NBS TECHNICAL NOTE 854

U.S. DEPARTMENT OF COMMERCE / National Bureau of Standards

A System of Fortran IV Computer Programs for Crystal Structure Computations

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards [1] 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:

Applied Mathematics — Electricity — Mechanics — Heat — Optical Physics — Nuclear Sciences [2] — Applied Radiation [2] — Quantum Electronics [3] — Electromagnetics [3] — Time and Frequency [3] — Laboratory Astrophysics [3] — Cryogenics [3].

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:

Analytical Chemistry — Polymers — Metallurgy — Inorganic Materials — Reactor Radiation — Physical Chemistry.

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:

Engineering and Product Standards — Weights and Measures — Invention and Innovation — Product Evaluation Technology — Electronic Technology — Technical Analysis — Measurement Engineering — Structures, Materials, and Life Safety [4] — Building Environment [4] — Technical Evaluation and Application [4] — Fire Technology.

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:

Office of Standard Reference Data — Office of Information Activities — Office of Technical Publications — Library — Office of International Relations.

---

[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

National Bureau of Standards Technical Note 854

Nat. Bur. Stand. (U.S.), Tech. Note 854, 133 pages (Feb. 1975)

CODEN: NBTNAE

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1975

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
(Order by SD Catalog No. C13.46:854). Price $2.00

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

	Page
RFINE4: A program for least squares refinement including general constraints.	1
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 ISC1 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_{obs}, the value of $\sin^2\theta/\lambda^2$ for the reflection, an integer specifying which scale factor applies to this reflection, β--the extinction parameter, a flag to mark reflections which had an intensity less than the minimum observable, the diffractometer angles χ, ϕ 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 Fouier 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 ISC1, 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 ISC1 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 $\sin \theta/\lambda$.

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

only when file ISC2 is being written, that is, either on the first cycle of a given run or when new structure factors are being read from cards. If this restriction is violated, the new curves are loaded to core and subsequently included in the data set for this structure, but they are never used to compute scattering factors for the reflections.

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:

$$F = \sum_{j}^{\text{atoms}} f_j \exp\left[i\left(2\pi \sum_{k} h_k x_k^j - \sum\sum\sum_{k\ell m} h_k h_\ell h_m c_{k\ell m}\right) - \sum\sum_{k\ell}\left(h_k h_\ell \beta_{k\ell}^j - \sum\sum_{mn} h_j h_\ell h_m h_n h_{k\ell mn}^j\right)\right]$$

x_k^j is the k^{th} fractional coordinate of atom j, β 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

$$w = 1/[\sigma^2 + (\rho F_{obs})^2].$$

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 BODAN1 and the thermal functions by routine ELVIB1.

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 σ_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

$$\frac{\partial A}{\partial p_n}' = \sum_m \frac{\partial A}{\partial p_m} \frac{\partial p_m}{\partial p_n}$$

and

$$\frac{\partial B}{\partial p_n}' = \sum_m \frac{\partial B}{\partial p_m} \frac{\partial p_m}{\partial p_n}$$

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

$$R_w^2 = \frac{\Sigma w \, (|Fo| - |Fc|)^2}{\Sigma \, w \, |Fo|^2}$$

for refinement on F or

$$R_w^2 = \frac{\Sigma w (|Fo|^2 - |Fc|^2)^2}{\Sigma w \, |Fo|^4}$$

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

$$R = \frac{\Sigma ||Fo| - |Fc||}{\Sigma \, |Fo|}$$

is computed for all reflections in each class. In addition, the ranges of variation of F_o 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^T represents the transpose of the vector of Miller indices and X'_{nj}

represents the jth symmetry transformed coordinate for the nth atom, then the argument of the trigonometric functions needed to calculate the contribution of this atom to the structure factor may be written as $H^T X'_{nj}$. Utilizing the expression for X' from page 3,

$$H^T X' = H^T(RX + T) = H^T RX + H^T T.$$

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

$$F = \sum_n a_n f_n \sum_j T_{nj} P_{nj}$$

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_n is an occupancy term, f_n is the value of the scattering curve for the nth atom, i is $\sqrt{-1}$ and β_n is the matrix of anisotropic thermal coefficients. When anomalous dispersion is included, the scattering factor becomes

$$f_n = f_{o_n} + f'_n + if''_n$$

with f_{o_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 $F = A + iB$, we find that

$$A = \sum_n \left[a_n\,(f_{o_n} + f'_n) \sum_j T_{nj} \cos 2\pi (H'^T_j X_n + H^T T_j) \right.$$

$$\left. - a_n f''_n \sum_j T_{nj} \sin 2\pi (H'^T_j X_n + H^T T_j) \right]$$

and

$$B = \sum_n \Big[a_n \, (f_{o_n} + f'_n) \sum_j \sin 2\pi (H'^T_j X_n + H^T T_j)$$

$$+ a_n f''_n \sum_j \cos 2\pi \, (H'^T_h X_n + H^T T_j) \Big].$$

The required partial derivatives are found by differentiating each of the above equations with respect to the components of X_n and β_j. After all sums have been accumulated, the subroutine MODIFY is called to perform any chainrule 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

$$F^*_c = F_c \, y^{1/2}$$

$$y = (1 + \beta(\theta) F_c^2 \, r^*)^{-1/2}$$

$$\beta(\theta) = \begin{cases} \dfrac{-2 \, e^4 \, 2}{m^2 c^4 V^2 A \sin 2\theta} \; \dfrac{(+ \cos^4 2\theta)}{(1 + \cos^2 2\theta)} \; \dfrac{dA}{d\mu} & \text{for x-rays} \\ \dfrac{\lambda^2}{V^2 \, A \sin 2\theta} \; \dfrac{dA}{d\mu} & \text{for neutrons} \end{cases}$$

λ is the wavelength, V is the cell volume, A is the transmission factor, μ 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 $r^* = (D'ZD)^{-1/2}$ with z_{ij} refined and c) Type II anisotropic with $r^* = (N'W\,N)^{-1/2}$ with W_{ij} refined. If $\bar{p}$ is a unit vector parallel to the incident beam and $\bar{s}$ is a unit vector parallel to the diffracted beam, $\bar{D}$ is a unit vector parallel to $\bar{p} \times \bar{s}$ 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

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

where n is the number of refined parameters and σ_{ij} is the covariance of the ith refined parameter with the jth 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 ϕ,χ, and ω. Here ω 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; χ 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; ϕ 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 ϕ, χ, and ω 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 transforms a second cumulant tensor in an orthonormal system to a set of anisotropic 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 $\underset{\sim}{T}$, $\underset{\sim}{L}$, and $\underset{\sim}{S}$.

Functions TKTLS, D3KDL, and D3KDS. compute the third cumulants and their derivatives with respect to $\underset{\sim}{L}$ and $\underset{\sim}{S}$.

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, FIJ, AXA, AXAXB, PIJKLM, EL4. are a set of functions used in the rigid body relations.

Common Blocks and Variables

1. Common Block /LSMAT/

B(200)	Normal equations vector
IEXT	Extinction type indicator
IPARA (1000)	Atom Parameter refinement indicators
MRANK	Order of the normal equations matrices
KCYL	Cycle number for the data set
LABEL (3)	12 character label for the data set

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)	ϕ, χ 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 auxillary 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 c. Employing the rules of Levy (1956), the following relationships are required for this atom:

$$\beta_{13} = \beta_{23} = 0$$

$$\beta_{11} = \beta_{22} = 2\ \beta_{12}$$

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

$$\beta_{22} = \beta_{11}$$

$$\beta_{12} = 1/2\ \beta_{11}.$$

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

Dependent		Independent		Constant	Comments (not on card)
6	6	6	5	1.0	$\beta_{22} = \beta_{11}$ for atom 6
6	8	6	5	0.5	$\beta_{12} = 0.5\ \beta_{11}$
0					End of list

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 mth site, a_{mn} the fractional occupancy for the nth species in the mth 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 $Fe_{6.2}Mg_{0.8}Si_8O_{22}(OH)_2$ and substitute into the above equation, we find that

$$4a_{1Mg} + 4A_{2Mg} + 2a_{3Mg} + 4a_{4Mg} = 1.6$$

Solving for a_{4Mg},

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

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

Dependent		Independent		Constant	Comments (not on card)
4	1	0	0	0.40	Initialization
4	1	1	1	-1.0	Site 1 dependence
4	1	2	1	-1.0	Site 2
4	1	3	1	-0.5	Site 3
0					End of list

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 $\underset{\sim}{T}$, $\underset{\sim}{L}$, and $\underset{\sim}{S}$ 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 $\underset{\sim}{T}$ and $\underset{\sim}{L}$ are given by Prince and Finger (1973), and the corresponding expressions including $\underset{\sim}{S}$ 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 $\underset{\sim}{T}$, 6 elements of $\underset{\sim}{L}$, and 9 elements of $\underset{\sim}{S}$, the constraint cards will be set up as follows: First, because of the so called "trace of $\underset{\sim}{S}$ singularity", the three diagonal elements of $\underset{\sim}{S}$ 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

```
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.
```

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 $\underset{\sim}{T}$, but may be dependent on any element of $\underset{\sim}{L}$ or $\underset{\sim}{S}$. 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 $\underset{\sim}{T}$, 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:

Column	Variable	Description
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.):

Column	Variable	Description
		1 - Ellipsoids computed for anisotropic atoms 2 - Ellipsoids and thermal corrections to bonds computed
6	MREF	Refinement type indicator 0 - Refine on F_2 1 - Refine on F^2
7	IFILE	Rewrite data set indicator 0 - Data set rewritten if this is the last cycle of the current run 1 - Data set not rewritten 2 - Data set rewritten. On input the standard deviations of parameters refined in previous cycles will be read in. 3 - Data set not rewritten. Standard deviations read in.
8	NEW(1)	If non-zero, read new cell parameters*
9	NEW(2)	If non-zero, read new bond distance limits
10	NEW(3)	If non-zero, read new symmetry operators
11	NEW(4)	If non-zero, read new scattering factors†
12	NEW(5)	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.
13	NEW(6)	If non-zero, read temperature factor type card
14	NEW(7)	If non-zero, read parameter selection card(s) and parameter dependency card(s)
15	NEW(8)	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.

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.

16-20	DMIN	Minimum bond distance in Angstroms. Needed only if NEW(2) $\neq$ 0.
21-25	DMAX	Maximum bond distance in Angstroms. Needed only if NEW(2) $\neq$ 0.

*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.):

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

3. Control Card (Cont.):

Column	Variable	Description
		NOTE: Values of IREJ from 5 to 9 allow the rejection mode to be changed even though NEW(8) = 0.
39-43	DELMAX	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.
44-53	JSCRF	Scale factor refinement switches (needed only if NEW(7) ≠ 0)
54-59	KEXTRF	Extinction factor refinement switches (needed only if NEW(7) ≠ 0)
60	JXPAR	Extra parameter indicator 0 - No extra parameters 1 - Extra parameters will be read (needed only if NEW(5) ≠ 0)
61	IFSGN	Orientation determination parameter 0 - Do not determine the orientation 1 - Change to orientation with lower weighted r (needed only if NEW(7) ≠ 0)

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

Columns	Variable	Description
1-8	A(1)	a axis length in angstroms
9-16	A(2)	b axis length
17-24	A(3)	c axis length
25-32	A(4)	Alpha in degrees
33-40	A(5)	Beta in degrees
41-48	A(6)	Gamma in degrees

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

Columns	Variable	Description
1-3	JTNS(1)	Translational part of x'
5-6	IRMAT(1,1)	First positional part of x'
8-9	IRMAT(2,1)	Second positional part of x'
11-13	JTNS(2)	Translational part of y'
15-16	IRMAT(1,2)	First positional part of y'
18-19	IRMAT(2,2)	Second positional part of y'
21-23	JTNS(3)	Translational part of z'
25-26	IRMAT(1,3)	First positional part of z'
28-29	IRMAT(2,3)	Second positional part of z'
34	IEF	End flag, non-zero on last card only 0 - Another symmetry card follows 1 - This is the last card

The translational operators are of the following form:
bbb 1/2 1/3 2/3 1/4 3/4 1/6 or 5/6. 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) ≠ 0.

Columns	Description
1-8	a_1
9-16	b_1
17-24	a_2
25-32	b_2
33-40	a_3
41-48	b_3
49-56	a_4
57-64	b_4
65-72	c
73-80	Identification

These coefficients must satisfy the equation

$$f(x) = c + \sum_{i=1}^{4} a_i \exp(-b_i x^2)$$

with $x = \sin\theta/\lambda$

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 (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) ≠ 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) ≠ 0 and NEWEXT ≠ 0. Format (6E12.6)

Columns	Value
1-12	r^* for isotropic extinction or r^*_{11} for anisotropic
13-24	r^*_{22}
25-36	r^*_{33}
37-48	r^*_{12}
49-60	r^*_{13}
61-72	r^*_{23}

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.

	Columns	Variable	Description
Card	1-6	TAG	Alphanumeric identifier
	7-12	SITE	Equipoint fraction
	13-18	OCCUP	Total occupancy of site
	19-24	OCCA	Occupancy of species A in this site Needed only if multiple occupancy
Card 2	1-2	ISCAT(1)	Number of scattering curve for species A
	3-4	ISCAT(2)	Number of scattering curve for species B (blank if single element in site).
	9-15	XYZ(1)	x of atom in fractional coordinates
	16-22	XYZ(2)	y
	23-29	XYZ(3)	z
	30-36	BETA(1)	B for an isotropic atom or β_{11} for an anisotropic atom
	37-43	BETA(2)	β_{22}
	44-50	BETA(3)	β_{33}
	51-57	BETA(4)	β_{12}
	58-64	BETA(5)	β_{13}
	65-71	BETA(6)	β_{23}
	73	IEF	Indicator of third cumulant
Card 3 - Needed if IEF ≠ 0.			
	1-8	C_{111}	
	9-16	C_{222}	
	17-24	C_{333}	
	25-32	C_{112}	
	33-40	C_{122}	
	41-48	C_{113}	
	49-56	C_{133}	
	57-64	C_{223}	
	65-72	C_{233}	
	73-78	C_{123}	
	80	IEF	Fourth cumulant ind.

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_{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-ω angle for transformation to standard, orthonormal coordinate system
5-χ angle for transformation
6-ϕ 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}
11-T_{22}
12-T_{33}
13-T_{12}
14-T_{13}
15-T_{23}
16-L_{11}
17-L_{22}
18-L_{33}
19-L_{12}
20-L_{13}
21-L_{23}
22-S_{11}
23-S_{22}
24-S_{33}
25-S_{12}
26-S_{13}
27-S_{23}
28-S_{21}
29-S_{31}
30-S_{32}

Columns 15-24 initial value of this parameter.

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

a)

Columns	Description
1-3	Integer specifying the dependent atom number.
5-6	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.

13. Parameter Constraint Cards(s) (Cont.):

a)

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

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) $\neq$ 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 F_{obs} and σ_F or other weight information. They may contain other information, such as a scale factor group identifier, the factor $\beta(\theta)$ 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 $\beta(\theta)$ will be computed according to the formula $\beta(\theta) = [(2\lambda^2/(V^2\sin^2\theta)] \times 10^8$, where V is the unit cell volume in $Å^3$. The refined quantity then will be the product $\bar{T}r^*$, with $\bar{T}$ 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.

Variable	Description
IH(1)	h for reflection
IH(2)	k
IH(3)	l
FOBS	Observed structure factor
SIGMA	Weight information consistent with IWT on Control Card
EXBETA	Factor $\beta(\theta)$ for extinction
I	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.
MREJ	Minimum Observable Parameter 1 - Intensity above minimum observable. 2 - Intensity below minimum observable.
IEF	End-of-list marker, zero on all data cards
TTH	2θ These angles are needed only if anisotropic
OMEG	ω extinction is to be calculated and must
CHI	χ conform to the diffractometer conventions
PHI	ϕ given by Busing and Levy (1967, p. 458).

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.

Column	Variable	Description
1	ICONT	0 - No more cycles. The data set on file ISCI will be updated if IFILE from the Control Card is zero 1 - Read a new title card and start another cycle
2	JPUN	0 - Do not punch any atom parameters 1 - Punch atom cards in form required by 'X-RAY67' 2 - Punch atom cards in form required by this program 3 - Punch atom cards in both forms
3-4	ISC1	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.

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)

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

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

This matrix is the matrix output by the lattice constant program of C. W. Burnham.

Column	Description
Card 1	
1-12	Variance of a (in angstroms**2)
13-24	Covariance of a with b
25-36	Covariance of a with c
37-48	Covariance of a with alpha (in angstrom-radians)
49-60	Covariance of a with beta
61-72	Covariance of a with gamma
Card 2	
1-12	Variance of b
13-24	Covariance of b with c
25-36	Covariance of b with alpha
37-48	Covariance of b with beta
49-60	Covariance of b with gamma
61-72	Variance of c
Card 3	
1-12	Covariance of c with alpha
13-24	Covariance of c with beta
25-36	Covariance of c with gamma
37-48	Variance of alpha (in radians**2)
49-60	Covariance of alpha with beta
61-72	Covariance of alpha with gamma
Card 4	
1-12	Variance of beta
13-24	Covariance of beta with gamma
25-36	Variance of gamma

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

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 0.4)

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

2. Parameter Card (cont.):

61-68	TMULT	Zero for difference Fourier. Multiplicity of a general reflection

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

Columns	Description
1-6	Multiplicity of h00 or 0k0
7-12	Multiplicity of 001
13-18	Multiplicity of hk0
19-24	Multiplicity of h01
25-30	Multiplicity of 0kl
31-36	Multiplicity of hh0
37-42	Multiplicity of h0h
43-48	Multiplicity of 0kk
49-54	Multiplicity of hhh
55-60	Multiplicity of hhl
61-66	Multiplicity of hkk or hkh
67-72	Multiplicity of hkl

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:

$$S = 3N_{ref} + 2(h'_{max} + 2)(2k'_{max} + 1)$$
$$+ 2(h'_{max} + 1)(P_x + 1)$$

where N_{ref} in the number of reflections included, h'_{max} is the maximum value of the index associated with the horizontal axis, k'_{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 $P_x P_y$. 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 of Cols. 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 product 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 F_{obs}

32 Number of print columns for F_{calc}

34 Number of print columns for σ_F

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 F_{calc}

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 F_{obs} 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

64 (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 σ_s 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 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 polynominal 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, CPLOT, 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)

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

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

Columns	Variable	Description
1-5	SBEG	Height of initial contour
6-10	CSTEP	Step between contours
11-15	SLIM	Height of final contour
16-20	NSIG	Number of digits after decimal point in contour label. If NSIG<0, contours will not be labelled.

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 $x_{i+1} \geq 0$. 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)

Columns	Variable	Description
1-5	INITZ	Sections from INITZ
6-10	ISTEPZ	to IFINZ in steps
11-15	IFINZ	of ISTEPZ will selected and contoured.

There should be NSECT of these cards.

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

Columns	Variable	Description
1-5	WX	x-grid coordinate
6-10	WY	y-grid coordinate
11-15	DX	Plotter x-axis offset
16-20	DY	Plotter y-axis offset
21-25	HGT	Label height in inches
26-31	LABL	Alphanumeric label

There should be NLABL of these cards.

Tape output for Calcomp plotter is on Unit 12.

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 $a_1x + a_2y + a_3z + a_4 = 0$ 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)

columns	Variable	Description
1-3	IO	If IO≠0, the output section will be written on Unit 12.
6	IZO	Section number of output
9	ISPC	If ISPC = 0 the output map will be single spaced; double spaced if ISPC=1.
12	IMRK	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.
13-15	IYI	Initial value of x in transformed section
16-18	IYF	Final value of x in transformed section
19-21	IZI	Initial value of y in transformed section
22-24	IZF	Final value of y in transformed section

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

Columns	Description
1-8	Length (in Å) of cell axis across page in original section
9-16	Length of cell axis down page in original section
17-24	Length of cell axis out of page in original section
25-32	Angle (in degrees) between axis 1 and 2
33-40	Angle between axes 1 and 3
41-48	Angle between axes 2 and 3

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)

Columns	Description
1-8	x- coordinate of first point
9-16	y- " " " "
17-24	z- " " " "
25-32	x- " " second "
33-40	y- " " " "
41-48	z- " " " "
49-56	x- " " third "
57-64	y- " " " "

4. Plane Card - (Cont.)

Columns	Description
65-72	z- coordinate of third point
74	Coordinate type indicator. If zero, the coordinates are given in grid spacings. If non-zero, the points are given in fractional coordinates.

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:

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

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.

Busing, W. R., and H. A. Levy (1962) A procedure for inverting large matrices. Comm. ACM, 5, 45-46.

_____ and _____ (1964) The effect of thermal motion on the estimation of bond lengths from diffraction measurements. Acta Crystallogr., 17, 142-146.

_____ and _____ (1967) Angle calculations for 3- and 4-circle x-ray and neutron diffractometers. Acta Crystallogr., 22, 457-464.

Coppens, P., and W. C. Hamilton (1970) Anisotropic extinction corrections in the Zachariasen approximation. Acta Crystallogr., A26, 71-83.

Cromer, D. T., and J. T. Waber (1965) Scattering factors computed from relativistic Dirac-Slater wave functions. (Acta Crystallogr., 18, 104-109.

_____ and J. B. Mann (1968) X-ray scattering factors computed from numerical Hartree-Fock wave functions. Acta Crystallogr., A24, 321-324.

Doyle, P. A., and P. S. Turner (1968) Relativistic Hartree-Fock X-ray and electron scattering factors. Acta Crystallogr., A24, 390-397.

Finger, L. W. (1969) Determination of cation distribution by least-squares refinement of single-crystal X-ray data. Carnegie Inst. Wash. Year Book 67, 216-217.

Forsyth, J. B., and M. Wells (1959). On an analytic approximation to the atomic scattering factor. Acta.Crystallogr., 12, 412.

Hamilton, W. C. (1959) On the isotropic temperature factor equivalent to a given anisotropic temperature factor. Acta crystallogr., 12, 609-610.

Levy, H. A. (1956) Symmetry relations among coefficients of the anisotropic temperature factor. Acta Crystallogr., 9, 679.

Nelmes, R. J., and Thornley, F. R. (1973) Anisotropic extinction in a synthetic boracite. To be published.

Zachariasen, W. H. (1967) A general theory of x-ray diffraction in crystals. Acta Crystallogr., 23, 558-564.

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 auxilliary functions called by them, including DIJKLM,DETERM, EL4,PIJKLM,BIJK,CIJKL,AIJ,VXV,SOLVE,EIJ,FIJ,IPAK,PDEGN,D3KDS, D2KDS,P3KDL,P2KDL,D2KDT,TKTLS,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 FORMAT (18A4)

Columns	Description
1 - 72	Identifier for problem. This should be any title which identifies the compound.

b. Cell Card FORMAT (6F10.5)

Columns	Description
1 - 10	a axis length, in Å
11 - 20	b axis length
21 - 30	c axis length
31 - 40	α in degrees
41 - 50	β
51 - 60	γ

c. Function Card FORMAT (5I4, 3X,A1, 6X, 3F10.5)

Columns	Description
1 - 4	Number of atoms in asymmetric unit (≤60)
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 FORMAT (3(A3,2(1X,A2),1X),I4)

Columns	Description
1 - 3	Translational part of x'
5 - 6	First positional part of x'
8 - 9	Second positional part of x'
11 - 13	Translational part of y'
15 - 16	First positional part of y'
18 - 19	Second positional part of y'
21 - 23	Translational part of z'
25 - 26	First positional part of z'
28 - 29	Second positional part of z'

Columns	Description
34	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	$\underline{a}$ with γ	
9 - 16	$\underline{b}$ with γ	
17 - 24	$\underline{c}$ with γ	
25 - 32	α with γ	
33 - 40	β with γ	
41 - 48	γ with γ	(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:

Columns	Description
1 - 6	Identification for atom
14 - 20	$\underline{x}$ in fractional coordinates
21 - 27	$\underline{y}$ in fractional coordinates
28 - 34	$\underline{z}$ in fractional coordinates
36	Temperature factor type 0 - isotropic 1 - anisotropic

Temperature factor card:

1 - 8	β_{11} or B
9 - 16	β_{22}
17 - 24	β_{33}
25 - 32	β_{12}
33 - 40	β_{13}
41 - 48	β_{23}

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

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

h. Temperature Factor Standard Deviations FORMAT (6F10.8)

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

Columns	Description
1 - 10	Standard Deviation of β_{11}
11 - 20	Standard Deviation of β_{22}
21 - 30	Standard Deviation of β_{33}

Temperature Factor Standard Deviations (Cont)

Columns	Description
31 - 40	Standard Deviation of β_{12}
41 - 50	Standard Deviation of β_{13}
51 - 60	Standard Deviation of β_{23}

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.

Columns	Correlation Coefficient of	
Card 1:		
1 - 8	β_{11} with β_{11}	(1.0)
Card 2:		
1 - 8	β_{11} with β_{22}	
9 - 16	β_{22} with β_{22}	(1.0)
Card 3:		
1 - 8	β_{11} with β_{33}	
9 - 16	β_{22} with β_{33}	
17 - 24	β_{33} with β_{33}	(1.0)
Card 4:		
1 - 8	β_{11} with β_{12}	
9 - 16	β_{22} with β_{12}	
17 - 24	β_{33} with β_{12}	
25 - 32	β_{12} with β_{12}	(1.0)
Card 5:		
1 - 8	β_{11} with β_{13}	
9 - 16	β_{22} with β_{13}	
17 - 24	β_{33} with β_{13}	
25 - 32	β_{12} with β_{13}	
33 - 40	β_{13} with β_{13}	(1.0)
Card 6:		
1 - 8	β_{11} with β_{23}	
9 - 16	β_{22} with β_{23}	
17 - 24	β_{33} with β_{23}	
25 - 32	β_{12} with β_{23}	
33 - 40	β_{13} with β_{23}	
41 - 48	β_{23} with β_{23}	(1.0)

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.

Standard Deviations of Atomic Positions (Cont)

Columns	Description
1 - 10	Standard deviation of x
11 - 20	Standard deviation of y
21 - 30	Standard deviation of z

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

k. Atomic Positions Correlation Matrix FORMAT (10F8.5)

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.

Columns	Correlation Coefficient of	
Card 1:		
1 - 8	x of atom 1 with x of atom 1	(1.0)
Card 2:		
1 - 8	x of atom 1 with y of atom 1	
9 - 16	y of atom 1 with y of atom 1	(1.0)
Card 3:		
1 - 8	x of atom 1 with z of atom 1	
9 - 16	y of atom 1 with x of atom 2	
17 - 24	z of atom 1 with z of atom 1	(1.0)
Card 4:		
1 - 8	x of atom 1 with x of atom 2	
9 - 16	y of atom 1 with x of atom 2	
17 - 24	x of atom 1 with x of atom 2	
25 - 32	x of atom 2 with x of atom 2	(1.0)

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:

$$Y = RX + T \qquad (1)$$

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

$$Z = X - Y \tag{2}$$

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

$$g_{ij} = \underline{a}_i \cdot \underline{a}_j \tag{3}$$

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

$$G = \begin{pmatrix} a^2 & ab\cos\gamma & ac\cos\beta \\ ab\cos\gamma & b^2 & bc\cos\alpha \\ ac\cos\beta & bc\cos\alpha & c^2 \end{pmatrix} \tag{4}$$

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:

$$s^2 = Z^T G Z \tag{5}$$

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 Θ for the angle $Y_1 - X - Y_2$ can be shown to satisfy the following relationship:

$$\cos \Theta = \frac{(Y_1-X)^T \, G(Y_2-X)}{S_{1X} \, S_{2X}} \qquad (6)$$

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:

$$(B - \lambda G^{-1})Q = 0 \qquad (7)$$

where B is the matrix of anisotropic temperature coefficients

λ is a characteristic value of the equation

G^{-1} is the reciprocal space metric tensor with elements

$g_{ij}^{-1} = \underline{b}_i \cdot \underline{b}_j$ where the $\underline{b}$'s are the reciprocal space axis vectors and

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.

$$(GB - \lambda I)Q = 0 \qquad (8)$$

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:

$$\det (GB - \lambda I) = 0 \qquad (9)$$

Solution of this equation gives three values of λ which satisfy equation (8). In order to have a meaningful physical situation, it is necessary that the λ'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 $\underline{q}$ and the angle, θ_i, between this vector $\underline{q}$ and the real space axis vector $\underline{a}_i$ may be found by forming the dot product of the two vectors:

$$\underline{a}_i \cdot \underline{q} = |\underline{a}_i||\underline{q}| \cos \theta_i \qquad (10)$$

This equation may be simplified since

$$\underline{a}_i \cdot \underline{q} = \underline{a}_i \cdot (q_1\underline{b}_1 + q_2\underline{b}_2 + q_3\underline{b}_3) \qquad (11)$$

and

$$\underline{a}_i \cdot \underline{b}_j = w_{ij} = \begin{matrix} 1 \text{ for } i = j \\ 0 \text{ for } i \neq j \end{matrix} \qquad (12)$$

thus

$$\underline{a}_i \cdot \underline{q} = q_i \qquad (13)$$

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

$$\cos \theta_i = \frac{q_i}{|\underline{a}_i||\underline{q}|} \qquad (14)$$

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

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

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:

$$y = y(x_1, x_2, \ldots\ldots, x_h) \qquad (16)$$

then the variance of y, denoted by σ_y^2 is found from the equation

$$\sigma_y^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial y}{\partial x_i} \frac{\partial y}{\partial x_j} \operatorname{cov}(x_i, x_j) \qquad (17)$$

where $\operatorname{cov}(x_i, x_j)$ 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:

$$\sigma_y^2 = \sum_{i=1} \sum_{j=1} \frac{\partial y}{\partial a_i} \frac{\partial y}{\partial a_j} \operatorname{cov}(a_i, a_j)$$

$$+ \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{\partial y}{\partial z_i} \frac{\partial y}{\partial z_j} \operatorname{cov}(z_i, z_j) \qquad (18)$$

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

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

H. PROGRAM NAMES AND DESCRIPTION

1. U M BADTEA:

   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     MAIN PROGRAM FOR RFINE                                                    ENGL  10
      INCLUDE LSMAT,LIST                                                        ENGL  20
      INCLUDE TAPE,LIST                                                         ENGL  30
      CALL CORSZ(ACI))                                                          ENGL  40
      WRITE(IBUT,2)                                                             ENGL  50
    2 FORMAT(' 14 FEBRUARY 1974 VERSION OF RFINE, WITH PROVISION FOR GENENGL  60
     1ERAL CONSTRAINTS.'/'     THIS VERSION WILL HANDLE 200 PARAMETERS.')ENGL  70
      READ(IN,4)LABEL                                                           ENGL  80
    4 FORMAT(3A4)                                                               ENGL  90
      CALL RFINE                                                                ENGL 100
      WRITE(IBUT,6)ISCI                                                         ENGL 110
    6 FORMAT('1PARAMETERS WRITTEN OUT TO UNIT ',I2,'.')                         ENGL 120
      STOP                                                                      ENGL 130
      END                                                                       ENGL 140
```

