
<td>Card 2:</td>
<td></td>
</tr>
<tr>
<td>1 - 8</td>
<td>( x ) of atom 1 with ( y ) of atom 1 (1.0)</td>
</tr>
<tr>
<td>9 - 16</td>
<td>( y ) of atom 1 with ( y ) of atom 1 (1.0)</td>
</tr>
<tr>
<td>Card 3:</td>
<td></td>
</tr>
<tr>
<td>1 - 8</td>
<td>( x ) of atom 1 with ( z ) of atom 1 (1.0)</td>
</tr>
<tr>
<td>9 - 16</td>
<td>( y ) of atom 1 with ( x ) of atom 2 (1.0)</td>
</tr>
<tr>
<td>17 - 24</td>
<td>( z ) of atom 1 with ( z ) of atom 1 (1.0)</td>
</tr>
<tr>
<td>Card 4:</td>
<td></td>
</tr>
<tr>
<td>1 - 8</td>
<td>( x ) of atom 1 with ( x ) of atom 2</td>
</tr>
<tr>
<td>9 - 16</td>
<td>( y ) of atom 1 with ( x ) of atom 2</td>
</tr>
<tr>
<td>17 - 24</td>
<td>( x ) of atom 1 with ( x ) of atom 2</td>
</tr>
<tr>
<td>25 - 32</td>
<td>( x ) of atom 2 with ( x ) of atom 2</td>
</tr>
</tbody>
</table>

This scheme should be continued until all elements of the matrix have been punched.

D. METHOD

1. Bond distances:

If \( X \) is a column matrix of fractional atomic coordinates for an atom in the asymmetric unit and \( Y \) is the column matrix of the coordinates of a symmetry related atom, then \( Y \) and \( X \) satisfy the following equation:

\[
Y = RX + T
\]  

(1)

where \( R \) is a rotation-like operation in that it involves axis interchanges and \( T \) is a translation operation which includes
glide and screw components as well as origin shifts. Defining Z as a column matrix whose coefficients are the differences in fractional coordinates between an atom at X and an atom at Y, then

\[ Z = X - Y \]  

(2)

If G is defined as the real space metric tensor which has elements

\[ g_{ij} = \mathbf{a}_i \cdot \mathbf{a}_j \]  

(3)

where the \( \mathbf{a}'s \) are the real space axis vectors, then G is the following matrix:

\[
G = \begin{pmatrix}
  a^2 & ab \cos \gamma & ac \cos \beta \\
  ab \cos \gamma & b^2 & bc \cos \alpha \\
  ac \cos \beta & bc \cos \alpha & c^2
\end{pmatrix}
\]  

(4)

If \( Z^T \) denotes the transpose of Z which involves an interchange of the rows and columns of Z, it may be shown that the square of the bond distance, \( s^2 \), between the atoms is determined by the following matrix equation:

\[ s^2 = Z^T G Z \]  

(5)

In the program, all atoms within a spherical shell centered on each atom in the asymmetric unit are located and the distances are calculated. The size of the spherical shell is determined by the minimum and maximum distances to be considered which are input parameters. The search for the contents of the shell is made efficient by enclosing the central atom with a unit cell shaped parallelopiped tangent to the sphere with all atomic locations outside this parallelopiped excluded. The position of those atoms which lie within the spherical shell are stored in a table. This table also includes the atom number and the number of the R matrix used to generate this position which is needed to properly compute the partial derivatives of the bond distance with respect to the positional coordinates. These two numbers are stored in packed form in the same storage location. Each time a new atom is generated, this table is searched to determine whether this atom has already been included. This search assures that atoms in special positions are properly treated although they may be generated in several different ways.

2. Bond angles:

All possible bond angles about each atom in the asymmetric unit are computed using the table of positions prepared during the bond distance calculation. Each pair of atoms from this table is checked to determine whether their distances from the central atom are within the limits, and if so, the bond angle is computed. If
X denotes the column matrix of the central atom with \( Y_1 \) and \( Y_2 \) denoting the column matrices of the peripheral atoms, the bond angle \( \Theta \) for the angle \( Y_1 - X - Y_2 \) can be shown to satisfy the following relationship:

\[
\cos \Theta = \frac{(Y_1 - X)^T G (Y_2 - X)}{S_{1X} S_{2X}}
\]

(6)

Where \( G \) is the real space metric tensor (see equation (4))

- \( S_{1X} \) is the bond distance from the atom of \( X \) to the atom at \( Y_1 \) and
- \( S_{2X} \) is the bond distance from the atom at \( X \) to the atom at \( Y_2 \).

3. Ellipsoids of Vibration:

The conversion of anisotropic thermal coefficients is a problem of finding the characteristic values and characteristic vectors of a particular matrix. Note that characteristic value, proper value, principal value and eigenvalue are equivalent terms. The matrix equation that must be solved is the following:

\[
(B - \lambda G^{-1})Q = 0
\]

(7)

where \( B \) is the matrix of anisotropic temperature coefficients

- \( \lambda \) is a characteristic value of the equation
- \( G^{-1} \) is the reciprocal space metric tensor with elements
  \[
g^{-1}_{ij} = b_i \cdot b_j \text{ where the } b's \text{ are the reciprocal space axis vectors}
\]

- \( Q \) is a column matrix of the coefficients of the characteristic vector. This is a vector in reciprocal space.

If equation (7) is premultiplied by \( G \), it is transformed into the standard form of a characteristic value problem.

\[
(GB - \lambda I)Q = 0
\]

(8)

where \( I \) is the identity matrix and

- \( G \) is the real space metric tensor defined in equations (3) and (4).

Equation (8) is a set of homogeneous equations of order 3 which have a solution if the determinant of the coefficients is zero. Thus the characteristic values may be found from the equation:
Solution of this equation gives three values of $\lambda$ which satisfy equation (8). In order to have a meaningful physical situation, it is necessary that the $\lambda$'s which are the roots of equation (9) be real and positive. If they are not, the set of temperature coefficients is said to be non-positive definite. This is sometimes referred to as complex amplitudes of vibration or negative temperature factors.

Each of the roots of equation (9) can then be inserted into equation (8) and the resulting equation solved for the characteristic vector by the method of cofactors. After the coefficients of $Q$ are determined, they may be used to form the reciprocal space vector $\mathbf{q}$ and the angle, $\theta_i$, between this vector $\mathbf{q}$ and the real space axis vector $\mathbf{a}_i$ may be found by forming the dot product of the two vectors:

$$\mathbf{a}_i \cdot \mathbf{q} = |\mathbf{a}_i| |\mathbf{q}| \cos \theta_i$$  \hspace{1cm} (10)

This equation may be simplified since

$$\mathbf{a}_i \cdot \mathbf{q} = \mathbf{a}_i \cdot (q_1 b_1 + q_2 b_2 + q_3 b_3)$$  \hspace{1cm} (11)

and

$$\mathbf{a}_i \cdot \mathbf{b}_j = w_{ij} = 1 \text{ for } i = j$$  \hspace{1cm} (12)

thus

$$\mathbf{a}_i \cdot \mathbf{q} = q_i$$  \hspace{1cm} (13)

Equation (12) is the definition of the reciprocal space axis vectors. Equation (10) can now be rewritten

$$\cos \theta_i = \frac{q_i}{|\mathbf{a}_i| |\mathbf{q}|}$$  \hspace{1cm} (14)

The root-mean-square amplitude of vibration along the $i$'th characteristic vector is called $\mu_i$ and is found from the $i$'th root, $\lambda_i$ of equation (9) by the following relationship:

$$\mu_i = (\lambda_i / 2\pi^2)^{1/2}$$  \hspace{1cm} (15)

4. Error Analysis:

If $y$ is a computed value which is a function of experimentally determined variables $x_1, x_2, \ldots, x_n$ such that:

$$y = y(x_1, x_2, \ldots, x_n)$$  \hspace{1cm} (16)
then the variance of \( y \), denoted by \( \sigma_y^2 \) is found from the equation

\[
\sigma_y^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial y}{\partial x_i} \frac{\partial y}{\partial x_j} \text{cov}(x_i, x_j)
\]

(17)

where \( \text{cov}(x_i, x_j) \) is the \( ij \)'th element in the variance-covariance matrix of the \( x \)'s.

In crystallographic calculations, the variables for which the variance-covariance matrix elements are known include the unit cell parameters, the atomic positions and the temperature factor coefficients. If the dependence of the bond distance on temperature factors is neglected, as is done in this program, the error in the bond distance will depend only on the cell parameters and the atomic positions. Similarly the ellipsoids will depend only on the unit cell parameters and the temperature factor coefficients. Thus, for crystallographic problems, equation (17) may be rewritten in the form:

\[
\sigma_y^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial y}{\partial a_i} \frac{\partial y}{\partial a_j} \text{cov}(a_i, a_j)
\]

\[
+ \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{\partial y}{\partial z_i} \frac{\partial y}{\partial z_j} \text{cov}(z_i, z_j)
\]

(18)

where the \( a \)'s are the unit cell parameters and the \( z \)'s are the atomic parameters. The variance-covariance matrix is the inverse of the normal equations least squares matrix. The error formula in equation (18) is a "full matrix" error formula. The so-called "diagonal" form includes only the terms in equation (18) which have \( i=j \). This approximation usually gives a smaller error. All partial derivatives required for computing the errors are analytically evaluated in this program. This procedure is better than numerical differentiation because it is more accurate, easier and faster to do and allows singularities in the derivatives to be detected easier. The necessary derivatives are evaluated using the chain rule of derivatives.

**E. OUTPUT**

1. General Output:

The first output consists of the information contained on the Title Card. Then the real space axis lengths, the cosines of the real space angles, the reciprocal space axis lengths and the cosines of the real space angles, the reciprocal space axis lengths and the cosines of the reciprocal angles are printed. The real space cell volume is also printed. Then the lattice type and
centric indicator is printed followed by the symmetry operations. Next, if errors are to be calculated, the standard deviations of the cell parameters and the cell parameter correlation matrix are printed. Then the atomic positions and temperature factors are printed. If ellipsoids and errors are to be calculated, the standard deviations of the thermal parameters and temperature factors are printed. If ellipsoids and errors are to be calculated, the standard deviations of the atomic positions and the correlations matrix for the positions are then printed. This completes the general output.

2. Bond Distance and Angle Output:

For each atom in the asymmetric unit, all atoms within the bond distance limits are generated. These atoms are added to a table of generated positions. For each bond distance within the allowable range, the atom identifier, the fractional coordinates, the bond distance, the error, if calculated, and the number of this atom in the table are printed. In the bond angle section of the program, all atoms except those in the asymmetric unit will be referenced by this atom number rather than by coordinates. This number makes it easy to determine quickly what angle has been calculated.

The bond angle section output consists of the atom identifier and coordinates of the central atom which is one of those in the asymmetric unit. The peripheral atoms are each referenced by an atom identifier and an atom table number. The bond angle and its error, if calculated, complete the output for each angle.

3. Ellipsoid of Vibration Output:

For each anisotropic atom, the equivalent isotropic B and its error are calculated and printed. Then for each of the characteristic values, the r.m.s. amplitude of vibration and the angles that the associated characteristic vector makes with the real cell axes as well as the errors in these quantities are printed. No output occurs for isotropic atoms.

4. Page Estimates:

The main program will produce approximately one page of output plus 1/2 page per atom if errors are to be calculated. The bond distance and angle subroutine will require about one page for each 60 distances or angles calculated. However, the user should keep the bond distance limits as small as possible since if n distances are within the limits, the number of angles computed will be approximately 1/2 n^2. The ellipsoids of vibration subroutine will require one page for each three atoms.
F. PROGRAM ERRORS AND ERROR STOPS

1. There are two kinds of errors which return control to the monitor through a STOP statement. If a symmetry card cannot be interpreted because the information is not of the correct form or because it is punched in the wrong columns, the programs executes a STOP 11 statement and returns to the monitor. The card in error will be the last symmetry card listed on the output. If the user attempts to read more than 24 symmetry cards, the program will return to monitor with a STOP 22 instruction. A different kind of error can occur in the bond distance calculation. If the distance limits are such that more than 1000 atom positions are generated, the program prints a diagnostic message and returns to the main program. This error does not terminate the program and ellipsoids of vibration may be calculated correctly after this condition occurs. However, bond angles will not be calculated.