```
      SUBROUTINE RFINE                                                          RFNE  10
                                                                                RFNE  20
C     THIS ROUTINE REVISED  APRIL 12, 1972                                      RFNE  30
                                                                                RFNE  40
C           STRUCTURE FACTOR CALCULATION AND LEAST SQUARES REFINEMENT           RFNE  50
C           MAXIMUM OF 60 ATOMS, 10 SCALE FACTORS,                              RFNE  60
C            20 SCATTERING FACTORS, 200 LEAST-SQUARES PARAMETERS                RFNE  70
C            AND 24 SYMMETRY OPERATIONS                                         RFNE  80
C                                                                               RFNE  90
C           JULY  1966    BY L. W. FINGER                                       RFNE 100
C           MODIFIED NOV 7,1966 FOR BLOCKED STRUCTURE FACTOR RECORDS            RFNE 110
C           MODIFIED JAN 30,1967 FOR EXTINCTION CORRECTION                      RFNE 120
C           SPECIAL PARAMETER REFINEMENT ADDED FEB. 1972                        RFNE 130
                                                                                RFNE 140
      DIMENSION HEAD(36),IBOFF(1023),IRMAT(2,3),ITNS(8),JRMAT(10),JTNS(3RFNE 150
     A),KREJ(4),KRMAT(10),NEW(8),R(36),TRS(8),FBUF(504),IFBUF(504)              RFNE 160
      DIMENSION EGGY(6)                                                         RFNE 170
      DIMENSION                                                                 RFNE 180
     $    BBETA(6,60),SIGB(6,60),SIGU(60),SIGXYZ(3,60)                         RFNE 190
      DIMENSION H(B(20),KB(20),JB(20)                                           RFNE 200
      DIMENSION TFB(6)                                                          RFNE 210
                                                                                RFNE 220
      INCLUDE ACOM,LIST                                                         RFNE 230
      COMMON/C/CELLP(3,4),ICENT,JCELL,NSYM,RMAT(3,3,24),TRANS(3,24)             RFNE 240
      INCLUDE DCOM,LIST                                                         RFNE 250
      COMMON/E/DMAX,OMIN,RECELL(6),EXANG(3)                                     RFNE 260
      COMMON/F/DELTA,FCALC,FOBS,FOMAX,FOMIN,IREJ,MREF,MREJ,NREJ,NOMBER(3RFNE 270
     I 6),ROEN(36),RHO,RHOMN,RHOMX,RNOM(36),WT,EXBETA,EXX,EXY                   RFNE 280
     2,F(SGN,IFSGN,EXTIN(6),KEXTRF(6),OELMAX                                    RFNE 290
      COMMON/G/BOFF(1023),FCURVE(9,20),FRMT(18),IWT,NSCAT,BETFAC,WVLNTH RFNE 300
     1,WTFAC                                                                    RFNE 310
      INCLUDE LSMAT,LIST                                                        RFNE 320
      INCLUDE TAPE,LIST                                                         RFNE 330
      INCLUDE XPAR,LIST                                                         RFNE 340
                                                                                RFNE 350
      EQUIVALENCE (IBUFF,BOFF),(K,DAOX),(FBUF,IFBUF)                            RFNE 360
      EQUIVALENCE (H(B                                                          RFNE 370
     1,DAOO),(KB,DADO(21)),(JB,DADO(41))                                        RFNE 380
                                                                                RFNE 390
      DATA (TNS/3H   ,3H1/2,3H1/3,3H2/3,3H1/4,3H3/4,3H1/6,3H5/6/                RFNE 400
      DATA TRS/0.0,0.5,0.3333333333,0.6666666667,0.25,0.75,0.1666666667,RFNE 410
     I 0.8333333333/                                                            RFNE 420
      DATA JRMAT/3H   ,3H  X,3H +X,3H -X,3H  Y,3H +Y,3H -Y,3H  Z,3H +Z,  RFNE 430
     1 3H -Z/                                                                   RFNE 440
      DATA KRMAT/0,1,1,-1,2,2,-2,3,3,-3/                                        RFNE 450
      DATA KREJ/2H  ,2H* ,2H  ,2H**/                                            RFNE 460
      DATA HEAD/1HX,1HY,1HZ,1H0,2H11,2H22,2H33,2H12,2H13,2H23,3H111,3H22RFNE 470
     A2,3H333,3H112,3H122,3H113,3H133,3H223,3H233,3H123,4H1111,4H2222,  RFNE 480
     B4H3333,4H1112,4H1113,4H1122,4H1123,4H1133,4H1222,4H1223,4H1233,4H1RFNE 490
     C333,4H2223,4H2233,4H2333,1H /                                             RFNE 500
                                                                                RFNE 510
    2 FORMAT(18A4)                                                              RFNE 520
    4 FORMAT(15I1,2F5.0,3I1,3(2,4I1,F5.0,1B11)                                  RFNE 530
    5   FORMAT('0 CONTROL PARAMETERS ARE  '/1512,2F7.2, 3I2,3I3,4I2,F7.2,  RFNE 540
     $1BI2)                                                                     RFNE 550
    6 FORMAT(9F8.0,2A4)                                                         RFNE 560
    7 FORMAT('0SYMMETRY CARDS'/)                                                RFNE 570
```

```
    8 FORMAT(1H1,18A4)                                                  RFNE 580
    9 FORMAT('OREAL CELL'8X,6F10.5)                                     RFNE 590
   10 FORMAT('ORECIPROCAL CELL'2X,6F10.5)                               RFNE 600
   11 FORMAT('OSCALE FACTURS - '10F9.4)                                 RFNE 610
   12 FORMAT(3(3A3,1X),I4)                                              RFNE 620
13    FORMAT(5X,3(3A3,1X),I4)                                          RFNE 630
   14 FORMAT(7F10.0)                                                    RFNE 640
   15 FORMAT(1X,2A4,9F11.6)                                             RFNE 650
   16 FORMAT(2A3,3F6.4/212,4X,9F7.5,I2)                                 RFNE 660
   17 FORMAT('+',107X,2F9.3)                                            RFNE 670
   18 FORMAT(80I1)                                                      RFNE 680
   19 FORMAT( 'OATOM LABEL      X',7X,'Y',7X,'Z',9X,'TEMPERATURE FACTUR ORFNE 690
     $ATA'//(6X,2A3,3F8.4,6F9.5))                                       RFNE 700
   20 FORMAT(1H1,18A4,5X,17HSTROCTURE FACTURS,10X,4HPAGE,I3/1HO,7X,1HH, RFNE 710
     1 4X,1HK,4X,1HL,6X,6HF(OBS),5X,7HF(CALC),5X,7HA(CALC),5X,7H8(CALC),RFNE 720
     2 5X,7HDELTA F,3X,11HDELTA/SIGMA,3X,11HEXT. FACTUR/)               RFNE 730
   21 FORMAT('OATOM LABEL   TF  SCAT.  SITE  OCCOP  OCCA'//(6X,2A3,3I3,3RFNE 740
     $F7.3))                                                            RFNE 750
   22 FORMAT(4X,3I5,F12.3,1X,A1,F10.3,3F12.3,2F14.4,2X,2A2)             RFNE 760
   23 FORMAT(5X,80I1)                                                   RFNE 770
90123 FORMAT('OPARAMETER SELECTION INFORMATION')                        RFNE 780
   24 FORMAT(1H1,18A4,5X,40HRESULTS OF STRUCTURE FACTOR CALCULATIONS)   RFNE 790
   25 FORMAT('OSPECIAL PARAMETERS'/(6I2X,2A4,F10.5))                    RFNE 800
   26 FORMAT(10X,10HWEIGHTED R,4X,2F15.2,110,F11.3//10X,12HUNWEIGHTED R,RFNE 810
     1 2X,2F15.2,110,F11.3//14X,16HRANGES OF F(OBS)/8I24X,2F15.2,I10,F11RFNE 820
     2 .3//14X,32HRANGES OF (SIN(THETA)/LAMBDA)**2/6(24X,2F15.2,I10,F11RFNE 830
     3 .3/))                                                            RFNE 840
   27 FORMAT(16E12.6)                                                   RFNE 850
   28 FORMAT(16HOALL REFLECTIONS,14X,9HNUMERATOR,4X,11HDENOMINATOR,4X,  RFNE 860
     1 6HNUMBER,10X,1HR/1H )                                            RFNE 870
   29 FORMAT('OEXTINCTION PARAMETERS - '6E14.5)                         RFNE 880
   30 FORMAT(23H ONREJECTED REFLECTIONS/1H )                            RFNE 890
   31 FORMAT('OSCATTERING CURVES'/' LABEL  ',4(5X,'A(',I1,')',7X,'B(',I1,RFNE 900
     $')',2X),5X,'C',10X,'FR',6X,'FI'/)                                 RFNE 910
   32 FORMAT( /10X,8HSUM FCAL,20X,F15.2/ 10X,31HSTANDARD DEV OF UNIT WEIRFNE 920
     1GHT OBS,F12.2)                                                    RFNE 930
   33 FORMAT('OINPUT PARAMETERS READ FROM FILE WITH LABEL **'3A4,'**')  RFNE 940
   34 FORMAT(13H MATRIX RANK I3,12H INSTEAD OF I3/17H CYCLE TERMINATED) RFNE 950
   36 FORMAT(15X,4HATOM,11X,9HPARAMETER,20X,3HOLD,10X,6HCHANGE,9X,3HNEW,RFNE 960
     1 10X,19HERROR   SHIFT/ERROR/)                                     RFNE 970
   38 FORMAT(7HO  ATOM,I3,4X,2A3,5X,2IHEQUIPOINT FRACTION = F6.3)       RFNE 980
   40 FORMAT(/27X,'SCALE FACTOR',I3,9X,F12.4,F15.4,F13.4,F14.5,F14.2,I5)RFNE 990
   44 FORMAT(27X,'OCCUPANCY SPECIES',I3,F16.5,F15.5,F13.5,F14.5,F14.2,I5RFNE1000
     1)                                                                 RFNE1010
   46 FORMAT(27X,17HOCCUPANCY SPECIESI3,F16.5,15X,F13.5,F14.7)          RFNE1020
   48 FORMAT(27X,25HTOTAL OCCUPANCY (FIXED) =F11.5)                     RFNE1030
   50 FORMAT(27X,A1,20X,2F15.7,F13.7,F14.7,F14.2,I5)                    RFNE1040
   51 FORMAT(27X,'BETA',A2,15X,2F15.7,F13.7,F14.7,F14.2,I5)             RFNE1050
   52 FORMAT(27X,A1,20X,F15.7,15X,F13.7,F14.7)                          RFNE1060
   53 FORMAT(27X,4HBETAA2,15X,F15.7,15X,F13.7,F14.7)                    RFNE1070
   54 FORMAT(1H1,18A4//10X,18HCORRELATION MATRIX/1H )                   RFNE1080
   56 FORMAT(15HO ROW,I4/(10(I4,F8.4)))                                RFNE1090
   57 FORMAT(1H118A4/'OCORRELATION COEFICIENTS WITH LARGEST MAGNITUDES'/RFNE1100
     $'4' PARAMETERS   VALUE    ')/(4(2I6,F9.4,2X)))                   RFNE1110
   58 FORMAT(I2)                                                        RFNE1120
   59 FORMAT(3X,'EXTINCTION CORRECTION')                                RFNE1130
   60 FORMAT(15X,'ISOTROPIC')                                           RFNE1140
   63 FORMAT(15X,'ANISOTROPIC - TYPE 'I1)                               RFNE1150
   64 FORMAT(31HOERROR IN SYMMETRY OPERATOR NO.,I3,1H.//16H JOB TERMINATRFNE1160
     1ED.)                                                              RFNE1170
   65 FORMAT(27X,'R-STAR 'A2,17X,E10.4,18X,E10.4)                       RFNE1180
   67 FORMAT(27X,'R-STAR ',A2,12X,2(5X,E10.4),3X,E10.4,4X,E10.4,F14.2,I5RFNE1190
     1)                                                                 RFNE1200
   68 FORMAT(25HOTOO MANY SYMMETRY CARDS.//16H JOB TERMINATED.)         RFNE1210
   70 FORMAT(63HOERROR IN NUMBER OF SCATTERING CURVES, SCALE FACTORS, ORRFNE1220
     1 ATOMS.//16H JOB TERMINATED.)                                     RFNE1230
   72 FORMAT(413,F8.0)                                                  RFNE1240
   74 FORMAT(///25X,22HMAXIMOM SHIFT/ERROR = F10.2//25X,22HAVERAGE SHIFTRFNE1250
     1/ERROR = F10.2)                                                   RFNE1260
   76 FORMAT(44H ESTIMATED STD. DEV. FOR OBS. OF UNIT WEIGHT5X,F12.2/)  RFNE1270
   78 FORMAT(/27X,12HSCALE FACTOR13,9X,F12.4,15X,F13.4)                 RFNE1280
   80 FORMAT(27X,'C(',A3,')',15X,2(6X,E9.4),4X,E9.4,5X,E9.4,F14.2,I5)   RFNE1290
   82 FORMAT(27X,'C(',A3,')',21X,E9.4,19X,E9.4,5X,E9.4)                 RFNE1300
   84 FORMAT(9F8.0,F7.0,I1)                                             RFNE1310
   85 FORMAT(10F8.0/5F8.0)                                              RFNE1320
   86 FORMAT(27X,'D(',A4,')',14X,2F15.9,F13.9,F14.9,F14.2,I5)           RFNE1330
   88 FORMAT(27X,'D(',A4,')',20X,E9.4,19X,E9.4,5X,E9.4)                 RFNE1340
   90 FORMAT(I4)                                                        RFNE1350
   95 FORMAT(/3X,'SPECIAL PARAMETERS')                                  RFNE1360
   96 FORMAT(27X,2A4,13X,2F15.8,F13.8,F14.8,F14.2,I5)                   RFNE1370
   97 FORMAT(27X,2A4,13X,F15.8,15X,F13.8,F14.8)                         RFNE1380
   98 FORMAT(/9X,'PRESENT ORIENTATION IS'F4.0,'  WTD R FOR THE OPPOSITE RFNE1390
     $ORIENTATION IS 'F7.4,'  THE NEW ORIENTATION IS'F4.0)              RFNE1400
C               INITIALIZE                                             RFNE1410
      PI=ACOS(-1.0)                                                     RFNE1420
      TEMPZZ=8.*PI*PI                                                   RFNE1430
      NPAR=0                                                            RFNE1440
      FISGN=1.0                                                         RFNE1450
      MAXPG=53                                                          RFNE1460
      IFIRST=-1                                                         RFNE1470
      REWIND ISC1                                                       RFNE1480
      REWIND ISC2                                                       RFNE1490
C               STARTING POINT FOR CYCLE                               RFNE1500
                                                                        RFNE1510
  100 READ(IN,2)TITLE                                                   RFNE1520
      WRITE(IOUT,8)TITLE                                                RFNE1530
      READ(IN,4)MFSEL,MSF,MCURR,MBOOAN,MELVIB,MREF,IFILE,NEW,(80FF(I),I=RFNE1540
     $1,13),(180FF(I),I=15,32)                                          RFNE1550
      WRITE(IOUT,5)MFSEL,MSF,MCURR,MBOOAN,MELVIB,MREF,IFILE,NEW,(80FF(I)RFNE1560
     $,I=1,13),(80FF(I),I=15,32)                                        RFNE1570
      IFIRST=IFIRST+1                                                   RFNE1580
      N80FF=0                                                           RFNE1590
      NFBR=0                                                            RFNE1600
C               CHECK IF FIRST CYCLE OF RUN                            RFNE1610
      IF(IFIRST)140,110,140                                             RFNE1620
  110                 DO 120 I=1,8                                      RFNE1630
      IF(NEW(I))120,130,120                                             RFNE1640
  120 CONTINUE                                                          RFNE1650
C               ALL PARAMETERS NEW, DO NOT READ SAVE UNIT              RFNE1660
      GO TO 140                                                         RFNE1670
C               READ SAVE UNIT TO START OF REFLECTION LIST             RFNE1680
  130 READ(ISC1)                                                        RFNE1690
     1KCYL,LABEL,RECELL,G,VOL,NSYM,NATOM,NSCAT,JCELL,MRANK,NPARA,NSF,OMIRFNE1700
     2N,OMAX,DELMAX,ICENT,(((RMAT(I,J,K),I=1,3),J=1,3),K=1,NSYM),(((TRANSRFNE1710
     3(I,J),I=1,3),J=1,NSYM),((CELLP(I,J),I=1,3),J=1,JCELL),((IPARA(I),I=RFNE1720
     41,NPARA),((FCURVE(I,J),I=1,9),J=1,NSCAT),(SCALE(I),I=1,NSF),IREJ,FRFNE1730
```

```
     5OMAX,FOMIN,RHOMN,RHOMX,(JSCRF(I),I=1,NSF),NRSF                    RFNE1740
     9,NPAR,(PAR(I),(PARNAM(J,I),J=1,2),IPAR(I),IRPAR(I),(=1,NPAR),FISGNRFNE1750
     0,IFSGN,EGGY,KEXTRF                                               RFNE1760
      READ(ISC1 )                                                      RFNE1770
     1((ANOM(J,I),J=1,2),I=1,NSCAT),IEXT,EXTIN,((TAG(J,I),J=1,2),ISOT(I))RFNE1780
     2,(ISCAT(J,I),J=1,2),OCCA(I),OCCUP(I),SITE(I),(XYZ(J,I),J=1,3),   RFNE1790
     3(BETA(J,I),J=1,6),(CR(J,I),J=1,10),(OR(J,I),J=1,15),I=1,NATOM),  RFNE1800
     4LDEGN,(KDEGN(I),I=1,LOEGN)                                       RFNE1810
      NOEGN=LDEGN/2                                                    RFNE1820
      WRITE(IOUT,33)LABEL                                              RFNE1830
C             CHECK FOR NEW PARAMETERS                                  RFNE1840
  140 IF(NEW(1))150,180,150                                            RFNE1850
C             READ NEW CELL PARAMETERS AND CALCULATE RECIPROCAL CELL    RFNE1860
  150 READ(IN,6)(A(I),I=1,6)                                           RFNE1870
      WRITE(IOUT,9)(A(I),I=1,6)                                        RFNE1880
                                  DO 160 I=4,6                         RFNE1890
      IF(ABS(A(I)-90.0)-0.001)154,155,155                              RFNE1900
  154 A(I)=0.0                                                         RFNE1910
      GO TO 160                                                        RFNE1920
  155 A(I)=COS(A(I)*PI/180.0)                                          RFNE1930
  160 CONTINUE                                                         RFNE1940
      II=0                                                             RFNE1950
                                  DO 170 I=1,2                         RFNE1960
      II=II+1                                                          RFNE1970
      G(I,I)=A(I)**2                                                   RFNE1980
      B(II)=G(I,I)                                                     RFNE1990
      IJ=I+1                                                           RFNE2000
                                  DO 170 J=IJ,3                        RFNE2010
      II=II+1                                                          RFNE2020
      IK=9-I-J                                                         RFNE2030
      G(I,J)=A(I)*A(J)*A(IK)                                           RFNE2040
      G(J,I)=G(I,J)                                                    RFNE2050
  170 B(II)=G(I,J)                                                     RFNE2060
      G(3,3)=A(3)**2                                                   RFNE2070
      B(6)=G(3,3)                                                      RFNE2080
      CALL SYMINV(B,3,A,0,A,1.0E-10,VOL,JRANK)                         RFNE2090
      VOL=SQRT(VOL)                                                    RFNE2100
      RECELL(1)=SQRT(B(1))                                             RFNE2110
      RECELL(2)=SQRT(B(4))                                             RFNE2120
      RECELL(3)=SQRT(B(6))                                             RFNE2130
      RECELL(4)=B(5)/(RECELL(2)*RECELL(3))                             RFNE2140
      RECELL(5)=B(3)/(RECELL(1)*RECELL(3))                             RFNE2150
      RECELL(6)=B(2)/(RECELL(1)*RECELL(2))                             RFNE2160
      EGGY(1)=4./RECELL(1)**2                                          RFNE2170
      EGGY(2)=4./RECELL(2)**2                                          RFNE2180
      EGGY(3)=4./RECELL(3)**2                                          RFNE2190
      EGGY(4)=4./(RECELL(1)*RECELL(2))                                 RFNE2200
      EGGY(5)=4./(RECELL(1)*RECELL(3))                                 RFNE2210
      EGGY(6)=4./(RECELL(2)*RECELL(3))                                 RFNE2220
      WRITE (IOUT,10)RECELL                                            RFNE2230
  180 IF(IFIRST.EQ.0.AND.NPAR.NE.0)CALL TRNFRM                         RFNE2240
      IF(NEW(2))190,200,190                                            RFNE2250
C             NEW BONO DISTANCE LIMITS                                  RFNE2260
  190 OMIN=BUFF(1)                                                     RFNE2270
      DMAX=BUFF(2)                                                     RFNE2280
  200 IF(NEW(3))210,420,210                                            RFNE2290
C             NEW SYMMETRY OPERATORS                                    RFNE2300
  210 ICENT=IBUFF(3)                                                   RFNE2310
      WRITE(IOUT,7)                                                    RFNE2320
                                  DO 220 I=1,3                         RFNE2330
  220 CELLP(I,1)=0.0                                                   RFNE2340
      JCELL=1                                                          RFNE2350
      ICELL=IBUFF(4)                                                   RFNE2360
      IF(ICELL)230,320,230                                             RFNE2370
  230 IF(ICELL-5)232,270,300                                           RFNE2380
  232                             DO 240 I=1,3                         RFNE2390
  240 CELLP(I,2)=0.5                                                   RFNE2400
      JCELL=2                                                          RFNE2410
      IF(ICELL-4)250,320,250                                           RFNE2420
  250 CELLP(ICELL,2)=0.0                                               RFNE2430
      GO TO 320                                                        RFNE2440
  270                             DO 290 I=2,4                         RFNE2450
                                  DO 280 J=1,3                         RFNE2460
  280 CELLP(J,I)=0.5                                                   RFNE2470
  290 CELLP(I-1,I)=0.0                                                 RFNE2480
      JCELL=4                                                          RFNE2490
      GO TO 320                                                        RFNE2500
  300 JCELL=3                                                          RFNE2510
                                  DO 310 I=2,3                         RFNE2520
      CELLP(1,I)=FLOAT(I-1)/3.0                                        RFNE2530
      CELLP(2,I)=FLOAT(4-I)/3.0                                        RFNE2540
  310 CELLP(3,I)=CELLP(2,I)                                            RFNE2550
C             READ NEW SYMMETRY CARDS                                   RFNE2560
  320 NSYM=0                                                           RFNE2570
  330 NSYM=NSYM+1                                                      RFNE2580
      READ(IN,12)(JTNS(I),(IRMAT(K,I),K=1,2),I=1,3),IEF                RFNE2590
      WRITE(IOUT,13)(JTNS(I),(IRMAT(K,I),K=1,2),I=1,3),IEF             RFNE2600
                                  DO 340 I=1,3                         RFNE2610
                                  DO 340 J=1,8                         RFNE2620
      IF(JTNS(I)-ITNS(J))340,350,340                                   RFNE2630
  340 CONTINUE                                                         RFNE2640
  342 WRITE(IOUT,64)NSYM                                               RFNE2650
      CALL EXIT                                                        RFNE2660
  350 TRANS(I,NSYM)=TRS(J)                                             RFNE2670
                                  DO 390 J=1,3                         RFNE2680
      RMAT(J,I,NSYM)=0.0                                               RFNE2690
                                  DO 390 K=1,2                         RFNE2700
      IJ=IRMAT(K,J)                                                    RFNE2710
                                  DO 360 L=1,10                        RFNE2720
      IF(JRMAT(L)-IJ)360,370,360                                       RFNE2730
  360 CONTINUE                                                         RFNE2740
      GO TO 342                                                        RFNE2750
  370 IF(I-IABS(KRMAT(L)))390,380,390                                  RFNE2760
  380 RMAT(J,I,NSYM)=ISIGN(1,KRMAT(L))                                 RFNE2770
  390 CONTINUE                                                         RFNE2780
      IF(IEF)420,400,420                                               RFNE2790
  400 IF(NSYM-24)330,410,410                                           RFNE2800
  410 WRITE(IOUT,68)                                                   RFNE2810
      CALL EXIT                                                        RFNE2820
  420 IF(NEW(4))430,460,430                                            RFNE2830
C             READ NEW SCATTERING FACTORS                               RFNE2840
  430 NSCAT=IBUFF(6)                                                   RFNE2850
      WRITE(IOUT,31)(I,I,I=1,4)                                        RFNE2860
      IF(NSCAT*(NSCAT-21))434,432,432                                  RFNE2870
  432 WRITE(IOUT,70)                                                   RFNE2880
      CALL EXIT                                                        RFNE2890
```

```
  434                        DO 450 I=1,NSCAT                                  RFNE2900
      ANOM(1,I)=0.0                                                           RFNE2910
      ANOM(2,I)=0.0                                                           RFNE2920
      READ(IN,6)(FCURVE(J,I),J=1,9),L,K                                       RFNE2930
      WRITE(IOUT,15)L,K,(FCURVE(J,I),J=1,9)                                   RFNE2940
      IF(IBUFF(5))440,450,440                                                 RFNE2950
  440 READ(IN,6)(ANOM(J,I),J=1,2)                                             RFNE2960
      WRITE(IOUT,17)(ANOM(J,I),J=1,2)                                         RFNE2970
  450 CONTINUE                                                                RFNE2980
  460 IF(NEW(5))470,540,470                                                   RFNE2990
C         CHECK IF SPECIAL PARAMETERS ONLY                                    RFNE3000
  470 IF(NEW(5).EQ.4)GO TO 479                                                RFNE3010
C         READ NEW SCALE, EXTINCTION AND ATOM PARAMETERS                      RFNE3020
      NATOM=IBUFF(7)                                                          RFNE3030
      NSF=IBUFF(8)                                                            RFNE3040
      IF(NSF*(NSF-11))471,432,432                                             RFNE3050
  471 IF(NATOM*(NATOM-61))472,432,432                                         RFNE3060
  472 READ(IN,6)(SCALE(I),I=1,NSF)                                            RFNE3070
      WRITE(IOUT,11)(SCALE(I),I=1,NSF)                                        RFNE3080
      IF(IBUFF(9).EQ.0)GO TO 4720                                             RFNE3090
C         READ EXTINCTION PARAMETERS                                          RFNE3100
      READ(IN,27)EXTIN                                                        RFNE3110
      WRITE(IOUT,29)EXTIN                                                     RFNE3120
 4720 IF(NEW(5)-2)473,476,480                                                 RFNE3130
C         READ ENTIRE ATOM LIST                                               RFNE3140
  473 DO 475 I=1,NATOM                                                        RFNE3150
      READ(IN,16)(TAG(J,I),J=1,2),SITE(I),OCCUP(I),OCCA(I),(ISCAT(J,I)        RFNE3160
     A,J=1,2),(XYZ(J,I),J=1,3),(BETA(J,I),J=1,6),IEF                          RFNE3170
      IF(IEF)474,475,474                                                      RFNE3180
C         READ THIRD CUMULANT COEFFICIENTS                                    RFNE3190
  474 READ((N,84)(CR(J,I),J=1,10),IEF                                         RFNE3200
C         READ FOURTH CUMULANT COEF.                                          RFNE3210
      IF(IEF.NE.0)READ(IN,85)(OR(J,I),J=1,15)                                 RFNE3220
  475 CONTINUE                                                                RFNE3230
      GO TO 479                                                               RFNE3240
C         READ SELECTED ATOMS                                                 RFNE3250
  476 READ(IN,58)I                                                            RFNE3260
      IF(I)480,480,477                                                        RFNE3270
  477 READ(IN,16)(TAG(J,I),J=1,2),SITE(I),OCCUP(I),OCCA(I),(ISCAT(J,I)        RFNE3280
     A,J=1,2),(XYZ(J,I),J=1,3),(BETA(J,I),J=1,6),IEF                          RFNE3290
      IF((IEF)476,476,478                                                     RFNE3300
  478 READ(IN,84)(CR(J,I),J=1,10),IEF                                         RFNE3310
C         READ FOURTH CUMULANT COEF.                                          RFNE3320
      IF(IEF.NE.0)READ(IN,85)(OR(J,I),J=1,15)                                 RFNE3330
      GO TO 476                                                               RFNE3340
  479 NPAR=0                                                                  RFNE3350
      IF(IBUFF(31).EQ.0)GO TO 480                                             RFNE3360
      READ(IN,90)NPAR                                                         RFNE3370
      CALL PAREXT                                                             RFNE3380
      CALL TRNFRM                                                             RFNE3390
C             READ NEW TEMPERATURE FACTOR TYPE                                RFNE3400
  480 READ(IN,18)J,(ISOT(I),I=1,NATOM)                                        RFNE3410
      IEXT=J                                                                  RFNE3420
      IF(J.LE.2)GO TO 489                                                     RFNE3430
C         CONVERT EXTINCTION TYPE                                             RFNE3440
      IF(J.GT.4)GO TO 486                                                     RFNE3450
C         CONVERT ISOTROPIC TO ANISOTROPIC                                    RFNE3460
      TEMP=EXTIN(1)**2                                                        RFNE3470
      IF(TEMP.GT.1.0E-20)TEMP=1.0/TEMP                                        RFNE3480
      DO 482 I=1,3                                                            RFNE3490
      EXTIN(I+3)=0.0                                                          RFNE3500
  482 EXTIN(I)=TEMP                                                           RFNE3510
      IEXT=J-2                                                                RFNE3520
      GO TO 488                                                               RFNE3530
  486 IF(J.GT.6)GO TO 487                                                     RFNE3540
C         CONVERT ANISO TO ISO                                                RFNE3550
      IEXT=0                                                                  RFNE3560
      EXTIN(1)=SQRT(1.0/(3.0*(EXTIN(1)+EXTIN(2)+EXTIN(3))))                   RFNE3570
      GO TO 488                                                               RFNE3580
C         SWITCH ANISO. TYPES                                                 RFNE3590
  487 IEXT=MAXO(9-J,2)                                                        RFNE3600
  488 WRITE(IOUT,29)EXTIN                                                     RFNE3610
  489                        DO 530 I=1,NATOM                                 RFNE3620
      IF(ISOT(I)-1)530,530,490                                                RFNE3630
C             TEMPERATURE FACTOR TYPE TO BE CHANGED                           RFNE3640
  490 IF(ISOT(I)-3)500,510,520                                                RFNE3650
C             ISOTRUPIC TO ANISOTRUPIC                                        RFNE3660
  500 ISOT(I)=1                                                               RFNE3670
      TEMP=BETA(1,I)                                                          RFNE3680
      BETA(6,I)=0.25*RECELL(2)*RECELL(3)*RECELL(4)*TEMP                       RFNE3690
      BETA(5,I)=0.25*RECELL(1)*RECELL(3)*RECELL(5)*TEMP                       RFNE3700
      BETA(4,I)=0.25*RECELL(1)*RECELL(2)*RECELL(6)*TEMP                       RFNE3710
      BETA(3,I)=0.25*RECELL(3)**2*TEMP                                        RFNE3720
      BETA(2,I)=0.25*RECELL(2)**2*TEMP                                        RFNE3730
      BETA(1,I)=0.25*RECELL(1)**2*TEMP                                        RFNE3740
      GO TO 530                                                               RFNE3750
C             ANISOTRUPIC TO ISOTROPIC                                        RFNE3760
  510 ISOT(I)=0                                                               RFNE3770
      BETA(1,I)=4.0*(G(1,1)*BETA(1,I)+G(2,2)*BETA(2,I)+G(3,3)*BETA(3,I)+RFNE3780
     1  2.0*(G(1,2)*BETA(4,I)+G(1,3)*BETA(5,I)+G(2,3)*BETA(6,I)))/3.0      RFNE3790
      GO TO 530                                                               RFNE3800
  520 IF(ISOT(I)-6)521,522,525                                                RFNE3810
  521 IF(ISOT(I)-5)524,522,524                                                RFNE3820
C         ISOT=5 OR 6, CHANGE TO THIRD CUM. FORM                              RFNE3830
  522 DO 523 J=1,10                                                           RFNE3840
  523 CR(J,I)=0.0                                                             RFNE3850
      ISOT(I)=ISOT(I)-1                                                       RFNE3860
  524 ISOT(I)=6-ISOT(I)                                                       RFNE3870
      GO TO 530                                                               RFNE3880
  525 IF(ISOT(I)-8)528,526,526                                                RFNE3890
C         CHANGE TO FOURTH CUMULANT FORM                                      RFNE3900
  526 DO 527 J=1,15                                                           RFNE3910
  527 OR(J,I)=0.0                                                             RFNE3920
      ISOT(I)=ISOT(I)-1                                                       RFNE3930
  528 ISOT(I)=10-ISOT(I)                                                      RFNE3940
  530 CONTINUE                                                                RFNE3950
      WRITE(IOUT,19)((TAG(J,I),J=1,2I,(XYZ(J,I),J=1,3I,(BETA(J,I,J=1,6)RFNE3960
     $,I=1,NATOM)                                                             RFNE3970
      WRITE(IOUT,21)((TAG(J,I),J=1,2I,ISOT(I),(ISCAT(J,I,J=1,2I,SITE(I)RFNE3980
     $,OCCUP(I),OCCA(I),I=1,NATOM)                                            RFNE3990
      GO TO 550                                                               RFNE4000
  540 IF(NEW(6))480,550,480                                                   RFNE4010
  550 IF(NEW(7))560,610,560                                                   RFNE4020
C             READ NEW PARAMETER SELECTION INFORMATION                        RFNE4030
  560 NPARA=0                                                                 RFNE4040
      IFSGN=IBUFF(32I                                                         RFNE4050
```

```
      MRANK=0                                                          RFNE4060
      NRSF=0                                                           RFNE4070
C        GET EXTINCTION REFINEMENT SWITCHES                            RFNE4080
      DO 565 I=1,6                                                     RFNE4090
      KEXTRF(I)=0                                                      RFNE4100
      IF(MOD(IBUFF(I+24),2).EQ.0)GO TO 565                             RFNE4110
      MRANK=MRANK+1                                                    RFNE4120
      KEXTRF(I)=MRANK                                                  RFNE4130
  565 CONTINUE                                                         RFNE4140
C        GET SCALE FACTOR REFINEMENT SWITCHES                          RFNE4150
      DO 575 I=1,NSF                                                   RFNE4160
      JSCRF(I)=0                                                       RFNE4170
      IF(MOD(IBUFF(I+14),2).EQ.0)GO TO 575                             RFNE4180
      MRANK=MRANK+1                                                    RFNE4190
      JSCRF(I)=MRANK                                                   RFNE4200
      NRSF=NRSF+1                                                      RFNE4210
  575 CONTINUE                                                         RFNE4220
      IF(NPAR.EQ.0)GO TO 579                                           RFNE4230
C        GET SPECIAL PARAMETER REFINEMENT SWITCHES                     RFNE4240
      READ(N,18)(IRPAR(I),(=1,NPAR)                                    RFNE4250
      DO 577 I=1,NPAR                                                  RFNE4260
      IF(IRPAR(I).EQ.0)GO TO 577                                       RFNE4270
      MRANK=MRANK+1                                                    RFNE4280
      IRPAR(I)=MRANK                                                   RFNE4290
  577 CONTINUE                                                         RFNE4300
  579 CONTINUE                                                         RFNE4310
                              DO 580 I=1,NATOM                         RFNE4320
      NPARA=NPARA+5                                                    RFNE4330
      IF((ISOT(I))570,580,570                                          RFNE4340
  570 NPARA=NPARA+5                                                    RFNE4350
      IF(ISOT(I)-2)580,576,576                                         RFNE4360
  576 NPARA=NPARA+10                                                   RFNE4370
      IF(ISOT(I).EQ.3) NPARA=NPARA+15                                  RFNE4380
  580 CONTINUE                                                         RFNE4390
      READ(N,18)(IPARA(I),I=1,NPARA)                                   RFNE4400
      WRITE(IOUT,9023)                                                 RFNE4410
      IF(NPAR.GT.0)WRITE(IOUT,23)(IRPAR(I),I=1,NPAR)                   RFNE4420
      WRITE(IOUT,23)(IPARA(I),I=1,NPARA)                               RFNE4430
                              DO 600 I=1,NPARA                         RFNE4440
      IF(IPARA(I))590,600,590                                          RFNE4450
  590 MRANK=MRANK+1                                                    RFNE4460
      IPARA(I)=MRANK                                                   RFNE4470
  600 CONTINUE                                                         RFNE4480
C           READ DEPENDENT PARAMETER INFORMATION                       RFNE4490
      CALL RDEGN                                                       RFNE4500
      IF(NDEGN.EQ.0)GO TO 610                                          RFNE4510
      KPARA=0                                                          RFNE4520
      DO 602 I=0,NATOM                                                 RFNE4530
      CALL RESET(I,KPARA)                                              RFNE4540
  602 CONTINUE                                                         RFNE4550
  610 IF(NDEGN.NE.0.AND.(MFSEL.NE.0.OR.(FIRST.EQ.0))CALL DERIVS(NEW(7)) RFNE4560
      IF(NDEGN.EQ.0)MDEGN=0                                            RFNE4570
      IF(MFSEL)612,616,612                                             RFNE4580
C           IF LEAST-SQUARES, CLEAR MATRICES                           RFNE4590
  612 IC=MDEGN                                                         RFNE4600
      MEMSZ=(MRANK*(MRANK+1))/2+IC                                     RFNE4610
C     SET SIZE OF LEAST SQUARES MATRICES                               RFNE4620
      CALL CORSZIA(MEMSZ))                                             RFNE4630
                              DO 614 IA=1,MRANK                        RFNE4640
      B(IA)=0.0                                                        RFNE4650
                              DO 614 IB=IA,MRANK                       RFNE4660
      IC=IC+1                                                          RFNE4670
  614 A(IC)=0.0                                                        RFNE4680
C           INITIALIZE FOR CYCLE                                       RFNE4690
  616 NSC9=NSCAT+12                                                    RFNE4700
      LBUFF=1022/NSC9                                                  RFNE4710
      LBUFF=NSC9*LBUFF+1                                               RFNE4720
      NPRINT=0                                                         RFNE4730
      SUMFC=0.0                                                        RFNE4740
      RMINUS=0.0                                                       RFNE4750
                              DO 630 IU=1,36                           RFNE4760
      RNUM(IU)=0.0                                                     RFNE4770
      RDEN(IU)=0.0                                                     RFNE4780
  630 NUMBER(IU)=0                                                     RFNE4790
      IF(MSF-1)634,634,632                                             RFNE4800
C*****************************************************************   RFNE4810
C        THIS SECTION SHOULD PREPARE TO WRITE A STRUCTURE FACTOR FILE  RFNE4820
C          FOR A FOURIER PROGRAM                                       RFNE4830
C             PREPARE BINARY STRUCTURE FACTOR TAPE                     RFNE4840
  632 REWIND ISC4                                                      RFNE4850
      WRITE(ISC4)NSYM,VOL,((RMAT((,J,K),I=1,3),J=1,3),K=1,NSYM),((TRANSRFNE4860
     1(J,I),J=1,3),I=1,NSYM)                                           RFNE4870
C*****************************************************************   RFNE4880
  634 IF((IBUFF(12)-5)636,635,635                                      RFNE4890
C           CHANGE REJECTION MODE                                      RFNE4900
  635 IBUFF(12)=IBUFF(12)-5                                            RFNE4910
      DELMAX=BUFF(13)                                                  RFNE4920
      IREJ=IBUFF(12)                                                   RFNE4930
      WR(TELIOUT,9635)IREJ,DELMAX                                      RFNE4940
 9635 FORMAT(' IREJ=',I2,' DELMAX=',F5.1)                              RFNE4950
  636 IF(NEW(8))665,640,665                                            RFNE4960
C           NEW STRUCTURE FACTORS NOT TO BE READ                       RFNE4970
  640 IF((IFIRST)660,650,660                                           RFNE4980
C           FIRST CYCLE, STRUCTURE FACTORS FROM SAVE TAPE              RFNE4990
  650 ASSIGN 700 TO INTYPE                                             RFNE5000
      GO TO 700                                                        RFNE5010
C           NOT FIRST CYCLE, STRUCTURE FACTORS FROM INTERMEDIATE UNIT  RFNE5020
  660 ASSIGN 830 TO INTYPE                                             RFNE5030
      GO TO 830                                                        RFNE5040
C           NEW STRUCTURE FACTORS                                      RFNE5050
  665 IREJ=IBUFF(12)                                                   RFNE5060
      IF((IBUFF(1))667,666,667                                         RFNE5070
C           NEW STRUCTURE FACTORS FROM STANDARD (INPUT UNIT            RFNE5080
  666 INN=IN                                                           RFNE5090
      GO TO 668                                                        RFNE5100
C           NEW STRUCTURE FACTORS FROM UNIT 4 (ISC3)                   RFNE5110
  667 INN=ISC3                                                         RFNE5120
  668 DELMAX=BUFF(13)                                                  RFNE5130
      IWT=IBUFF(10)+1                                                  RFNE5140
      FOMAX=0.0                                                        RFNE5150
      FOMIN=1.0E+6                                                     RFNE5160
      RHOMN=1.0E+6                                                     RFNE5170
      RHOMX=0.0                                                        RFNE5180
      READ(IN,2)FRMT                                                   RFNE5190
      WRITE(IOUT,9668)FRMT                                             RFNE5200
 9668 FORMAT('0FORMAT FOR STRUCTURE FACTOR INPUT IS'/1X,18A4)          RFNE5210
```

```
C      IF IWT.EQ.6 AN 'IGNURANCE FACTOR' WILL BE READ IN FOR COMPUTING   RFNE5220
C      WEIGHTS                                                           RFNE5230
       IF(IWT.EQ.6)READ(IIN,6)WTFAC                                      RFNE5240
       IF((WT.EQ.6)WRITE(IOUT,9669)WTFAC                                 RFNE5250
 9669 FORMAT('0SIGMA WILL BE COMPUTED BY SIGMA=SQRT(SIGMA**2+(',F6.4,'**FRFNE5260
     1)**2)')                                                            RFNE5270
C      NEW(8) SPECIFIES THE INPUT FORMAT AS FOLLOWS. 1=OUTPUT OF ABSORB  RFNE5280
C      (MY VERSION), 2=XRAY67 HKL CARD SEQUENCE, 3=925 TAPES (BETA TO BE RFNE5290
C      COMPUTED).                                                        RFNE5300
       IF(NEW(8).NE.3) GO TO 669                                         RFNE5310
       READ(IIN,6)WVLNTH                                                 RFNE5320
       BETFAC=(2.*WVLNTH**2/VOL**2)*1.0E+8                               RFNE5330
       WRITE(IOUT,9670)WVLNTH                                            RFNE5340
 9670 FORMAT('0WAVELENGTH IS ',F7.4)                                     RFNE5350
  669 ASSIGN 671 TO INTYPE                                               RFNE5360
       IBUFF(1)=0                                                        RFNE5370
       NBUFF=1                                                           RFNE5380
  671 IEF=NEW(8)                                                         RFNE5390
       CALL INPUT(IEF)                                                   RFNE5400
       IF(LBUFF-NBUFF)672,672,673                                        RFNE5410
  672 WRITE(ISC2)(IBUFF(I),I=1,LBUFF)                                    RFNE5420
       NBUFF=1                                                           RFNE5430
  673 IBUFF(NBUFF+1)=IH(1)                                               RFNE5440
       IBUFF(NBUFF+2)=IH(2)                                              RFNE5450
       IBUFF(NBUFF+3)=IH(3)                                              RFNE5460
       BUFF(NBUFF+4)=FOBS                                                RFNE5470
C  HERE WT = SIGMA F                                                     RFNE5480
       BUFF(NBUFF+5)=WT                                                  RFNE5490
       BUFF(NBUFF+6)=RHO                                                 RFNE5500
       IBUFF(NBUFF+7)=ISCL                                               RFNE5510
       BUFF(NBUFF+8)=EXBETA                                              RFNE5520
       IBUFF(NBUFF+9)=MREJ                                               RFNE5530
       DO 6740 I=10,12                                                   RFNE5540
 6740 BUFF(NBUFF+I)=EXANG(I-9)                                           RFNE5550
                                  DO 674 I=1,NSCAT                       RFNE5560
  674 BUFF(I+NBUFF+12)=SCAT(I)                                           RFNE5570
       NBUFF=NBUFF+NSC9                                                  RFNE5580
       IF(IEF)675,670,675                                                RFNE5590
  675 IBUFF(1)=1                                                         RFNE5600
       IBUFF(NBUFF-NSC9+1)=3000                                          RFNE5610
       WRITE(ISC2)(IBUFF(I),I=1,LBUFF)                                   RFNE5620
  691 END FILE ISC2                                                      RFNE5630
       IF(IFILE.LT.2)GO TO 2000                                          RFNE5640
C      READ SAVED SIGMA. FIRST SKIP OVER CONSTRAINTS AND VAR-COV MATRIX. RFNE5650
       READ(ISC1)IBUFF(1)                                                RFNE5660
       READ((ISC1)IBUFF(1)                                               RFNE5670
       READ(ISC1)((BBETA(J,I),J=1,6),SIGO(I),(SIGXYZ(J,I),J=1,3),(SIGB(J,RFNE5680
      1I),J=1,6),I=1,NATOM)                                              RFNE5690
       GO TO 2000                                                        RFNE5700
C             READ STRUCTURE FACTORS FROM SAVE UNIT                      RFNE5710
  700 IF(NBUFF)708,701,708                                               RFNE5720
  701 READ(ISC1)(IBUFF(I),I=1,LBUFF)                                     RFNE5730
       IF(NEW(4))702,707,702                                             RFNE5740
C             NEW SCATTERING FACTORS                                     RFNE5750
  702 ITEMP=7-NSC9                                                       RFNE5760
  703 ITEMP=ITEMP+NSC9                                                   RFNE5770
       IF((TEMP-LBUFF)704,707,707                                        RFNE5780
  704 RHO=BUFF(ITEMP)                                                    RFNE5790
                                  DO 706 I=1,NSCAT                       RFNE5800
       TEMP=FCURVE(9,I)+ANOM(1,I)                                        RFNE5810
                                  DO 705 J=1,4                           RFNE5820
  705 TEMP=TEMP+FCURVE(2*J-1,I)*EXP(-FCURVE(2*J,I)*RHO)                  RFNE5830
  706 BUFF(I+ITEMP+6)=TEMP                                               RFNE5840
       GO TO 703                                                         RFNE5850
  707 WRITE(ISC2)(IBUFF(I),I=1,LBUFF)                                    RFNE5860
       NBUFF=1                                                           RFNE5870
  708 IH(1)=IBUFF(NBUFF+1)                                               RFNE5880
       IH(2)=IBUFF(NBUFF+2)                                              RFNE5890
       IH(3)=IBUFF(NBUFF+3)                                              RFNE5900
       FOBS=BUFF(NBUFF+4)                                                RFNE5910
C  HERE WT = SIGMA F                                                     RFNE5920
       WT=BUFF(NBUFF+5)                                                  RFNE5930
       RHO=BUFF(NBUFF+6)                                                 RFNE5940
       ISCL=IBUFF(NBUFF+7)                                               RFNE5950
       EXBETA=BUFF(NBUFF+8)                                              RFNE5960
       MREJ=IBUFF(NBUFF+9)                                               RFNE5970
       DO 712 I=10,12                                                    RFNE5980
  712 EXANG(I-9)=BUFF(NBUFF+I)                                           RFNE5990
                                  DO 709 I=1,NSCAT                       RFNE6000
  709 SCAT(I)=BUFF(I+NBUFF+12)                                           RFNE6010
       NBUFF=NBUFF+NSC9                                                  RFNE6020
       IF(LBUFF-NBUFF)710,710,711                                        RFNE6030
  710 NBUFF=0                                                            RFNE6040
  711                             DO 770 I=1,3                           RFNE6050
  770 H(I)=IH(I)                                                         RFNE6060
       IF(IH(1)-3000)870,691,870                                         RFNE6070
C             READ FROM INTERMEDIATE TAPE                              RFNE6080
  830 IF(NBUFF)832,831,832                                               RFNE6090
  831 READ(ISC2)(IBUFF(I),I=1,LBUFF)                                     RFNE6100
       NBUFF=1                                                           RFNE6110
  832 IH(1)=IBUFF(NBUFF+1)                                               RFNE6120
       IH(2)=IBUFF(NBUFF+2)                                              RFNE6130
       IH(3)=IBUFF(NBUFF+3)                                              RFNE6140
       FOBS=BUFF(NBUFF+4)                                                RFNE6150
       WT=BUFF(NBUFF+5)                                                  RFNE6160
       RHO=BUFF(NBUFF+6)                                                 RFNE6170
       ISCL=IBUFF(NBUFF+7)                                               RFNE6180
       EXBETA=BUFF(NBUFF+8)                                              RFNE6190
       MREJ=IBUFF(NBUFF+9)                                               RFNE6200
       DO 836 I=10,12                                                    RFNE6210
  836 EXANG(I-9)=BUFF(NBUFF+I)                                           RFNE6220
                                  DO 833 I=1,NSCAT                       RFNE6230
  833 SCAT(I)=BUFF(I+NBUFF+12)                                           RFNE6240
       NBUFF=NBUFF+NSC9                                                  RFNE6250
       IF(LBUFF-NBUFF)834,834,835                                        RFNE6260
  834 NBUFF=0                                                            RFNE6270
  835                             DO 840 I=1,3                           RFNE6280
  840 H(I)=IH(I)                                                         RFNE6290
       IF(IH(1)-3000)870,2000,870                                        RFNE6300
C             COMPUTE STRUCTURE FACTOR                                   RFNE6310
  870 CALL SFAC                                                          RFNE6320
       FCALC=ACALC**2+BCALC**2                                           RFNE6330
C        CALCULATE EXTINCTION FACTOR                                     RFNE6340
       CALL EXCALC                                                       RFNE6350
       EXY=1.0/SQRT(1.0+EXX)                                             RFNE6360
       FCALC=SQRT(FCALC*EXY)                                             RFNE6370
```

```
      IF((IFSGN.NE.0)FMINUS=SQRT(EXY*(AM(NUS**2+BMINUS**2))                RFNE6380
      SUMFC=SUMFC+FCALC                                                    RFNE6390
      NREJ=0                                                               RFNE6400
      FOBS=FOBS/SCALE(ISCL)                                                RFNE6410
      SIGF=WT/SCALE(ISCL)                                                  RFNE6420
      IF(IREJ)880,930,880                                                  RFNE6430
  880 GO TO (890,900,902,920),IREJ                                         RFNE6440
C             REJECT IF LESS THAN MIN OBS                                  RFNE6450
  890 NREJ=MREJ                                                            RFNE6460
      GO TO 930                                                            RFNE6470
C             REJECT IF LESS THAN MIN OBS OR DELTA GR DELMAX               RFNE6480
  900 IF(MREJ)890,902,890                                                  RFNE6490
C             REJECT IF DELTA F GR DELMAX                                  RFNE6500
  902 IF(ABS(FOBS-FCALC)/SIGF-DELMAX)930,930,904                           RFNE6510
  904 NREJ=1                                                               RFNE6520
      GO TO 930                                                            RFNE6530
C             USE SPECIAL ROUTINE REJECT                                   RFNE6540
  920 CALL REJECT                                                          RFNE6550
C             CALCULATE CONTRIBUTION TO R                                  RFNE6560
C* * * RCALC ALSO CONVERTS 'WT' TO A WEIGHT  (SIGMA F TO HERE)            RFNE6570
  930 CALL RCALC(SCALE(ISCL))                                              RFNE6580
      IF(NREJ.NE.0.OR.IFSGN.EQ.0)GO TO 931                                 RFNE6590
      TEMP=FOBS-FMINUS                                                     RFNE6600
      IF(MREF.NE.0)TEMP=FOBS**2-FMINUS**2                                  RFNE6610
      RMINUS=RMINUS+WT*TEMP**2                                             RFNE6620
  931 IF(MSF)932,980,932                                                   RFNE6630
  932 IF(MSF-3)934,970,934                                                 RFNE6640
C             WRITE BCO OUTPUT TAPE                                        RFNE6650
  934 NPAGE=NPRINT/MAXPG+1                                                 RFNE6660
      IF(MAXPG*(NPAGE-1)-NPRINT)950,940,950                                RFNE6670
C             PAGE OUTPUT                                                  RFNE6680
  940 WRITE(IOUT,20)TITLE,NPAGE                                            RFNE6690
C             PRINT STRUCTURE FACTOR                                       RFNE6700
  950 TEMP=FOBS-FCALC                                                      RFNE6710
      TEMP1=DELTA/SIGF                                                     RFNE6720
      WRITE(IOUT,22)IH,FOBS,KREJ(MREJ+1),FCALC,ACALC,BCALC,TEMP,TEMP1,     RFNE6730
     I EXY,KREJ(NREJ+1),KREJ(MREJ+3)                                       RFNE6740
      NPRINT=NPRINT+1                                                      RFNE6750
  960 IF(MSF-2)980,970,970                                                 RFNE6760
C*****************************************************************       RFNE6770
C         THIS SECTION SHOULD ADD THE CURRENT REFLECTION TO THE FOURIER     RFNE6780
C           FILE                                                           RFNE6790
C             WRITE BINARY STRUCTURE FACTOR TAPE                           RFNE6800
  970 IFBUF(NFBR+1)=IH(1)                                                  RFNE6810
      IFBUF(NFBR+2)=IH(2)                                                  RFNE6820
      IFBUF(NFBR+3)=IH(3)                                                  RFNE6830
      FBUF(NFBR+4)=FOBS                                                    RFNE6840
      FBUF(NFBR+5)=FCALC                                                   RFNE6850
      FBUF(NFBR+6)=ACALC                                                   RFNE6860
      FBUF(NFBR+7)=BCALC                                                   RFNE6870
      IFBUF(NFBR+8)=NREJ                                                   RFNE6880
C       ADD SIGMA F AND EXTINCTION PARAMETER TO LIST                       RFNE6890
      FBUF(NFBR+9)=SIGF                                                    RFNE6900
      FBUF(NFBR+10)=EXY                                                    RFNE6910
      NFBR=NFBR+10                                                         RFNE6920
      IF(NFBR-500)980,972,972                                              RFNE6930
  972 WRITE(ISC4)FBUF                                                      RFNE6940
      NFBR=0                                                               RFNE6950
C*****************************************************************       RFNE6960
  980 IF(MFSEL)1000,990,1000                                               RFNE6970
C             GO TO GET NEW STRUCTURE FACTOR                               RFNE6980
  990 GO TO INTYPE,(830,700,671)                                           RFNE6990
 1000 IF(NREJ)990,1010,990                                                 RFNE7000
C             ADD TO LEAST SQUARES MATRICES IF REFLECTION NOT REJECTED     RFNE7010
 1010 CALL MATRIX                                                          RFNE7020
      GO TO 990                                                            RFNE7030
C             END OF LIST FOUND, REWIND TAPES                              RFNE7040
 2000 REWIND ISC1                                                          RFNE7050
      REWIND ISC2                                                          RFNE7060
      IF(MSF-2)2020,2010,2010                                              RFNE7070
C * * * * * * * * * * * * * * THIS SECTION ENDS THE FOURIER FILE * * *    RFNE7080
 2010 IFBUF(NFBR+1)=99                                                     RFNE7090
      WRITE(ISC4)FBUF                                                      RFNE7100
C             COMPUTE AND PRINT OUT R FACTORS                              RFNE7110
 2020 WRITE(IOUT,24)TITLE                                                  RFNE7120
      NUMBER(2)=NUMBER(1)                                                  RFNE7130
      NUMBER(20)=NUMBER(19)                                                RFNE7140
                                  DO 2030 I=1,36                           RFNE7150
      IF(NUMBER(I))2024,2022,2024                                          RFNE7160
 2022 R(I)=0.0                                                             RFNE7170
      GO TO 2030                                                           RFNE7180
 2024 J=(I/19)*18+1                                                        RFNE7190
      IF(I.NE.2.AND.I.NE.20)ROEN(I)=ROEN(J)*(FLOAT(NUMBER(I))/FLOAT(NUMBRFNE7200
     1ER(J)))                                                              RFNE7210
      R(I)=SQRT(RNUM(I)/ROEN(I))                                           RFNE7220
 2030 CONTINUE                                                             RFNE7230
      P(2)=R(2)**2                                                         RFNE7240
      R(20)=R(20)**2                                                       RFNE7250
      WRITE(IOUT,28)                                                       RFNE7260
      WRITE(IOUT,26)(RNUM(I),ROEN(I),NUMBER(I),R(I),I=1,18)                RFNE7270
      TEMP=SQRT(RNUM(19)/FLOAT(NUMBER(19)-MRANK))                          RFNE7280
      WRITE(IOUT,30)                                                       RFNE7290
      WRITE(IOUT,26)(RNUM(I),ROEN(I),NUMBER(I),R(I),I=19,36)               RFNE7300
      WRITE(IOUT,32)SUMFC,TEMP                                             RFNE7310
      IF(MFSEL)2040,3000,2040                                              RFNE7320
C             LEAST-SQUARES MATRIX INVERSION                               RFNE7330
 2040 N=-MRANK                                                             RFNE7340
      NJ=MRANK+1                                                           RFNE7350
C             CONDITION MATRICES                                           RFNE7360
                                  DO 2041 I=1,MRANK                        RFNE7370
      N=N+NJ                                                               RFNE7380
      NJ=NJ-1                                                              RFNE7390
 2041 OBOB(I,1)=1.0/SQRT(A(N+MOEGN))                                       RFNE7400
      N=MOEGN                                                              RFNE7410
                                  DO 2042 I=1,MRANK                        RFNE7420
      OAOO(I)=B(I)                                                         RFNE7430
      B(I)=B(I)*OBOB(I,1)                                                  RFNE7440
                                  DO 2042 J=1,MRANK                        RFNE7450
      N=N+1                                                                RFNE7460
 2042 A(N)=A(N)*OBOB(I,1)*OBOB(J,1)                                        RFNE7470
      CALL SYMINV(A(MOEGN+1),MRANK,OAOB,1,0,1.0E-10,DET,JRANK)             RFNE7480
      IF(JRANK-MRANK)2050,2060,2050                                        RFNE7490
 2050 WRITE(IOUT,34)JRANK,MRANK                                            RFNE7500
      GO TO 3000                                                           RFNE7510
 2060 WRITE(IOUT,8)TITLE                                                   RFNE7520
      WRITE(IOUT,36)                                                       RFNE7530
```

```
      II=1+MDEGN                                                        RFNE7540
      SUM=0.0                                                           RFNE7550
      TERM=TEMP**2                                                      RFNE7560
C             CONVERT MATRIX TO COVARIANCE MATRIX                       RFNE7570
                                  DO 2070 I=1,MRANK                     RFNE7580
      DADB(I,1)=SQRT(A(II)*TERM)*DBOB(I,1)                              RFNE7590
      B(I)=B(I)*DADB(I,1)                                               RFNE7600
      SUM=SUM+B(I)*DADB(I)                                              RFNE7610
                                  DO 2070 J=I,MRANK                     RFNE7620
      A(I)=A(II)*TERM*DBOB(I,1)*DBOB(J,1)                               RFNE7630
 2070 II=II+1                                                           RFNE7640
      SUM=SQRT(RNUM(19)-SUM)/FLOAT(NUMBER(19)-MRANK))                   RFNE7650
      WRITE(IOUT,76)SUM                                                 RFNE7660
      IJ=1                                                              RFNE7670
      SUM=0.0                                                           RFNE7680
      SEMAX=0.0                                                         RFNE7690
C             COMPUTE AND PRINT NEW PARAMETERS                          RFNE7700
      WRITE(IOUT,59)                                                    RFNE7710
      IF(IEXT)2074,2072,2074                                            RFNE7720
C        ISOTROPIC EXTINCTION                                           RFNE7730
 2072 WRITE(IOUT,60)                                                    RFNE7740
      J=1                                                               RFNE7750
      K=35                                                              RFNE7760
      GO TO 2076                                                        RFNE7770
C        ANISOTROPIC                                                    RFNE7780
 2074 WRITE(IOUT,63)IEXT                                                RFNE7790
      J=6                                                               RFNE7800
      K=4                                                               RFNE7810
 2076 DO 2090 I=1,J                                                     RFNE7820
      IF(KEXTRF(I))2080,2082,2080                                       RFNE7830
C             EXTINCTION CORRECTION REFINED                             RFNE7840
 2080 TEMP=EXTIN(I)                                                     RFNE7850
      EXTINII)=EXTINII)+B(IJ)                                           RFNE7860
      TEMP1=B(IJ)/DADB(IJ,1)                                            RFNE7870
      IF(ABS(TEMP1).GT.ABS(SEMAX)) SEMAX=TEMP1                          RFNE7880
      SUM=SUM+ABS(TEMP1)                                                RFNE7890
      WRITE(IOUT,67)HEAD(I+K),TEMP,B(IJ),EXTIN(I),DADB(IJ,1),TEMP1      RFNE7900
     1,IJ                                                               RFNE7910
      IJ=IJ+1                                                           RFNE7920
      GO TO 2090                                                        RFNE7930
 2082 WRITE(IOUT,65)HEAD(I+K),EXTIN(I),EXTINII)                         RFNE7940
 2090 CONTINUE                                                          RFNE7950
      IF(IFSGN.EQ.0)GO TO 2095                                          RFNE7960
C        CHECK ABSOLUTE ORIENTATION                                     RFNE7970
      TEMP=FISGN                                                        RFNE7980
      IF(0.99*RNUM(19).GT.RMINUS)FISGN=-FISGN                           RFNE7990
      RMINUS=SQRT(RMINUS/RDEN(19))                                      RFNE8000
      WRITE(IOUT,98)TEMP,RMINUS,FISGN                                   RFNE8010
 2095                             DO 2100 I=1,NSF                     RFNE8020
      IF(JSCRF(I))2096,2097,2096                                        RFNE8030
 2096 TEMP=SCALE(I)                                                     RFNE8040
      SCALE(I)=SCALE(I)+B(IJ)                                           RFNE8050
      TEMP1=B(IJ)/DADB(IJ,1)                                            RFNE8060
      WRITE(IOUT,40)(,TEMP,B(IJ),SCALE(I),DADB(IJ,I),TEMP1,IJ          RFNE8070
      IF(ABS(TEMP1).GT.ABS(SEMAX)) SEMAX=TEMP1                          RFNE8080
      SUM=SUM+ABS(TEMP1)                                                RFNE8090
      IJ=IJ+1                                                           RFNE8100
      GO TO 2100                                                        RFNE8110
 2097 WRITE(IOUT,78)I,SCALE(I),SCALE(I)                                 RFNE8120
 2100 CONTINUE                                                          RFNE8130
      IF(NDEGN.EQ.0)GO TO 2109                                          RFNE8140
C        RESTORE THE UNALTERED DEPENDENCY LIST                          RFNE8150
      REWIND ISC5                                                       RFNE8160
      READ(ISC5)LDEGN,(KOEGNIL),L=1,LOEGN)                              RFNE8170
      IF(NPAR.EQ.0)GO TO 2109                                           RFNE8180
C        SAVE OLD SPECIAL PARAMETERS                                    RFNE8190
      DO 6020 I=1,NPAR                                                  RFNE8200
      SAVE(I)=PAR(I)                                                    RFNE8210
C        UPDATE THE SPECIAL PARAMETERS                                  RFNE8220
      IJ=IRPAR(I)                                                       RFNE8230
      IF(IJ.GT.0)PAR(I)=PAR(I)+B(IJ)                                    RFNE8240
 6020 CONTINUE                                                          RFNE8250
      CALL RESETIO,0)                                                   RFNE8260
C        NOW PRINT OUT THE SPECIAL PARAMETERS                           RFNE8270
      WRITE(IOUT,95)                                                    RFNE8280
      DO 2108 I=1,NPAR                                                  RFNE8290
      TEMP=SAVE(I)                                                      RFNE8300
      IJ=IRPAR(I)                                                       RFNE8310
      IF(IJ)6040,6040,6030                                              RFNE8320
 6030 TEMP1=B(IJ)/DADB(IJ,1)                                            RFNE8330
      WRITE(IOUT,96)(PARNAM(J,I),J=1,2),TEMP,B(IJ),PAR(I),DADB(IJ,1),   RFNE8340
     1 TEMP1,IJ                                                         RFNE8350
      IF(ABS(TEMP1).GT.ABS(SEMAX))SEMAX=TEMP1                           RFNE8360
      SUM=SUM+ABS(TEMP1)                                                RFNE8370
      GO TO 2108                                                        RFNE8380
 6040 WRITE(IOUT,97)(PARNAM(J,I),J=1,2),SAVE(I),PAR(I)                  RFNE8390
 2108 CONTINUE                                                          RFNE8400
 2109 IK=0                                                              RFNE8410
                                  DO 2290 I=1,NATOM                     RFNE8420
C        SAVE THE OLD PARAMETERS FOR THIS ATOM                          RFNE8430
      SAVE(1)=OCCUP(I)                                                  RFNE8440
      SAVE(2)=OCCA(I)                                                   RFNE8450
      DO 6060 J=1,3                                                     RFNE8460
 6060 SAVE(J+2)=XYZ(J,I)                                                RFNE8470
      SAVE(6)=BETA(1,I)                                                 RFNE8480
      IF(ISOT(I).LT.1)GO TO 6070                                        RFNE8490
      DO 6062 J=2,6                                                     RFNE8500
 6062 SAVE(J+5)=BETA(J,I)                                               RFNE8510
      IF(ISOT(I).LT.2)GO TO 6070                                        RFNE8520
      DO 6064 J=1,10                                                    RFNE8530
 6064 SAVE(J+11)=CRI(J,I)                                               RFNE8540
      IF(ISOT(I).LT.3)GO TO 6070                                        RFNE8550
      DO 6066 J=1,15                                                    RFNE8560
 6066 SAVE(J+21)=OR(J,I)                                                RFNE8570
C        NOW UPDATE ALL REFINED PARAMETERS FOR THIS ATOM.               RFNE8580
 6070 J=1                                                               RFNE8590
      IJ=IPARA(IK+J)                                                    RFNE8600
      IF(ISCAT(2,I))6090,6080,6090                                      RFNE8610
C        SINGLE OCCUPANCY                                               RFNE8620
 6080 IF(IJ.GT.0)OCCUP(I)=OCCUP(I)+B(IJ)                                RFNE8630
      GO TO 6100                                                        RFNE8640
 6090 IF(IJ.GT.0)OCCA(I)=OCCA(I)+B(IJ)                                  RFNE8650
C        POSITIONAL PARAMETERS                                          RFNE8660
 6100 DO 6110 J=1,3                                                     RFNE8670
      (J=IPARA(IK+J+1)                                                  RFNE8680
      IF(IJ.GT.0)XYZ(J,I)=XYZ(J,I)+B(IJ)                                RFNE8690
```