G. COMMON VARIABLES AND DESCRIPTION

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XYZ(3,60)</td>
<td>Fractional coordinates of atoms in asymmetric unit</td>
</tr>
<tr>
<td>BETA(6,60)</td>
<td>Temperature factor coefficients</td>
</tr>
<tr>
<td>A(3)</td>
<td>Real cell axis lengths</td>
</tr>
<tr>
<td>ANG(3)</td>
<td>Cosines of the real cell angles</td>
</tr>
<tr>
<td>AS(3)</td>
<td>Reciprocal axis lengths</td>
</tr>
<tr>
<td>CS(3)</td>
<td>Cosines of the reciprocal cell angles</td>
</tr>
<tr>
<td>ATOM(2,60)</td>
<td>Atom identifiers</td>
</tr>
<tr>
<td>SIGA(6,6)</td>
<td>Variance-covariance matrix for real cell parameters</td>
</tr>
<tr>
<td>SIGB(6,6,60)</td>
<td>Variance-covariance matrix for thermal coefficients</td>
</tr>
<tr>
<td>RS(3,3,48)</td>
<td>Rotational matrix part of symmetry operator</td>
</tr>
<tr>
<td>TS(3,48)</td>
<td>Translational part of symmetry operator</td>
</tr>
<tr>
<td>TP(4,3)</td>
<td>Location of lattice points in fractional coordinates</td>
</tr>
<tr>
<td>ITF(60)</td>
<td>Temperature factor type indicator</td>
</tr>
<tr>
<td>CORR(16290)</td>
<td>Variance-covariance matrix for atomic positions.</td>
</tr>
<tr>
<td></td>
<td>This is stored in packed form.</td>
</tr>
<tr>
<td>XYZP(3,1000)</td>
<td>Fractional coordinates of atoms generated in the bond distance subprogram.</td>
</tr>
<tr>
<td>IATP(1000)</td>
<td>Atom identifier of atoms generated in bond distance calculation.</td>
</tr>
<tr>
<td>NA</td>
<td>Number of atoms in asymmetric unit</td>
</tr>
<tr>
<td>NSS</td>
<td>Number of symmetry operations</td>
</tr>
<tr>
<td>VOL</td>
<td>Volume of real space unit cell</td>
</tr>
<tr>
<td>DLIMIT(3)</td>
<td>Array of bond distance limits</td>
</tr>
<tr>
<td>PI</td>
<td>The value of the constant ( \pi )</td>
</tr>
<tr>
<td>G(3,3)</td>
<td>Real space metric tensor</td>
</tr>
<tr>
<td>NLP</td>
<td>Number of lattice points in the unit cell</td>
</tr>
<tr>
<td>IERR</td>
<td>Error calculation indicator</td>
</tr>
<tr>
<td>TITLE(18)</td>
<td>Alphanumeric title information</td>
</tr>
</tbody>
</table>
H. PROGRAM NAMES AND DESCRIPTION

1. **UMBADTEA:**
   
   This is the main program which handles the input and calls the bond distance and ellipsoid of vibration programs.

2. **BODAN(IBOD):**
   
   This subprogram calculates bond distances and errors and if the argument IBOD is equal to 2, calculates the bond angles and errors.

3. **ELVIB:**
   
   This subprogram calculates the ellipsoid of vibration parameters and associated errors.

4. **VARIAN(A,B,C,D,E,II):**
   
   This subprogram calculates the variance of a computed value where the partial derivatives of the value with respect to the cell parameters are stored in array B, the partial derivatives with respect to the structure parameters are stored in array C, the cell convariance matrix is stored in array D, the structure parameters covariance matrix is in array E, the argument II contains the atom number when the variance of ellipsoid parameters is calculated otherwise it contains 1 and the computed variance is output in argument A.

5. **CELLD(B,A,AN,R,CS,VOL):**
   
   This subprogram computes the partial derivatives of the reciprocal cell parameters with respect to the real cell parameters. These derivatives are used in computing the ellipsoid errors. Array B contains the partial derivatives after execution. Arrays A, AN, R and CS are the input values of the real cell axis lengths, the real cell cosines, the reciprocal axis lengths and cosines of the reciprocal cell angles respectively. The argument VOL is the input value of the real cell volume.

6. **DERDET(A,B,C):**
   
   This subprogram calculates the partial derivatives of matrix B with respect to the thermal coefficients and stores them in array A. Array C contains an intermediate set of partial derivatives.
7. CORR1(SIGZ,NN,II,JJ,CORR,KK,LL,MM):

This subprogram prepares the atomic positional correlation matrix for atoms II and JJ from the packed matrix CORR of rank MM and stores it at the KK,LL position of the array SIGZ which is of rank NN.

I. SYSTEM REQUIREMENTS

U M BADTEA requires two tape units, a standard input tape and a standard output tape. All input is BCD from Logical Unit Number 5 while output is BCD to Logical Unit Number 6. The program requires approximately 32K storage locations. If the modified ORFFE tape is to be used, it is read from logical unit 4.
Appendix C

LISTINGS OF PROGRAMS

Fortran listings of all programs and subprograms are included herein. Example runstreams, with the output they produce, are available on request from either author.
C IF (WT.EQ.0) AN 'IGNORANCE FACTOR' WILL BE READ IN FOR COMPUTING
C WEIGHTS.
C IF (WT.EQ.0) READ (11,10) W
C IF (WT.EQ.0) WRITE (11,10) W

C SIGMA WILL BE COMPUTED BY SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
C IF (WT.EQ.0) SIGMA=SUM(SIGMA**2)*F0.4,
SUBROUTINE INPUT (IEP)  INFT 10  IEP 20  INFT 30
C  THIS ROUTINE REVISED  MARCH 10, 1970
C  SUBROUTINE TS READ STRUCTURE FACTOR CARDS, COMPUTE SCATTERING FACTORS
C  -SIGN FACTORS AND WEIGHTS
DATA RAD,0.74532/
DIMENSION HK(3)
INCLUDE DCMLIST
COMMON/E/DMAX,DM,NRECV(6),EXANG(3)
COMMON/P/DELTA,POH,POMAX,PMEN,PEP,MEP,MEB,MEJ,NREB,NRMB(3)
COMMON/K/HUPP(1023),FCURVE(20),PSTRU(18),IT,WSCAT,SCPAT,WHMTH
1,WTAC
INCLUDE TAP LIST
DATA LEN/22/
06 IT(1,2,3),IEF
1 READ INP,PSTRU,END=120)INP,POBS,PELMA,1,IEP,MCB,EXOS,TTB,OMEG,CEI:IEP=200
1,IEP
C  CALCULATE ANGLES FOR ANISOTROPIC EXTINCTION
EXANG(1)*CHI+RAD
EXANG(2)*POH+RAD
EXANG(3)*RAD(#MHO Paper 0.5*TTH)
IPM IE,NO.0)NO TO 120
06 IT 10
C  WEIGHT ON CARD
IPM IE,NO.0)NO TO 120
1 READ INP,PSTRU,END=120)INP,MEJ,1,POBS,EXOS,SIOMA
IPM IE,NO.0)NO TO 120
06 IT 10
C  WEIGHT ON CARD
IPM IE,NO.0)NO TO 120
1 READ INP,PSTRU,END=120)INP,MEJ,1,POBS,EXOS,SIOMA
IPM IE,NO.0)NO TO 120
06 IT 10
C  INTEGRATE "IOMGBCHANGE FACTOR,*
R5 IPM (MEJ,NO.0)XIP(SORT(WT**2)XWTAC*PERS)**2
C  SET SCALE FACTOR PDB REFLECTION
90 IPM I)100,110,100
150 IESC+1
RETURN
110 IESC+1
RETURN
120 IEP+1
RETURN
END
SCAT(L)*FCURVE(4,L)*ANOM(1,L)
40 SCAT(L)*SCAT(L)*FCURVE(4,L)*EXP(-FCURVE(4,L)*RNO)
WT*SIGMA
C  COMPUTE SIGMA(s) FROM INFORMATION ON CARD
06 IT (90,50,60,90,80,56),IT
C  SIGMA(P+2) ON CARDS
50 WT*0.5*SIGMA/POBS
06 IT 90
C  WEIGHT ON CARD
60 WT=1.0/SOMIT(2)
IF(INEB)700,70,70
WT=2.0*POBS*WT
06 IT 90
C  USE SPECIAL ROUTINE
80 CALL WEIGHT
IPM(4)100,50,60
C  INTEGRATE "IOMGBCHANGE FACTOR,*
R5 IPM (MEJ,NO.0)XIP(SORT(WT**2)XWTAC*POHS)**2
C  SET SCALE FACTOR PDB REFLECTION
90 IPM I)100,110,100
150 IESC+1
RETURN
110 IESC+1
RETURN
120 IEP+1
RETURN
END
SCAT(L)*FCURVE(4,L)*ANOM(1,L)
40 SCAT(L)*SCAT(L)*FCURVE(4,L)*EXP(-FCURVE(4,L)*RNO)
WT*SIGMA
C  COMPUTE SIGMA(s) FROM INFORMATION ON CARD
06 IT (90,50,60,90,80,56),IT
C  SIGMA(P+2) ON CARDS
50 WT*0.5*SIGMA/POBS
06 IT 90
C  WEIGHT ON CARD
60 WT=1.0/SOMIT(2)
IF(INEB)700,70,70
WT=2.0*POBS*WT
06 IT 90
C  USE SPECIAL ROUTINE
80 CALL WEIGHT
IPM(4)100,50,60
C  INTEGRATE "IOMGBCHANGE FACTOR,*
R5 IPM (MEJ,NO.0)XIP(SORT(WT**2)XWTAC*POHS)**2
C  SET SCALE FACTOR PDB REFLECTION
90 IPM I)100,110,100
150 IESC+1
RETURN
110 IESC+1
RETURN
120 IEP+1
RETURN
END
SCAT(L)*FCURVE(4,L)*ANOM(1,L)
40 SCAT(L)*SCAT(L)*FCURVE(4,L)*EXP(-FCURVE(4,L)*RNO)
WT*SIGMA
C  COMPUTE SIGMA(s) FROM INFORMATION ON CARD
06 IT (90,50,60,90,80,56),IT
C  SIGMA(P+2) ON CARDS
50 WT*0.5*SIGMA/POBS
06 IT 90
C  WEIGHT ON CARD
60 WT=1.0/SOMIT(2)
IF(INEB)700,70,70
WT=2.0*POBS*WT
06 IT 90
C  USE SPECIAL ROUTINE
80 CALL WEIGHT
IPM(4)100,50,60
C  INTEGRATE "IOMGBCHANGE FACTOR,*
R5 IPM (MEJ,NO.0)XIP(SORT(WT**2)XWTAC*POHS)**2
C  SET SCALE FACTOR PDB REFLECTION
90 IPM I)100,110,100
150 IESC+1
RETURN
110 IESC+1
RETURN
120 IEP+1
RETURN
END
SCAT(L)*FCURVE(4,L)*ANOM(1,L)
40 SCAT(L)*SCAT(L)*FCURVE(4,L)*EXP(-FCURVE(4,L)*RNO)
WT*SIGMA
C  COMPUTE SIGMA(s) FROM INFORMATION ON CARD
06 IT (90,50,60,90,80,56),IT
C  SIGMA(P+2) ON CARDS
50 WT*0.5*SIGMA/POBS
06 IT 90
C  WEIGHT ON CARD
60 WT=1.0/SOMIT(2)
IF(INEB)700,70,70
WT=2.0*POBS*WT
06 IT 90
C  USE SPECIAL ROUTINE
80 CALL WEIGHT
IPM(4)100,50,60
C ROUTINE TO DETERMINE THE PROPER VALUES OF CRYSTALLOGRAPHIC

PARAMETERS THAT ARE FUNCTIONS OF OTHER PARAMETERS.

DEFINE 11(x,FLD0,9,8)
DEFINE 12(x,FLD0,9,8)
DEFINE 13(x,FLD0,9,8)

INCLUDE ASCD,LIST
INCLUDE DCDW,LIST
INCLUDE XPAR,LIST
INCLUDE LSMDAT,LIST
INCLUDE INCLUDE,LIST

LDECN=1+NDENV
IPERA=1

C ARE THERE ANY SPECIAL PARAMETERS THAT ARE THEMSELVES DEPENDENT
C (AS IN NON-CRYSTALLOGRAPHIC SYMMETRY, ISOTROPIC TRANSFORMATION OR
C ANISOTROPY, ETC.), IF THESE DEPENDENCIES MUST BE SIMPLE MULTIPLIERS.

DO 10 1=1,NDENV

10 IF(11(1),NDENV,9)+10 TO 90
IF(12(1),NDENV,9)+10 TO 90
IF(13(1),NDENV,9)+10 TO 90

PPAR=PAR(1,1)
PPAR=PPAR(1,1)

C IF DEPENDENT PARAMETER IS 0,0, INITIALIZE TO CONSTANT.
KK=1

P11=1+10NDENV
P12=1+10NDENV
P13=1+10NDENV
P14=1+10NDENV

C RECURSIVE CALLS TO OTHER PARAMETER DEFINITIONS

50 CONTINUE
90 CONTINUE

CPPAR=P11
CPPAR=P12
CPPAR=P13
CPPAR=P14

C COMPUTE NEW VALUES OF PARAMETERS FOR ATOM NA.

C IS THIS PARAMETER DEPENDENT ON ANY OTHERS?
DO 500 K=1,NDENV

500 CONTINUE

C CONTINUE

C IS PARAMETER INDEPENDENT?

C IF(12(K),PEPAR,5)+10 TO 900
C IF(13(K),PEPAR,5)+10 TO 900
C IF(14(K),PEPAR,5)+10 TO 900

C DEPENDENT PARAMETER, SET IT.

IPARAPARJ=1
PPAR=PPAR+1

C INDEPENDENT PARAMETER SPECIAL, SET ALL DEPENDENT PARAMETERS FOR THIS
C ATOM WHICH DEPEND ON THIS SET OF SPECIAL PARAMETERS.