```
 6110 CONTINUE                                                          RFNE8700
C         THERMAL PARAMETERS                                           RFNE8710
      IJ=IPARA(IK+5)                                                    RFNE8720
      IF(IJ.GT.0)BETA(1,I)=BETA(1,I)+B(IJ)                              RFNE8730
      IF(ISUT(I).LT.1)GO TO 6150                                        RFNE8740
      DO 6120 J=2,6                                                     RFNE8750
      IJ=IPARA(IK+J+4)                                                  RFNE8760
      IF(IJ.GT.0)BETA(J,I)=BETA(J,I)+B(IJ)                              RFNE8770
 6120 CONTINUE                                                          RFNE8780
      IF(ISOT(I).LT.2)GO TO 6150                                        RFNE8790
      DO 6130 J=1,10                                                    RFNE8800
      IJ=IPARA(IK+J+10)                                                 RFNE8810
      IF(IJ.GT.0)CR(J,I)=CR(J,I)+B(IJ)                                  RFNE8820
 6130 CONTINUE                                                          RFNE8830
      IF(ISOT(I).LT.3)GO TO 6150                                        RFNE8840
      DO 6140 J=1,15                                                    RFNE8850
      IJ=IPARA(IK+J+20)                                                 RFNE8860
      IF(IJ.GT.0)DR(J,I)=DR(J,I)+B(IJ)                                  RFNE8870
 6140 CONTINUE                                                          RFNE8880
C         UPDATE DEPENDENT PARAMETERS                                  RFNE8890
 6150 KPARA=IK                                                          RFNE8900
      CALL RESET(I,KPARA)                                               RFNE8910
C         NOW PRINT OUT UPDATED PARAMETERS                             RFNE8920
      WRITE(IOUT,38)I,(TAG(J,I),J=1,2),SITE(I)                          RFNE8930
      IF(ISCAT(2,I))2140,2110,2140                                      RFNE8940
C             SINGLE OCCUPANCY                                         RFNE8950
 2110 TEMP=SAVE(1)                                                      RFNE8960
      IJ=IPARA(IK+1)                                                    RFNE8970
      IF(IJ.LE.0)GO TO 2130                                             RFNE8980
      TEMP1=B(IJ)/OAOB(IJ,1)                                            RFNE8990
      IF(ABS(TEMP1).GT.ABS(SEMAX)) SEMAX=TEMP1                          RFNE9000
      SUM=SUM+ABS(TEMP1)                                                RFNE9010
      WRITE(IOUT,44)ISCAT(1,I),TEMP,B(IJ),OCCUP(I),OAOB(IJ,I),TEMP1     RFNE9020
     1,IJ                                                               RFNE9030
      SIGO(I)=OAOB(IJ,1)                                                RFNE9040
      GO TO 2180                                                        RFNE9050
 2130 WRITE(IOUT,46)ISCAT(1,I),TEMP,TEMP                                RFNE9060
      GO TO 2180                                                        RFNE9070
C             MULTIPLE OCCUPANCY                                        RFNE9080
 2140 TEMP=SAVE(2)                                                      RFNE9090
      TEMP1=OCCUP(I)-TEMP                                               RFNE9100
      IJ=IPARA(IK+1)                                                    RFNE9110
      IF(IJ.GT.0)GO TO 2160                                             RFNE9120
      TEMP2=UCCUP(I)-UCCA(I)                                            RFNE9130
      SIGMA=ERRCAL(I,1)                                                 RFNE9140
      IF(SIGMA.GT.0.0000001)GO TO 2150                                  RFNE9150
      WRITE(IOUT,46)ISCAT(1,I),TEMP,OCCA(I)                             RFNE9160
 2145 WRITE(IOUT,46)ISCAT(2,I),TEMP1,TEMP2                              RFNE9170
      GO TO 2170                                                        RFNE9180
 2150 WRITE(IOUT,46)ISCAT(1,I),TEMP,OCCA(I),SIGMA                       RFNE9190
      GO TO 2145                                                        RFNE9200
 2160 SIGO(I)=OAOB(IJ,1)                                                RFNE9210
      TEMP2=OCCUP(I)-OCCA(I)                                            RFNE9220
      TEMP3=B(IJ)/OAOB(IJ,1)                                            RFNE9230
      SUM=SUM+ABS(TEMP3)                                                RFNE9240
      IF(ABS(TEMP3).GT.ABS(SEMAX)) SEMAX=TEMP3                          RFNE9250
      WRITE(IOUT,44)ISCAT(1,I),TEMP,B(IJ),OCCA(I),OAOB(IJ,1),TEMP3      RFNE9260
     1,IJ                                                               RFNE9270
      WRITE(IOUT,46)ISCAT(2,I),TEMP1,TEMP2                              RFNE9280
 2170 WRITE(IOUT,48)OCCUP(I)                                            RFNE9290
C             POSITIONAL PARAMETERS                                    RFNE9300
 2180                          DO 2210 J=1,3                            RFNE9310
      TEMP=SAVE(J+2)                                                    RFNE9320
      IJ=IPARA(IK+J+1)                                                  RFNE9330
      IF(IJ.GT.0)GO TO 2200                                             RFNE9340
      SIGMA=ERRCAL(I,J+1)                                               RFNE9350
      IF(SIGMA.GT.0.0000001)GO TO 2190                                  RFNE9360
      WRITE(IOUT,52)HEAO(J),TEMP,XYZ(J,I)                               RFNE9370
      GO TO 2210                                                        RFNE9380
 2190 WRITE(IOUT,52)HEAO(J),TEMP,XYZ(J,I),SIGMA                         RFNE9390
      GO TO 2210                                                        RFNE9400
 2200 TEMP1=B(IJ)/OAOB(IJ,1)                                            RFNE9410
      IF(ABS(TEMP1).GT.ABS(SEMAX)) SEMAX=TEMP1                          RFNE9420
      SUM=SUM+ABS(TEMP1)                                                RFNE9430
      WRITE(IOUT,50)HEAO(J),TEMP,B(IJ),XYZ(J,I),OAOB(IJ,1),TEMP1        RFNE9440
     1,IJ                                                               RFNE9450
      SIGXYZ(J,I)=OAOB(IJ,I)                                            RFNE9460
 2210 CONTINUE                                                          RFNE9470
C             THERMAL PARAMETERS                                       RFNE9480
                               DO 2280 J=1,6                            RFNE9490
      TEMP=SAVE(J+5)                                                    RFNE9500
      IJ=IPARA(IK+J+4)                                                  RFNE9510
      IF(ISUT(I))2250,2220,2250                                         RFNE9520
C             ISOTROPIC ATOM                                           RFNE9530
 2220 IF(IJ)2240,2240,2230                                              RFNE9540
 2230 TEMP1=B(IJ)/DAOB(IJ,I)                                            RFNE9550
      SUM=SUM+ABS(TEMP1)                                                RFNE9560
      IF(ABS(TEMP1).GT.ABS(SEMAX)) SEMAX=TEMP1                          RFNE9570
      WRITE(IOUT,50)HEAD(4),TEMP,B(IJ),BETA(1,I),OAOB(IJ,1),TEMP1       RFNE9580
     1,IJ                                                               RFNE9590
      GO TO 2290                                                        RFNE9600
 2240 SIGMA=ERRCAL(I,5)                                                 RFNE9610
      IF(SIGMA.GT.0.0000001)GO TO 2245                                  RFNE9620
      WRITE(IOUT,52)HEAD(4),TEMP,BETA(1,I)                              RFNE9630
      GOTO 2290                                                         RFNE9640
 2245 WRITE(IOUT,52)HEAO(4),TEMP,BETA(1,I),SIGMA                        RFNE9650
      GO TO 2290                                                        RFNE9660
C             ANISOTROPIC ATOM                                         RFNE9670
 2250 IF(IJ)2270,2270,2260                                              RFNE9680
 2260 TEMP1=B(IJ)/OAOB(IJ,1)                                            RFNE9690
      SUM=SUM+ABS(TEMP1)                                                RFNE9700
      IF(ABS(TEMP1).GT.ABS(SEMAX)) SEMAX=TEMP1                          RFNE9710
      WRITE(IOUT,51)HEAO(J+4),TEMP,B(IJ),BETA(J,I),OAUB(IJ,1),TEMP1     RFNE9720
     1,IJ                                                               RFNE9730
      BETAB=BETA(J,I)*EGGY(J)                                           RFNE9740
      BBETA(J,I)=BETA(J,I)*EGGY(J)/TEMPZZ                               RFNE9750
      SIGB(J,I)=OAOB(IJ,1)*EGGY(J)/TEMPZZ                               RFNE9760
      SHIFT=OAOB(IJ,1)*EGGY(J)                                          RFNE9770
      WRITE(IOUT,7070)BETAB,SH(FT,BBETA(J,I),SIGB(IJ,I)                 RFNE9780
 7070 FORMAT(1JX,2(F10.4,3X,F10.6,10X))                                 RFNE9790
      GO TO 2280                                                        RFNE9800
 2270 SIGMA=ERRCAL(I,J+4)                                               RFNE9810
      IF(SIGMA.GT.0.0000001)GO TO 2275                                  RFNE9820
      WRITE(IOUT,53)HEAO(J+4),SAVE(J+5),BETA(J,I)                       RFNE9830
      GO TO 2280                                                        RFNE9840
 2275 WRITE(IOUT,53)HEAO(J+4),SAVE(J+5),BBETA(J,I),SIGMA                RFNE9850
```

```
 2280 CONTINUE                                                          RFNE9860
      IF(ISOT(I)-2)2290,2281,2281                                       RFNE9870
C        THIRU CUMULANT PARAMETERS                                      RFNE9880
 2281 DO 2284 J=1,10                                                    RFNE9890
      TEMP=SAVE(J+11)                                                   RFNE9900
      IJ=IPARA(IK+J+10)                                                 RFNE9910
      IF(IJ)2283,2283,2282                                              RFNE9920
 2282 TEMP1=B(IJ)/OAOB(IJ,1)                                            RFNE9930
      SUM=SUM+ABS(TEMP1)                                                RFNE9940
      IF(ABS(TEMP1).GT.ABS(SEMAX)) SEMAX=TEMP1                          RFNE9950
      WRITE(IOUT,80)HEAO(J+10),TEMP,B(IJ),CR(J,I),OAOB(IJ,1),TEMPI      RFNE9960
     I,IJ                                                               RFNE9970
      GO TO 2284                                                        RFNE9980
 2283 SIGMA=ERRCAL(I,J+10)                                              RFNE9990
      IF(SIGMA.GT.0.0000001)GO TO 6283                                  RFNE   0
      WRITE(IOUT,82)HEAO(J+10),SAVE(J+11),CR(J,I)                       RFNE  10
      GO TO 2284                                                        RFNE  20
 6283 WRITE(IOUT,82)HEAD(J+10),SAVE(J+11),CR(J,I),SIGMA                 RFNE  30
 2284 CONTINUE                                                          RFNE  40
      IF(ISOT(I)-3)2290,2285,2290                                       RFNE  50
C        FOORTH CUMULANT PARAMETRES                                     RFNE  60
 2285 DO 2288 J=1,15                                                    RFNE  70
      TEMP=SAVE(J+21)                                                   RFNE  80
      IJ=IPARA(IK+J+20)                                                 RFNE  90
      IF(IJ)2287,2287,2286                                              RFNE 100
 2286 TEMP1=B(IJ)/OAOB(IJ,1)                                            RFNE 110
      SUM=SUM+ABS(TEMP1)                                                RFNE 120
      IF(ABS(TEMP1).GT.ABS(SEMAX)) SEMAX=TEMP1                          RFNE 130
      WRITE(IOUT,86)HEAO(J+20),TEMP,B(IJ),OR(J,I),OAOB(IJ,1),TEMPI      RFNE 140
     1,IJ                                                               RFNE 150
      GO TO 2288                                                        RFNE 160
 2287 SIGMA=ERRCAL(I,J+20)                                              RFNE 170
      IF(SIGMA.GT.0.0000001)GO TO 6287                                  RFNE 180
      WRITE(IOUT,88)HEAO(J+20),TEMP,OR(J,I)                             RFNE 190
      GO TO 2288                                                        RFNE 200
 6287 WRITE(IOUT,88)HEAO(J+20),TEMP,OR(J,I),SIGMA                       RFNE 210
 2288 CONTINUE                                                          RFNE 220
 2290 IK=KPARA                                                          RFNE 230
      SUM=SUM/FLOAT(MKANK)                                              RFNE 240
      WRITE(IOUT,74)SEMAX,SUM                                           RFNE 250
C        CALCULATE CORRELATION MATRIX                                   RFNE 260
      IF(MCORR.NE.0)WRITE(IOUT,54)TITLE                                 RFNE 270
      IJ=MOEGN+1                                                        RFNE 280
      KNT=0                                                             RFNE 290
      DO 2300 I=1,20                                                    RFNE 300
 2300 HIB(I)=0.0                                                        RFNE 310
                                    DO 2320 I=1,MRANK                   RFNE 320
                                    DO 2310 J=I,MRANK                   RFNE 330
      B(J)=A(IJ)/(OAOB(I,1)*OAOB(J,1))                                  RFNE 340
      IF(I.EQ.J)GO TO 2310                                              RFNE 350
C        GET OFF-DIAGONAL CORRELATIONS WITH LARGEST MAGNITUOES          RFNE 360
      IF(KNT.LT.20)KNT=KNT+1                                            RFNE 370
      DO 2301 K=1,KNT                                                   RFNE 380
C        COMPARE CURRENT VALUE WITH LIST                                RFNE 390
      IF(ABS(B(J)).GT.ABS(HIB(K)))GO TO 2303                            RFNE 400
 2301 CONTINUE                                                          RFNE 410
      GO TO 2310                                                        RFNE 420
 2303 L=KNT-K                                                           RFNE 430
      IF(L.LT.1)GO TO 2305                                              RFNE 440
C        MOVE LAST ENTRIES UP IN THE LIST                               RFNE 450
      DO 2304 M=1,L                                                     RFNE 460
      N=KNT-M                                                           RFNE 470
      HIB(N+1)=HIB(N)                                                   RFNE 480
      KB(N+1)=KB(N)                                                     RFNE 490
 2304 JB(N+1)=JB(N)                                                     RFNE 500
C        INSERT ENTRY                                                   RFNE 510
 2305 HIB(K)=B(J)                                                       RFNE 520
      KB(K)=I                                                           RFNE 530
      JB(K)=J                                                           RFNE 540
 2310 IJ=IJ+1                                                           RFNE 550
      IF(MCORR.EQ.0)GO TO 2320                                          RFNE 560
C        PRINT ENTIRE CORRELATION MATRIX                                RFNE 570
      WRITE(IOUT,56)I,(J,B(J),J=I,MRANK)                                RFNE 580
 2320 CONTINUE                                                          RFNE 590
C        PRINT THE LARGEST VALUES                                       RFNE 600
      WRITE(IOUT,57)ITITLE,(KB(J),JB(J),HIB(J),J=1,KNT)                 RFNE 610
C              CHECK TO SEE IF AUXILLARY FUNCTIONS TO BE DONE           RFNE 620
 3000 IF(MBOOAN)3010,3020,3010                                          RFNE 630
 3010 CALL BOOANI(MBOOAN)                                               RFNE 640
 3020 IF(MELVIB)3030,3040,3030                                          RFNE 650
 3030 CALL ELVIBI(MELVIB,MBOOAN)                                        RFNE 660
 3040 IF((MBOOAN.EQ.0.AND.MELVIB.EQ.0).OR.NOEGN.EQ.0)GO TO 3042         RFNE 670
      REWIND ISC5                                                       RFNE 680
      READ(ISC5)LOEGN,IKOEGN(I),I=1,LOEGN)                              RFNE 690
 3042 READ(IN,18)IEF,NFLAG,ITEM1,ITEM2                                 RFNE 700
      IF(ITEM1.NE.0.OR.ITEM2.NE.0)ISC1=10*ITEM1+ITEM2                   RFNE 710
C        CHECK FUR LEGAL RANGE OF 'NFLAG'                               RFNE 720
      IF(NFLAG.LT.0.OR.NFLAG.GT.3)GO TO 7086                            RFNE 730
C        NO CARDS IF NFLAG=0                                            RFNE 740
      IF(NFLAG.EQ.0)GO TO 7083                                          RFNE 750
C        PUNCH XRAY67 CARDS IF NFLAG=1 OR 3                             RFNE 760
      IFINFLAG.EQ.2)GO TO 7093                                          RFNE 770
      DO 7091 I=1,NATOM                                                 RFNE 780
      DO 7090 J=1,6                                                     RFNE 790
 7090 TFB(J)=BETA(J,I)*EGGY(J)                                          RFNE 800
 7091 WRITE(IPUN,7092)(TAG(J,I),J=1,2),(XYZ(J,I),J=1,3),ITAG(J,I),J=1,2)RFNE 810
     $,TFB                                                              RFNE 820
 7092 FORMAT('ATOM',3X,2A3,3FB.4/'BIJ',4X,2A3,6FB.4,1X,'4')             RFNE 830
 7093 IF(NFLAG.EQ.1)GO TO 7083                                          RFNE 840
C        PUNCH 'RFINE' CAROS IF NFLAG=2 OR 3                            RFNE 850
C        PUNCH SCALE FACTOR AND EXTINCTION CAROS                        RFNE 860
      WRITE(IPUN,7097)(SCALE(I),I=1,NSF)                                RFNE 870
      WRITE(IPUN,27)EXTIN                                               RFNE 880
 7097 FORMAT(10FB.4)                                                    RFNE 890
      DO 7096 I=1,NATOM                                                 RFNE 900
      K=0                                                               RFNE 910
      IF(ISOT(I).GT.1)K=1                                               RFNE 920
      IFBUF(I)=I-ISOT(I)**3+6*ISOT(I)**2-2*ISOTII))/3                   RFNE 930
      WRITE(IPUN,16)(TAG(J,I),J=1,2),SITE(I),OCCUP(I),OCCAII),IISCAT(J,IRFNE 940
     $),J=1,2),(XYZIJ,I),J=1,3),(BETA(J,I),J=1,6),K                     RFNE 950
      IF(K.EQ.0) GO TO 7096                                             RFNE 960
C        PUNCH 3RO CUMULANT                                             RFNE 970
      K=ISOT(I)-2                                                       RFNE 980
      WRITE(IPUN,7094)(CR(J,I),J=1,10),K                                RFNE 990
 7094 FORMAT(9FB.6,F6.4,I2)                                             RFNE1000
      IF(K.EQ.0)GO TO 7096                                              RFNE1010
```

```
C        PUNCH 4TH CUMULANT                                              RFNE1020
      WRITE(IPUN,7095)(OR(J,(),J=1,15)                                   RFNE1030
 7095 FORMAT(10F8.6)                                                     RFNE1040
 7096 CONTINUE                                                           RFNE1050
      (F(NPAR.EQ.0)GO TO 7100                                            RFNE1060
C        PUNCH THE SPECIAL PARAMETERS                                    RFNE1070
      WR(TE(IPUN,90)NPAR                                                 RFNE1080
      CALL PUNPAR                                                        RFNE1090
C        PUNCH THE EXT(NCTIUN AND TEMPERATURE FACTOR TYPES               RFNE1100
 7100 WRITE((PUN,18)IEXT,(IF8UF(I),I=I,NATUM)                           RFNE1110
      GO TO 7083                                                         RFNE1120
 7086 WRITE(IOUT,7087)NFLAG                                              RFNE1130
 7087 FORMAT(' WRUNG VALUE OF NFLAG,ACTUAL VALUE (S',(3)                 RFNE1140
 7083 CONTINUE                                                           RFNE1150
      IF(MSF-2)9980,9970,9970                                            RFNE1160
 9970 WR(TE(ISC4)NATUM,((TAG(J,(),J=1,2),ISUT(I),((SCAT(J,I),J=1,2),    RFNE1170
     $OCCA(I),OCCUP(I),S(TE((),(XYZ(J,I),J=1,3),(88ETA(J,(),J=1,6),      RFNE1180
     $I=1,NATOM),(S(GU((),(SIGXYZ(J,I),J=1,3),(SIG8(J,I),J=1,6),I=1,NATORFNE1190
     $M)                                                                 RFNE1200
      ENO FILE ISC4                                                      RFNE1210
 9980 CUNTINUE                                                           RFNE1220
      (F(IEF(100,4000,100                                                RFNE1230
C            ENO OF CALCULAT(ON                                          RFNE1240
 4000 REWINO (SC1                                                        RFNE1250
      (F(MUO(IFILE,2))4080,4010,4080                                     RFNE1260
C            WRITE SAVE UNIT                                             RFNE1270
 4010 LOEGN=2*NOEGN                                                      RFNE1280
      WRITE(ISC1)                                                        RFNE1290
     1KCYL,LA8EL,RECELL,G,VOL,NSYM,NATOM,NSCAT,JCELL,MRANK,NPARA,NSF,OMIRFNE1300
     2N,OMAX,OELMAX,ICENT,(((RMAT((,J,K),I=1,3),J=1,3),K=1,NSYM),((TRANSRFNE1310
     3(I,J),I=1,3),J=1,NSYM),(((CELLP((,J),I=1,3),J=1,JCELL),(IPARA(),I=RFNE1320
     41,NPARA),((FCURVE(I,J),I=1,9),J=1,NSCAT),(SCALE((),(=1,NSF),IREJ,FRFNE1330
     5OMAX,FOMIN,RHOMN,RHOMX,(JSCRF(I),(=1,NSF),NKSF                    RFNE1340
     9,NPAR,(PAR(I),(PARNAM(J,I),J=1,2),IPAR(I),IRPAR(I),(=1,NPAR),FISGNRFNE1350
     0,IFSGN,EGGY,KEXTRF                                                RFNE1360
      WRITE(ISC1)                                                        RFNE1370
     1(((ANOM(J,(),J=1,2),I=1,NSCAT),IEXT,EXTIN,((TAG(J,I),J=1,2),(SOT(I)RFNE1380
     2,((SCAT(J,(),J=1,2),OCCA((),OCCUP((),S(TE(I),(XYZ(J,(),J=1,3),    RFNE1390
     3(8ETA(J,I),J=1,6),(CR(J,(),J=1,10),(OR(J,I),J=1,15),(=1,NATOM),    RFNE1400
     4LOEGN,(KOEGN((),I=1,LOEGN)                                        RFNE1410
 4020 READ(ISC2)(I8UFF(I),(=1,L8UFF)                                     RFNE1420
      WRITE(ISC1)(I8UFF(I),I=I,L8UFF)                                    RFNE1430
      (F(I8UFF(1))4060,4020,4060                                         RFNE1440
 4060 REW(NO ISC2                                                        RFNE1450
      IF(IFIRST.EQ.0.ANO.MFSEL.EQ.0(GO TO 4062                           RFNE1460
      (K=MOEGN+1                                                         RFNE1470
      (J=MOEGN+(MRANK*(MRANK+1))/2                                       RFNE1480
      WRITE(ISC1)MDEGN,(A(J),J=1,MOEGN)                                  RFNE1490
      WRITE(ISC1)(A(J),J=IK,IJ(                                          RFNE1500
      WRITE(ISC1)((88ETA(J,I),J=1,6),SIGO((),(S(GXYZ(J,I),J=1,3),(SIG8(JRFNE1510
     1,I),J=1,6),I=1,NATOM)                                              RFNE1520
 4062 ENO F(LE (SC1                                                      RFNE1530
      REWINO ISC1                                                        RFNE1540
C            END OF THIS ROUT(NE IF NEW PARAMETERS ARE TO 8E SAVEO      RFNE1550
      RETURN                                                             RFNE1560
C            ENO OF RUN FOR NEW PARAMETERS NOT SAVEO                    RFNE1570
 4080 CALL EXIT                                                          RFNE1580
      ENO                                                                RFNE1590
```

```
      BLOCK DATA                                                         INCM  10
      INCLUDE IND23,LIST                                                 INCM  20
      DATA IND2/1,1,2,2,3,3,1,2,1,3,2,3/IND3/1,1,1,2,2,2,3,3,3,1,1,2,1, INCM  30
     12,2,1,1,3,1,3,3,2,2,3,2,3,3,1,2,3/                                 INCM  40
      INCLUDE TAPE,LIST                                                  INCM  50
      DATA IN,IOUT,ISCI,ISC2,ISC3,ISC4,ISC5,IPUN/5,6,10,29,11,12,28,1/   INCM  60
      END                                                                INCM  70
```

```
      SUBROUTINE INPUT (IEP)                                                INPT  10
                                                                            INPT  20
C        THIS ROUTINE REVISED   MARCH 10, 1970                              INPT  30
                                                                            INPT  40
C             SUBROUTINE TO READ STRUCTURE FACTOR CARDS, COMPUTE SCATTER    INPT  50
C              -ING FACTORS AND WEIGHTS                                     INPT  60
                                                                            INPT  70
      DATA RAD/.0174533/                                                    INPT  80
      DIMENSION BK(3)                                                       INPT  90
                                                                            INPT 100
      INCLUDE DCOM.LIST                                                     INPT 110
      COMMON/E/DMAX,DMIN,RECELL(6),EXANG(3)                                 INPT 120
      COMMON/P/DELTA,FCALC,POBS,PRMAX,PRMIN,IREJ,MREP,MREJ,NREJ,NUMBER(3    INPT 130
     1 6),HDEN(36),RHO,RHOMN,RHOMX,RNUM(36),WT,EXBETA,EXX,EXY               INPT 140
      COMMON/G/HUPP(1023),FCURVE(9,20),PRMT(18),IWT,NSCAT,HETPAC,WVLNTH     INPT 150
     1,WTPAC                                                                INPT 160
      INCLUDE TAPE.LIST                                                     INPT 170
                                                                            INPT 180
      DATA LEN/22/                                                          INPT 190
      GO TO(1,2,3),IEP                                                      INPT 200
1     READ(INN,PRMT,END=120)IH,POBS,SIGMA,I,IEP,MREJ,EXBETA,TTH,OMEG,CHI    INPT 210
     1,PHI                                                                  INPT 220
C        CALCULATE ANGLES FOR ANISOTROPIC EXTINCTION                        INPT 230
      EXANG(1)=CHI*RAD                                                      INPT 240
      EXANG(2)=PHI*RAD                                                      INPT 250
      EXANG(3)=RAD*(OMEO*0.5*TTH)                                           INPT 260
      IP(IEP.NE.0)GO TO 120                                                 INPT 270
      GO TO 10                                                              INPT 280
2     READ(INN,PRMT,END=120)IH,MREJ,I,POBS,EXBETA,SIGMA                     INPT 290
      IP(IH(1).EQ.0.AND.IH(2).EQ.0.AND.IH(3).EQ.0)GO TO 2                   INPT 300
      GO TR 10                                                              INPT 310
3     READ(INN,PRMT,END=120)IH,MREJ,POBS,SIGMA                              INPT 320
      IP(IH(1).EQ.0.AND.IH(2).EQ.0.AND.IH(3).EQ.0)GO TO 3                   INPT 330
      I=1                                                                   INPT 340
   10 HHO=0.0                                                               INPT 350
      MREJ=MREJ-1                                                           INPT 360
      IP(SIGMA)16,14,16                                                     INPT 370
   14 SIGMA=1.0                                                             INPT 380
   16                          DO 20 K=1,3                                  INPT 390
      H(K)=IH(K)                                                            INPT 400
      BK(K)=H(K)*RECELL(K)                                                  INPT 410
   20 HHO=HHO+BK(K)**2                                                      INPT 420
      HHO=(HHO+2.0*(BK(1)*BK(2)*RECELL(6)+BK(1)*BK(3)*RECELL(5)+BK(2)*BK    INPT 430
     1 (3)*RECELL(4)))*0.25                                                 INPT 440
      IP(PORS-PRMAX)22,22,21                                                INPT 450
   21 PRMAX=PORR                                                            INPT 460
   22 IP(POBS-PRMIN)23,24,24                                                INPT 470
   23 PRMIN=PORS                                                            INPT 480
   24 IP(RHO-RHOMX)26,26,25                                                 INPT 490
   25 RHOMX=RHO                                                             INPT 500
   26 IP(RHR-RHRMN)27,28,28                                                 INPT 510
   27 RHOMN=RHO                                                             INPT 520
   28 SIG=SQRT(HHO)                                                         INPT 530
      IP(IEP.EQ.3)EXBETA=RETPAC/(SIN(2.*ASIN(SIG*WVLNTH)))                  INPT 540
      IEP=0                                                                 INPT 550
C             COMPUTE SCATTERING FACTORS                                    INPT 560
                          DO 40 L=1,NSCAT                                   INPT 570
      SCAT(L)=FCURVE(9,L)*ANOM(1,L)                                         INPT 580
                          DO 40 M=1,R,2                                     INPT 590
   40 SCAT(L)=SCAT(L)+PCURVE(M,L)*EXP(-PCURVE(M+1,L)*RHO)                   INPT 600
      WT=SIGMA                                                              INPT 610
C        COMPUTE SIGMA(F) FROM INFORMATION ON CARD                          INPT 620
      GO TO (90,50,60,RO,RO,B5),IWT                                         INPT 630
C        SIGMA(P**2) ON CARDS                                               INPT 640
   50 WT=0.5*SIGMA/POBS                                                     INPT 650
      GO TO 90                                                              INPT 660
C        WEIGHT ON CARD                                                     INPT 670
   60 WT=1.0/SQRT(WT)                                                       INPT 6RO
      IF(MREP)70,90,70                                                      INPT 690
   70 WT=2.0*PORS*WT                                                        INPT 700
      GO TO 90                                                              INPT 710
C        USR SPECIAL ROUTINE                                                INPT 720
   B0 CALL WEIGHT                                                           INPT 730
      IP(IWT-4)60,90,60                                                     INPT 740
C     INCREPORATE 'IONORANCE FACTOR'.                                       INPT 750
   R5 IP(MREJ.EQ.0)WT=SQRT(WT**2+(WTPAC*POBS)**2)                           INPT 760
C        SET SCALE FACTOR FOR REPLECTION                                    INPT 770
   90 IP(I)100,110,100                                                      INPT 7RO
  100 ISCL=I                                                                INPT 790
      RETURN                                                                INPT R00
  110 ISCL=1                                                                INPT R10
      RETURN                                                                INPT R20
  120 IEP=1                                                                 INPT R30
      RETURN                                                                INPT R40
      END                                                                   INPT R50
```