C DEPENDENT PARAMETER IS A LINEAR FUNCTION OF SPECIAL PARAMETERS.

C
IP(134 KEDGN(NK)),H,N,0,0,144 KEDGN(NK)),H,N,0,144 TO 210D
IP(I11 KEDGN(NK)),H,N,0,0,114 TO 210D
K142 KEDGN(NK))
TEMP=TEMP+PAR(NK)+N*1)

2100 CONTINUE
GO TO 170
2200 WRITE(16,9200)
9200 FORMAT('falUTY DEPENDENCY INFORMATION, FOR ABDUCTION')
C GET SET DESIGNAUnS, AND CLEAR ALL ARRAYS.
210 SET*(11,IPAM(NK))
DM 214 JJJ=1,3
GXYZ(IJ)*0.
GET(IJJ)*0.
UM(IJJ)*0.
DM 214 JJJ=1,3
TII,JJJ=0.
LL(IJJ,JJJ)=0.
214 SHJ,JJJ=0.
C FILL IN VALUES FROM THIS SET.
DM 220 IJJ,NPAR
IP(11 KEDGN(NK)),H,N,0,0,114,IPAM(NK),PAR(NK),NA)
220 CONTINUE
C DO WE NEED TO TRANSFORM XYZ FROM CRYSTAL TO SPECIAL ANGLE 17?
C IF NO XYZ IS MENTIONED AS A DEPENDENT PARAMETER,
OM 230 IJJ,11,LOGON,2
JJ=134 KEDGN(NK))
K=114 KEDGN(NK))
L=L1=114 KEDGN(NK))
PFF,0,0,NO,ANG,0,E,2,AND,0,E,4,AND,0,E,0,0,0,0 TO 240
230 CONTINUE
C XYZ DEPENDENT ON SPECIAL PARAMETER, COMPUTE THEN.
OM 240 IJJ,11,LOGON,2
K=114 KEDGN(NK))
L=L1=114 KEDGN(NK))
P(144 KEDGN(NK)),H,N,0,0,144 TO 240
240 CONTINUE
C XYZ DEPENDENT ON SPECIAL PARAMETER, COMPUTE THEN.
OM 240 IJJ,11,LOGON,2
K=114 KEDGN(NK))
L=L1=114 KEDGN(NK))
P(144 KEDGN(NK)),H,N,0,0,144 TO 240
240 CONTINUE
C XYZ DEPENDENT ON SPECIAL PARAMETER, COMPUTE THEN.
C Elements of T Tensor, Only Retas Involved, K

400 IDENT=IDENT=9
JK=IND[1, IDEN]
JL=IND[2, IDEN]
DM 410 L=1,6
JW=IND[1,1]
JN=IND[2,1]
410 BTM[1] . DZKTY(JM,JN,JK,JL)
CALL TPNZT(TMP, TMT, TSM)
420 KE=1(KDGN (1,1))
IFK=GT[1]G0 TO 999
425 KE=1(KDGN (1,1))
IFK=J350, 350,999
C Elements of L Tensor, Retas and Gammas Involved, K

500 IDENT=IDENT=15
JK=IND[1, IDEN]
JL=IND[2, IDEN]
CALL CLN(LM, TMT, EL, SMTZ)
IFNPNW[GT], G0 TO 550
DM 510 L=1,6
JW=IND[1,1]
JN=IND[2,1]
510 BTM[1] . DZKTY(JM, EL, SJ, JN, JK, JL)
CALL TPNZT(TMP, TMT, TSM)
520 KE=1(KDGN (1,1))
IFK=GT[1]G0 TO 550
IFK=LT, G0 TO 999
AAK[1] . DZKTY(JM, JN, JK, JL)
DKV=160
575 K=1
580 IFJ=AJ350, 350, 999
585 JK=IND[1,1]
JN=IND[2,1]
590 GTM[1]G0 TO 999
CALL TPNZT(TMP, TMT, TSM)
595 KE=1(KDGN (1,1))
IFK=GT[1]G0 TO 999
AAK[1] . DZKTY(JM, JN, JK, JL)

1. IF I<KDEGN(1),NE,0,300 TO 2000
   J=I*KDEGN(1)
   I=I+1
   IF I<KDEGN(1)
   D=1000  J=1, K=1, NODGN, 2
   IF I<I(KDEGN(1)),NE,0,121 (KDEGN(J),NE,1,1)GO TO 2000
   K=1
   L=1
   I=1440
   NF=1
   IF I<I(KDEGN(1)),NE,0,300 TO 1500
   IF I<I(KDEGN(1)),NE,0,121 (KDEGN(I),NE,4,5)GO TO 1440
   IF I(I(KDEGN(1)),NE,0,AND,12 (KDEGN(J),NE,1,1)KDEGN(J)=0
   1400 CONTINUE
   1500 CONTINUE
   2000 D=2000  J=1, K=1, NODGN, 2
   IF I<I(KDEGN(1)),NE,0,300 TO 2001
   IF I<I(KDEGN(1)),NE,0,121 (KDEGN(J),NE,1,1)GO TO 2000
   2900 CONTINUE
   2990 CONTINUE
   C HOW ELIMINATE ALL DEPENDENT SPECIAL PARAMETERS, RESET IF NECESSARY,
   C SAVY, AND PACK THE LIST.
   C ALSO ELIMINATE INITIALIZING CONSTANTS.
   3001 D=2000  J=1, K=1, NODGN, 2
   IF I<I(KDEGN(1)),NE,0,300 TO 21D1
   IF I<I(KDEGN(1)),NE,0,AND,12 (KDEGN(J),NE,0,0)GO TO 21D0
   21D0 NF=2
   21D1 CONTINUE
   21D2 CONTINUE
   C REPLACE INDEPENDENT PARAMETER DESIGNATIONS BY THEIR MATRIX POINTERS
   JPAK=0
   JATYM=0
   D=2400  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   J=I2 (KDEGN(I)
   K=2*INPAK(J)
   G0 TO 2350
   2320 IF I<I(KDEGN(1)),EQ,JATYM,1 TO 2330
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
   D=2350  J=1, K=1, NODGN, 2
   K=I*KDEGN(1)
   K=I*KDEGN(1)
   IF I<I(KDEGN(1)),NE,0,300 TO 2320
   IF JATYM,1,121 J=1,2,3,4,5
   IF (150) JATYM,1,121 J=1,2,3,4,5
   JPAK=JPAK+5
C IF IFSGN.EQ.0.JG0 TO 396
C EVALUATE STRUCTURE FACTOR FOR LAMANTHORNE ORIENTATION
AMINUS=TEMP**4*(TEMP**AING+TEMP**88)+AMINUS
BMINUS=TEMP**4*(TEMP**B88+TEMP**88)+BMINUS
C EVALUATE INDIVIDUAL ATOM CONTRIBUTIONS
B=401(TEMP**4)*(TEMP**AING+TEMP**88)B
B=401(TEMP**4)*(TEMP**B88+TEMP**88)B
ACALC=ACALC+B
BCALC=BCALC+B
C MODIFY DERIVATIVES
C 430 TEMPS=TWLP*TEMP4
SUM=SUM+TEMP*SUM
CAO=AOXIAA+AOXIAA+AOXIAA+AOXIAA
C DERIVATIVE WITH RESPECT TO ISOTROPIC TEMPERATURE FACTOR
C 420 DAOOB(LJ)=K*AM0NIO1
C DERIVATIVES WITH RESPECT TO ANISOTROPIC TEMPERATURE FACTORS
C 430 DAOOB(LJ)=TEMP4*(TEMP+OAIH)+TEMP4*(TEMP+OAIH)
C 440 DBOOB(LJ)=TEMP4*(TEMP+OAIH)+TEMP4*(TEMP+OAIH)
C 450 CONTINUE
C 470 MSELS=SAVE
RETURN
C CALCULATE EXTINCTION FACTOR
C IF (EXT1<1.D0 OR EXT1>1.D0)
C 1000 EXE=EXT1*EXT1
C 1010 SGH=SGH+SGH
C 1020 VI1=VI1+VI1
C 1030 V12=V12+V12
C 1040 VI3=VI3+VI3
C 1050 VI4=VI4+VI4
C 1060 SUM=SUM+SUM
C 1070 RETURN
C C CALCULATE DERIVATIVES OF THE FORM OF-SIGME-1
C 1080 SUM=SUM+SUM
C 1090 EXE=EXE*EXE*EXE*EXE
C 1100 SGH=SGH+SGH
C 1110 VI1=VI1+VI1
C 1120 V12=V12+V12
C 1130 VI3=VI3+VI3
C 1140 VI4=VI4+VI4
C 1150 SUM=SUM+SUM
C 1160 RETURN
C C ANISOTROPIC EXTINCTION - CALCULATE SIG FUNCTIONS OF ANGLES
C 1170 SIGH=SIGH+SIGH
C 1180 VI1=VI1+VI1
C 1190 V12=V12+V12
C 1200 VI3=VI3+VI3
C 1210 RETURN
C C TYPE 2 ANISOTROPIC
C 1220 SNU=SIGME(SME)
C 1230 CONTINUE
C 1240 MSELS=SAVE
RETURN
C C FOR PRODUCTS OF COEFFICIENTS
1040 UN 1050 =13
1050 V11=V11+C1PHI+2*C1PHI+C1PHI+C1
V12=V12+C1PHI+C1PHI+C1PHI+C1
V13=V13+C1PHI+C1PHI+C1PHI+C1
V14=V14+C1PHI+C1PHI+C1PHI+C1
RETURN
END
PROGRAM H(9) INPUT 1,2,A,B,M
15 PRINT "H(9)=(1/SQRT(16+16*M*X*M))" RETURN
}
**SUBROUTINE DWM(A,B,C)**

MULTIPLY 2 3X3 MATRICES TOGETHER.

DIMENSION A(I,J),B(I,J),C(I,J),D(I,J,E(3))

DIM 10 I=1,3
DIM 10 J=1,3
C(I,J)=0,

ENTRY DWM(A,B,C)

10 C(I,J)=C(I,J)+A(I,K)*B(K,J)

RETURN

**FUNCTION DWM(A,B,C)**

**SUBROUTINE EULR(S,T,A)***

**SUBROUTINE TFW2(T,A,N)**

INCLUDE IND2DLIST

DIMENSION A(I,J),H(0, 1, 2), I(1, 2), T(2, 3), B(3, 3), C(J, J, I), D(J, J, I), E(J, J, I)

DATA P1, 3, 4, 0, 9260/0

DO 40 1=1,3

TFW 10

TFW 20

TFW 30

TFW 40

TFW 50

TFW 60

TFW 70

TFW 80

TFW 90

TFW 100

TFW 110

TFW 120

TFW 130

TFW 140

TFW 150

TFW 160

TFW 170

TFW 180

TFW 190

TFW 200

TFW 210

TFW 220

TFW 230

TFW 240

TFW 250

TFW 260

TFW 270

TFW 280

TFW 290

TFW 300

TFW 310

TFW 320

TFW 330

TFW 340

TFW 350

TFW 360

TFW 370

TFW 380

TFW 390

TFW 400

END
FUNCTION DJE(LM, R, J, L, E, L, L, W)
DIMENSION H(3)
IN = DJE(LM, L, E)
IF(WOD(IN, 6), EO, AND, IN.LE.18) IOD TO 1080
IF(IN, EO, 60, EO, EO, 60, IN, EO, 60, 36) IOD TO 1060
IF(JL, NE, EI, EI, EI, EI, EI, NE, EI, NE, H, 1095 TO 1400
DIV = D
GO TO 1100
1040 DIV = 4,
GO TO 1100
1060 DIV = 6,
GO TO 1100
1080 DIV = 12,
1100 IFO(WOD(IN, 3), H, 0) IOD TO 1500
1125 IFO(WOD(IN, 2), H, 0) IOD TO 1500
1150 J = 1
1250 IFO(WOD(IN, 2), H, 0) IOD TO 1375
1375 L = 1
GO TO 1150
1500 IFO(WOD(IN, 2), H, 0) IOD TO 1750
1750 M = 1
GO TO 1375
1900 GO TO (100, 250, 300), I
1100 GO TO (110, 150, 2000), J
1200 GO TO (125, 125, 2000), L
1250 GO TO (1000, 4000, 2000), L
1300 IF(P = 2) IOD 2000, 2000
1400 IF(P = 2) IOD 2000, 2000
1450 IF(P = 2) IOD 2000, 2000
1500 IF(P = 2) IOD 2000, 2000
1600 IF(P = 2) IOD 2000, 2000
1700 IF(P = 2) IOD 2000, 2000
200 GO TO (210, 250, 2000), J
DIJE 10
210 GO TO (220, 240, 1000), X
DIJE 20
220 GO TO (225, 230, 5000), L
DIJE 30
225 GO TO (2000, 3000, 1000), W
DIJE 40
230 IF(N = 2) IOD 2000, 2000
DIJE 50
240 IF(N = 2) IOD 2000, 2000
DIJE 60
245 IF(N = 2) IOD 2000, 2000
DIJE 70
250 IF(N = 2) IOD 2000, 2000
DIJE 80
260 IF(N = 2) IOD 2000, 2000
DIJE 90
270 IF(N = 2) IOD 2000, 2000
DIJE 100
300 GO TO (310, 350, 1000), J
DIJE 110
310 GO TO (320, 340, 4000), X
DIJE 120
320 GO TO (325, 330, 300), L
DIJE 130
325 GO TO (3000, 3000, 1000), W
DIJE 140
330 IF(N = 2) IOD 2000, 2000
DIJE 150
340 IF(N = 2) IOD 2000, 2000
DIJE 160
345 IF(N = 2) IOD 2000, 2000
DIJE 170
350 IF(N = 2) IOD 2000, 2000
DIJE 180
360 IF(N = 2) IOD 2000, 2000
DIJE 190
370 IF(N = 2) IOD 2000, 2000
DIJE 200
1000 DJE(LM + 1),
RETURN
DIJE 220
2000 FAC = 1,
DIJE 230
2010 DJE(LM + FAC * R(I) / (24, * DIV))
DIJE 240
RETURN
DIJE 250
3000 W = W + (1, 1, 3) * 1
DIJE 260
3010 DJE(LM + R(W) / (24, * DIV))
DIJE 270
RETURN
DIJE 280
4000 N = N + (1, 3) * 1
DIJE 290
GO TO 3010
DIJE 300
5000 FAC = 2,
DIJE 310
GO TO 2010
END
FUNCTION DJE(LM, R, J, L, E, L, W)
DIMENSION H(3)
PIJE(LM + 12, * (J, E, L, W))
RETURN
END
GO TO 410
330 CONTINUE
C 1 NEW ATOM, COMPUTE DISTANCE
S=0.0
DO 340 I=1,3
340 TPEXZ(K,11)=TPEXZ(K,11)*S
DO 350 N=1,3
350 EXP=EXP+TPEXZ(K,11)
DO 360 N=1,3
360 EXP=EXP+EXPXZ(K,11)
IF(K-KK)IL(K,NN)=1,410,410,370
370 END (K)
C 2 CHECK IF LIST ARRAYS FULL
LIST=300,300,380
GO TO 410
C 3 ADD ATOM TO LIST
390 LIST=LIST+1
400 LIST=LST,*64+J+J
DO 400 I=1,3
410 EXPXZ(K,LUST)=EXPXZ(K,LUST)+1
420 WRITE10,6(HS(I,J),I=1,2),EXPXZ,K,LIST
430 CONTINUE
440 RETURN
C 4 COMPUTE BOND ANGLES
450 WRITE10,10,TITLE
I=LIST-1
C 5 LOOPS THROUGH ATOMS
DO 460 I=1,11,1
WRITE10,12(HS(I,J),I=1,2),(1(1,J),J=1,3)
C 6 FIRST LOOP THROUGH LIST
DO 470 J=1,11,1
470 EXP=EXPXZ(K,LUST)+EXPXZ(K,LUST)
DO 480 N=1,3
480 EXP=EXP+EXPXZ(K,LUST)
DO 490 N=1,3
490 EXP=EXPXZ(K,LUST)+EXPXZ(K,LUST)
IF(K-KK)IL(K,NN)=1,410,410,370
440 RETURN
C 7 SECOND LOOP THROUGH LIST
DO 500 I=1,11
500 EXP=EXP+EXPXZ(K,LUST)
DO 510 N=1,3
510 EXP=EXPXZ(K,LUST)+EXPXZ(K,LUST)
DO 520 N=1,3
520 EXP=EXP+EXPXZ(K,LUST)
530 EXP=EXPXZ(K,LUST)+EXPXZ(K,LUST)
```plaintext
9220 FORMAT(1X,M4+3,F10.4))
F1=5
IP:1,E,NDEON 160 TO 210
RETURN
END

SUBROUTINE WEIGHT
C THIS ROUTINE REVISED AND OR RESEQUENCED ON OCT. 28, 1968
C
DUWAY ROUTINE
RETURN
ENTRY REJECT
C
DUWAY ROUTINE
RETURN
D U T Y 1 0 0

I PAR

PRGC
C M B N W/ A X A R/N F A E , A T X I ( 3 , 3 ) , A T X ( 3 , 3 ) , P A M ( 2 , 5 0 ) , P A R N A M ( 2 , 5 0 ) , I P A M ( 5 0 )
1 , I N P A M ( 5 0 ) , I X T ( 3 ) , S Y T ( 3 , 3 ) , S Y T ( 3 ) , T T L ( 3 , 3 ) , T T L ( 3 , 3 ) , T T L ( 3 , 3 ) , T T L ( 3 , 3 )
2 , T T L ( 3 , 3 ) , T T L ( 3 , 3 ) , T T L ( 3 , 3 ) , T T L ( 3 , 3 ) , T T L ( 3 , 3 ) , T T L ( 3 , 3 ) , T T L ( 3 , 3 )
T A P E

PRGC
L SH A T

PRGC
C M B N W/ L S H A T / W 2 0 0 , I S T X , I P A R M 1 0 0 0 , I N P A K , K E T C L , L A B E L 1 1 , JS C R E ( 1 0
1 , J S C R E ( 1 0 )
A C M

PRGC
C M B N W/ A C T A ( 6 , 6 0 ) , G I ( 3 , 3 ) , I B I T ( 6 0 ) , N A T O N , T A H I ( 2 , 6 0 ) , T I T L E ( 1 8 ) , A T X
1 3 , G O P , R I ( 3 , 6 0 ) , O M ( 1 5 , 6 0 )
D O M

PRGC
C M B N W/ O A C A L C , A I N G ( 6 0 ) , A N U M ( 2 , 2 0 ) , I C A L C , B I N M ( 6 0 ) , D A M ( 6 , 6 0 ) , D A M
1 6 0 , D A R X ( 3 , 6 0 ) , D A R M ( 6 , 6 0 ) , D A R M ( 3 , 6 0 ) , D A R M ( 6 , 6 0 ) , D A R M ( 3 , 6 0 ) , D A R M ( 6 , 6 0 )
3 , D A R M ( 6 0 ) , D S H I ( 1 0 , 6 9 ) , D A R M ( 1 5 , 6 0 ) , D A R M ( 3 0 , 6 0 ) , A N I N C , A M I N S
A N D ( 3 0 , 6 0 ) , D A R M ( 2 5 6 0 ) , A N D ( 6 4 0 0 )
D M I S S I O N SAV/I/D/DEGEN ( 2 6 4 0 ) , M A 2 6 4 0 0
MO/I/N F N C : ( S A V E , A I N D ) ( E D K O N , A A , G A D C ( 1 , 3 7 )
I N D 2 3

PRGC
C M B N W/ I N D 2 / I N D 2 ( 2 , 6 0 ) , I N D ( 3 , 1 0 )
END

ROUTE TO ADJUST THE SIZE OF THE MEMORY FIELD OF AN ACTIVE PROGRAM
CALLING SEQUENCE
'ARRAY' MUST BE AT THE END OF THE 'IF' OR 'TO' BANK OF THE PROGRAM.
N IS THE LARGEST INDEX REQUIRED BY THE PROGRAM.
AXRZ

S(1)
CHWZ= L A O , O , X I I - GET FINAL ADDRESS
A O , L A S T - SAME SIZE AS BEFORE?
J X II
A I , L A S T - YES = RETURN.
S A D , L A S T - NEED MORE CORR?
J C R Z 1
F R L O M E S - YES
J 2 , X I I - NO, REDUCE NEEDS.
C R Z 1 - GET MORE.
F R L O M E S - RETURN.
J 2 , X I I - *0
LAST
END
```
210 FORMAT(12X,A1,7X,I10,P5.5)
211 FORMAT(20X,I10,P5.5)

FPF 3=1/3,1/3,1/3
FPF 4=2/4,2/4,2/4
FPF 7=1/7,1/7,1/7
PI=ACOS(-1.0)

READ IN, TITLE
WRITE OUT, TITLE
READ IN, ANG
C
C CALCULATE SINES AND COSINES OP DIRECT SPACE ANGLES
DO 4 I=1,3
IN(CASS ANG(I))=90.0,60.0,30.0,0.0 TO 1000
S(I)=1.0
ANG(1)=90.0
IN 4 5 6

1000 ANG(1)*ANG(1)/PI/180.0
SE(I)=SIN(ANG(I))
ANG(I)=CHAND(ANG(I))
C
C MATRICES I, G IS THE REAL SPACE METRIC TENSOR
4 G(I,1)=A(I)**2
DO 144 I=1,3
144 G(I,1)=G(I,3)
DO 146 J=1,3
146 G(I,J)=A(I)*A(J)*ANG(I)*ANG(J)
C
C COMPUTE REAL SPACE CELL VOLUME
VOL=(1.0-ANG(2)*ANG(3))**2*GRT(1,1)**2-ANG(2)**2-ANG(3)**2
C
C COMPUTER RECIPROCAL SPACE CELL PARAMETERS
AS(1)=A(2)/3/3/3/3/VOL
AS(2)=A(3)/3/3/3/3/VOL
AS(3)=A(1)/3/3/3/3/VOL
C
C SG(1)=ANG(2)+ANG(3)+ANG(1)/2/2/2
C
C SIG(1)=ANG(2)+ANG(3)/2/2/2
C
C SIG2=ANG(2)+ANG(3)/2/2/2
C
C SIG3=ANG(2)/2/2/2
C
C W I T H I N T, SPACEL I T E R, ANG, AS, CD, VOL
C
C READ CONTROL CARD
READ IN,5 NA, I C H T, IM B, I E F, I P T Y P E, D I M I T
IP(NDIMIT)=1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
C
C WRITE OUT, ONLY I P T Y P E, J C E L L (1,1)
C
C READ SYMMETRY CARDS
DO 7 I=1,6
READ IN,6 MTR X, Y, Z, IMPS X, Y, Z, I E F
C
C WRITE OUT, ONLY I P T Y P E, I M P S X, Y, Z
C
C CLEAR ROTATIONAL MATRICES
DO 13 I=1,3
DO 10 J=1,3
10 R(S,I,J)=0.0
C
C CONVERT TRANSLATIONAL OPERATOR
DO 11 I=1,3
IP(I,Y)=-IP(I,X)
C
C I N T I N U I T E
C
C CARD CHARACTERS ARE NOT IN THE ACCEPTABLE LIST
STOP 11
190  IJI=1K+1
K=INDEX(JM(JA,JA))