```
      COMPILER (FLD=0)                                                  RDGN  10
      SUBROUTINE RDEGN                                                  RDGN  20
C     SUBROUTINE TO READ IN THE DEPENDENCY RELATIONSHIPS AND ARRANGE    RDGN  30
C     THEM IN ORDER OF INDEPENDENT PARAMETERS.                          RDGN  40
      DEFINE I1(K)=FLD(0,9,K)                                           RDGN  50
      DEFINE I2(K)=FLD(9,9,K)                                           RDGN  60
      DEFINE I3(K)=FLD(18,9,K)                                          RDGN  70
      DEFINE I4(K)=FLD(27,9,K)                                          RDGN  80
      INCLUDE OCOM,LIST                                                 RDGN  90
      INCLUDE TAPE,LIST                                                 RDGN 100
      DIMENSION IDPAR(2),INPAR(2,5),CNPAR(5)                            RDGN 110
      NDEGN=0                                                           RDGN 120
      KMAX=2640                                                         RDGN 130
      KP=1                                                              RDGN 140
10    READ(IN,9000)IDPAR,((INPAR(I,J),I=1,2),CNPAR(J),J=1,5)            RDGN 150
9000  FORMAT(2I3,5(2I3,F8.0))                                           RDGN 160
      IF(IDPAR(1).EQ.0.AND.IDPAR(2).EQ.0)GO TO 100                      RDGN 170
      DO 20 I=1,5                                                       RDGN 180
      IF(INPAR(1,I).EQ.0.AND.INPAR(2,I).EQ.0.AND.CNPAR(I).EQ.0.)GO TO 1DRDGN 190
      IF(INPAR(2,I).LT.0)GO TO 12                                       RDGN 200
      KDEGN(KP)=IPAK(INPAR(1,I),INPAR(2,I),IDPAR(1),IDPAR(2))           RDGN 210
      AA(KP*1)=CNPAR(I)                                                 RDGN 220
      KP=KP*2                                                           RDGN 230
      NDEGN=NDEGN*1                                                     RDGN 240
      IF(KP.LT.KMAX)GO TO 20                                            RDGN 250
C     TOO MANY RELATIONS                                                RDGN 260
11    WRITE(IOUT,9011)                                                  RDGN 270
9011  FORMAT('0TOO MANY DEPENDENCY RELATIONS. RUN ABORTED.')            RDGN 280
      STOP                                                              RDGN 290
C     'RUN' INDICATED ON DEPENDENCY CARD.                               RDGN 300
12    IF(NOD(I,2).NE.0.OR.INPAR(1,I).NE.0)GO TO 16                      RDGN 310
      JMAX=-INPAR(2,I)                                                  RDGN 320
      JMIN=INPAR(2,I-1)*1                                               RDGN 330
      DO 15 J=JMIN,JMAX                                                 RDGN 340
      KDEGN(KP)=IPAK(INPAR(1,I),J,IDPAR(1),IDPAR(2))                    RDGN 350
      AA(KP*1)=0.                                                       RDGN 360
      KP=KP*2                                                           RDGN 370
      NDEGN=NDEGN*1                                                     RDGN 380
      IF(KP-KMAX)15,11,11                                               RDGN 390
15    CONTINUE                                                          RDGN 400
      GO TO 20                                                          RDGN 410
C     ERROR ON DEPENDENCY CARD                                          RDGN 420
16    WRITE(IOUT,9016)                                                  RDGN 430
9016  FORMAT('0IMPROPER DEPENDENCY INFORMATION. RUN ABORTED.')          RDGN 440
      STOP                                                              RDGN 450
20    CONTINUE                                                          RDGN 460
      GO TO 10                                                          RDGN 470
C     RESHUFFLE CONSTRAINT INDICATORS TO GROUP THEM ACCORDING TO THE    RDGN 480
C     INDEPENDENT PARAMETER.                                            RDGN 490
100   NDEGN=NDEGN-1                                                     RDGN 500
      DO 11D I=1,II                                                     RDGN 510
      JJ=I*1                                                            RDGN 520
      DO 110 J=JJ,NDEGN                                                 RDGN 530
      IF(KDEGN(2*I-1)-KDEGN(2*J-1))11D,110,                             RDGN 540
      ITEMP=KDEGN(2*I-1)                                                RDGN 550
      TEMP=AA(2*I)                                                      RDGN 560
      KDEGN(2*I-1)=KDEGN(2*J-1)                                         RDGN 570
      AA(2*I)=AA(2*J)                                                   RDGN 580
      KDEGN(2*J-1)=ITEMP                                                RDGN 590
      AA(2*J)=TEMP                                                      RDGN 600
110   CONTINUE                                                          RDGN 610
200   CONTINUE                                                          RDGN 620
      RETURN                                                            RDGN 630
      END                                                               RDGN 640
```

```
      COMPILER (FLD=0)                                                  RSET  10
      SUBROUTINE RESPT(NA,KPARA)                                        RSET  20
C     SUBROUTINE TO DETERMINE THE PROPER VALUES OF CRYSTALLOGRAPHIC     RSET  30
C     PARAMETERS THAT ARE FUNCTIONS OF OTHER PARAMETERS..               RSET  40
      DEFINE I1(K)=FLD(0,9,K)                                           RSET  50
      DEFINE I2(K)=FLD(9,9,K)                                           RSET  60
      DEFINE I3(K)=FLD(18,9,K)                                          RSET  70
      DEFINE I4(K)=FLD(27,9,K)                                          RSET  80
      INCLUDE ACOM,LIST                                                 RSET  90
      INCLUDE DCOM,LIST                                                 RSET 100
      INCLUDE XPAR,LIST                                                 RSET 110
      INCLUDE LSMAT,LIST                                                RSET 120
      INCLUDE IND23,LIST                                                RSET 130
      INCLUDE TAPE,LIST                                                 RSET 140
      LDEGN=2*NDEGN                                                     RSET 150
      IP(NA).,.100                                                      RSET 160
C     ARE THERE ANY SPECIAL PARAMETERS THAT ARE THEMSELVES DEPENDENT    RSET 170
C     (AS IN NON-CRYSTALLOGRAPHIC SYNMETRY, ISOTROPIC TRANSLATION OR    RSET 180
C     LIBRATION, ETC.)? THESE DEPENDENCIES MUST BE SIMPLE MULTIPLIERS.  RSET 190
      DO 90 I=1,LOEGN,2                                                 RSET 200
      IP(I3(KDEGN(I)).NE.0)GO TO 90                                     RSET 210
C     A DEPENDENT SPECIAL PARAMETER. HAS IT ALREADY BEEN SET?           RSET 220
      J=I4(KDEGN(I))                                                    RSET 230
      IP(IRPAR(J).NE.0)GO TO 90                                         RSPT 240
C     NO IT HASN'T. PROCEED TO SET IT.                                  RSET 250
      IRPAR(J)=-I                                                       RSET 260
      PAR(J)=0.                                                         RSET 270
C     IF INDEPENDENT PARAMETER IS 0.0, INITIALIZE TO CONSTANT.          RSET 280
      KK=I                                                              RSET 290
      IP(I2(KDEGN(I)).NE.0)GO TO 20                                     RSET 300
      KK=KK*2                                                           RSET 310
      PAR(J)=AA(I*1)                                                    RSET 320
20    IP(KK.GT.LDEGN)GO TO 90                                           RSET 330
      DO 50 K=KK,LDEGN,2                                                RSET 340
      IP(I3(KDEGN(K)).NE.0.OR.I4(KDEGN(K)).NE.J)GO TO 50                RSET 350
      LL=I2(KDEGN(K))                                                   RSET 360
      PAR(J)=PAR(J)*PAR(LL)*AA(K*1)                                     RSET 370
50    CONTINUE                                                          RSET 380
90    CONTINUE                                                          RSET 390
      DO 95 I=1,NPAR                                                    RSET 400
      IF(IPPAR(I).LT.0)IRPAR(I)=0                                       RSET 410
95    CONTINUE                                                          RSET 420
      RETURN                                                            RSET 430
C     COMPUTE NEW VALUES OF PARAMETERS FOR ATOM NA.                     RSET 440
100   NPARA=5                                                           RSET 450
      IP(ISOT(NA).GE.1)NPARA=NPARA*5                                    RSET 460
      IP(ISOT(NA).GE.2)NPARA=NPARA*10                                   RSET 470
      IP(ISOT(NA).GE.3)NPARA=NPARA*15                                   RSET 480
      IP(LDPGN.LE.0)GO TO 1001                                          RSET 490
      DO 1000 J=1,NPARA                                                 RSET 500
C     IS PARAMETER INDEPENDENT?                                         RSET 510
      IF(IPARA(KPARA*J).GT.0)GO TO 1000                                 RSET 520
C     IS THIS PARAMETER DEPENDENT ON ANY OTHERS?                        RSET 530
      DO 900 K=I,LDEGN,2                                                RSET 540
      IP(I3(KDEGN(K)).NE.NA.OR.I4(KDEGN(K)).NE.J)GO TO 900              RSET 550
C     DEPENDENT PARAMETER. SET IT.                                      RSET 560
      IP(IPARA(KPARA*J).LT.0)GO TO 900                                  RSET 570
      IPARA(KPARA*J)=-I                                                 RSET 580
      TEMPAR=0.                                                         RSET 590
      KK=K                                                              RSET 600
      IP(I1(KDEGN(K)).NE.0.OR.I2(KDEGN(K)).NE.0)GO TO 110               RSET 610
      TEMPAR=AA(K*1)                                                    RSET 620
      KK=KK*2                                                           RSET 630
110   IF(KK.GT.LOEGN)GO TO 170                                          RSET 640
      DO 150 L=KK,LDEGN,2                                               RSET 650
      IF(I3(KDEGN(L)).NE.NA.OR.I4(KOEGN(L)).NE.J)GO TO 150              RSET 660
C     IS THE INDEPENDENT PARAMETER SPECIAL?                             RSET 670
      MK=I2(KDEGN(L))                                                   RSET 680
      IF(I1(KDEGN(L)).EQ.0)GO TO 200                                    RSET 690
C     NO. LINEAR DEPENDENCIES ONLY. WHAT KIND OF PARAMETER?             RSET 700
      MJ=I1(KDEGN(I))                                                   RSET 710
      IF(MK-2)112,,                                                     RSET 720
      IF(MK-5)116,,                                                     RSET 730
      IF(MK-11)120,,                                                    RSET 740
      IF(MK-21)124,128,128                                              RSET 750
112   TEMPAR=TEMPAR*OCCA(MJ)*AA(L*1)                                    RSET 760
      ITYPE=1                                                           RSET 770
      GO TO 150                                                         RSET 780
116   TEMPAR=TEMPAR*XYZ(MK-1,MJ)*AA(L*I)                                RSET 790
      ITYPE=2                                                           RSET 800
      GO TO 150                                                         RSET 810
120   TEMPAR=TEMPAR*BETA(MK-4,MJ)*AA(L*1)                               RSET 820
      ITYPE=3                                                           RSET 830
      GO TO 150                                                         RSET 840
124   TEMPAR=TEMPAR*OR(MK-10,MJ)*AA(L*I)                                RSET 850
      ITYPE=4                                                           RSET 860
      GO TO 150                                                         RSET 870
128   TEMPAR=TEMPAR*DR(MK-20,MJ)*AA(L*I)                                RSET 880
      ITYPE=5                                                           RSET 890
150   CONTINUE                                                          RSET 900
151   GO TO(152,154,156,158,160),ITYPE                                  RSET 910
152   OCCA(NA)=TEMPAR                                                   RSET 920
      GO TO 1000                                                        RSET 930
154   XYZ(J-1,NA)=TEMPAR                                                RSET 940
      GO TO 1000                                                        RSET 950
156   BETA(J-4,NA)=TEMPAR                                               RSET 960
      GO TO 1000                                                        RSET 970
158   OR(J-10,NA)=TEMPAR                                                RSET 980
      GO TO 1000                                                        RSET 990
160   OR(J-20,NA)=TEMPAR                                                RSET1000
      GO TO 1000                                                        RSET1010
170   IF(J.EQ.1)ITYPE=1                                                 RSET1020
      IF(J.GE.2)ITYPE=2                                                 RSET1030
      IF(J.GE.5)ITYPE=3                                                 RSET1040
      IF(J.GE.11)ITYPE=4                                                RSET1050
      IF(J.GE.21)ITYPE=5                                                RSET1060
      GO TO 151                                                         RSET1070
C     INDEPENDENT PARAMETER SPECIAL, SET ALL DEPENDENT PARAMETERS FOR   RSET1080
C     THIS ATOM WHICH DEPEND ON THIS SET OF SPECIAL PARAMETERS.         RSET1090
200   IF(I1(IPAR(MK)).EQ.1)GO TO 210                                    RSET1100
      IF(I1(IPAR(MK)).EQ.0)GO TO 2000                                   RSET1110
      CALL SPVAL(NA,IPAR(MK),J)                                         RSET1120
      GO TO 1000                                                        RSET1130
C     DEPENDENT PARAMETER IS A LINEAR FUNCTION OF SPECIAL PARAMETERS.   RSET1140
2000  DO 2100 NK=L,LOEGN,2                                              RSET1150
```

```
      IF(13(KOEGN(NK)),NE.NA.OR.14(KDEGN(NK)),NE.J)GO TO 210D          RSET1160
      IF(11(KDEGN(NK)),NE.0)GO TO 220D                                 RSET1170
      MK=12(KDEGN(NK))                                                 RSET1180
      TEMPAR=TEMPAR*PAR(MK)**A(NK*1)                                   RSET1190
2100  CONTINUE                                                         RSET1200
      GO TO 170                                                        RSET1210
2200  WRITE(10UT,9200)                                                 RSET1220
9200  FORMAT('DEAULTY DEPENDENCY INFORMATION. RUN ABORTED')            RSET1230
      STOP                                                             RSET1240
C     GET SET DESIGNATOR. AND CLEAR ALL ARRAYS.                        RSET1250
210   ISET=12(IPAR(MK))                                                RSET1260
      DO 214 II=1,3                                                    RSET1270
      OXYZ(II)=0.                                                      RSET1280
      SXYZ(II)=0.                                                      RSET1290
      OXP(II)=0.                                                       RSET1300
      DO 214 JJ=1,3                                                    RSET1310
      T(II,JJ)=D.                                                      RSET1320
      EL(II,JJ)=0.                                                     RSET133D
214   S(II,JJ)=0.                                                      RSET1340
C     FILL IN VALUES FROM THIS SET.                                    RSET1350
      DO 220 II=1,NPAR                                                 RSET1360
      IF(12(IPAR(II)),NE.ISET)GO TO 220                                RSET1370
      CALL FILLIN(II,14(IPAR(II)),PAR(II),NA)                          RSET1380
220   CONTINUE                                                         RSET1390
      CALL EULER(OXP(3),OXP(2),OXP(1),TRNS)                            RSET1400
      CALL MULMN(XTOA,TRNS,TTRNS)                                      RSET1410
C     DO WE NEED TO TRANSFORM XYZ FROM CRYSTAL TO SPECIAL? ANSWER IS YESRSET1420
C     IF NO XYZ IS MENTIONED AS A DEPENDENT PARAMETER.                 RSET1430
      DO 23D II=1,LOEGN,2                                              RSET1440
      JJ=13(KDEGN(II))                                                 RSET1450
      KK=14(KDEGN(II))                                                 RSET1460
      LL=11(KDEGN(II))                                                 RSET1470
      IF(JJ.EQ.NA.AND.KK.GE.2.AND.KK.LE.4.AND.LL.EQ.0)GO TO 240        RSET148D
230   CONTINUE                                                         RSET1490
      CALL TRNSPS(TRNS,TRANSI)                                         RSET1S00
      CALL MULMN(TRANSI,ATOX,TTRNS1)                                   RSET1S10
      DO 232 II=1,3                                                    RSET1S20
232   RTEM(II)=XYZ(II,NA)-OXYZ(II)                                     RSET1S30
      CALL MULNV(TTRNSI,RTEM,SXYZ)                                     RSET1S40
      GO TO 250                                                        RSET1S50
C     XYZ DEPENDENT ON SPECIAL PARAMETER. COMPUTE THEM.                RSET1S60
240   CALL MULNV(TRNS,SXYZ,RTEM)                                       RSET1S70
      DO 244 II=1,3                                                    RSET158D
244   XYZ(II,NA)=RTEM(II)+OXYZ(II)                                     RSET1S90
      DO 246 II=2,4                                                    RSET160D
246   IPARA(KPARA*II)=-1                                               RSET161D
C     IF BETAS ARE DEPENDENT, COMPUTE THEM.                            RSET1620
250   DO 268 II=1,LOEGN,2                                              RSET1630
      JJ=11(KDEGN(II))                                                 RSET1640
      KK=13(KDEGN(II))                                                 RSET1650
      LL=14(KDEGN(II))                                                 RSET1660
      IF(JJ.NE.0.OR.KK.NE.NA.OR.LL.LT.S.OR.LL.GT.10)GO TO 268          RSET167D
C     APPLY LIBRATION CORRECTION TO SXYZ                               RSET168D
      CALL CORLIB(RTEM,EL,SXYZ)                                        RSET169D
      DO 260 JJ=1,6                                                    RSET1700
      JK=1ND2(1,JJ)                                                    RSET1710
      JL=1ND2(2,JJ)                                                    RSET1720
258   RTEM(JJ)=SKTLS(RTEM,JK,JL,T,EL,S)                                RSET173D
260   CONTINUE                                                         RSET1740
      CALL TFM2(TTRNS,RTEM,BETA(1,NA))                                 RSET175D
      DO 264 JJ=S,10                                                   RSET1760
264   IPARA(KPARA*JJ)=-1                                               RSET1770
      IF(ISOT(NA)-2)900,280,280                                        RSET178D
268   CONTINUE                                                         RSET179D
C     IF GAMMAS ARE DEPENDENT, COMPUTE THEM.                           RSET180D
27D   DO 298 II=1,LDEGN,2                                             RSET181D
      JJ=II(KDEGN(II))                                                 RSET182D
      KK=13(KDEGN(II))                                                 RSET1830
      LL=14(KDEGN(II))                                                 RSET1840
      IF(JJ.NE.D.OR.KK.NE.NA.OR.LL.LT.11.OR.LL.GT.20)GO TO 298         RSET1850
280   DO 290 JJ=1,10                                                   RSET1860
      JK=1ND3(1,JJ)                                                    RSET1870
      JL=1ND3(2,JJ)                                                    RSET188D
      JN=1ND3(3,JJ)                                                    RSET189D
282   GTEM(JJ)=TKTLS(RTEM,JK,JL,JM,EL,S)                               RSET19DD
29D   CONTINUE                                                         RSET191D
      CALL TFM3(TTRNS,GTEN,CR(1,NA))                                   RSET1920
      DO 294 JJ=11,20                                                  RSET193D
294   IPARA(KPARA*JJ)=-1                                               RSET1940
      GO TO 900                                                        RSET19SD
298   CONTINUE                                                         RSET196D
9D0   CONTINUE                                                         RSET1970
100D  CONTINUE                                                         RSET1980
1D01  DO 1D0S I=1,NPARA                                                RSET1990
      IF(IPARA(KPARA*I),LT.0)IPARA(KPARA*I)=D                          RSET20DD
10DS  CONTINUE                                                         RSET201D
      KPARA=KPARA*NPARA                                                RSET2020
      RETURN                                                           RSET2D30
      END                                                              RSET2040
```

```
      COMPILER (FLD=Q)                                                  DRVS  10
      SUBROUTINE DERIVS(NEW7)                                           DRVS  20
C     SUBROUTINE TO COMPUTE THE DERIVATIVES OF DEPENDENT, CRYSTALLO-    DRVS  30
C     GRAPHIC PARAMETERS WITH RESPECT TO THE SPECIAL PARAMETERS.        DRVS  40
      DEFINE I1(K)=FLD(0,9,K)                                           DRVS  50
      DEFINE I2(K)=FLD(9,9,K)                                           DRVS  60
      DEFINE I3(K)=FLD(18,9,K)                                          DRVS  70
      DEFINE I4(K)=FLD(27,9,K)                                          DRVS  80
      INCLUDE XPAR,LIST                                                 DRVS  90
      INCLUDE LSMAT,LIST                                                DRVS 100
      INCLUDE ACOM,LIST                                                 DRVS 110
      INCLDDE IND23,LIST                                                DRVS 120
      INCLUDE TAPE,LIST                                                 DRVS 130
      INCLUDE DCOM,LIST                                                 DRVS 140
      MDEGN=2*NDEGN                                                     DRVS 150
      IF(NPAR.EQ.0) GO TO 1001                                          DRVS 160
      I=-1                                                              DRVS 170
10    I=I+2                                                             DRVS 180
      IF(I1(KDEGN(I)).NE.0)GO TO 1001                                   DRVS 190
      IF(I2(KDEGN(I)).EQ.0.OR.I3(KDEGN(I)).EQ.0)GO TO 1000              DRVS 200
      II=I2(KDEGN(I))                                                   DRVS 210
      NA=I3(KDEGN(I))                                                   DRVS 220
      NPMIN=I4(KDEGN(I))                                                DRVS 230
      DO 20 J=1,MDEGN,2                                                 DRVS 240
      IF((J+2).GT.MDEGN)GO TO 22                                        DRVS 250
      IF(I3(KDEGN(J+2)).NE.NA)GO TO 22                                  DRVS 260
      IF(I2(KDEGN(J+2)).NE.II)GO TO 22                                  DRVS 270
20    CONTINUE                                                          DRVS 280
22    NPMAX=I4(KDECN(J))                                                DRVS 290
C     I POINTS AT THE FIRST PARAMETER OF ATOM NA WHICH IS DEPENDENT ON  DRVS 300
C     SPECIAL PARAMETER II. J POINTS AT THE LAST.                       DRVS 310
C     GET SET DESIGNATOR, CLEAR ARRAYS, AND COMPUTE TRANSFORMATION.     DRVS 320
      IF(I1(IPAR(II)).EQ.0)GO TO 990                                    DRVS 330
      IF(II(IPAR(II)).EQ.1)GO TO 30                                     DRVS 340
      CALL SPDER1(II,NA,NPMIN,NPMAX,1,J)                                DRVS 350
      GO TO 990                                                         DRVS 360
30    ISET=I2(IPAR(II))                                                 DRVS 370
      DO 34 K=1,3                                                       DRVS 380
      DXYZ(K)=0.                                                        DRVS 390
      SXYZ(K)=0.                                                        DRVS 400
      DXP(K)=0.                                                         DRVS 410
      DO 34 L=1,3                                                       DRVS 420
      T(K,L)=0.                                                         DRVS 430
      EL(K,L)=0.                                                        DRVS 440
34    S(K,L)=0.                                                         DRVS 450
C     FILL IN VALUES FROM THIS SET.                                     DRVS 460
      DO 40 K=1,NPAR                                                    DRVS 470
      IF(I2(IPAR(K)).NE.ISET)GO TO 40                                   DRVS 480
      CALL FILLIN(K,I4(IPAR(K)),PAR(K),NA)                              DRVS 490
40    CONTINUE                                                          DRVS 500
      CALL EDLER(DXP(3),DXP(2),DXP(1),TRNS)                             DRVS 510
C     DO WE NEED TO FILL IN KYZ                                         DRVS 520
      IF(NPMIN.LE.4)GO TO 50                                            DRVS 530
      CALL TRNSPS(TRNS,TRANSI)                                          DRVS 540
      CALL MULMM(TRANSI,ATOX,TTRNS1)                                    DRVS 550
      DO 42 K=1,3                                                       DRVS 560
42    RTEM(K)=XYZ(K,NA)-DXYZ(K)                                         DRVS 570
      CALL MULMV(TTRNSI,RTEM,SXYZ)                                      DRVS 580
C     WHAT KIND IS INDEPENDENT PARAMETER?                               DRVS 590
50    IDEN=I4(IPAR(II))                                                 DRVS 600
      JDEN=(IDEN+2)/3                                                   DRVS 610
      IF(JDEN.GE.3)CALL MULMM(XTOA,TRNS,TTRNS)                          DRVS 620
      GO TO (100,200,300,400,400,500,500,600,600,600),JDEN             DRVS 630
C     SPECIAL ORIGIN                                                    DRVS 640
100   DO 110 K=1,J,2                                                    DRVS 650
110   AA(K+1)=1.                                                        DRVS 660
      GO TO 990                                                         DRVS 670
C     ORIENTATION OF SPECIAL SYSTEM WRT CRYSTAL ORTHONORMAL.            DRVS 680
200   K=I                                                               DRVS 690
      IF(IDEN-5)210,230,250                                             DRVS 700
C     OMEGA ANGLE                                                       DRVS 710
210   CALL OTDOMG(DXP(3),DXP(2),DXP(1),TRNS)                            DRVS 720
212   CALL MULMM(XTOA,TRNS,TTRNS)                                       DRVS 730
      IF(NPMIN.GE.5)GO TO 215                                           DRVS 740
C     DERIVATIVES OF POSITION PARAMETERS WRT OMEGA,CHI, OR PHI.         DRVS 750
      CALL MULMV(TTRNS,SXYZ,RTEM)                                       DRVS 760
214   KK=I4(KDEGN(K))                                                   DRVS 770
      IF(KK.GE.5)GO TO 216                                              DRVS 780
      AA(K+1)=RTEM(KK-1)                                                DRVS 790
      K=K+2                                                             DRVS 800
      IF(K-J)214,214,990                                                DRVS 810
215   IF(NPMIN.GE.11)GO TO 221                                          DRVS 820
216   IF(NPMAX.LE.4)GO TO 990                                           DRVS 830
C     DERIVATIVES OF BETAS WRT OMEGA, CHI, OR PHI.                      DRVS 840
C     CORRECT SXYZ FOR LIBRATION.                                       DRVS 850
      CALL CORLIB(RTEM,EL,SXYZ)                                         DRVS 860
      DO 218 L=1,6                                                      DRVS 870
      JK=IND2(1,L)                                                      DRVS 880
      JL=IND2(2,L)                                                      DRVS 890
218   RTEM(L)=SKTLS(RTEM,JK,JL,T,EL,S)                                  DRVS 900
      CALL TPM2(TTRNS,RTEM,RTEM)                                        DRVS 910
220   KK=I4(KDEGN(K))                                                   DRVS 920
      IF(KK.GE.11)GO TO 222                                             DRVS 930
      AA(K+1)=RTEM(KK-4)                                                DRVS 940
      K=K+2                                                             DRVS 950
      IF(K-J)220,220,990                                                DRVS 960
221   CALL CORLIB(RTEM,EL,SXYZ)                                         DRVS 970
222   IF(NPMAX.LE.10)GO TO 990                                          DRVS 980
C     DERIVATIVES OF GAMMAS WRT OMEGA, CRI, OR PNI.                     DRVS 990
      DO 224 L=1,10                                                     DRVS1000
      JK=IND3(1,L)                                                      DRVS1010
      JL=IND3(2,L)                                                      DRVS1020
      JM=IND3(3,L)                                                      DRVS1030
224   GTEM(L)=TKTLS(RTEM,JK,JL,JM,EL,S)                                 DRVS1040
      CALL TPM3(TTRNS,GTEM,GTEM)                                        DRVS1050
226   KK=I4(KDEGN(K))                                                   DRVS1060
      IF(KK.GT.20)GO TO 990                                             DRVS1070
      AA(K+1)=GTEM(KK-10)                                               DRVS1080
      K=K+2                                                             DRVS1090
      IF(K-J)226,226,990                                                DRVS1100
C     CRI ANGLE                                                         DRVS1110
230   CALL OTDCNI(DXP(3),DXP(2),DXP(1),TRNS)                            DRVS1120
      GO TO 212                                                         DRVS1130
C     PRI ANGLE                                                         DRVS1140
250   CALL OTDPHI(DXP(3),DXP(2),DXP(1),TRNS)                            DRVS1150
```

```
      GØ TØ 212                                                    DRVS1160
C     PØS1TIØN IN SPECIAL SYSTEM                                   DRVS1170
300   K=1                                                          DRVS1180
      IF(NPMIN.GE.5)GØ TØ 350                                      DRVS1190
      DØ 310 L=1,3                                                 DRVS1200
310   BTEM(L)=0.                                                   DRVS1210
      BTEM(IDEN-6)=1.                                              DRVS1220
      CALL MULMV(TTRNS,BTEM,RTEM)                                  DRVS1230
320   KK=I4(KDEGN(K))                                              DRVS1240
      IF(KK.GT.4)GØ TØ 360                                         DRVS1250
      AA(K+1)=RTEM(KK-1)                                           DRVS1260
      K=K+2                                                        DRVS1270
      IF(K-J)320,320,990                                           DRVS1280
C      WE SHALL NØW IGNØRE THE DERIVATIVES ØF BETAS AND GAMMAS WRT  DRVS1290
C     SXYZ, ALTHØUGH THESE MAY ACTUALLY BE IMPØRTANT. WE WILL FILL THEM DRVS1300
C     IN LATER.                                                    DRVS1310
350   KK=I4(KDEGN(K))                                             DRVS1320
360   IF(KK.GT.10)GØ TØ 370                                        DRVS1330
      AA(K+1)=0.                                                   DRVS1340
      K=K+2                                                        DRVS1350
      IF(K-J)350,350,990                                           DRVS1360
365   KK=I4(KDEGN(K))                                              DRVS1370
370   IF(KK.GT.20)GØ TØ 990                                        DRVS1380
      AA(K+1)=0.                                                   DRVS1390
      K=K+2                                                        DRVS1400
      IF(K-J)365,365,990                                           DRVS1410
C     ELEMENTS ØF T TENSØR, ØNLY BETAS INVØLVED.                   DRVS1420
400   K=I                                                          DRVS1430
      IDEN=IDEN-9                                                  DRVS1440
      JK=IND2(1,IDEN)                                              DRVS1450
      JL=IND2(2,IDEN)                                              DRVS1460
      DØ 410 L=1,6                                                 DRVS1470
      JM=IND2(1,L)                                                 DRVS1480
      JN=IND2(2,L)                                                 DRVS1490
410   BTEM(L)=D2KDT(JM,JN,JK,JL)                                   DRVS1500
      CALL TFM2(TTRNS,BTEM,BTEM)                                   DRVS1510
420   KK=I4(KDEGN(K))                                              DRVS1520
      IF(KK.GT.10)GØ TØ 990                                        DRVS1530
      IF(KK.LT.5)GØ TØ 425                                         DRVS1540
      AA(K+1)=BTEM(KK-4)                                           DRVS1550
425   K=K+2                                                        DRVS1560
      IF(K-J)420,420,990                                           DRVS1570
C     ELEMENTS ØF L TENSØR, BETAS AND GAMMAS INVØLVED.             DRVS1580
500   K=I                                                          DRVS1590
      IDEN=IDEN-15                                                 DRVS1600
      JK=IND2(1,IDEN)                                              DRVS1610
      JL=IND2(2,IDEN)                                              DRVS1620
      CALL CØRLIB(RTEM,EL,SXYZ)                                    DRVS1630
      IF(NPMIN.GT.10)GØ TØ 560                                     DRVS1640
      DØ 510 L=1,6                                                 DRVS1650
      JM=IND2(1,L)                                                 DRVS1660
      JN=IND2(2,L)                                                 DRVS1670
510   BTEM(L)=D2KDL(RTEM,EL,S,JM,JN,JK,JL)                         DRVS1680
      CALL TFM2(TTRNS,BTEM,BTEM)                                   DRVS1690
520   KK=I4(KDEGN(K))                                              DRVS1700
      IF(KK.GT.10)GØ TØ 560                                        DRVS1710
      IF(KK.LT.5)GØ TØ 525                                         DRVS1720
      AA(K+1)=BTEM(KK-4)                                           DRVS1730
525   K=K+2                                                        DRVS1740
      IF(K-J)520,520,950                                           DRVS1750
560   DØ 570 L=1,10                                                DRVS1760
      JM=IND3(1,L)                                                 DRVS1770
      JN=IND3(2,L)                                                 DRVS1780
      JP=IND3(3,L)                                                 DRVS1790
570   GTEM(L)=D3KDL(RTEM,EL,S,JM,JN,JP,JK,JL)                      DRVS1800
      CALL TFM3(TTRNS,GTEM,GTEM)                                   DRVS1810
580   KK=I4(KDEGN(K))                                              DRVS1820
      IF(KK.GT.20)GØ TØ 990                                        DRVS1830
      AA(K+1)=GTEM(KK-10)                                          DRVS1840
      K=K+2                                                        DRVS1850
      IF(K-J)580,580,990                                           DRVS1860
C     ELEMENTS ØF S TENSØR.                                        DRVS1870
600   K=1                                                          DRVS1880
      IDEN=IDEN-21                                                 DRVS1890
      CALL CØRLIB(RTEM,EL,SXYZ)                                    DRVS1900
      IF(IDEN.GT.6)GØ TØ 605                                       DRVS1910
      JK=IND2(1,IDEN)                                              DRVS1920
      JL=IND2(2,IDEN)                                              DRVS1930
      GØ TØ 608                                                    DRVS1940
605   JL=IND2(1,IDEN-3)                                            DRVS1950
      JK=IND2(2,IDEN-3)                                            DRVS1960
608   DØ 610 L=1,6                                                 DRVS1970
      JM=IND2(1,L)                                                 DRVS1980
      JN=IND2(2,L)                                                 DRVS1990
610   BTEM(L)=D2KDS(RTEM,EL,JM,JN,JK,JL)                           DRVS2000
      CALL TFM2(TTRNS,BTEM,BTEM)                                   DRVS2010
620   KK=I4(KDEGN(K))                                              DRVS2020
      IF(KK.GT.10)GØ TØ 650                                        DRVS2030
      IF(KK.LT.5)GØ TØ 625                                         DRVS2040
      AA(K+1)=BTEM(KK-4)                                           DRVS2050
625   K=K+2                                                        DRVS2060
      IF(K-J)620,620,990                                           DRVS2070
650   DØ 660 L=1,10                                                DRVS2080
      JM=IND3(1,L)                                                 DRVS2090
      JN=IND3(2,L)                                                 DRVS2100
      JP=IND3(3,L)                                                 DRVS2110
660   GTEM(L)=D3KDS(RTEM,EL,S,JM,JN,JP,JK,JL)                      DRVS2120
      CALL TFM3(TTRNS,GTEM,GTEM)                                   DRVS2130
670   KK=I4(KDEGN(K))                                              DRVS2140
      IF(KK.GT.20)GØ TØ 990                                        DRVS2150
      AA(K+1)=GTEM(KK-10)                                          DRVS2160
      K=K+2                                                        DRVS2170
      IF(K-J)670,670,990                                           DRVS2180
990   I=J                                                          DRVS2190
1000  IF(I+2-MDEGN)10,,                                            DRVS2200
1001  CØNTINUE                                                     DRVS2210
C     IF THERE ARE ANY SPECIAL PARAMETERS THAT DEPEND ØN ØTHER SPECIAL DRVS2220
C     PARAMETERS, THE DERIVATIVES MUST FIRST BE CØRRECTED SØ THAT ALL DRVS2230
C     DERIVATIVES REFLECT ALL DEPENDENCIES. FIRST SAVE THE UNALTERED DRVS2240
C     LIST                                                         DRVS2250
      REWIND ISC5                                                  DRVS2260
      WRITE(ISC5)MDEGN,(KDEGN(I),I=1,MDEGN)                        DRVS2270
      IF(NPAR.NE.0.ØR.NEW7.NE.0)CALL PDEGN(NEW7)                   DRVS2280
      IF(NPAR.EQ.0)GØ TØ 2001                                      DRVS2290
      DØ 2000 I=1,MDEGN,2                                          DRVS2300
      IF(II(KDEGN(I)).NE.0)GØ TØ 2002                              DRVS2310
```

```
      IF(I3(KDEGN(I)).NE.0)GO TO 20D0                                   DRVS232D
      JJ=I4(KDEGN(I))                                                   DRVS233D
      II=I2(KDEGN(I))                                                   DRVS234D
      IJ=I+2                                                            DRVS235D
      DO 1500 J=IJ,MDEGN,2                                              DRVS236D
      IF(I1(KDEGN(J)).NE.D.OR.I2(KDEGN(J)).NE.II)GO TO 2D0D             DRVS2370
      KK=I3(KDEGN(J))                                                   DRVS238D
      LL=I4(KDEGN(J))                                                   DRVS2390
      DO 14DD K=J,MDEGN,2                                               DRVS2400
      IF(I1(KDEGN(K)).NE.0)GO TO 150D                                   DRVS241D
      IF(I4(KDEGN(K)).NE.LL.OR.I3(KDEGN(K)).NE.KK)GO TO 14DD            DRVS242D
      IF(I2(KDEGN(K)).EQ.JJ)AA(J+1)=AA(J+1)+AA(K+1)*AA(I+1)             DRVS243D
1400  CONTINUE                                                          DRVS2440
15DD  CONTINUE                                                          DRVS2450
2D0D  CONTINDE                                                          DRVS246D
20D2  DO 299D I=1,MDEGN,2                                             DRVS247D
      IF(I1(KDEGN(I)).NE.D)GO TO 2D01                                   DRVS2480
      IF(I3(KDEGN(I)).NE.D)GO TO 2990                                   DRVS2490
      II=I4(KDEGN(I))                                                   DRVS250D
      DO 2980 J=1,MDEGN,2                                               DRVS251D
      IF(I1(KDEGN(J)).EQ.0.ANO.I2(KDEGN(J)).EQ.II)KDEGN(J)=0            DRVS2520
2980  CONTINUE                                                          DRVS253D
299D  CONTINUE                                                          DRVS254D
C     NOW ELIMINATE ALL DEPENDENT SPECIAL PANANETERS. RESORT IF NECES-  DRVS255D
C     SARY, AND PACK THE LIST.                                          DRVS2560
C     ALSO ELIMINATE INITIALIZING CONSTANTS.                            DRVS2570
2001  DO 210D I=1,MDEGN,2                                               DRVS258D
      IF(II(KDEGN(I)).NE.0)GO TO 21D1                                   DRVS2590
      IF(I3(KDEGN(I)).NE.0.AND.I2(KDEGN(I)).NE.0)GO TO 21D0             DRVS2600
      NPACK=2                                                           ORVS2610
      J=I                                                               DRVS2620
2005  IF(I3(KDEGN(J+NPACK)).NE.D.AND.I2(KDEGN(J+NPACK)).NE.D)GO TO 2010 DRVS2630
      NPACK=NPACK+2                                                     DRVS2640
      IF(J+NPACK-MDEGN)2005,2005,2D95                                   DRVS2650
201D  KDEGN(J)=KDEGN(J+NPACK)                                           DRVS2660
209D  AA(J+I)=AA(J+I+NPACK)                                             ORVS2670
      J=J+2                                                             DRVS2680
      IF(J+NPACK-MDEGN)2005,2005,2095                                   DRVS2690
2095  MDEGN=MDEGN-NPACK                                                 DRVS27DD
      GO TO 2101                                                        DRVS2710
21D0  CONTINUE                                                          DRVS2720
21DI  CONTINUE                                                          DRVS273D
C     REPLACE INDEPENDENT PARAMETER DESIGNATORS BY THEIR MATRIX POINTERSDRVS274D
      JPAR=0                                                            DRVS275D
      JATOM=0                                                           DRVS276D
      DO 24D0 I=1,MDEGN,2                                               DRVS2770
      K3=I3(KDEGN(I))                                                   DRVS2780
      K4=I4(KDEGN(I))                                                   DRVS279D
      IF(II(KDEGN(I)).NE.0)GO TO 2320                                   ORVS2800
      J=I2(KDEGN(I))                                                    DRVS2810
      K2=INPAR(J)                                                       DRVS282D
      GO TO 2350                                                        DRVS283D
2320  IF(II(KDEGN(I)).EQ.JATOM)GO TO 233D                               DRVS284D
      IF(JATOM.EQ.0)GO TO 2329                                          DRVS285D
      IF(ISOT(JATOM-1)2324,2322,                                        DRVS286D
      IF(ISOT(JATOM).GT.2)JPAR=JPAR*15                                  DRVS287D
      JPAR=JPAR*1D                                                      DRVS288D
2322  JPAR=JPAR*5                                                       DRVS2890
2324  JPAR=JPAR*5                                                       DRVS2900
2329  JATOM=JATOM*1                                                     DRVS29I0
      GO TO 2320                                                        ORVS292D
233D  K1=I2(KDEGN(I))                                                   DRVS2930
      K2=IPARA(JPAR*K1)                                                 DRVS2940
2350  KDEGN(I)=IPAK(0,K2,K3,K4)                                         DRVS2950
24D0  CONTINUE                                                          DRVS296D
C     SHIFT ALTERED DEPENDENCY LIST UP INTO 'A' ARRAY.                  DNVS2970
      CALL CORSZ(A(NDEGN))                                              DRVS2980
      DO 25DD I=1,MDEGN                                                 DRVS2990
2500  A(I)=AA(I)                                                        DRVS3000
      RETURN                                                            ORVS3010
      END                                                               DRVS302D
```

```
      SUBROUTINE SFAC                                                   SFAC  10
                                                                        SFAC  20
C THIS ROUTINE REVISED FEB. 10, 1972                                    SFAC  30
                                                                        SFAC  40
C             STRUCTURE FACTOR GENERATION                               SFAC  50
                                                                        SFAC  60
      INCLUDE ACOM,LIST                                                 SFAC  70
      COMMON/C/CELLP(3,4),ICENT,JCELL,NSYM,RMAT(3,3,24),TRANS(3,24)     SFAC  80
      INCLUDE OCOM,LIST                                                 SFAC  90
      COMMON/E/OMAX,OMIN,RECELL(6),EXANG(3)                             SFAC 100
      COMMUN/F/UELTA,FCALC,FUBS,FOMAX,FOMIN,IREJ,MREF,MREJ,NREJ,NUMBER(3SFAC 110
     1 6),ROEN(36),RHO,RHOMN,RHOMX,RNUM(36),WT,EXBETA,EXX,EXY           SFAC 120
     2,FISGN,IFSGN,EXTIN(6),KEXTRF(6)                                   SFAC 130
      COMMON/LS/OAB(6),OAX(3),OBB(6),OBX(3),HCP(4),HHJ(6),HP(3,24),HPR( SFAC 140
     1 3),TPR(24),HHHJ(10),OAC(10),OBC(10),HHHH(15),OAO(15),OBO(15)     SFAC 150
      INCLUDE LSMAT,LIST                                                SFAC 160
      OIMENSION V(3),VV(6)                                              SFAC 170
      EQUIVALENCE (V,OAX),(VV,HHJ)                                      SFAC 180
                                                                        SFAC 190
      OATA TWOPI/6.2831 85307/                                          SFAC 200
                                                                        SFAC 210
      ISAVE=MFSEL                                                       SFAC 220
      IF(MFSEL.NE.0.ANO.(IREJ.EO.1.OR.IREJ.EO.2).ANO.MREJ.EO.1)MFSEL=0  SFAC 230
C             GENERATE TRANSFORMED INDICES                              SFAC 240
                              00 10 I=1,NSYM                            SFAC 250
      TPR(I)=0.0                                                        SFAC 260
                              00 10 J=1,3                               SFAC 270
      HP(J,I)=0.0                                                       SFAC 280
      TPR(I)=TPR(I)+H(J)*TRANS(J,I)                                     SFAC 290
                              00 10 K=1,3                               SFAC 300
   10 HPIJ,()=HPIJ,I)+HIK)*RMATIK,J,I)                                  SFAC 310
      ACALC=0.0                                                         SFAC 320
      BCALC=0.0                                                         SFAC 330
      AMINUS=0.0                                                        SFAC 340
      8MINUS=0.0                                                        SFAC 350
C             START LOOP THROUGH ATOMS                                  SFAC 360
                              00 450 L=1,NATOM                          SFAC 370
C             INITIALIZE                                                SFAC 380
      TF=JCELL*I2-ICENT)                                                SFAC 390
      EXPIJ=1.0                                                         SFAC 400
      AAINO=0.0                                                         SFAC 410
      88INO=0.0                                                         SFAC 420
      IF(MFSEL)20,60,20                                                 SFAC 430
   20                         00 30 M=1,3                             SFAC 440
      OAXIM)=0.0                                                        SFAC 450
   30 08X(M)=0.0                                                        SFAC 460
      IF(ISOT(L))40,70,40                                               SFAC 470
   40                         00 50 N=1,6                             SFAC 480
      OAB(N)=0.0                                                        SFAC 490
   50 088(N)=0.0                                                        SFAC 500
      IF(ISOT(L)-1)80,80,52                                             SFAC 510
   52 00 54 N=1,10                                                      SFAC 520
      OAC(N)=0.0                                                        SFAC 530
   54 OBC(N)=0.0                                                        SFAC 540
      IF(ISOT(L)-2)80,80,55                                             SFAC 550
   55 00 56 N=1,15                                                      SFAC 560
      OAO(N)=0.0                                                        SFAC 570
```

```
      COMPILER (FLO=0)                                                  PRXT  10
      SUBROUTINE PAREXT                                                 PRXT  20
C     SUBROUTINE TO READ IN SPECIAL PARAMETER INFORMATION,              PRXT  30
C     READ 1 CARD FOR EACH OF N SPECIAL PARAMETERS, CONTAINING THE      PRXT  40
C     FOLLOWING INFORMATION.                                            PRXT  50
C     COL 1-8       NAME OF PARAMETER - UP TO 8 ALPHANUMERIC CHARACTERS. PRXT  60
C     COL 9-10      TYPE OF PARAMETER, 1=SPECIAL COORDINATE SYSTEM (SHAPE PRXT  70
C                   CONSTRAINT) OR RIGID BODY THERMAL MOTION, 2=DEFINED BYPRXT  80
C                   USER IN SUBROUTINE SPDERI (SUPPLIED AS DUMMY).      PRXT  90
C     0 INDICATES A LINEAR FUNCTION, WITH COEFFICIENTS AS ON CONSTRAINT PRXT 100
C     CARDS                                                             PRXT 110
C     COL 11-12     NUMBER INDICATING WHICH OF SEVERAL SPECIAL COORDINATEPRXT 120
C                   OR RIGID BODY SYSTEMS THIS PARAMETER BELONGS TO.    PRXT 130
C     COL 13-14     PARAMETER DESIGNATOR, IF TYPE 1, THE CODE IS AS     PRXT 140
C                   FOLLOWS 1-X COORDINATE OF ORIGIN (IN CRYSTAL SYSTEM), PRXT 150
C                   2-Y COORDINATE OF ORIGIN, 3-Z COORDINATE OF ORIGIN, PRXT 160
C                   4-OMEGA ANGLE FOR TRANSFORM TO STANDARD, ORTHONORMAL PRXT 170
C                   SYSTEM, 5-CHI ANGLE FOR TRANSFORM, 6-PHI ANGLE FOR  PRXT 180
C                   TRANSFORM, 7-X COORDINATE IS SPECIAL SYSTEM, 8-Y    PRXT 190
C                   COORDINATE IS SPECIAL SYSTEM, 9-Z COORDINATE IN SPECI-PRXT 200
C                   AL SYSTEM, 10-T11, 11-T22, 12-T33,13-T12, 14-T13,   PRXT 210
C                   15-T23, 16-L11, 17-L22, 18-L33, 19-L12, 20-L13, 21-L23PRXT 220
C                   22-S11, 23-S22, 24-S33, 25-S21                      PRXT 230
C                   22-S11, 23-S22, 24-S33, 25-12, 26-313, 27-S23, 28-S21,PRXT 240
C                   29-S31, 30-S32.                                     PRXT 250
C     COL 15-24     INITIAL VALUE OF THIS PARAMETER.                    PRXT 260
      OEFINE I1(K)=FLD(0,9,K)                                           PRXT 270
      DEFINE I2(K)=FLDI9,9,K)                                           PRXT 280
      OEFINE I4(K)=FLO(27,9,K)                                          PRXT 290
      INCLUDE XPAR,LIST                                                 PRXT 300
      INCLUDE TAPE,LIST                                                 PRXT 310
      OIMENSION ITEM(3)                                                 PRXT 320
      IF(NPAR.EQ.0)RETURN                                               PRXT 330
      WRITE(IOUT,9001)                                                  PRXT 340
      00 100 I=1,NPAR                                                   PRXT 350
      READ(IN,9000)(PARNAM(J,I),J=1,2),ITEM,PAR(I)                     PRXT 360
      IPAR(I)=IPAK(ITEM(1),ITEM(2),0,ITEM(3))                           PRXT 370
      WRITE(IOUT,9002)(PARNAM(J,I),J=1,2),ITEM,PAR(I)                   PRXT 380
  100 CONTINUE                                                          PRXT 390
      RETURN                                                            PRXT 400
 9000 FORMAT(2A4,3I2,F10.4)                                             PRXT 410
 9001 FORMAT('0SPECIAL PARAMETERS')                                     PRXT 420
 9002 FORMAT(1X,2A4,3(2X,I2),2X,F10.4)                                  PRXT 430
      ENTRY PUNPAR                                                      PRXT 440
      00 200 I=1,NPAR                                                   PRXT 450
      ITEM(1)=I1(IPAR(I))                                               PRXT 460
      ITEM(2)=I2(IPAR(I))                                               PRXT 470
      ITEM(3)=I4(IPAR(I))                                               PRXT 480
      WRITE(IPUN,9000)(PARNAM(J,I),J=1,2),ITEM,PAR(I)                   PRXT 490
  200 CONTINUE                                                          PRXT 500
      END                                                               PRXT 510
```

```
   56 080(N)=0.0                                                        SFAC 580
      GO TO 80                                                          SFAC 590
   60 IF((SOT(L))80,70,80                                               SFAC 600
C             CALCULATE ISOTROPIC TEMPERATURE FACTOR                   SFAC 610
   70 TF=EXP(-8ETA(1,L)*RHO)*TF                                         SFAC 620
C             START LOOP THROUGH SYMMETRY OPERATIONS                   SFAC 630
   80                          DO 280 IA=1,NSYM                      SFAC 640
      ARG=TPR(IA)                                                       SFAC 650
                               DO 90 I8=1,3                          SFAC 660
      HPR(I8)=HP(I8,IA)                                                 SFAC 670
   90 ARG=ARG+HPR(I8)*XYZ(I8,L)                                         SFAC 680
      CUM3=0.0                                                          SFAC 690
      IF(ISOT(L))100,130,100                                            SFAC 700
C             GENERATE TRANSFORMED INDEX PRODUCTS FOR ANISO. ATOMS     SFAC 710
  100                          DO 110 IC=1,3                         SFAC 720
  110 HHJ(IC)=HPR(IC)**2                                                SFAC 730
      HHJ(4)=2.0*HPR(1)*HPR(2)                                          SFAC 740
      HHJ(5)=2.0*HPR(1)*HPR(3)                                          SFAC 750
      HHJ(6)=2.0*HPR(2)*HPR(3)                                          SFAC 760
C             CALCULATE ANISOTROPIC TEMPERATURE FACTOR                 SFAC 770
      TEMP=0.0                                                          SFAC 780
                               DO 120 IO=1,6                         SFAC 790
  120 TEMP=TEMP-HHJ(IO)*8ETA(IO,L)                                      SFAC 800
      CUM4=0.0                                                          SFAC 810
      IF(ISOT(L)-1)129,129,122                                          SFAC 820
C        CALCULATE THIRD CUMULANT INDEX PRODUCTS                       SFAC 830
  122 DO 123 IO=1,3                                                     SFAC 840
  123 HHHJ(IO)=HPR(IO)*HHJ(IO)                                          SFAC 850
      HHHJ(4)=3.0*HHJ(1)*HPR(2)                                         SFAC 860
      HHHJ(5)=3.0*HHJ(2)*HPR(1)                                         SFAC 870
      HHHJ(6)=3.0*HHJ(1)*HPR(3)                                         SFAC 880
      HHHJ(7)=3.0*HHJ(3)*HPR(1)                                         SFAC 890
      HHHJ(8)=3.0*HHJ(2)*HPR(3)                                         SFAC 900
      HHHJ(9)=3.0*HHJ(3)*HPR(2)                                         SFAC 910
      HHHJ(10)=3.0*HHJ(4)*HPR(3)                                        SFAC 920
      DO 124 IO=1,10                                                    SFAC 930
  124 CUM3=CUM3+HHHJ(IO)*CR(IO,L)                                       SFAC 940
      IF(ISOT(L)-2)129,129,125                                          SFAC 950
C        CALCULATE FOURTH CUMULANT INDEX PRODUCTS                      SFAC 960
  125 DO 126 IO=1,3                                                     SFAC 970
  126 HHHH(IO)=HHJ(IO)**2                                               SFAC 980
      HHHH( 4)= 4.0*HHHJ(1)*HPR(2)                                      SFAC 990
      HHHH( 5)= 4.0*HHHJ(1)*HPR(3)                                      SFAC1000
      HHHH( 6)= 6.0*HHJ(1)*HHJ(2)                                       SFAC1010
      HHHH(7)=12.0*HHJ(1)*HPR(2)*HPR(3)                                 SFAC1020
      HHHH(8)=6.0*HHJ(1)*HHJ(3)                                         SFAC1030
      HHHH(9)=4.0*HHHJ(2)*HPR(1)                                        SFAC1040
      HHHH(10)=12.0*HHJ(2)*HPR(1)*HPR(3)                                SFAC1050
      HHHH(11)=12.0*HHJ(3)*HPR(1)*HPR(2)                                SFAC1060
      HHHH(12)=4.0*HHHJ(3)*HPR(1)                                       SFAC1070
      HHHH(13)=4.0*HHHJ(2)*HPR(3)                                       SFAC1080
      HHHH(14)=6.0*HHJ(2)*HHJ(3)                                        SFAC1090
      HHHH(15)=4.0*HHHJ(3)*HPR(2)                                       SFAC1100
      DO 127 IO=1,15                                                    SFAC1110
  127 CUM4=CUM4+HHHH(IO)*OR(IO,L)                                       SFAC1120
C        CALCULATE TEMPERATURE FACTOR                                  SFAC1130
  129 EXP(J=EXP(TEMP+CUM4)                                              SFAC1140
  130 T=TWOPI*ARG-CUM3                                                  SFAC1150
      COSIJ=COS (T)*EXPIJ                                               SFAC1160
      SINIJ=SIN (T)*EXPIJ                                               SFAC1170
      AAINO=AA(NO+COSIJ                                                 SFAC1180
      (F(ICENT)140,150,140                                              SFAC1190
  140 88INO=88INO+S(NIJ                                                 SFAC1200
  150 IF(MFSEL)190,280,190                                              SFAC1210
C             CALCULATE POSITIONAL DERIVATIVES                         SFAC1220
  190                          DO 200 IAA=1,3                        SFAC1230
  200 OAX(IAA)=OAX(IAA)-HPR(IAA)*SINIJ                                  SFAC1240
      IF(ICENT)210,230,210                                              SFAC1250
  210                          DO 220 IA8=1,3                        SFAC1260
  220 08X(IA8)=08X(IA8)+HPR(IA8)*COSIJ                                  SFAC1270
  230 IF(ISOT(L))240,280,240                                            SFAC1280
C             CALCULATE THERMAL DERIVATIVES FOR ANISOTROPIC ATOMS      SFAC1290
  240                          DO 250 IAC=1,6                        SFAC1300
  250 OA8(IAC)=OA8(IAC)-HHJ(IAC)*COSIJ                                  SFAC1310
      IF(ICENT)260,270,260                                              SFAC1320
  260 DO 265 IO=1,6                                                     SFAC1330
  265 088(IO)=088(IO)-HHJ(IO)*SINIJ                                     SFAC1340
  270 IF(ISOT(L)-1)280,280,271                                          SFAC1350
C        CALCULATE THIRD CUM. DERIVS.                                  SFAC1360
  271 DO 272 IO=1,10                                                    SFAC1370
  272 OAC(IO)=OAC(IO)+HHHJ(IO)*S(NIJ                                    SFAC1380
      IF(ICENT)273,275,273                                              SFAC1390
  273 DO 274 IO=1,10                                                    SFAC1400
  274 OBC(IO)=OBC(IO)-HHHJ(IO)*COSIJ                                    SFAC1410
  275 IF(ISOT(L)-2)280,280,276                                          SFAC1420
C        CALCULATE FOURTH CUMULANT DERIVS.                             SFAC1430
  276 DO 277 IO=1,15                                                    SFAC1440
  277 OAO(IO)=OAO(IO)+HHHH(IO)*COS(J                                    SFAC1450
      IF(ICENT)278,280,278                                              SFAC1460
  278 DO 279 IO=1,15                                                    SFAC1470
  279 OBO(IO)=OBO(IO)+HHHH(IO)*S(NIJ                                    SFAC1480
  280 CONTINUE                                                          SFAC1490
  360 ITY1=ISCAT(1,L)                                                   SFAC1500
      ITY2=ISCAT(2,L)                                                   SFAC1510
      TEMP4=TF*S(TE(L)                                                  SFAC1520
      IF((TY2)370,380,370                                               SFAC1530
C             EVALUATE SCATTERING FACTOR FOR MULTIPLE OCCUPANCY        SFAC1540
  370 OCCB=UCCUP(L)-OCCA(L)                                             SFAC1550
      REAL1=SCAT((TY1)                                                  SFAC1560
      REAL2=SCAT((TY2)                                                  SFAC1570
      TEMP=UCCA(L)*REAL1+OCCB*REAL2                                     SFAC1580
      TEMP1=OCCA(L)*ANOM(2,ITY1)+OCCB*ANUM(2,(TY2)                      SFAC1590
      TEMP2=REAL1-REAL2                                                 SFAC1600
      TEMP3=ANOM(2,ITY1)-ANUM(2,ITY2)                                   SFAC1610
      GO TO 390                                                         SFAC1620
C             EVALDATE SCATTERING FACTOR FOR SINGLE OCCUPANCY          SFAC1630
  380 REAL1=SCAT1(ITY1)                                                 SFAC1640
      TEMP=UCCUP(L)*REAL1                                               SFAC1650
      TEMP3=ANOM(2,(TY1)                                                SFAC1660
      TEMP1=OCCUP(L)*TEMP3                                              SFAC1670
      TEMP2=REAL1                                                       SFAC1680
  390 88INU=FISGN*88(NO                                                 SFAC1690
      (FIMFSEL)392,396,392                                              SFAC1700
C             EVALUATE DERIVATIVES WITH RESPECT TO OCCUPANCY           SFAC1710
  392 OAOO(L)=TEMP4*(TEMP2*AAINO-TEMP3*88(NO)                           SFAC1720
      OBOO(L)=TEMP4*(TEMP2*88INO+TEMP3*AAINO)                           SFAC1730
```

```
      IF(IFSGN.EQ.0)GO TO 396                                          SFAC1740
C        EVALUATE STRUCTURE FACTOR FOR ENANTIOMORPHIC ORIENTATION       SFAC1750
      AMINUS =TEMP4*(TEMP*AAINO+TEMP1*BBINO)+AMINUS                    SFAC1760
      BMINUS=TEMP4*(-TEMP*BBINO+TEMP1*AAINO)+BMINUS                    SFAC1770
C              EVALUATE INDIVIDUAL ATOM CONTRIBUTIONS                   SFAC1780
  396 AINO(L)=TEMP4*(TEMP*AAIND-TEMP1*BBINO)                           SFAC1790
      BINO(L)=TEMP4*(TEMP*BBINO+TEMP1*AAINO)                           SFAC1800
      ACALC=ACALC+AINO(L)                                              SFAC1810
      BCALC=BCALC+BINO(L)                                              SFAC1820
      IF(MFSEL)400,450,400                                             SFAC1830
C              MODIFY DERIVATIVES                                       SFAC1840
  400 TEMP5=TWOPI*TEMP4                                                SFAC1850
                                  DO 410 IAG=1,3                       SFAC1860
      DAOX(IAG,L)=TEMP5*(TEMP*DAX(IAG)-TEMP1*OBX(IAG))                 SFAC1870
  410 OBOX(IAG,L)=TEMP5*(TEMP*OBX(IAG)+TEMP1*DAX(IAG))                 SFAC1880
      IF(ISOT(L))430,420,430                                           SFAC1890
C              DERIVATIVE WITH RESPECT TO ISOTROPIC TEMPERATURE FACTOR  SFAC1900
  420 DAOB(I,L)=-RHO*AINO(L)                                           SFAC1910
      OBOB(I,L)=-RHO*BINO(L)                                           SFAC1920
      GO TO 450                                                        SFAC1930
C              DERIVATIVES WITH RESPECT TO ANISOTROPIC TEMPERATURE FACTOR SFAC1940
  430                             DO 440 IAH=1,6                       SFAC1950
      DAOB(IAH,L)=TEMP4*(TEMP*DAB(IAH)-TEMP1*OBB(IAH))                 SFAC1960
  440 OBOB(IAH,L)=TEMP4*(TEMP*OBB(IAH)+TEMP1*DAB(IAH))                 SFAC1970
      IF(ISOT(L)-1)450,450,441                                         SFAC1980
C        DERIVATIVES WRT THE THIRD CUMULANT                             SFAC1990
  441 DO 442 IO=1,10                                                   SFAC2000
      DAOC(IO,L)=TEMP4*(TEMP*DAC(IO)-TEMP1*OBC(IO))                    SFAC2010
  442 OBOC(IO,L)=TEMP4*(TEMP*OBC(IO)+TEMP1*DAC(IO))                    SFAC2020
      IF(ISOT(L)-2)450,450,443                                         SFAC2030
C        DERIVATIVES WRT THE FOURTH CUMULANT                            SFAC2040
  443 DO 444 IO=1,15                                                   SFAC2050
      DAOO(IO,L)=TEMP4*(TEMP*DAO(IO)-TEMP1*OBO(IO))                    SFAC2060
  444 OBOO(IO,L)=TEMP4*(TEMP*OBO(IO)+TEMP1*DAO(IO))                    SFAC2070
  450 CONTINUE                                                         SFAC2080
  470 MFSEL=ISAVE                                                      SFAC2090
      RETURN                                                           SFAC2100
                                                                       SFAC2110
      ENTRY EXCALC                                                     SFAC2120
C        CALCULATE EXTINCTION FACTOR                                    SFAC2130
      IF(IEXT)1010,1000,1010                                           SFAC2140
C        ISOTROPIC FORM                                                 SFAC2150
 1000 EXX=EXBETA*EXTIN(1)*FCALC                                        SFAC2160
      OROE(1)=1.0                                                      SFAC2170
      RETURN                                                           SFAC2180
C        ANISOTROPIC EXTINCTION - CALCULATE TRIG FUNCTIONS OF ANGLES    SFAC2190
 1010 SCHI=SIN(EXANG(I))                                               SFAC2200
      CCHI=COS(EXANG(I))                                               SFAC2210
      SPHI=SIN(EXANG(2))                                               SFAC2220
      CPHI=COS(EXANG(2))                                               SFAC2230
      IF(IEXT-1)1030,1020,1030                                         SFAC2240
C        TYPE 1 CORRECTION                                              SFAC2250
 1020 V(1)=SPHI*SCHI                                                   SFAC2260
      V(2)=-CPHI*SCHI                                                  SFAC2270
      V(3)=-CCHI                                                       SFAC2280
      GO TO 1040                                                       SFAC2290
C        TYPE 2 ANISOTROPIC                                             SFAC2300
 1030 SNU=SIN(EXANG(3))                                                SFAC2310
      CNU=COS(EXANG(3))                                                SFAC2320
      V(1)=-CCHI*SPHI*CNU-SNU*CPHI                                     SFAC2330
      V(2)=CCHI*CPHI*CNU-SNU*SPHI                                      SFAC2340
      V(3)=-SCHI*CNU                                                   SFAC2350
C        FOR PRODUCTS OF COEFFICIENTS                                   SFAC2360
 1040 DO 1050 I=1,3                                                    SFAC2370
 1050 VV(I)=V(I)**2                                                    SFAC2380
      VV(4)=2.0*V(1)*V(2)                                              SFAC2390
      VV(5)=2.0*V(1)*V(3)                                              SFAC2400
      VV(6)=2.0*V(2)*V(3)                                              SFAC2410
      SUM=0.0                                                          SFAC2420
      DO 1060 I=1,6                                                    SFAC2430
 1060 SUM=SUM+VV(I)*EXTIN(I)                                           SFAC2440
      IF(SUM.LE.0.0)SUM=1.0                                            SFAC2450
      TEMP=-1.0/SQRT(SUM)                                              SFAC2460
      EXX=EXBETA*FCALC*ABS(TEMP)                                       SFAC2470
C        CALCULATE DERIVATIVES OF THE FORM DR-STAR/DE-IJ                SFAC2480
      DO 1070 I=1,6                                                    SFAC2490
 1070 OROE(I)=0.5*VV(I)*TEMP/SUM                                       SFAC2500
      RETURN                                                           SFAC2510
      END                                                              SFAC2520
```