200  DO 210  KJ,J6
   IJI=IJI+1
   JCR=IPAR(A(IJI))
   IF(IJR)EQ.210,B200
   SIGD(JJ,1)=C(N(VK))
   SIGD(JJ,1)=C(N(VK))
   CONTINUE
210  CONTINUE
220  CONTINUE
230  IJI=EP(I)
232  IL*IR*KHNT(IJ=1)*I
C  MODIFIER SGR IF NECESSARY
IP=MODON(E,0.0) 00 TO 239
C  LORP THROUGH DEPENDENCIES
DO 235  I=1,MODON(E,2)
IPAR=IPAR(CNH(1))
IF(IPAR.GT.I+1000) 1000 235
IA=1(IPAR)
IA=1(IPAR)
CALL CRH(1)
DO 235  I=1,MODON(E,2)
IPAR=IPAR(CNH(1))
1000 CONTINUE
C  DEP. PARAMETER IS AN AMI, CEM.
C  TEMPERATURE FACTOR MATRIX FOR THE SAME ATOM IF SO. THE
C  DEPENDENCY PARAMETER IS A MEMBER OF THE ANISOTROPIC
C  COVARIANCE OF THE DEPENDENCY PARAMETER WITH THE DEPENDENT
C  ONE MUST BE COMPUTED.
DO 1010  JI=1,6
JI=IPAR(IJ,1)
IPAR2(IPAR,1)=IPAR2(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)
C  CORR. COVARIANCE OF THE DEPENDENCY PARAMETER WITH THE DEPENDENT
C  ONE MUST BE COMPUTED.
DO 1010  JI=1,6
JI=IPAR(IJ,1)
IPAR2(IPAR,1)=IPAR2(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)
1010  CONTINUE
C  COMPUTE COVARIANCE OF DEPENDENT PARAMETERS WITH EACH OTHER.
1011  DO 1500  I=1,MODON(E,2)
IPAR3(IPAR,1)=IPAR(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)+IPAR(IPAR,1)
1500  CONTINUE
235  CONTINUE
235  LATT=-Z
235  CONTINUE
C  SKIP NEXT SECTION IF NO BOND FUNCTIONS
IP=MODON(EQ.0) 00 TO 750
C  START UNPACKING POSITION MATRIX
IPAR=IPAR2(IPAR)+I
C  EA WILL BE THE INDEX OF THE NEXT LOCATION IN THE UNPACKED ARRAY.
C  INCR IS THE STEP NEEDED TO ADVANCE EA AN ENTIRE ROW
C  EA WILL INDEX THE NEXT LOCATION IN THE PACKED MATRIX
EA=EA+3*EA+1
LATT=3*EA+1/Z2
380 EA=EA1
390 DO 410 L=1,3
EC=ECJP
JP=JP1
KEPAP=KEPA1
IP(IPAR,KEPAP)=400,410,400
410 CONTINUE
L=ITP(E)
NB=CNCH(L,1)
C CONTINUE
420 DO 430 EA=EA1
IP(IPAR,EA1)=440,460,460
C ADVANCE COUNTER
430 EA=EA1
440 CONTINUE
450 COUNTS
470 CONTINUE
C LOOP THROUGH TEMPERATURE FACTORS
DO 440 D=460 L=1,48
KEPAP=KEPA1
IP(IPAR,KEPAP)=440,460,460
C ADVANCE COUNTER
450 EA=EA1
460 CONTINUE
470 CONTINUE
C LOOP THROUGH VARIOUS TEMPERATURE FACTORS
DO 530 J=1,981
IPAR=IPAR1
IP(IPAR,IPAR1)=520,530,520
C ADVANCE INDEX AND INCREASE INCREMENT
520 EA=EA+INCR
INCR=INCR-1
530 CONTINUE
540 CONTINUE
C ATOMIC VARIANCE-COVARIANCE MATRIX IS EXTRACTED, NOT MODIFIED FOR
C COUPLING
IP(MDGN,EQ,0) GO TO 570
NB=NB1A
DO 550 I=1,MDGN,2
IPAR=14(CNSTM(I,1))
IP(IPAR,LT,0,0,IPAR,GT,1,0,0) GO TO 555
IAT=14(CNSTM(I,1))
IF(IPAR*2=IAT-1) IPAR-
DO 550 E=1,981
IP(IPAR,IPAR,IPAR1) GO TO 560
C COMPUTE COVARIANCE OF POSITION WITH A REPIN2 POSITION.
EC=INDEXEK(IPAR)
EA=INDEXED(J,JPMEK)
CV(EK)=CV(EK)+CV(EA)+CV(DIM(1)+1)
550 CONTINUE
C COMPUTE COVARIANCE OF UNREPIN2 POSITION WITH OTHER.
DO 600 E=1,MDGN,2
IPAR=14(CNSTM(EK))
IP(IPAR,LT,0,0,IPAR,GT,1,0,0) GO TO 560
JAT=14(CNSTM(EK))
L=14(CNSTM(EK))
IPAR*2=IAT-1 IPAR-
EA=INDEXED(J,L)
EC=INDEXEK(IPAR,IPAR,IPAR1)
CV(EK)=CV(EK)+CV(EK)+CV(EA)+CV(DIM(1)+1)+CV(EK)
560 CONTINUE
C CONTINUE
570 DB=NA+1,LETZ
SOGX(I,J,CV(EK))
580 DB=NB+1,
C CONTRAST MATRIX TO THE MINIMUM REQUIRED
790 CALL CONSUM(SOGX(LETZ))
C BREAD AND CONVERT TO CER RELAXATION MATRIX
800 READ IN,6,S(SOGX(1,1),J1,J1,1,1) *
C GET REAL SPACE CELL QUANTITIES
810 NH=1
DO 800 I=1,9
800 N(I)=SOGX(I,1,1)
C EXPAND SYMMETRY OPERATORS FOR A CRYSTAL
850 IPCICB=IP(480,850,850)
C END OF PROGRAM
920 CALL EXIT
C END OF PROGRAM
930 CONTINUE
940 CONTINUE
950 CONTINUE
960 CONTINUE
970 CONTINUE
980 CONTINUE
990 CONTINUE
IF(IBM, 0, 1) RETURN
WRITE(IOU, 9002) TITLE, (DIST(I), I=1, 2)
SET UPPER LIMIT OF DISTANCE TO CONTACT DISTANCE
SX=DIST(I)
DO 28 I=1, NA
IF(NI=1) GO TO 28
WRITE(IOU, 9010) X, Y, Z, M, X12, X13, X11, X21, X23, X31, X32, X33
IF(EB=0, 0, 0) GOTO 1006
C GET SELF VA_COV MATRIX
DO 124 N=1, 3
L=1
SIGV(N, 3, 4)=CORR(N)
IF(EQ, 1) GOTO 124
DO 25 K=2, N
L=L+1
25 SIGV(N, K+3)=CORR(N)
DO 124 J=J+1,N-1
C OUTSIDE LOOP THROUGH GENERATED ATOM LIST
1006 ITEP=1+ATOM-1
DO 29 J=1, ITEP
DO 30 N=1, 3
30 X(IPX(N))=X(N, 1)
30 Y(IPX(N))=Y(N, 1)
30 Z(IPX(N))=Z(N, 1)
C COMPUTE ANG CHECK DISTANCE
5=S
DO 31 N=1, 3
DO 32 K=1, N-1
32 X(IPX(N))=X(N, 1)
32 Y(IPX(N))=Y(N, 1)
32 Z(IPX(N))=Z(N, 1)
C COMPUTE VA_COV MATRICES
C DO 124 N=1, NA
C DO 25 K=2, N
C L=L+1
C 25 SIGV(N, K+3)=CORR(N)
C DO 124 J=J+1,N-1
C OUTSIDE LOOP THROUGH GENERATED ATOM LIST
C 1006 ITEP=1+ATOM-1
C DO 29 J=1, ITEP
C DO 30 N=1, 3
C 30 X(IPX(N))=X(N, 1)
C 30 Y(IPX(N))=Y(N, 1)
C 30 Z(IPX(N))=Z(N, 1)
C COMPUTE ANG CHECK DISTANCE
C 5=S
C DO 31 N=1, 3
C DO 32 K=1, N-1
C 32 X(IPX(N))=X(N, 1)
C 32 Y(IPX(N))=Y(N, 1)
C 32 Z(IPX(N))=Z(N, 1)
C COMPUTE VA_COV MATRICES
SUBROUTINE DERBI(A,B,C)
C THIS ROUTINE REVISED AND PRERESTORED ON OCT. 28, 1969
C
DIMENSION A(6),B(3,3),C(6,3,3)
DO 1 I=1,6
A(I)=0
DO 1 J=1,3
B(I,J)=0
DO 1 K=1,3
C(I,J,K)=0
1 A(I,J,K)=B(I,J)*B(J,K)+B(J,K)*B(K,I)
RETURN
END

SUBROUTINE VARIAN(A,B,C,D,B,II)
C THIS ROUTINE REVISED AND PRERESTORED ON OCT. 28, 1969
C
DIMENSION B(6),C(6),D(6,6),II(6,6,40)
II=II
F=II(I)=2*II(I,J)+C(I,J)=2*II(I,J)
II=II(I,J,K)=2*II(I,J,K)+C(I,J,K)=2*II(I,J,K)
RETURN
END 10
DEED 10
DEED 20
DEED 30
DEED 40
DEED 50
DEED 60
DEED 70
DEED 80
DEED 90
DEED 100
DEED 110
DEED 120
DEED 130
DEED 140
DEED 150
DEED 160
DEED 170
DEED 180
VARM 10
VARM 20
VARM 30
VARM 40
VARM 50
VARM 60
VARM 70
VARM 80
VARM 90
VARM 100
VARM 110
VARM 120
VARM 130
VARM 140
VARM 150
VARM 160
VARM 170
VARM 180
COMPILER (PLD-Q)
SUBROUTINE LIBON(XZ,TZ,PAR,EPAR,HPAR,0,0,0,NDGC,ANSTR)
LCC 10
LCC 20
LCC 30
LCC 40
LCC 50
LCC 60
LCC 70
LCC 80
LCC 90
LCC 100
LCC 110
LCC 120
LCC 130
LCC 140
LCC 150
LCC 160
LCC 170
LCC 180
LCC 190
LCC 200
LCC 210
LCC 220
LCC 230
LCC 240
LCC 250
LCC 260
LCC 270
LCC 280
LCC 290
LCC 300
LCC 310
LCC 320
LCC 330
LCC 340
LCC 350
LCC 360
LCC 370
LCC 380
LCC 390
LCC 400
LCC 410
LCC 420
LCC 430
LCC 440
LCC 450
LCC 460
LCC 470
LCC 480
LCC 490
LCC 500
LCC 510
LCC 520
LCC 530
LCC 540
LCC 550
LCC 560
LCC 570

I

II

**C**

MOVE REFLECTION DATA TO START OF BUFFER AREA

FOUR1740

00 610 I=1+1NINDEX

FOUR1750

610 FBUFL(=ARRAY(1))

FOUR1760

IPN=INDEX

FOUR1770

WRITE(OUT1,10)IPN

FOUR1780

10 1=1,INDEX

FOUR1790

XX=TOWOFTLOAT(I)

FOUR1800

YY=TOWOFTLOAT(I)

FOUR1810

ZZ=TOWOFTLOAT(I)

FOUR1820

iaux=ifix(XX)

FOUR1830

iaux+1

FOUR1840

iaux=ifix(YY)

FOUR1850

iaux+1

FOUR1860

iaux=ifix(ZZ)

FOUR1870

iaux+1

FOUR1880

iaux=iaux(iaux+iaux+iaux)

FOUR1890

iaux=iaux(iaux+iaux+iaux)

FOUR1900

iaux=iaux(iaux+iaux+iaux)

FOUR1910

iaux=iaux(iaux+iaux+iaux)

FOUR1920

iaux=iaux(iaux+iaux+iaux)

FOUR1930

iaux=iaux(iaux+iaux+iaux)

FOUR1940

iaux=iaux(iaux+iaux+iaux)

FOUR1950

iaux=iaux(iaux+iaux+iaux)

FOUR1960

iaux=iaux(iaux+iaux+iaux)

FOUR1970

iaux=iaux(iaux+iaux+iaux)

FOUR1980

iaux=iaux(iaux+iaux+iaux)

FOUR1990

iaux=iaux(iaux+iaux+iaux)

FOUR2000

iaux=iaux(iaux+iaux+iaux)

FOUR2010

iaux=iaux(iaux+iaux+iaux)

FOUR2020

iaux=iaux(iaux+iaux+iaux)

FOUR2030

iaux=iaux(iaux+iaux+iaux)

FOUR2040

iaux=iaux(iaux+iaux+iaux)

FOUR2050

iaux=iaux(iaux+iaux+iaux)

FOUR2060

iaux=iaux(iaux+iaux+iaux)

FOUR2070

iaux=iaux(iaux+iaux+iaux)

FOUR2080

iaux=iaux(iaux+iaux+iaux)

FOUR2090

iaux=iaux(iaux+iaux+iaux)

FOUR2100

iaux=iaux(iaux+iaux+iaux)

FOUR2110

iaux=iaux(iaux+iaux+iaux)

FOUR2120

iaux=iaux(iaux+iaux+iaux)

FOUR2130

iaux=iaux(iaux+iaux+iaux)

FOUR2140

iaux=iaux(iaux+iaux+iaux)

FOUR2150

iaux=iaux(iaux+iaux+iaux)

FOUR2160

iaux=iaux(iaux+iaux+iaux)

FOUR2170

iaux=iaux(iaux+iaux+iaux)

FOUR2180

iaux=iaux(iaux+iaux+iaux)

FOUR2190

iaux=iaux(iaux+iaux+iaux)

FOUR2200

iaux=iaux(iaux+iaux+iaux)

FOUR2210

iaux=iaux(iaux+iaux+iaux)

FOUR2220

iaux=iaux(iaux+iaux+iaux)

FOUR2230

iaux=iaux(iaux+iaux+iaux)

FOUR2240

iaux=iaux(iaux+iaux+iaux)

FOUR2250

iaux=iaux(iaux+iaux+iaux)

FOUR2260

iaux=iaux(iaux+iaux+iaux)

FOUR2270

iaux=iaux(iaux+iaux+iaux)

FOUR2280

iaux=iaux(iaux+iaux+iaux)

FOUR2290

iaux=iaux(iaux+iaux+iaux)

FOUR2300

iaux=iaux(iaux+iaux+iaux)

FOUR2310

iaux=iaux(iaux+iaux+iaux)

FOUR2320

iaux=iaux(iaux+iaux+iaux)

FOUR2330

iaux=iaux(iaux+iaux+iaux)

FOUR2340

iaux=iaux(iaux+iaux+iaux)

FOUR2350

iaux=iaux(iaux+iaux+iaux)

FOUR2360

iaux=iaux(iaux+iaux+iaux)

FOUR2370

iaux=iaux(iaux+iaux+iaux)

FOUR2380

iaux=iaux(iaux+iaux+iaux)

FOUR2390

iaux=iaux(iaux+iaux+iaux)

FOUR2400

iaux=iaux(iaux+iaux+iaux)

FOUR2410

iaux=iaux(iaux+iaux+iaux)

FOUR2420

iaux=iaux(iaux+iaux+iaux)

FOUR2430

iaux=iaux(iaux+iaux+iaux)

FOUR2440

iaux=iaux(iaux+iaux+iaux)

FOUR2450

iaux=iaux(iaux+iaux+iaux)

FOUR2460

iaux=iaux(iaux+iaux+iaux)

FOUR2470

iaux=iaux(iaux+iaux+iaux)

FOUR2480

iaux=iaux(iaux+iaux+iaux)

FOUR2490

iaux=iaux(iaux+iaux+iaux)

FOUR2500

iaux=iaux(iaux+iaux+iaux)

FOUR2510

iaux=iaux(iaux+iaux+iaux)

FOUR2520

iaux=iaux(iaux+iaux+iaux)

FOUR2530

iaux=iaux(iaux+iaux+iaux)

FOUR2540

iaux=iaux(iaux+iaux+iaux)

FOUR2550

iaux=iaux(iaux+iaux+iaux)

FOUR2560

iaux=iaux(iaux+iaux+iaux)

FOUR2570

iaux=iaux(iaux+iaux+iaux)

FOUR2580

iaux=iaux(iaux+iaux+iaux)

FOUR2590

iaux=iaux(iaux+iaux+iaux)

FOUR2600

iaux=iaux(iaux+iaux+iaux)

FOUR2610

iaux=iaux(iaux+iaux+iaux)

FOUR2620

iaux=iaux(iaux+iaux+iaux)

FOUR2630

iaux=iaux(iaux+iaux+iaux)

FOUR2640

iaux=iaux(iaux+iaux+iaux)

FOUR2650

iaux=iaux(iaux+iaux+iaux)

FOUR2660

iaux=iaux(iaux+iaux+iaux)

FOUR2670

iaux=iaux(iaux+iaux+iaux)

FOUR2680

iaux=iaux(iaux+iaux+iaux)

FOUR2690

iaux=iaux(iaux+iaux+iaux)

FOUR2700

iaux=iaux(iaux+iaux+iaux)

FOUR2710

iaux=iaux(iaux+iaux+iaux)

FOUR2720

iaux=iaux(iaux+iaux+iaux)

FOUR2730

iaux=iaux(iaux+iaux+iaux)

FOUR2740

iaux=iaux(iaux+iaux+iaux)

FOUR2750

iaux=iaux(iaux+iaux+iaux)

FOUR2760

iaux=iaux(iaux+iaux+iaux)

FOUR2770

iaux=iaux(iaux+iaux+iaux)

FOUR2780

iaux=iaux(iaux+iaux+iaux)

FOUR2790

iaux=iaux(iaux+iaux+iaux)

FOUR2800

iaux=iaux(iaux+iaux+iaux)

FOUR2810

iaux=iaux(iaux+iaux+iaux)

FOUR2820

iaux=iaux(iaux+iaux+iaux)

FOUR2830

iaux=iaux(iaux+iaux+iaux)

FOUR2840

iaux=iaux(iaux+iaux+iaux)

FOUR2850

iaux=iaux(iaux+iaux+iaux)

FOUR2860

iaux=iaux(iaux+iaux+iaux)

FOUR2870

iaux=iaux(iaux+iaux+iaux)

FOUR2880

iaux=iaux(iaux+iaux+iaux)

FOUR2890
C
READ SECTION
35 READ(ISC,BUFF,11,1=1,NUM)
DO 90 1=1,12A
J=J+1
IZ=IZ+1-1
IZ=JCC
C
GET I INTO ACTUAL SET COMPUTED
345 IF(IZ.LE.5,J=2)GO TO 344
IF(FCZ.GE.JCC-1)AND(JZ.LE.JZFGO)TO 344
GO TO 343
344 IF(JZC,NE.KJC)GO TO 90
J=J+1
IZ=IZ+2=0
JYC=JYI
40 IF(JYC.GT.IYFIGO TO 90
JYC=JYI
GET Y INTO ACTUAL SET COMPUTED
41 IF(JYI.JE.JC).AND.JYC.LE.JYFIGO TO 42
IF(JYI.GE.0)GO TO 320
JYC=J-1=SIGLY
GO TO 41
42 INDEX NUM=((IZ-IZI)+YI+1)
X=(1)*YI+A(3)*IZ+A(4)
J=JYC-JYI*JX-1=1
I=X
FX=X-1
JDOE[4]=I-1
JDOE[2]=I
JDOE[3]=I+1
JDOE[4]=I+2
DO 60 1=1,4
50 IF(JDOE[1].GE.IXI).AND.JDOE[2].LE.IXFIGO TO 60
IF(FCZ.0)GO TO 70
GO TO 50
AA=LINE[1]/2/2+LINE[1]/2/2+LINE[1]/2/2
BB=LINE[1]/2/2+LINE[1]/2/2+LINE[1]/2/2
CC=LINE[1]/2/2+LINE[1]/2/2+LINE[1]/2/2
BUFF INDEX=AA+FX+BB+FX+CC+FX+LINE[1]/2
DO 80 3=1,2
70 BUIF INDEX=11111
80 JYC=-1
GO TO 40
90 CONTINUE
C
CHECK IF ANY SECTIONS LEFT TO FIND
DO 91 1=1,12A
 IF(JZC(11)+NUM2)330,91,30
91 CONTINUE
100 N=0
DO 120 2=1,12A
N=M+1
L=MIN(102,1+1)
J=1=1
M=0
DO 105 2=1,12A
N=M+1
105 IF(N=M+1)
WRITE(OUT,)return}
123) IRET=KRET
ICOUNT=KCOUNT-KSAVE
K+KSAVE=1
KSAVE=1
GO TO 120
C
END
CPLT1160
CPLT1170
CPLT1180
CPLT1190
CPLT1200
CPLT1210
CPLT1220
SUBROUTINE ICOUNT I0DATA, NROW, NCOL, NRUS, NCUS, GCREG, CLIM, ACSTP)
C
CONTURING SUBPROGRAM BY S. W ZISK, MIT LINCOLN LAB, MARCH 1960.
C
REWITTEN IN USA$1 BASIC FORTRAN BY NN BRENNER MIT LL, MARCH 1968.
C
CSTP IS ACSTP WITH THE SIGN CHOSEN TO INCREASE IN THE CORRECT
C
DIRECTION FROM CREG TO CLIM.
C
FOR EACH XLEV, PLOTS A CONTOUR OF DATA, I.J., OUTPUT BEING RETURNED.
C
C VIA SUCCESSIVE CALLS TO SUBROUTINES CPLUT0 OR 1-3, FOR PEN
C
UP OR PEN DOWN TRACES. CPLUT02-3 FOR LABELING AT THIS POSITION.
C
C IF DESIRED, FULL CALL IS CALL CPLUT02PLLOT, ROW, COL, XLEV, WHERE
C
ROW IS FROM 1, TO FLOAT(MAXY). COL IS FROM 1, TO FLOAT(MAXY).
C
C CONT EXTRUDES ONE POINT AT A TIME TO CPLUT. IF DESIRED FOR
C
EFFICIENCY, CPLUT MAY SAVE UP POINTS IN A SCRATCH ARRAY.
C
C WHEN IPOINT IS 0, CPLUT SHOULD JUST REMEMBER THIS (ROW, COL) POINT.
C
C WHEN IPOINT=1, CPLUT SHOULD DRAW A LINE SEGMENT FROM THE REMEMBER-
C
ED POINT TO THE NEW POINT, AND THEN MAKE THE NEW POINT THE REME-
C
BERED POINT.
C
C A CONTOUR LABEL IS CALLED FROM CPLUT IF IPOINT=2 ONLY ONCE PER
C
COMPLETE CONTOUR SEGMENT, SO THAT IF SEVERAL DIFFERENT LOOPS ARE
C
AT HEIGHT XLEV, THEY WILL EACH GET A LABEL.
C
C THIS PROGRAM IS WRITTEN IN USA$1 BASIC FORTRAN IN AN ATTEMPT TO BE
C
COMPATIBLE WITH AS MANY EXISTING FORTRAN COMPILERS AS POSSIBLE.
C
C AS WRITTEN, THIS PROGRAM ASSUMES DATA IS A FLOATING POINT ARRAY.
C
DIMENSION DATA(111)
REAL MARK
C
DIMENSION DATA(NROW,NCOL), OF WHICH ONLY MAXX, MAXY ARE Used.
C
NOTE--THE LUM ORDER BIT OF EACH ENTRY DATA(I,J) IS LIALE TO CHANGE.
C
THIS WILL NEGIGIBLY AFFECT A FLOATING POINT ARRAY.
C
C CORRECTED AND MODIFIED FEB. 1970. G F REID, SG.
C
C IRT AND JUP INDICATE THE DIRECTION OF THE LAST CONTOUR STEP.
C
C IRT=01 INDICATES THAT THE LAST STEP HAD X DECREASING, INCREASING.
C
C JUP=01 INDICATES THAT THIS STEP HAD Y DECREASING, INCREASING.
C
C DIRECTION OF THE NEXT CONTOUR STEP IS DETERMINED BY THE VALUES AT
C
C A, B, C, AND D WITH THE NEXT STEP ENDOG ON A OR AT D ISPECIAL CASE.
C
C A AND B, C, D ARE THE VERTICES OF A SQUARE LABELLED IN ORDER.
C
C A TEST BASED ON THE VALUE USE CAN ESTIMATE THE VALUE AT THE CENTER.
C
C OF THE SQUARE WOULD ADD TO THE CONTOURING MORE CONSISTENT.
C
C
C MAXX=NRUS
C
C MAXY=NCUS
C
C CSTP=SIGN ACSTP, CLIM, CREG
C
CTR=CREG
C
C CLEAR LEAST SIGNIFICANT BIT
C
C 101 XLEV=UNMARK(CTR)
C
C JLOOP=1
C
C JUP=JLOOP
C
C I=1
C
C IY=1
C
C NO=1
C
C 00 102 11, 110
C
C 102 DATA(I,J)=UNMARK(I,J)
C
C 103 ICNT 100
C THE LEAST SIGNIFICANT BIT OF DATA(1) IS USED TO MARK THE LEFT-HAND
C EDGE OF THE CONTOUR WHEN IT Crosses THAT ROW.
C LOOKING FOR THE FIRST POINT ON THE XLEV CONTOUR.