```
      COMPILER (FLD*0)                                                  MATX  10
      SUBROUTINE MATRIX                                                 MATX  20
      DEFINE I2(K)*FLD(9,9,K)                                           MATX  30
      DEFINE I3(K)*FLD(18,9,K)                                          MATX  40
      DEFINE I4(K)*FLD(27,9,K)                                          MATX  50
                                                                        MATX  60
C         THIS ROUTINE REVISED FEB 10, 1972                             MATX  70
                                                                        MATX  80
C               ROUTINE TO FORM LEAST SQUARES MATRIX                    MATX  90
                                                                        MATX 100
      INCLUDE ACOM,LIST                                                 MATX 110
      INCLUDE XPAR,LIST                                                 MATX 120
      INCLUDE DCOM,LIST                                                 MATX 130
      COMMON/F/DELTA,FCALC,FOBS,FOMAX,FOMIN,IREJ,MREF,NREJ,NREJ,NUMBER(3MATX 140
     1 6),RDEN(36),RHO,RHOMN,RHOMX,RNDM(36),WT,XDBETA,EXX,EXY           MATX 150
     2,FISGN,IESGN,IXTIN(6),KEXTPI(6)                                   MATX 160
      COMMON/LS/DERIV(200)                                              MATX 170
      INCLUDE LSMAT,LIST                                                MATX 180
      DIMENSION LDLGN(200)                                              MATX 190
      EQUIVALENCE(IDEGN,A)                                              MATX 200
      DIMENSION DADTL(60),DBDTL(60)                                     MATX 210
      EQUIVALENCE (DADTL,AIND),(DBDTL,BIND)                             MATX 220
                                                                        MATX 230
      CONS=-0.5*(EXY*EXY)**3I/FCALC                                     MATX 240
      CONS2=0.25*FCALC**3*EXX*BETA*EXY                                  MATX 250
      IF(NREF)80,90,80                                                  MATX 260
   8D CONS=2.0*FCALC*CONS                                               MATX 270
      CONS2=2.0*FCALC*CONS2                                             MATX 280
   9D J=1                                                               MATX 29D
      IF(IEXTR(NI,0))J=6                                                MATX 300
      DO 100 I=1,J                                                      MATX 310
      IJ=KEXTRE(I)                                                      MATX 320
      IF(IJ,NE,0)DERIV(IJ)=CONS2*DBDF(I)                                MATX 330
  10D CONTINUE                                                          MATX 340
      IF(NREF,EQ,0)GO TO 140                                            MATX 350
      DO 120 I=1,NRSF                                                   MATX 360
      IJ=JSCRF(I)                                                       MATX 370
      IF(IJ,NE,0)DERIV(IJ)=0.                                           MATX 380
  120 CONTINUE                                                          MATX 390
C               DERIVATIVE WITH RESPECT TO SCALE FACTOR                 MATX 400
      IF(JSCRF(ISCL),EQ,0) GO TO 140                                    MATX 410
      J=JSCRF(ISCL)                                                     MATX 420
      DERIV(J)=-FOBS /SCALE(ISCL)                                       MATX 430
      IF(NREF)130,140,130                                               MATX 440
C               REFINEMENT ON F**2                                      MATX 450
  130 DERIV(J)=-2.0*FOBS *02/SCALE(ISCL)                                MATX 460
  140 ACALC=ACALC*CONS                                                  MATX 470
      BCALC=BCALC*CONS                                                  MATX 480
      II=1                                                              MATX 490
      IF(NPAR,EQ,0)GO TO 149                                            MATX 500
      DO 145 I=1,NPAR                                                   MATX 510
      IJ=I2PAR(I)                                                       MATX 520
      IF(IJ,LE,0)GO TO 145                                              MATX 530
      DERIV(IJ)=0.                                                      MATX 540
  141 IF(I2(LDEGN(II))-IJ),142,145                                      MATX 550
 1415 II=II*2                                                           MATX 560
      IF(II-MDEGN)141,141,149                                           MATX 570
  142 CALL MODIFY(DERIV(IJ),I3(LDEGN(II)),I4(LDEGN(II)),A(II*I)I        MATX 580
      GO TO 1415                                                        MATX 590
  145 CONTINUE                                                          MATX 600
  149 IK=0                                                              MATX 61D
                                  DO 240 J=1,NATOM                      MATX 620
      IK=IK*I                                                           MATX 630
      IJ=IPARA(IK)                                                      MATX 640
      IF(IJ)150,160,150                                                 MATX 650
C               DERIVATIVE WRT OCCUPANCY                                MATX 66D
  150 DERIV(IJ)=ACALC*DADO(J)+BCALC*DBDO(J)                             MATX 670
  152 IF(II-MDEGN),,160                                                 MATX 680
      IF(I2(LDEGN(II))-IJ),156,160                                      MATX 690
  154 II=II*2                                                           MATX 700
      GO TO 152                                                         MATX 710
  156 CALL MODIFY(DERIV(IJ),I3(LDEGN(II)),I4(LDEGN(III),A(II*I II       MATX 720
      GO TO 154                                                         MATX 730
  160                             DO 180 K=1,3                          MATX 740
      IK=IK*I                                                           MATX 750
      IJ=IPARA(IK)                                                      MATX 760
      IF(IJ)170,180,170                                                 MATX 770
C               DERIVATIVE WRT POSITION                                 MATX 780
  170 DERIV(IJ)=ACALC*DADX(K,J)+BCALC*DBDX(K,J)                         MATX 790
  172 IF(II-MDEGN),,180                                                 MATX 80D
      IF(I2(LDEGN(II))-IJ),176,180                                      MATX 810
  174 II=II*2                                                           MATX 820
      GO TO 172                                                         MATX 830
  176 CALL MODIFY(DERIV(IJ),I3(LDEGN(II)I,I4(LDEGN(III),A(II*I)I        MATX 84D
      GO TO 174                                                         MATX 850
  180 CONTINUE                                                          MATX 860
      IF(ISOT(J)1210,190,210                                            MATX 870
  190 IX=IK*I                                                           MATX 880
      IJ=IPARA(IK)                                                      MATX 890
      IF(IJ1200,240,200                                                 MATX 900
C               DERIVATIVE WRT ISOTROPIC TEMPERATURE FACTOR             MATX 910
  200 DERIV(IJ)=ACALC*DADB(I,J)+BCALC*DBDB(I,J)                         MATX 920
  202 IF(II-MDEGN),,240                                                 MATX 930
      IF(I2(LDEGN(II)I-IJ),206,240                                      MATX 940
  204 II=II*2                                                           MATX 950
      GO TO 202                                                         MATX 960
  206 CALL MODIFY(DERIV(IJ),I3(LDEGN(III),I4(LDEGN(II)I,A(II*I))       MATX 970
      GO TO 204                                                         MATX 980
  210                             DO 230 L=I,6                          MATX 990
      IK=IK*1                                                           MATX1000
      IJ=IPARA(IK)                                                      MATXID10
      IF(IJ)220,230,220                                                 MATXI020
C               DERIVATIVE WRT ANISOTROPIC TEMPERATURE FACTOR           MATX1D30
  220 DERIV(IJ)=ACALC*DADH(L,J)+BCALC*DBDH(L,J)                         MATX1D40
  222 IF(II-MDEGN),,230                                                 MATX1050
      IF(I2(LDEGN(II))-IJI,226,230                                      MATXI060
  224 II=II*2                                                           MATXI070
      GO TO 222                                                         MATXI080
  226 CALL MODIFY(DERIV(IJ),I3(LDEGN(II)I,I4(LDEGN(II)),A(II*I))       MATXI090
      GO TO 224                                                         MATXI100
  230 CONTINUE                                                          MATXI110
      IF(ISOT(J)-I)240,240,231                                          MATXI120
C        DERIVATIVES WRT TO THE THIRD CUMULANT                          MATXI130
  231 DO 233 L=I,10                                                     MATXI140
      IX=IX*1                                                           MATXI150
```

```
      IJ=IPARA(IK)                                                      MATX1160
      IF(IJ)232,233,232                                                 MATX1170
  232 DERIV(IJ)=ACALC*OAOC(L,J)+BCALC*DBOC(L,J)                         MATX1180
 2322 IF(I1-MDEGN),,233                                                 MATX1190
      IF(I2(LDEGN(I1))-IJ),2326,233                                     MATX1200
 2324 II=II+2                                                           MATX1210
      GO TO 2322                                                        MATX1220
 2326 CALL MODIFY(DERIV(IJ),I3(LOEGN(II)),I4(LDEGN(I1)),A(I1+1))       MATX1230
      GO TO 2324                                                        MATX1240
  233 CONTINUE                                                          MATX1250
      IF(IS8T(J)-2)240,240,234                                          MATX1260
C        DERIVATIVES WRT TO THE FOURTH CUMULANT                         MATX1270
  234 DO 236 L=1,15                                                     MATX1280
      IK=IK+1                                                           MATX1290
      IJ=IPARA(IK)                                                      MATX1300
      IF(IJ)235,236,235                                                 MATX1310
  235 DERIV(IJ)=ACALC*DADD(L,J)+BCALC*DBDD(L,J)                         MATX1320
 2352 IF(I1-MDEGN),,236                                                 MATX1330
      IF(I2(LDEGN(I1))-IJ),2356,236                                     MATX1340
 2354 II=11+2                                                           MATX1350
      GO TO 2352                                                        MATX1360
 2356 CALL MODIFY(DERIV(IJ),I3(LOEGN(II)),I4(LDEGN(I1)),A(I1+1))       MATX1370
      GO TO 2354                                                        MATX1380
  236 CONTINUE                                                          MATX1390
  240 CONTINUE                                                          MATX1400
C           FORM MATRICES                                               MATX1410
      INDEX=MDEGN+1                                                     MATX1420
                              DO 280 M=1,MRANK                          MATX1430
      TEMP=DERIV(M)*WT                                                  MATX1440
      IF(TEMP)260,250,260                                               MATX1450
  250 INDEX=INDEX+MRANK-M+1                                             MATX1460
      GO TO 280                                                         MATX1470
  260                         DO 270 N=M,MRANK                          MATX1480
      A(INDEX)=A(INDEX)+TEMP*DERIV(N)                                   MATX1490
  270 INDEX=INDEX+1                                                     MATX1500
      B(M)=B(M)-DELTA*TEMP                                              MATX1510
  280 CONTINUE                                                          MATX1520
      RETURN                                                            MATX1530
      END                                                               MATX1540
```

```
      SUBROUTINE MODIFY(DERIV,NA,NP,TEMP)                               MDFY  10
      INCLUDE OCOM,LIST                                                 MDFY  20
      IF(NP.GE.2)GO TO 100                                              MDFY  30
      DERIV=DERIV+TEMP*(ACALC*OADD(NA)+BCALC*DBDD(NA))                  MDFY  40
      RETURN                                                            MDFY  50
  100 IF(NP.GE.5)GO TO 200                                              MDFY  60
      DERIV=DERIV+TEMP*(ACALC*OAOX(NP-1,NA)+BCALC*OBOX(NP-1,NA))        MDFY  70
      RETURN                                                            MDFY  80
  200 IF(NP.GE.11)GO TO 300                                             MDFY  90
      DERIV=DERIV+TEMP*(ACALC*DADB(NP-4,NA)+BCALC*DBDB(NP-4,NA))        MDFY 100
      RETURN                                                            MDFY 110
  300 IF(NP.GE.21)GO TO 400                                             MDFY 120
      DERIV=DERIV+TEMP*(ACALC*OADC(NP-10,NA)+BCALC*OBOC(NP-10,NA))      MDFY 130
      RETURN                                                            MDFY 140
  400 DERIV=DERIV+TEMP*(ACALC*OADD(NP-20,NA)+BCALC*OBOC(NP-20,NA))      MDFY 150
      RETURN                                                            MDFY 160
      END                                                               MDFY 170
```

```
C     DUMMY SUBROUTINES SPVAL AND SPOERI                                SPVL  10
      SUBROUTINE SPVAL                                                  SPVL  20
      RETURN                                                            SPVL  30
      ENTRY SPOERI                                                      SPVL  40
      RETURN                                                            SPVL  50
      END                                                               SPVL  60
```

```
      SUBROUTINE RCALC(SCFF)                                            RCAL  10
                                                                        RCAL  20
C THIS ROUTINE REVISED MARCH 10, 1970                                   RCAL  30
                                                                        RCAL  40
C               ROUTINE TO CALCULATE CONTRIBUTIONS TO R                 RCAL  50
                                                                        RCAL  60
      COMMON/F/DELTA,FCALC,FOBS,FOMAX,FOMIN,IREJ,NREF,NREJ,NREJ,NUMBER(3RCAL  70
     1 6),RDEN(36),RHO,RHOMN,RHOMX,RNUM(36),WT,LXDELTA,EXX,EXY          RCAL  80
     2,FISGN,IESGN,FXTIN(6),KEXTRF(6),DELMAX                            RCAL  90
                                                                        RCAL 100
C               CALCULATE RANGE OF FOBS AND RHO                         RCAL 110
      SCF=SCFF                                                          RCAL 120
      IRNG1=(SCF*FOBS-FOMIN)/(FOMAX-FOMIN*0.00001)**8.0+1.0            RCAL 130
      IF(IRNG1)98,98,99                                                 RCAL 140
   98 IRNG)=1                                                           RCAL 150
      GO TO 110                                                         RCAL 160
   99 IF(IRNG1-8)110,110,100                                            RCAL 170
  100 IRNG1=8                                                           RCAL 180
  110 IRNG2=     (RHO-RHOMN)/(RHOMX-RHOMN*0.00001)**8.0+1.0            RCAL 190
      IF(IRNG2)102,102,104                                              RCAL 200
  102 IRNG2=1                                                           RCAL 210
      GO TO 130                                                         RCAL 220
  104 IF(IRNG2-8)130,130,120                                            RCAL 230
  120 IRNG2=8                                                           RCAL 240
C               CALCULATE NUMERATOR AND DENOMINATOR FOR WEIGHTED R      RCAL 250
  130 WT=SCF**2/WT**2                                                   RCAL 260
      IF(NREJ.NE.0.OR.IREJ.LT.2)GO TO 135                               RCAL 270
      WT=WT*(1.-3.*((WT*((FOBS-FCALC)/DELMAX)                           RCAL 280
     1**2)**2)**2.*((WT*((FOBS-FCALC)/DELMAX)**2)**3))                  RCAL 290
  135 IF(NREF)150,140,150                                               RCAL 300
C          REFINE ON F                                                  RCAL 310
  140 DELTA=FOBS-FCALC                                                  RCAL 320
      FDEN=WT*FOBS**2                                                   RCAL 330
      GO TO 160                                                         RCAL 340
C               REFINEMENT OF F**2                                      RCAL 350
  150 WT=0.25*WT/FOBS**2                                                RCAL 360
      DELTA=FOBS**2-FCALC**2                                            RCAL 370
      FDEN=FOBS**4*WT                                                   RCAL 380
  160 RNUM=WT*DELTA**2                                                  RCAL 390
      DELF=AUS(FOBS-FCALC)                                              RCAL 400
                                    DO 200 I=1,19,)8                    RCAL 410
      RDEN(I)=RDEN(I)+FDEN                                              RCAL 420
      RDEN(I+1)=RDEN(I+1)+FOBS                                          RCAL 430
      RNUM(I)=RNUM(I)+RNUM                                              RCAL 440
      RNUM(I+1)=RNUM(I+1)+DELF                                          RCAL 450
      NUMBER(I)=NUMBER(I)+1                                             RCAL 460
      J=I+IRNG1+1                                                       RCAL 470
      )1=0                                                              RCAL 480
  170 RNUM(J)=RNUM(J)+RNUM                                              RCAL 490
      RDEN(J)=RDEN(J)+FDEN                                              RCAL 500
      NUMBER(J)=NUMBER(J)+1                                             RCAL 510
      II=II+1                                                           RCAL 520
      IF()1-)190,180,190                                                RCAL 530
  180 J=1+IRNG2*5                                                       RCAL 540
      GO TO 170                                                         RCAL 550
  190 IF(NREJ)2)0,200,210                                               RCAL 560
  200 CONTINUE                                                          RCAL 570
  210 RETURN                                                            RCAL 580
      END                                                               RCAL 590
```

```
      COMPILER (FLD=0)                                                  FLLN  10
      SUBROUTINE FILLIN(IK,IP,F,NA)                                     FLLN  20
C     SUBROUTINE TO FILL IN THE SPECIAL PARAMETERS FROM THE LIST.       FLLN  30
C     IP=TYPE OF PARAMETER, F=CURRENT VALUE, IK IS POSITION IN PARAMETERFLLN  40
C     LIST.                                                             FLLN  50
      DEFINE I1(K)=FLD(0,9,K)                                           FLLN  60
      DEFINE I2(K)=FLO(9,9,K)                                           FLLN  70
      DEFINE I3(K)=FLO(18,9,K)                                          FLLN  80
      DEFINE I4(K)=FLO(27,9,K)                                          FLLN  90
      INCLUDE XPAR,LIST                                                 FLLN 100
      INCLUDE LSMAT,LIST                                                FLLN 110
      INCLUDE IND23,LIST                                                FLLN 120
      INCLUDE OCOM,LIST                                                 FLLN 130
      JO=(IP+2)/3                                                       FLLN 140
      GO TO(100,200,300,400,400,500,500,600,600,600),JO                 FLLN 150
C     SPECIAL ORIGIN                                                    FLLN 160
  100 OXYZ(IP)=F                                                        FLLN 170
      RETURN                                                            FLLN 180
C     ORIENTATION                                                       FLLN 190
  200 OXP(IP-3)=F                                                       FLLN 200
      RETURN                                                            FLLN 210
C     SPECIAL XYZ. WE MUST DETERMINE IF IT APPLIES TO ATOM NA.          FLLN 220
  300 LDEGN=2*NDEGN                                                     FLLN 230
      DO 310 I=1,LDEGN,2                                                FLLN 240
      IF(I2(KDEGN(I)),NE.IK)GO TO 310                                   FLLN 250
      IF(I1(KDEGN(I)),NE,0)GO TO 310                                    FLLN 260
      IF(I3(KDEGN(I)),NE.NA) GO TO 310                                  FLLN 270
      GO TO 320                                                         FLLN 280
  310 CONTINUE                                                          FLLN 290
      RETURN                                                            FLLN 300
  320 SXYZ(IP-6)=F                                                      FLLN 310
      RETURN                                                            FLLN 320
C     T TENSOR ELEMENT                                                  FLLN 330
  400 I=IP-9                                                            FLLN 340
      J=IN02(1,I)                                                       FLLN 350
      K=IN02(2,I)                                                       FLLN 360
      T(J,K)=F                                                          FLLN 370
      IF(J.NE,K)T(K,J)=F                                                FLLN 380
      RETURN                                                            FLLN 390
C     L TENSOR ELEMENT                                                  FLLN 400
  500 I=IP-15                                                           FLLN 410
      J=IN02(1,I)                                                       FLLN 420
      K=IN02(2,I)                                                       FLLN 430
      EL(J,K)=F                                                         FLLN 440
      IF(J.NE.K)EL(K,J)=F                                               FLLN 450
      RETURN                                                            FLLN 460
C     ELEMENT OF S TENSOR                                               FLLN 470
  600 I=IP-21                                                           FLLN 480
      IF(I.GT.6)GO TO 605                                               FLLN 490
      J=IN02(1,I)                                                       FLLN 500
      K=IN02(2,I)                                                       FLLN 510
      GO TO 610                                                         FLLN 520
  605 K=IN02(1,I-3)                                                     FLLN 530
      J=IN02(2,I-3)                                                     FLLN 540
  610 S(J,K)=F                                                          FLLN 550
      RETURN                                                            FLLN 560
      END                                                               FLLN 570
```

```
      COMPILER (FLO=Q)                                                  ERCL  10
      FUNCTION ERRCAL(NA,NP)                                            ERCL  20
      DEFINE I1(K)=FLO(0,9,K)                                           ERCL  30
      DEFINE I2(K)=FLO(9,9,K)                                           ERCL  40
      DEFINE I3(K)=FLO(18,9,K)                                          ERCL  50
      DEFINE I4(K)=FLO(27,9,K)                                          ERCL  60
      INCLUDE LSMAT,LIST                                                ERCL  70
      DIMENSION LISPAR(30),OERI(30),KOEGN(200)                          ERCL  80
      EQUIVALENCE (KDEGN,A)                                             ERCL  90
      IP=0                                                              ERCL 100
      IF(MOEGN)101,101,                                                 ERCL 110
      DO 100 I=1,MOEGN,2                                                ERCL 120
      IF(I3(KDEGN(I)).NE.NA.OR.I4(KDEGN(I)).NE.NP)GO TO 100             ERCL 130
      IP=IP+1                                                           ERCL 140
      IF(IP.GT.30)GO TO 101                                             ERCL 150
      LISPAR(IP)=I2(KOEGN(I))                                           ERCL 160
      OERI(IP)=A(I+1)                                                   ERCL 170
  100 CONTINUE                                                          ERCL 180
  101 IP(IP.NE.0)GO TO 150                                              ERCL 190
      ERRCAL=0.                                                         ERCL 200
      RETURN                                                            ERCL 210
  150 SUM=0.                                                            ERCL 220
      DO 200 I=1,IP                                                     ERCL 230
      DO 200 J=I,IP                                                     ERCL 240
      II=LISPAR(I)                                                      ERCL 250
      JJ=LISPAR(J)                                                      ERCL 260
      KK=MDEGN*((II-1)*(2*MRANK-II*2))/2+JJ-II+1                        ERCL 270
      TERM=OERI(I)*OERI(J)*A(KK)                                        ERCL 280
      IP(I.NE.J)SUM=SUM+TERM                                            ERCL 290
  200 SUM=SUM+TERM                                                      ERCL 300
      ERRCAL=SQRT(SUM)                                                  ERCL 310
      RETURN                                                            ERCL 320
      END                                                               ERCL 330
```

```
      SUBROUTINE EULER(PHI,CHI,OMEGA,TRANS)                             EULR  10
C     SUBROUTINE TO GENERATE THE MATRIX B-INVERSE FROM THE EULERIAN     EULR  20
C     ANGLES PHI,CHI,AND OMEGA.                                         EULR  30
      DATA RAD/57.29577792/                                             EULR  40
      DIMENSION TRANS(3,3)                                              EULR  50
      ASSIGN 10 TO KGO                                                  EULR  60
      GO TO 100                                                         EULR  70
   10 TRANS(1,1)=CP*CX*CO-SP*SO                                         EULR  80
      TRANS(1,2)=-CP*CX*SO-SP*CO                                        EULR  90
      TRANS(1,3)=CP*SX                                                  EULR 100
      TRANS(2,1)=SP*CX*CO+CP*SO                                         EULR 110
      TRANS(2,2)=-SP*CX*SO+CP*CO                                        EULR 120
      TRANS(2,3)=SP*SX                                                  EULR 130
      TRANS(3,1)=-SX*CO                                                 EULR 140
      TRANS(3,2)=SX*SO                                                  EULR 150
      TRANS(3,3)=CX                                                     EULR 160
      RETURN                                                            EULR 170
      ENTRY OTDPBI(PHI,CHI,OMEGA,TRANS)                                 EULR 180
      ASSIGN 20 TO KGO                                                  EULR 190
      GO TO 100                                                         EULR 200
   20 TRANS(1,1)=-SP*CX*CO-CP*SO                                        EULR 210
      TRANS(1,2)=SP*CX*SO-CP*CO                                         EULR 220
      TRANS(1,3)=-SP*SX                                                 EULR 230
      TRANS(2,1)=CP*CX*CO-SP*SO                                         EULR 240
      TRANS(2,1)=-CP*CX*SO-SP*CO                                        EULR 250
      TRANS(2,3)=CP*SX                                                  EULR 260
      DO 25 I=1,3                                                       EULR 270
      DO 23 J=1,2                                                       EULR 280
   23 TRANS(J,I)=TRANS(J,I)/RAD                                         EULR 290
   25 TRANS(3,I)=0.                                                     EULR 300
      RETURN                                                            EULR 310
      ENTRY DTDCBI(PHI,CHI,OMEGA,TRANS)                                 EULR 320
      ASSIGN 30 TO KGO                                                  EULR 330
      GO TO 100                                                         EULR 340
   30 TRANS(1,1)=-CP*SX*CO                                              EULR 350
      TRANS(1,2)=CP*SX*SO                                               EULR 360
      TRANS(1,3)=CP*CX                                                  EULR 370
      TRANS(2,1)=-SP*SX*CO                                              EULR 380
      TRANS(2,2)=SP*SX*SO                                               EULR 390
      TRANS(2,3)=SP*CX                                                  EULR 400
      TRANS(3,1)=-CX*CO                                                 EULR 410
      TRANS(3,2)=CX*SO                                                  EULR 420
      TRANS(3,3)=-SX                                                    EULR 430
      DO 35 I=1,3                                                       EULR 440
       DO 35 J=1,3                                                      EULR 450
   35 TRANS(I,J)=TRANS(I,J)/RAD                                         EULR 460
      RETORN                                                            EULR 470
      ENTRY OTDOMG(PHI,CHI,OMEGA,TRANS)                                 EULR 480
      ASSIGN 40 TO KGO                                                  EULR 490
      GO TO 100                                                         EULR 500
   40 TRANS(1,1)=-CP*CX*SO-SP*CO                                        EULR 510
      TRANS(1,2)=-CP*CX*CO+SP*SO                                        EULR 520
      TRANS(2,1)=-SP*CX*SO+CP*CO                                        EULR 530
      TRANS(2,2)=-SP*CX*CO-CP*SO                                        EULR 540
      TRANS(3,1)=SX*SO                                                  EULR 550
      TRANS(3,2)=SX*CO                                                  EULR 560
      DO 45 I=1,3                                                       EULR 570
```

```
      DO 43 J=1,2                                                    EULR 580
   43 TRANS(I,J)=TRANS(I,J)/RAD                                      EULR 590
45    TRANS(I,3)=0.                                                  EULR 600
      RETURN                                                         EULR 610
100   CP=COS(PHI/RAD)                                                EULR 620
      CX=COS(CHI/RAD)                                                EULR 630
      CO=COS(OMEGA/RAD)                                              EULR 640
      SP=SIN(PHI/RAD)                                                EULR 650
      SX=SIN(CHI/RAD)                                                EULR 660
      SO=SIN(OMEGA/RAD)                                              EULR 670
      GO TO KGO(10,20,30,40)                                         EULR 680
      END                                                            EULR 690
```

```
      SUBROUTINE MULMM(A,B,C)                                        MULM  10
C     MULTIPLY 2 3X3 MATRICES TOGETHER.                              MULM  20
      DIMENSION A(3,3),B(3,3),C(3,3),D(3),E(3)                       MULM  30
      DO 10 I=1,3                                                    MULM  40
      DO 10 J=1,3                                                    MULM  50
      C(I,J)=0.                                                      MULM  60
      DO 10 K=1,3                                                    MULM  70
10    C(I,J)=C(I,J)+A(I,K)*B(K,J)                                    MULM  80
      RETURN                                                         MULM  90
      ENTRY MULMV(A,D,E)                                             MULM 100
      DO 20 I=1,3                                                    MULM 110
      E(I)=0.                                                        MULM 120
      DO 20 J=1,3                                                    MULM 130
20    E(I)=E(I)+A(I,J)*D(J)                                          MULM 140
      RETURN                                                         MULM 150
      END                                                            MULM 160
```

```
      SUBROUTINE TFM2(T,A,B)                                         TFM   10
      INCLUDE IND23,LIST                                             TFM   20
      DIMENSION A(6),B(6),C(10),D(10),T(3,3),BTEM(3,3),GTEM(3,3,3),J(3) TFM  30
      EQUIVALENCE(BTEM(1,1),GTEM(1,1,1)),(JI,J(1)),(J2,J(2)),(J3,J(3)) TFM  40
      DATA PI/3.14159265/                                            TFM   50
      DO 10 I=1,6                                                    TFM   60
      DO 5 K=1,2                                                     TFM   70
5     J(K)=IND2(K,1)                                                 TFM   80
      BTEM(J1,J2)=A(I)                                               TFM   90
      BTEM(J2,JI)=A(I)                                               TFM  100
10    CONTINUE                                                       TFM  110
      DO 20 I=1,6                                                    TFM  120
      B(I)=0.                                                        TFM  130
      DO 15 K=1,2                                                    TFM  140
15    J(K)=IND2(K,I)                                                 TFM  150
      DO 18 K=1,3                                                    TFM  160
      DO 18 L=1,3                                                    TFM  170
18    B(I)=B(I)+T(J1,K)*T(J2,L)*BTEM(K,L)                            TFM  180
20    B(I)=B(I)*2.*PI**2                                             TFM  190
      RETURN                                                         TFM  200
      ENTRY TFM3(T,C,D)                                              TFM  210
      DO 30 I=1,10                                                   TFM  220
      DO 25 K=1,3                                                    TFM  230
25    J(K)=IND3(K,I)                                                 TFM  240
      GTEM(JI,J2,J3)=C(I)                                            TFM  250
      GTEM(J2,J3,J1)=C(I)                                            TFM  260
      GTEM(J3,J1,J2)=C(I)                                            TFM  270
      GTEM(J3,J2,J1)=C(I)                                            TFM  280
      GTEM(J2,J1,J3)=C(I)                                            TFM  290
30    GTEM(J1,J3,J2)=C(I)                                            TFM  300
      DO 40 I=1,10                                                   TFM  310
      D(I)=0.                                                        TFM  320
      DO 35 K=1,3                                                    TFM  330
35    J(K)=IND3(K,I)                                                 TFM  340
      DO 38 K=1,3                                                    TFM  350
      DO 38 L=1,3                                                    TFM  360
      DO 38 M=1,3                                                    TFM  370
38    D(I)=D(I)*T(J1,K)*T(J2,L)*T(J3,M)*GTEM(K,L,M)                  TFM  380
40    D(I)=D(I)*4.*PI**3/3.                                          TFM  390
      RETURN                                                         TFM  400
      END                                                            TFM  410
```