103 IUP=IUP
    ILOOP=JLOOP
    I=I
    IQUIP=0
    IF (IX-MAXX) 104,131,122
104 L=IX+NRW*(IY-1)
C IX,IY SAVE THE LAST POINT THAT WAS SEARCHED, TO RESTART THE
C SEARCH AFTER FINISHING THE CURRENT CONTOUR LINE.
C
105 OA1=DATA(I)
    OA1=UNMARK(OA1)
    IF (OA1-XLEV) 106,116,106
106 IF (IX-1) 213,107,116
107 IF (IX=1) 214,112,108
108 LMX=L-NROW
    OA2=DATA(LMA)
    OID=UNMARK(OA2)
    IF (OID-XLEV) 111,109,111
109 IF (OID-OJDA2) 111,110,111
110 YPT=IY-1
    GO TO 115
111 IF (IX-MAXX) 112,116,1215
C LOOKING FOR AN INTERSECTION IN THE LEFT-HAND COLUMN BEFORE
C SCANNING FOR INTERSECTIONS IN THE CURRENT ROW.
C
112 LPX=L-NROW
    OID=UNMARK(OA2-LPX)
    IF (OID-XLEV) 113,116,113
113 IF (OID-OJDA2) 114,114,116
C ARE BOTH OID AND OJDA1 ON THE SAME SIDE OF THE CONTOUR
114 GEL=OID-OJDA1
    IF (GEL<0.0) 115,107,108
    YPT=FLOAT(IY)+XLEV-OJDA1/DEL
115 XPT=IX
    YI=YPT
    ILOOP=0
    IF OID=DATA(LPX) 130,136,130
C IF TWO ROWS ARE EACH FILLED WITH HIGH AND LOW NUMBERS RESPECTIVELY
C WE HAVE JUST FOUND THE INTER-ROW CONTOUR.
C
116 OJDA2=DATA(I)
    OJDA2=UNMARK(OJDA2)
    IF (OJDA2-XLEV) 130,117,130
117 IF (OJDA2-XLEV) 129,118,128
118 IF (OJDA2-XLEV) 119,121,119
119 XPT=IX
C X= XPT
    IF (IX-1) 216,120,137
C CONTINUE WHEN IT TOUCHES THAT ROW.
C
C FINDING THE LOWER OR UPPER CORNER OF A PLATEAU THAT HITS THE
C RIGHT EDGE.
C
123 IF (OJDA2-OJDA2) 131,124,131
124 L=I+1
    LMX=L-NROW
    OJDA2=UNMARK(OJDA2-LMX)
    IF (OJDA2-XLEV) 125,126,125
    IUP=0
125 GO TO 127
126 IF (OJDA2-XLEV) 127,131,127
C IF BOTH UPPER AND LOWER POINTS ARE ALSO ON, THEN DUMP THIS ONE.
C
127 IRT=0
    ILOOP=0
    XPT=XPT
    GO TO 137
C SKIPS ANY BLOCK OF ALL XLEV
C FOR 1ST POINT Greater THAN XLEV
C
128 IF (XLEV-OJDA2) 130,130,130
C
129 IF (XLEV-OJDA2) 130,130,130
C
130 IF (XLEV-OJDA2) 130,130,130
C EMITS ANY RIGHT HANG EDGE COLUMN OF ALL XLEV
C
131 IF (IX-MAXX) 135,131,131
C
132 IJUMP=1
    GO TO 207
133 IX=IX
    IY=IY
    L=IX+NRW*(IY-1)
    IF (IY-MAXY) 105,134,219
134 JLOOP=0
    ILOOP=JLOOP
    JUP=ILOOP
135 1X=1X+1
L=1+1
DJA=DJAJ
DIA=DIVA
GOTO 116
136 DEL=DOA2=DOA1
I=FA[S1000=I.0=-4.0=DEL=SIGN(1.0-4.0,DEL)
XPT=FLOAT(IX1)*XLEV=DOA1/DEL
XI=XPT
C SET LAST SIGNIFICANT BIT
137 DIA(1)=MARK(DATA1)
C MARK DATA1
C YPT=IV
Y1=YPT
138 CALL (PLAGD 10,XPT,YPT,XLEV)
IX=IX+1
IF IIY=11 220,139,140
139 ILOOP=0
C TO START THE NEXT SEARCH AT A NEW POINT.
140 IF ILOOP=0 207,141,207
C X1, Y1 ARE THE STARTING POINT OF THIS CONTOUR LINE.
C ILOOP=0 IF WE START ON AN EDGE, AND I IF OTHERWISE.
C CONPT FINDS THE NEXT POINT ON THE CONTOUR, WORKING FROM THE LAST
C POINT THAT WAS FOUND AND THE GENERAL DIRECTION UPR OR DOWN, RIGHT
C OR LEFT) THAT THE CONTOUR IS PROGRESSING. IT ALSO NOTES ANY
C CHANGE IN THIS DIRECTION AND STORES IT FOR THE NEXT CALCULATION.
141 IONENT=1
IXPT=XPT
YPT=YPT
IF (YPT-FLOAT(I1YPT)) 142,143,142
C GO TO 2030 IF IT'S ON A ROW.
142 IY=IYT+1UP
IYB=IY
IYC=IYPT+1-1UP
IXA=IXPT
IXB=IXPT+1+1RT+1RT
IXC=IXB
[IND]=1
ID=IXA+SIGN[I1YC-1]
GO TO 140
143 IF (XPT-FLOAT(I1XPT)) 144,145,144
C GO TO 2050 IF IT'S ON A ROW AND ON A COLUMN.
144 IY=IXT+1RT
GO TO 147
145 IONENT=0
IXA=IXPT+1RT+1RT
IXC=IX+1RT+1RT
IYA=IYPT
IYB=IYT+1UP+1UP
IXC=IX
IY=IY
[IND]=2
146 IF (X=MARK(I1DATA1))
X=MARK(I1DATA1)
147 IF IDIA=XLEV) 149,156,161
C WE TEST IN ORDER IDA=DATA1 AND POSSIBLY) IDA TO DETERMINE THE
C NEXT POINT ON THE CONTOUR. IDA TEST IS TO TRY TO PREVENT THE
C CONTOUR CROSSING A SADDLE OR TRUGH.
C IDA TO XLEV)
148 IF (IDIA=XLEV) 150,159,155
149 IF (IDIA=XLEV) 160,151,152
C IF (IDIA-XLEV)
150 IF (IDIA-XLEV) 160,151,152
151 IF (IDIA-XLEV) 190,180,180
152 IF (IDIA-XLEV) 190,180,180
153 IF (IDIA-XLEV) 161,180,180
154 IF (IDIA-XLEV) 161,180,180
155 IF (IDIA-XLEV) 161,180,180
156 IF (IDIA-XLEV) 161,180,180
157 IF (IDIA-XLEV) 179,180,180
158 IF (IDIA-XLEV) 179,180,180
159 IF (IDIA-XLEV) 179,180,180
160 IF (IDIA-XLEV) 179,180,180
161 IF (IDIA-XLEV) 179,180,180
162 IF (IDIA-XLEV) 180,180,180
163 IF (IDIA-XLEV) 180,180,180
164 IF (IDIA-XLEV) 180,180,180
165 IF (IDIA-XLEV) 180,180,180
166 IF (IDIA-XLEV) 180,180,180
167 IF (IDIA-XLEV) 180,180,180
168 IF (IIDENT) 190,169,170
169 IF (IIDENT) 190,169,170
170 IF (IIDENT) 177,173,173
171 IF (IIDENT) 177,173,173
172 IF (IIDENT) 177,173,173
173 I0UMP=3
174 IF (IIP=1RT=2) 176,172,175
C X-LIMIT CHECK
C GO TO 146
C GO TO 140
C GO TO 2050 IF IT'S ON A ROW AND ON A COLUMN.
C GO TO 147
C X-LIMIT CHECK
C GO TO 146
C GO TO 140
C GO TO 2050 IF IT'S ON A ROW AND ON A COLUMN.
175 IF (XPT+1RT=MAXX) 172,172,176
176 IDENT=IDENT+2
C GO TO 171
C
C Y-LIMIT CHECK
C
177 IF (YPT+1UP-2) 173,172,178
178 IF (YPT+1UP-MAXY) 172,172,173
179 XPT=IYA
C YPT=IYA
C GO TO 194
180 XPT=IYC
C YPT=IYC
C IF (IDENT) 182,194,194
C
DON'T CHANGE DIRECTION IF IT'S AN OFF-TO-ON...
C
182 IF (IND-1) 183,192,183
183 IR=1-IR
C GO TO 194
184 DEL=DEL-DIA
C IFABS(DEL-LT.1.0=40) DEL=SIGN1.0-4.0DEL
C DEL=XLEV-DIA/DEL
C IF (IND-2) 185,186,185
185 XPT=FLOAT(XAI)+DEL#FLOAT(IRT+IRI-1)
C YPT=IYA
C GO TO 194
186 YPT=FLOAT(YAI)+DEL#FLOAT(IUP+IUP-1)
C XPT=IAY
C GO TO 194
187 DEL=DIR-DIC
C IFABS(DIR-LT.1.0=40) DEL=SIGN1.0-4.0DEL
C DEL=XLEV-DIC/DEL
C IF (IND-2) 188,189,188
188 YPT=FLOAT(YCI)+DEL#FLOAT(IUP+IUP-1)
C XPT=IAY
C GO TO 194
189 XPT=FLOAT(XCI)+DEL#FLOAT(IRT-1)
C YPT=IYB
C GO TO 194
190 DEL=DIC-DIR
C IFABS(DIR-LT.1.0=40) DEL=SIGN1.0-4.0DEL
C DEL=XLEV-DIC/DEL
C IF (IND-2) 191,192,191
191 XPT=FLOAT(XCI)-DEL#FLOAT(IRT-1)
C YPT=IYB
C
192 IUP=I-1UP
C GO TO 194
193 XPT=FLOAT(YCI)-DEL#FLOAT(IUP+IUP-1)
C YPT=IAY
C GO TO 183
194 CALL CPT ((1.xpt+ypt.xev)
C
C PLOTS A LINE TO THIS LOCATION.
C
C
195 IF (IUMP=3) 195,20,206
C 196 YPT=IYB
C 197 IF (XPT-1MUN=IYPT-1)
C (IFABS(YPT-1FLOAT(YPT-1)0.01.196.196.198)
C 198 IF YPT IS AN INTEGER, MARK THE POINT IMMEDIATELY.
C 199 ATELY TO THE LEFT. IF IT IS ALREADY MARKED, THIS
C HAS BEEN PLOTTED BEFORE.
C
196 IFDATA(MARKDATA(I)) 120,2,197,202
C 200 IF THIS POINT HAS BEEN TRAVERSED BEFORE...
C
197 DATA(MARKDATA(I))
C
C IF NOT, KEYS THIS POINT FOR FUTURE REFERENCE.
C
198 IF (XPT-1.O=221,2040,199
C
199 IF (XPT-1.0) 200,204,222
C
200 IF (XPT-1.0) 223,204,201
C
201 IF (XPT-1.0) 141,204,224
C
202 IF (XPT-1.0) 204,203,204
C
203 IF (XPT-1.0) 204,203,204
C
204 DATA(MARKDATA(I))
C
205 IF (IUMP=3) 205,206
C
206 IF (IUMP=3) 206,205,206
C
C IF THE CONTOUR HITS EITHER AN EDGE OR ITSELF IN THE MIDDLE,
C IT RETURNS ONCE TO THE STARTING POINT AND BEGINS CONTURING IN
C THE OTHER DIRECTION. IF THE STARTING POINT IS ON AN EDGE, IT'S T
C
207 IF (IUMP=3) 207,208
C
208 IF (IUMP=3) 208,209,210
C
209 IF (IUMP=3) 209,209,210
C
210 IF (IUMP=3) 210,200,206
C
210 CALL CPT (10.xpt+ypt.xev)
C
C PLOTS A LINE TO THIS LOCATION.
A5=3,*((A-(X^C))-A9-A9-A9
A6=3,*((Y^C)-A10-A10-A10
=1,
U=0.
104 U=UNIT
WZ=(A2+AB)4*A2+AB
CALL PLOT(WX, WY, 2)
I=I+1
IF (I.LE.NP) GO TO 104
IC=IC+1
IF (IC.LE.IN) GO TO 103
RETURN
C END

C COMMENT CARDS FOR PINGPC
C THIS PROGRAM IS AN ADAPTION OF THE EKAY 67 PROGRAM LIST PC, IT BAK
C BEEN ALTERED TO "STAND ALONE" AND READ THE PREVIOUS OUTPUT TAPE WTRIT C
C PAIRE, THE LEAST SQUARES PROGRAM WRITTEN BY L., W. PINTER AT TUB
C GEOGRAPHICAL LAB, CARNEGIE INSTITUTE OF WASHINGTON.
C THE OUTPUT OF THE PROGRAM HAS BEEN CHANGED SO THAT THE "FRASB IN
C WILLICYCLE" IS REPLACED AT SIGMA(F0), SCALLED UP AT A FACTOR OF LSTF
C 10 TO MAKE IT CONSISTENT WITH PP.
C CARD ONE
C COL 1-6 COMPOUND IDENTIFICATION, 7-7B ANT TITL
C CARD TWO
C 14 (1/2/3) FOR R INDEX (MOST/NEXT MOST/LEAST) RAPIDLY LSTF 130
C I C ODE TO TELL HOW INPUT DATA AND OUTPUTS
C 15 SORTING FREQUENCY FOR L INDEX
C 16 SORTING FREQUENCY FOR L INDEX
C (NOT THAT SUM AND PRODUCT OF NUMBERS IN COLUMNS 14,15, AND 16 WSLSTF 170
C EQUAL 1)
C 17-20 NUMBER OF LINES PER "LISTPC" PAGE LSTF 190
C 21 (BLANK)/A4) FOR DO/ABSOLUTELY DO NOT) INTERNALLY CHANGE THE LSTF 200
C OF THE NUMBER OF LINES IN ORDER TO MAKE THE BOTTOM OF THE LAST PAGE LSTF 210
C AS EVEN AS POSSIBLE
C 22-24 NUMBER OF LIST COLUMNS FOR "LISTPC" PAGE LSTF 220
C NOTE THE PRODUCT OF THE NUMBER OF COLUMNS AND THE NUMBER OF COLUMNS PLSTF 240
C "LISTPC" PAGE CANNOT EXCEED 3000. THAT IS, NO MORE THAN 3000 FOR TLSTF 250
C TOTAL OF REFLECTIONS AND READINGS (WITH THEIR SPACES), FOR PAGE, LSTF 260
C ZERO (0) BLANK IN ANY OF THE FIVE FOLLOWING FIELDS MEANS JUST THAT LSTF 270
C 26 NUMBER OF BLANK PRINT COLUMNS BEFORE THE LISTPC COLUMN LSTF 280
C 28 NUMBER OF PRINT COLUMNS FOR MOST RAPIDLY CHANGING INDEX LSTF 290
C 30 NUMBER OF PRINT COLUMNS FOR (0) TIMES 10 LSTF 300
C 32 NUMBER OF PRINT COLUMNS FOR PC TIMES 10 LSTF 310
C 34 NUMBER OF PRINT COLUMNS FOR SIDEB TIMES 10 LSTF 320
C 36 SPECIAL FLAG FOR LESS TRANS (BLACK + * ) (JCODE > 2) LSTF 330
C 38 SPECIAL FLAG FOR EXTINCT (BLACK + * ) (JCODE = 3) LSTF 340
C 40 SPECIAL FLAG FOR SPECIAL REFLECTION (JCODE > 4) LSTF 350
C THE NEXT NINE FIELDS ARE (BLANK)/A1) FOR DO NOT)/DO LSTF 360
C 42 PRINT SYMBOL FOR LESS THAN OR EXTINCT REFLECTIONS LSTF 370
C THIS SYMBOL SWITCH ADDS ONE WORK PRINT COLUMN LSTF 380
C 44 FOR DIFFERENT STRUCTURES, ATTACH SIGN OF A TO PC LSTF 390
C 46 DOUBLE SPACE THE LINES LSTF 400
C 48 WRITE EACH "LISTPC" PAGE TO BE THE TOP OF A PRINTER PAGE LSTF 410
C 50 PRINT CURRENT "TITLE" AT TOP OF EACH PRINTER PAGE LSTF 420
C 52 PUNCH A SET OF "CARDPC" CARDS (HAS I,EL,F0,PC,AB,JCODE, AND LSTF 430
C LEVEL INDICATORS)
C 56 PRINT MENUS SIGN ON THE PC OR OF THE REFLECTIONS WHICH ARE LESS TRANS LSTF 450
C 59 WRITE A SEPARATE COPY OF THE PC LIST ON TAPE A LSTF 460
C 62 NUMBER OF TIMES TO TRY TO GET HEADING AT THE TOP OF ALL COLUMNS LSTF 470
C 64 (1/2/3) FOR HEADINGS SEPARATED BY (NO BLANK LINES)/1 BLANK LSTF 490
C LINE ABOVE /1 ABOVE AND BELOW /1 BLANK /1 BLANK LSTF 490
C 65-72 FACTOR F's ARE TO BE MULTIPLIED BY, BLANK = 10 LSTF 500
C 73-B0 FACTOR SIGN IS TO BE MULTIPLIED BY, BLANK = 10 LSTF 510
C AS PRESENTLY WRITTEN, THE PROGRAM WILL NOT READ D VALUES FROM CARDSTF 520
C BUT THIS CAN EASILY BE ADAPTED BY MAKING A USER SPECIFIC VERSION LSTF 530
C OF SUBROUTINE PING LSTF 540
C C----VERSION AND UPDATING ESTABLISHED 1 JUNE 1969 LSTF 550
C ALTERATIONS MADE AS FOLLOWS... 20 JUN 69, 23 SBF 69 LSTF 570
C----IN A FINAL SET IF THE NEADING ARRAY IX MAY LOOK AS FOLLOWS
C IX(1) = 28N X
C IX(2) = BLANKES
C IX(3) = BLANKES
C IX(4) = IX,
C IX(5) = IX,
C IX(6) = 28K,
C IX(7) = IX,
C IX(8) = BLANKES
C IX(9) = IX
C IX(10) = BLANKES
C==ICAM(11) = IN
C ICAM(12) = 2N X,
C IX(5) = ICAM(11)
C IX(6) = ICAM(12)
GO TO 421
C----IT MUST BE L
412 ICAM2 = 4
ICAM4P = 2
ICAM7P = 1
C----ICAM(13) = IX,
C ICAM(14) = 2N
C IX(7) = ICAM(14)
421 CONTINUE
C---IPACAS 451, 451, 425
C----INTIALIZE THE SCALE BAND TO 1.0
425 CONTINUE
C----INFORMATION FROM TAPE
451 CONTINUE
100*0
NP*1
IPF = 500
C TOP OF PAGE
4001 CONTINUE
DR 4002 J = 1, ITMTH
IF (J) = 2
4002 CONTINUE
IPF (ITMTH + 1) = -2
IPF = 999
IPZ = 9999
ICOUNT = 0
ATP SP LMP FH STORING REFLECTIONS
C----DATA ON BINARY TAPE. ----READ IT
4501 CONTINUE
IF (IPF, LT, 500 JG04502)
IPF = 0
CALL FINGETAPE (A, BUP, F)
4502 CONTINUE
IPF (IPF, DE, 500 JG04501)
L = (UDYH (IPF - 1))
IF (L = 0.00) 90 TO 4503
L = (UDYH (IPF))
IL = (UDYH (IPF + 3))
FEEL = BUF (IPF)
JCMOE = 18000 (IPF + 8)
LSTF2900
LSTF2910
LSTF2920
LSTF2930
LSTF2940
LSTF2950
LSTF2960
LSTF2970
LSTF2980
LSTF2990
LSTF3000
LSTF3010
LSTF3020
LSTF3030
LSTF3040
LSTF3050
LSTF3060
LSTF3070
LSTF3080
LSTF3090
LSTF3100
LSTF3110
LSTF3120
LSTF3130
LSTF3140
LSTF3150
LSTF3160
LSTF3170
LSTF3180
LSTF3190
LSTF3200
LSTF3210
LSTF3220
LSTF3230
LSTF3240
LSTF3250
LSTF3260
LSTF3270
LSTF3280
LSTF3290
LSTF3300
LSTF3310
LSTF3320
LSTF3330
LSTF3340
LSTF3350
LSTF3360
LSTF3370
LSTF3380
LSTF3390
LSTF3400
LSTF3410
LSTF3420
LSTF3430
LSTF3440
LSTF3450
LSTF3460
LSTF3470
LSTF3480
LSTF3490
LSTF3500
LSTF3510
LSTF3520
LSTF3530
LSTF3540
LSTF3550
LSTF3560
LSTF3570
LSTF3580
LSTF3590
LSTF3600
LSTF3610
LSTF3620
LSTF3630
LSTF3640
LSTF3650
LSTF3660
LSTF3670
LSTF3680
LSTF3690
LSTF3700
LSTF3710
LSTF3720
LSTF3730
LSTF3740
LSTF3750
LSTF3760
LSTF3770
LSTF3780
LSTF3790
LSTF3800
LSTF3810
LSTF3820
LSTF3830
LSTF3840
LSTF3850
LSTF3860
LSTF3870
LSTF3880
LSTF3890
LSTF3900
LSTF3910
LSTF3920
LSTF3930
LSTF3940
LSTF3950
LSTF3960
LSTF3970
LSTF3980
LSTF3990
LSTF4000
LSTF4010
LSTF4020
LSTF4030
LSTF4040
LSTF4050
CSE IF IPAGE IS ALL LOADED IN

/* COUNT-WATRUX Macro, 6001, 6006, 6009
C----- IF NOT IN CASO WE'RE OS READ ANOTHER CARD, IN TAPE, TAPE.
C
C----- PREPARE TO SHARE THE STACK FOR PRINTING NICETIES.
6001 CONTINUE

C----- FIGURE OUT NUMBER OF BLANKS AT THE BOTTOM OF THE STACK

I = WATRUX * 1
IBLAT = 0
00 6003 J = 1,WATRUX
I = I - 1
IF(IP(1)) = 1 6002,6004,6006
6002 IBLAT = IBLAT * I
6003 CONTINUE

6004 CONTINUE

IP (ICL) 6012,6016,6005
6006 IF(IPAGE-ICLK(15)) 6007,6012,6007

C----- ALL REFLECTIONS ARE NNN IN - EVEN THE BOTTOM OF THE PAGE

C----- UNLESS OF COURSE APAGE FLAG SAYS NO.
6007 IF(IPBLAT/ICLPP) 6011,6016,6008
6008 IBLAT = IBLAT - ICLPP
LINE = LINE - 1
IF(LINE) 6009,6009,6007
6009 LINE = 1
6011 CONTINUE

WATRUX = ICLPPLINE
6012 CONTINUE

C----- SHARE THE STACK TO MAKE COLUMNS AS HEAT AS POSSIBLE.
LDF = 1
ICLCL * 1
C----- SEARCH THE TABLE COLUMN BY LOUSEST COLUMN, WHERE'S THE IEL?
6095 CONTINUE

LDFT = ICLCL+LINE
ITOP = I907 - LINE * 1
C----- TEST IF TOP ENTRY IN THE STACK IS A BLANK
C----- IF IT IS, SEE IF THE ONE BELOW IT IS AN INDX
IF(IPMHTOP) 1 7001,7001,7001
7001 IF(IPMHTOP) = 1 11 7001,7001,7001
C----- POP THE WHOLE STACK FROM START POINT ONWARD UP ONE SPACE
7002 CONTINUE

O0 7005 J = ITOP,WATRUX
INDEX[I] = INDEX[I]+1
P0(J) = IPG(J)+1
IPCP(J) = IPCP(J)+1
ISIG(J) = ISIG(J)+1
IPSH(J) = IPSH(J)+1
7005 CONTINUE

IBLAT = IBLAT * I
C----- SEE IF THE BOTTOM OF THIS COLUMN IS A HEADING OR A BLANK
7011 IF(IPJWATRUX) 1 7012,7014,7031
C----- IT'S A BLANK, SEE IF ANY ABOVE IS A HEADING,
7012 IF(IPJWATRUX) = 1 11 7031,7031,7031
C----- IT'S SO SET THE 'PAPER TO 2
7013 IFP = 2
GO TO 7015
C----- HEADING IS AT THE BOTTOM OF A COLUMN, SET PAPER TO 1
7014 IFP(1) = IFP(J) = 1 7031,7021,7021
LSTP4560
LSTP4570
LSTP4580
LSTP4590
LSTP4600
LSTP4610
LSTP4620
LSTP4630
LSTP4640
LSTP4650
LSTP4660
LSTP4670
LSTP4680
LSTP4690
LSTP4700
LSTP4710
LSTP4720
LSTP4730
LSTP4740
LSTP4750
LSTP4760
LSTP4770
LSTP4780
LSTP4790
LSTP4800
LSTP4810
LSTP4820
LSTP4830
LSTP4840
LSTP4850
LSTP4860
LSTP4870
LSTP4880
LSTP4890
LSTP4900
LSTP4910
LSTP4920
LSTP4930
LSTP4940
LSTP4950
LSTP4960
LSTP4970
LSTP4980
LSTP4990
LSTP5000
LSTP5010
LSTP5020
LSTP5030
LSTP5040
LSTP5050
LSTP5060
LSTP5070
LSTP5080
LSTP5090
LSTP5100
LSTP5110
LSTP5120
LSTP5130
LSTP5140
LSTP5150
LSTP5160
LSTP5170
LSTP5180
LSTP5190
LSTP5200
LSTP5210

C----- THERE ARE BLANKS AT ENO, PUSH THE STACK DOWNWARD.
7021 CONTINUE
I1LAT = IBLAT = IPGP
O0 7024 K = 1,IPGP
I1STRT = IMAG = IPGP + K
I = WATRUX * 1
O0 7023 J = I1STRT,WATRUX
I = I - 1
INDEX[I] = INDEX[I]+1
IPG(J) = IPG(J)+1
IPCP(J) = IPCP(J)+1
ISIG(J) = ISIG(J)+1
IPSH(J) = IPSH(J)+1
7023 CONTINUE

LSTP4560
LSTP4570
LSTP4580
LSTP4590
LSTP4600
LSTP4610
LSTP4620
LSTP4630
LSTP4640
LSTP4650
LSTP4660
LSTP4670
LSTP4680
LSTP4690
LSTP4700
LSTP4710
LSTP4720
LSTP4730
LSTP4740
LSTP4750
LSTP4760
LSTP4770
LSTP4780
LSTP4790
LSTP4800
LSTP4810
LSTP4820
LSTP4830
LSTP4840
LSTP4850
LSTP4860
LSTP4870
LSTP4880
LSTP4890
LSTP4900
LSTP4910
LSTP4920
LSTP4930
LSTP4940
LSTP4950
LSTP4960
LSTP4970
LSTP4980
LSTP4990
LSTP5000
LSTP5010
LSTP5020
LSTP5030
LSTP5040
LSTP5050
LSTP5060
LSTP5070
LSTP5080
LSTP5090
LSTP5100
LSTP5110
LSTP5120
LSTP5130
LSTP5140
LSTP5150
LSTP5160
LSTP5170
LSTP5180
LSTP5190
LSTP5200
LSTP5210
**Title and Subtitle:**
A System of Fortran IV Computer Programs for Crystal Structure Computations

**Author(s):**
Larry W. Finger and E. Prince

**Performing Organization Name and Address:**
National Bureau of Standards
Department of Commerce
Washington, D.C. 20234

**Sponsoring Organization Name and Complete Address (Street, City, State, ZIP):**
Same

**Abstract:**
This report gives detailed descriptions and instructions for use of a system of programs for crystallographic calculations, including least-squares refinement with generalized systems of constraints, calculation of bond distances and angles with errors, Fourier synthesis, plotting of contours in Fourier maps, and preparation of structure factor tables for publication.

**Keywords:**
Computer programs; contour plotting; constrained refinement; crystallographic calculations; Fourier section; Fourier synthesis; least squares.
NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH reports National Bureau of Standards research and development in physics, mathematics, and chemistry. It is published in two sections, available separately:

- Physics and Chemistry (Section A)
- Mathematical Sciences (Section B)

Papers of interest primarily to scientists working in these fields. This section covers a broad range of physical and chemical research, with major emphasis on standards of physical measurement, fundamental constants, and properties of matter. Issued six times a year. Annual subscription: Domestic, $17.00; Foreign, $21.25.

• Mathematical Sciences (Section B)

Studies and compilations designed mainly for the mathematician and theoretical physicist. Topics in mathematical statistics, theory of experiment design, numerical analysis, theoretical physics and chemistry, logical design and programming of computers and computer systems. Short numerical tables. Issued quarterly. Annual subscription: Domestic, $9.00; Foreign, $11.25.

DIMENSIONS/NBS (formerly Technical News Bulletin)—This monthly magazine is published to inform scientists, engineers, businessmen, industry, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on the work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing. Annual subscription: Domestic, $9.45; Foreign, $11.85.

NONPERIODICALS

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications with interested industries, professional organizations, and regulatory bodies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a world-wide program coordinated by NBS. Program under authority of National Standard Data Act (Public Law 90-386).

NOTE: At present the principal publication outlet for these data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St. N.W., Wash. D.C. 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The purpose of the standards is to establish nationally recognized requirements for products, and to provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.


Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service (Springfield, Va. 22161) in paper copy or microfiche form.


BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau: Cryogenic Data Center Current Awareness Service

A literature survey issued weekly. Annual subscription: Domestic, $20.00; foreign, $25.00.


Electromagnetic Metrology Current Awareness Service

Issued monthly. Annual subscription: $100.00 (Special rates for multi-subscriptions). Send subscription order and remittance to Electromagnetics Division, National Bureau of Standards, Boulder, Colo. 80302.