```
      SUBROUTINE TRNFRM                                                 TRNF  10
C     SUBROUTINE CALLED ONCE PER JOB, AFTER CELL CONSTANTS HAVE BEEN    TRNF  20
C     READ IN, TO SET UP TRANSFORMS TO TAKE VECTORS FROM CRYSTAL AXES   TRNF  30
C     TO AN ORTHONORMAL COORDINATE SYSTEM, AND TO TAKE THE SECOND AND   TRNF  40
C     THIRD COMULANTS BACK TO THE CRYSTAL SYSTEM.                       TRNF  50
      INCLUDE ACOM,LIST                                                 TRNF  60
      INCLUDE XPAR,LIST                                                 TRNF  70
      ATOX(1,1)=SQRT(G(1,1))                                            TRNF  80
      ATOX(1,2)=G(1,2)/ATOX(1,1)                                        TRNP  90
      ATOX(1,3)=G(1,3)/ATOX(1,1)                                        TRNF 100
      ATOX(2,1)=0.                                                      TRNF 110
      ATOX(2,2)=SQRT(G(2,2)-ATOX(1,2)**2)                               TRNF 120
      ATOX(2,3)=(G(2,3)-ATOX(1,2)*ATOX(1,3))/ATOX(2,2)                  TRNF 130
      ATOX(3,1)=0.                                                      TRNF 140
      ATOX(3,2)=0.                                                      TRNF 150
      ATOX(3,3)=SQRT(G(3,3)-ATOX(1,3)**2-ATOX(2,3)**2)                  TRNF 160
      DO 10 I=1,3                                                       TRNF 170
10    XTOA(I,I)=1./ATOX(I,I)                                            TRNF 180
      XTOA(1,2)=-ATOX(1,2)/(ATOX(1,1)*ATOX(2,2))                        TRNF 190
      XTOA(2,3)=-ATOX(2,3)/(ATOX(2,2)*ATOX(3,3))                        TRNF 200
      XTOA(1,3)=(ATOX(1,2)*ATOX(2,3)/ATOX(2,2)-ATOX(1,3))/(ATOX(1,1)*ATOTRNF 210
     1X(3,3))                                                           TRNF 220
      XTOA(2,1)=0.                                                      TRNF 230
      XTOA(3,1)=0.                                                      TRNF 240
      XTOA(3,2)=0.                                                      TRNF 250
      RETURN                                                            TRNF 260
      END                                                               TRNF 270
```

```
      SUBROUTINE TRNSPS(A,B)                                            TSPS  10
C     TRANSPOSE MATRIX A AND STORE IN B.                                TSPS  20
      DIMENSION A(3,3),B(3,3)                                           TSPS  30
      DO 100 I=1,3                                                      TSPS  40
      DO 100 J=1,3                                                      TSPS  50
100   B(J,I)=A(I,J)                                                     TSPS  60
      RETURN                                                            TSPS  70
      END                                                               TSPS  80
```

```
      COMPILER (FLD=Q)                                                  IPAK  10
      FUNCTION IPAK(I1,I2,I3,I4)                                        IPAK  20
      FLD(0,9,ITEM)=I1                                                  IPAK  30
      FLD(9,9,ITEM)=I2                                                  IPAK  40
      FLD(18,9,ITEM)=I3                                                 IPAK  50
      FLD(27,9,ITEM)=I4                                                 IPAK  60
      IPAK=ITEM                                                         IPAK  70
      RETURN                                                            IPAK  80
      END                                                               IPAK  90
```

```
      FUNCTION SKTLS(R,I,J,T,EL,S)                                      SKTL  10
      DIMENSION R(3),T(3,3),EL(3,3),S(3,3)                              SKTL  20
      SUM=T(I,J)                                                        SKTL  30
      DO 1000 K=1,3                                                     SKTL  40
      SUM=SUM+AIJ(R,J,K)*S(K,I)*AIJ(R,I,K)*S(K,J)                       SKTL  50
      DO 1000 L=1,3                                                     SKTL  60
      SUM=SUM+AIJ(R,I,K)*AIJ(R,J,L)*EL(K,L)                             SKTL  70
      DO 1000 M=1,3                                                     SKTL  80
      SUM=SUM+3.*EL(L,M)*(S(K,I)*CIJKL(R,J,K,L,M)*S(K,J)*CIJKL(R,I,K,L,MSKTL  90
     1    ))                                                            SKTL 100
      DO 1000 N=1,3                                                     SKTL 110
      SUM=SUM+EL(K,L)*EL(M,N)*(3.*(AIJ(R,I,K)*CIJKL(R,J,L,M,N)*AIJ(R,J,KSKTL 120
     I    )*CIJKL(R,I,L,M,N))*2.*BIJK(R,I,K,M)*BIJK(R,J,L,N))           SKTL 130
1000  CONTINUE                                                          SKTL 140
      SKTLS=SUM                                                         SKTL 150
      RETURN                                                            SKTL 160
      END                                                               SKTL 170
```

```
      FUNCTION D2KDT(I,J,K,L)                                           D2DT  10
      D2KDT=0.                                                          D2DT  20
      IF(I.EQ.K.AND.J.EQ.L)D2KDT=1.                                     D2DT  30
      RETURN                                                            D2DT  40
      END                                                               D2DT  50
```

```
      FUNCTION D2KDL(R,EL,S,I,J,K,L)                                    D2DL  10
      DIMENSION R(3),EL(3,3),S(3,3)                                     D2DL  20
      DEFINE C(I,J,K,L)=CIJKL(R,I,J,K,L)                                D2DL  30
      DEFINE C6(I,K,L,M)=6.*C(I,K,L,M)                                  D2DL  40
      DEFINE ACC(I,J,K,L,M,N)=3.*AIJ(R,I,K)*C(J,L,M,N)                  D2DL  50
      SUM=AIJ(R,I,K)*AIJ(R,J,L)*AIJ(R,I,L)*AIJ(R,J,K)                   D2DL  60
      DO 1000 M=1,3                                                     D2DL  70
      SUM=SUM+S(M,I)*C6(J,K,L,M)*S(M,J)*C6(I,K,L,M)                     D2DL  80
      DO 1000 N=1,3                                                     D2DL  90
      SUM=SUM+EL(M,N)*(ACC(I,J,K,L,M,N)*ACC(I,J,L,K,M,N)*ACC(J,I,K,L,M,ND2DL 100
     1      )*ACC(J,I,L,K,N,N)*ACC(I,J,M,N,K,L)*ACC(J,I,M,N,K,L)*ACC(I,JD2DL 110
     2      ,M,N,L,K)*ACC(J,I,M,N,L,K)*4.*(BIJK(R,I,K,N)*BIJK(R,J,L,N)*  D2DL 120
     3      BIJK(R,I,L,M)*BIJK(R,J,K,N)))                               D2DL 130
1000  CONTINUE                                                          D2DL 140
      D2KDL=SUM                                                         D2DL 150
      IF(K.EQ.L)D2KDL=SUM/2.                                            D2DL 160
      RETURN                                                            D2DL 170
      END                                                               D2DL 180
```

```
      FUNCTION D2KDS(R,EL,I,J,K,L)                                      D2DS  10
      DIMENSION R(3),EL(3,3)                                            D2DS  20
      DEFINE CCC(J,K,L,M)=3.*CIJKL(R,J,K,L,M)                           D2DS  30
      SUM=0.                                                            D20S  40
      IF(I.NE.L)GO TO 1000                                              D2DS  50
      SUM=SUM+AIJ(R,J,K)                                                D2DS  60
      DO 500 M=1,3                                                      D2DS  70
      DO 500 N=1,3                                                      D2DS  80
  500 SUM=SUM+EL(M,N)*CCC(J,K,M,N)                                      02DS  90
 1000 IF(J.NE.L)GO TO 2000                                              D2DS 100
      SUM=SUM+AIJ(R,I,K)                                                D2DS 110
      DO 1500 M=1,3                                                     D2DS 120
      DO 1500 N=1,3                                                     D2DS 130
 1500 SUM=SUM+EL(M,N)*CCC(I,K,M,N)                                      D2DS 140
 2000 D2KDS=SUM                                                         D2DS 150
      RETURN                                                            D2DS 160
      END                                                               D2DS 170
```

```
      FUNCTION TKTLS(R,I,J,K,EL,S)                                      TKTL  10
      DIMENSION R(3),EL(3,3),S(3,3)                                     TKTL  20
      INTEGER P                                                         TKTL  30
      DEFINE B(K,L,M)=BIJK(R,K,L,M)                                     TKTL  40
      DEFINE A(I,J)=AIJ(R,I,J)                                          TKTL  50
      SUM=0.                                                            TKTL  60
      DO 1000 L=1,3                                                     TKTL  70
      DO 1000 M=1,3                                                     TKTL  80
      SUM=SUM+2.*(S(L,I)*S(M,J)*B(K,L,M)*S(L,I)*S(M,K)*B(J,L,M)*S(L,J)* TKTL  90
     1     S(M,K)*B(I,L,M))                                             TKTL 100
      DO 1000 N=1,3                                                     TKTL 110
      SUM=SUM+2.*EL(M,N)*(S(L,I)*(A(J,M)*B(K,L,N)+A(K,M)*B(J,L,N))*S(L,JTKTL 120
     1     )*(A(I,M)*B(K,L,N)+A(K,M)*B(I,L,N))+S(L,K)*(A(I,M)*B(J,L,N)*  TKTL 130
     2     A(J,M)*B(I,L,N)))                                            TKTL 140
      DO 1000 P=1,3                                                     TKTL 150
      SUM=SUM+EL(N,P)*(S(L,J)*S(M,K)*PIJKLM(R,I,L,M,N,P)*S(L,I)*S(M,K)* TKTL 160
     1     PIJKLM(R,J,L,M,N,P)+S(L,I)*S(M,J)*PIJKLM(R,K,L,M,N,P)*2.*EL(L,TKTL 170
     2     M)*(A(I,L)*A(J,N)*B(K,M,P)+A(I,L)*A(K,N)*B(J,M,P)+A(J,L)*A(K,NTKTL 180
     3     )*B(I,M,P)))                                                 TKTL 190
 1000 CONTINUE                                                          TKTL 200
      TKTLS=SUM                                                         TKTL 210
      RETURN                                                            TKTL 220
      END                                                               TKTL 230
```

```
      FUNCTION D3KDL(R,EL,S,I,J,K,L,M)                                  D3DL  10
      DIMENSION R(3),EL(3,3),S(3,3)                                     D3DL  20
      DEFINE ABB(J,K,L,M,N)=2.*AIJ(R,J,L)*BIJK(R,K,M,N)                 D3DL  30
      DEFINE AAB(I,J,K,L,M,N,NN)=AIJ(R,I,L)*AIJ(R,J,N)*BIJK(R,K,M,NN)   D3DL  40
      INTEGER P                                                         D3DL  50
      DEFINE PP(I,J,K,L,M)=PIJKLM(R,I,J,K,L,M)*PIJKLM(R,I,J,K,M,L)      D3DL  60
      SUM=0.                                                            D3DL  70
      DO 1000 N=1,3                                                     03DL  80
      SUM=SUM+S(N,I)*(ABB(J,K,L,M,N)*ABB(K,J,L,M,N)*ABB(J,K,M,L,N)*ABB(KD3DL  90
     1     ,J,M,L,N))*S(N,J)*(ABB(I,K,L,M,N)*ABB(K,I,L,M,N)*ABB(I,K,M,L,ND3DL 100
     2     )*ABB(K,I,M,L,N))*S(N,K)*(ABB(I,J,L,M,N)*ABB(J,I,L,M,N)*ABB(I,D3DL 110
     3     J,M,L,N)*ABB(J,I,M,L,N))                                     D3DL 120
      DO 1000 P=1,3                                                     D30L 130
      SUM=SUM+S(N,J)*S(P,K)*PP(I,N,P,L,M)*S(N,I)*S(P,K)*PP(J,N,P,L,M)   D3DL 140
     1     *S(N,I)*S(P,J)*PP(K,N,P,L,M)*2.*EL(N,P)*(AAB(I,J,K,L,M,N,P)  D3DL 150
     2     *AAB(I,K,J,L,M,N,P)*AAB(J,K,I,L,M,N,P)*AAB(I,J,K,N,P,L,M)     D3DL 160
     3     *AAB(I,K,J,N,P,L,M)*AAB(J,K,I,N,P,L,M))                      D3DL 170
 1000 CONTINUE                                                          D3DL 180
      D3KDL=SUM                                                         D3DL 190
      IP(L.EQ.M)D3KDL=SUM/2.                                            D3DL 200
      RETURN                                                            D3DL 210
      END                                                               D3DL 220
```

```
      FUNCTION D3KDS(R,EL,S,I,J,K,LP,MP)                                D3DS  10
      DIMENSION R(3),EL(3,3),S(3,3)                                     D3DS  20
      INTEGER P                                                         D3DS  30
      DEFINE BB(K,L,M)=2.*BIJK(R,K,L,M)                                 D3DS  40
      DEFINE ABB(J,K,L,M,N)=2.*AIJ(R,J,L)*BIJK(R,K,M,N)                 03DS  50
      SUM=0.                                                            D3DS  60
      DO 1000 L=1,3                                                     D3DS  70
      DO 1000 M=1,3                                                     D3DS  80
      IF(L.EQ.LP.AND.I.EQ.MP)SUM=SUM+S(M,J)*BB(K,L,M)*S(M,K)*BB(J,L,M)  D3DS  90
      IF(M.EQ.LP.AND.J.EQ.MP)SUM=SUM+S(L,I)*BB(K,L,M)                   03DS 100
      IF(M.EQ.LP.AND.K.EQ.MP)SUM=SUM+S(L,I)*BB(J,L,M)*S(L,J)*BB(I,L,M)  D3DS 110
      IF(L.EQ.LP.AND.J.EQ.MP)SUM=SUM+S(M,K)*BB(I,L,M)                   D3DS I20
      DO 1000 N=1,3                                                     D30S 130
      IP(L.NE.LP)GO TO 500                                              D3DS 140
      IP(I.EQ.MP)SUM=SUM+EL(M,N)*(ABB(J,K,M,L,N)+ABB(K,J,M,L,N))        D3DS 150
      IP(J.EQ.MP)SUM=SUM+EL(M,N)*(ABB(I,K,M,L,N)+ABB(K,I,M,L,N))        D3DS 160
      IF(K.EQ.MP)SUM=SUM+EL(M,N)*(ABB(I,J,M,L,N)+ABB(J,I,M,L,N))        D3DS 170
      GO TO 1000 P=1,3                                                  D3DS 180
  500 IP(L.NE.LP)GO TO 750                                              03DS 190
      IP(J.EQ.MP)SUM=SUM+EL(N,P)*S(M,K)*PIJKLM(R,I,L,M,N,P)             D3DS 200
      IF(I.EQ.MP)SUM=SUM+EL(N,P)*(S(M,K)*PIJKLM(R,J,L,M,N,P)*S(M,J)*    D3DS 210
     1                 PIJKLM(R,K,L,M,N,P))                           D3DS 220
      GO TO 1000                                                        03DS 230
  750 IP(M.NE.LP)GO TO 1000                                             03DS 240
      IF(K.EQ.MP)SUM=SUM+EL(N,P)*(S(L,J)*PIJKLM(R,I,L,M,N,P)*S(L,I)*    D3DS 250
     I                 PIJKLM(R,J,L,M,N,P))                           D3DS 260
      IF(J.EQ.MP)SUM=SUM+EL(N,P)*S(L,I)*PIJKLM(R,K,L,M,N,P)             D3DS 270
 1000 CONTINUE                                                          D3DS 280
      D3KDS=SUM                                                         D3DS 290
      RETURN                                                            D3DS 300
      END
```

```
      FUNCTION DIJKLM(R,I,JI,KI,LI,MI)                                  DIJK  10
      DIMENSION R(3)                                                    DIJK  20
      IN=JI*KI*LI*MI                                                    DIJK  30
      IF(MOD(IN,6).EQ.0.AND.IN.LE.18)GO TO 1080                         DIJK  40
      IF(IN.EQ.4.OR.IN.EQ.9.OR.IN.EQ.36)GO TO 1060                      DIJK  50
      IF(JI.NE.KI.OR.KI.NE.LI.OR.LI.NE.MI)GO TO 1040                    DIJK  60
      DIV=1.                                                            DIJK  70
      GO TO 1100                                                        DIJK  80
 1040 DIV=4.                                                            DIJK  90
      GO TO 1100                                                        DIJK 100
 1060 DIV=6.                                                            DIJK 110
      GO TO 1100                                                        DIJK 120
 1080 DIV=12.                                                           DIJK 130
 1100 IF(MOD(IN,3).NE.0)GO TO 1500                                      DIJK 140
      M=3                                                               DIJK 150
      IN=IN/3                                                           DIJK 160
      IF(MOD(IN,3).NE.0)GO TO 1250                                      DIJK 170
      L=3                                                               DIJK 180
      IN=IN/3                                                           DIJK 190
      IF(MOD(IN,3).NE.0)GO TO 1125                                      DIJK 200
      K=3                                                               DIJK 210
      J=IN/3                                                            DIJK 220
      GO TO 1900                                                        DIJK 230
 1125 IF(MOD(IN,2).NE.0)GO TO 1150                                      DIJK 240
      K=2                                                               DIJK 250
      J=IN/2                                                            DIJK 260
      GO TO 1900                                                        DIJK 270
 1150 K=1                                                               DIJK 280
      J=1                                                               DIJK 290
      GO TO 1900                                                        DIJK 300
 1250 IF(MOD(IN,2).NE.0)GO TO 1375                                      DIJK 310
      L=2                                                               DIJK 320
      IN=IN/2                                                           DIJK 330
      IF(MOD(IN,2).NE.0)GO TO 1150                                      DIJK 340
      K=2                                                               DIJK 350
      J=IN/2                                                            DIJK 360
      GO TO 1900                                                        DIJK 370
 1375 L=1                                                               DIJK 380
      GO TO 1150                                                        DIJK 390
 1500 IF(MOD(IN,2).NE.0)GO TO 1750                                      DIJK 400
      M=2                                                               DIJK 410
      IN=IN/2                                                           DIJK 420
      GO TO 1250                                                        DIJK 430
 1750 M=1                                                               DIJK 440
      GO TO 1375                                                        DIJK 450
 1900 GO TO (100,200,300),I                                             DIJK 460
  100 GO TO (110,150,2000),J                                            DIJK 470
  110 GO TO (120,140,3000),K                                            DIJK 480
  120 GO TO (125,130,2000),L                                            DIJK 490
  125 GO TO (1000,4000,3000),M                                          DIJK 500
  130 IF(M-2)1000,2000,1000                                             DIJK 510
  140 IF(L-2)1000,145,4000                                              DIJK 520
  145 IF(M-2)1000,4000,3000                                             DIJK 530
  150 IF(K-2)1000,160,1000                                              DIJK 540
  160 IF(L-2)1000,170,5000                                              DIJK 550
  170 IF(M-2)1000,2000,1000                                             DIJK 560
  200 GO TO (210,250,2000),J                                            DIJK 570
  210 GO TO (220,240,1000),K                                            DIJK 580
  220 GO TO (225,230,5000),L                                            DIJK 590
  225 GO TO (2000,3000,1000),M                                          DIJK 600
  230 IF(M-2)1000,2000,4000                                             DIJK 610
  240 IF(L-2)1000,245,3000                                              DIJK 620
  245 IF(M-2)1000,3000,1000                                             DIJK 630
  250 IF(K-2)1000,260,4000                                              DIJK 640
  260 IF(L-2)1000,270,2000                                              DIJK 650
  270 IF(M-2)1000,1000,4000                                             DIJK 660
  300 GO TO (310,350,1000),J                                            DIJK 670
  310 GO TO (320,340,4000),K                                            DIJK 680
  320 GO TO (325,330,2000),L                                            DIJK 690
  325 GO TO (2000,1000,4000),M                                          DIJK 700
  330 IF(M-2)1000,5000,3000                                             DIJK 710
  340 IF(L-2)1000,345,1000                                              DIJK 720
  345 IF(M-2)1000,1000,4000                                             DIJK 730
  350 IF(K-2)1000,360,3000                                              DIJK 740
  360 IF(L-2)1000,370,2000                                              DIJK 750
  370 IF(M-2)1000,2000,3000                                             DIJK 760
 1000 DIJKLM=0.                                                         DIJK 770
      RETURN                                                            DIJK 780
 2000 FAC=1.                                                            DIJK 790
 2010 DIJKLM=FAC*R(I)/(24.*DIV)                                         DIJK 800
      RETURN                                                            DIJK 810
 3000 MM=MOD(I+1,3)+1                                                   DIJK 820
 3010 DIJKLM=-R(MM)/(24.*DIV)                                           DIJK 830
      RETURN                                                            DIJK 840
 4000 MM=MOD(I,3)+1                                                     DIJK 850
      GO TO 3010                                                        DIJK 860
 5000 FAC=2.                                                            DIJK 870
      GO TO 2010                                                        DIJK 880
      END                                                               DIJK 890
```

```
      FUNCTION FIJKLM(R,I,J,K,L,M)                                      FIJK  10
      DIMENSION R(3)                                                    FIJK  20
      DEFINE D(J,K,L,M)=DIJKLM(R,I,J,K,L,M)                             FIJK  30
      FIJKLM=12.*D(J,K,L,M)                                             FIJK  40
      RETURN                                                            FIJK  50
      END                                                               FIJK  60
```

```
      SUBROUTINE BOOAN1(MBOOAN)                                          BOOAN  10
                                                                         BOOAN  20
C        THIS ROUTINE REVISED  JULY 28, 1970                             BOOAN  30
                                                                         BODN   40
C              BOND DISTANCE AND ANGLE SUBPROGRAM FOR U M REFINE         BODN   50
C              OUTPUT DELETED IF ARGUMENT NEGATIVE                       BOON   60
                                                                         BOON   70
      INCLUDE ACOM,LIST                                                  BODN   80
      COMMON/C/CELLP(3,4),ICENT,JCELL,NSYM,RMAT(3,3,24),TRANS(3,24)      BODN   90
      COMMON/O/8(9),OXX(3),OXY(3),OXZ(3),  P(6),XP(3),XXP(3),XYP(3),YP(3BOON  100
     1),ATEMP(65),NUMB(300),XPR(3,300),LIST,SN,SX,NXX(6),ZTEMP(390)      BODN   110
      COMMON/E/DMAX,DMIN,RECELL(6)                                       BOON   120
      INCLUDE TAPE,LIST                                                  BODN   130
                                                                         BOON   140
      EQUIVALENCE (NXX(1),NXL),(NXX(2),NYL),(NXX(3),NZL),(NXX(4),NXU),(NBOON  150
     1 XX(5),NYU),(NXX(6),NZO)                                           HODN   160
                                                                         BOON   170
     2 FORMAT(10X,2A3,3F10.5,54X,I5)                                     BOON   180
     4 FORMAT(1H1,18A4/15X,23HBOND DISTANCES BETWEEN F5.2,4H ANO,F5.2,5H BODN   190
      1ANG.//10X,6HATOM 1,8X,1NX,9X,1NY,9X,1NZ,3X,6HATOM 2,8X,1NX,9X,1NY,BODN   200
     2 9X,1HZ,4X,8HDISTANCE,4X,6HNUMBER/)                                BOON   210
     6 FORMAT(48X,2A3,3F10.5,F11.5,I10)                                  HODN   220
     8 FORMAT(20HOATOM ARRAY OVERFLOW)                                   HOON   230
    10 FORMAT(1H1,10X,18A4//10X,11HBOND ANGLES)                          HODN   240
    12 FORMAT(1HO,5X,12HCENTRAL ATOM,5X,1NX,9X,1NY,9X,1NZ//9X,2A3,3F10.5/BOOAN  250
      1/15X,6HATOM 1,2X,6HNUMBER,10X,6HATOM 2,2X,6HNUMBER,10X,5HANGLE/)  BOOAN  260
     14 FORMAT(15X,2A3,I6,12X,2A3,I6,F17.3)                              BOON   270
                                                                         BODN   280
      NUM=1                                                              BODN   290
      IF(ICENT)102,100,102                                               BOON   300
  100 NUM=2                                                              BOON   310
  102 IF(MBODAN)112,112,110                                              BODN   320
  110 WRITE(IOUT,4)TITLE,OMIN,OMAX                                       BOON   330
  112 SX=OMAX**2                                                         BODN   340
      SN=OMIN**2                                                         BOON   350
      LIST=0                                                             BOON   360
                              DO 120 I=1,3                               BOON   370
  120 DXX(I)=OMAX*RECELL(I)                                              BOON   380
C             FIRST LOOP THROUGH ATOM LIST                               BOON   390
                              DO 420 II=1,NATOM                          BOON   400
                              DO 130 J=1,3                               BOON   410
      OXY(J)=XYZ(J,II)-OXX(J)                                            BOON   420
  130 OXZ(J)=XYZ(J,II)-OXX(J)                                            BOON   430
      IP(II-1)135,170,135                                                BODN   440
C             CHECK IF ATOM II IN GENERATED LIST                         BODN   450
  135                         DO 160 I=1,LIST                          BODN   460
      IP(NOMB(I)/64-II)160,140,160                                       BOON   470
  140                         DO 150 J=1,3                             BODN   480
      IF(A85(XPR(J,I)-XYZ(J,II))-0.00001)150,160,160                     BOUN   490
  150 CONTINUE                                                           BOON   500
      GO TO 190                                                          BODN   510
  160 CONTINUE                                                           BOON   520
C             ADD ATOM II TO LIST                                        BOUN   530
  170 LIST=LIST+1                                                        ROON   540
      I=LIST                                                             BOUN   550
      NUMB(LIST)=64*II+1                                                 BODN   560
                              DO 180 J=1,3                               BOUN   570
  180 XPR(J,LIST)=XYZ(J,II)                                              BODN   580
  190 IF(MBODAN)194,192,192                                              BODN   590
  192 WRITE(IOUT,2)(TAG(J,II),J=1,2),(XYZ(J,II),J=1,3),I                 BODN   600
C             LOOP THROUGH GENERATED LIST                                BOON   610
  194                         DO 240 J=1,LIST                          BOON   620
      S=0.0                                                              BODN   630
                              DO 200 K=1,3                               BODN   640
  200 XP(K)=XYZ(K,II)-XPR(K,J)                                           BODN   650
                              DO 220 N=1,3                               BODN   660
      XZP=0.0                                                            BODN   670
                              DO 210 K=1,3                               BODN   680
  210 XZP=XZP+G(K,N)*XP(K)                                               BODN   690
  220 S=S+XZP*XP(N)                                                      BODN   700
      IP((S-SX)*(SN-S))240,240,230                                       BODN   710
  230 S=SQRT(S)                                                          BODN   720
      IJ=NUMB(J)/64                                                      BODN   730
      IF(MBODAN)240,238,238                                              BODN   740
  238 WRITE(IOUT,6)(TAG(K,IJ),K=1,2),(XPR(K,J),K=1,3),S,J                BODN   750
  240 CONTINUE                                                           BODN   760
C             START SECOND LOOP THROUGH ATOM LIST                        BODN   770
                              DO 420 JJ=1,NATOM                          BODN   780
C             START LOOP THROUGH SYMMETRY OPERATIONS                     BODN   790
                              DO 420 J=1,NSYM                            BODN   800
                              DO 250 L=1,3                               BODN   810
      XP(L)=TRANS(L,J)                                                   BODN   820
                              DO 250 M=1,3                               BOON   830
  250 XP(L)=XP(L)+RMAT(L,M,J)*XYZ(M,JJ)                                  BODN   840
C             LOOP FOR CENTER OF SYMMETRY                                BODN   850
                              DO 420 ITY=1,NUM                           BOON   860
      GO TO (280,260),ITY                                                BODN   870
  260                         DO 270 L=1,3                             BOON   880
  270 XP(L)=-XP(L)                                                       BOON   890
C             LOOP THROUGH LATTICE POINTS                                BOON   900
  280                         DO 420 KK=1,JCELL                        BOON   910
                              DO 290 L=1,3                               BODN   920
  290 XXP(L)=XP(L)+CELLP(L,KK)                                           BOON   930
C             SET TRANSLATION LIMITS                                     BODN   940
                              DO 300 I=1,3                               BOON   950
      NXX(I)=5-IFIX(XXP(I)-DXZ(I))                                       HODN   960
      NXX(I+3)=5.0+DXY(I)-XXP(I)                                         BODN   970
      IP(NXX(I)-NXX(I+3))300,300,420                                     BODN   980
  300 CONTINUE                                                           BODN   990
      NXU=NXX(4)                                                         BODN1000
      NYU=NXX(5)                                                         BODN1010
      NZU=NXX(6)                                                         BODN1020
C             STEP THROUGH UNIT CELLS                                    BODN1030
                              DO 410 IA=NXL,NXU                          BODN1040
      XYP(1)=XXP(1)+FLOAT(IA-5)                                          BODN1050
                              DO 410 IB=NYL,NYU                          BODN1060
      XYP(2)=XYP(2)+FLOAT(IB-5)                                          BODN1070
                              DO 410 IC=NZL,NZU                          BODN1080
      XYP(3)=XXP(3)+FLOAT(IC-5)                                          BODN1090
C             CHECK FOR ATOM DUPLICATION                                 BODN1100
                              DO 330 I=1,LIST                          BODN1110
      IP(NUMB(I)/64-JJ)330,310,330                                       BODN1120
  310                         DO 320 L=1,3                             BODN1130
      IF(A85(XYP(L)-XPR(L,I))-0.00001)320,330,330                        BODN1140
  320 CONTINUE                                                           BODN1150
```

```
      GO TO  410                                                        BODN1160
  330 CONTINUE                                                          BODN1170
C               NEW ATOM, COMPOTE DISTANCE                            BODN1180
      S=0.0                                                             BODN1190
                                DO 340 K=1,3                            BODN1200
  340 YP(K)=XYZ(K,II)-XYP(K)                                            BODN1210
                                DO 360 N=1,3                            BODN1220
      XZP=0.0                                                           BODN1230
                                DO 350 K=1,3                            BODN1240
  350 XZP=XZP+YP(K)*G(K,N)                                              BODN1250
  360 B=S+XZP*YP(N)                                                     BODN1260
      IF((S-SX)*(SN-S))410,410,370                                      BODN1270
  370 S=SQRT(B)                                                         BODN1280
C               CHECK IF LIST ARRAYS FULL                             BODN1290
      IF(LIST-3DO)390,3ED,3BO                                           BODN1300
  3BO WRITE(IOUT,E)                                                     BODN1310
      GO TO 410                                                         BODN1320
C               ADD ATOM TO LIST                                      BODN1330
  390 LIST=LIST+1                                                       BODN1340
      NUMB(LIST)=64*JJ+J                                                BODN1350
                                DO 400 K=1,3                            BODN1360
  4DO XPR(K,LIST)=XYP(K)                                                BODN1370
      IF(MBODAN)410,402,402                                             BODN13BO
  402 WRITE(IOUT,6)(TAG(K,JJ),K=1,2),XYP,S,LIST                        BODN1390
  41D CONTINUE                                                          BODN1400
  420 CONTINUE                                                          BODN1410
  430 IF(MBODAN-1)440,440,450                                           BODN1420
  440 RETURN                                                            BODN1430
C               COMPUTE BOND ANGLES                                   BODN1440
  450 WRITE(IOOT,10)TITLE                                               BODN1450
      IJ=LIST-1                                                         BODN1460
C               LOOP THROUGH ATOMS                                    BODN1470
                                DO 5E0 I=1,NATOM                        BODN14B0
      WRITE(IOOT,12)(TAG(K,I),K=1,2),(XYZ(J,I),J=1,3)                   BODN1490
C               FIRST LOOP THROUGH LIST                               BODN1500
                                DO 5B0 J=1,IJ                           BODN1510
                                DO 46D K=1,3                            BODN1520
  46D XP(K)=XPR(K,J)-XYZ(K,I)                                           BODN1530
      B=0.0                                                             BODN1540
                                DO 4E0 N=1,3                            BODN1550
      XXP(N)=0.0                                                        BODN1560
                                DO 470 K=1,3                            BODN1570
  47D XXP(N)=XXP(N)+G(K,N)*XP(K)                                        BODN1580
  48D B=S*XXP(N)*XP(N)                                                  BODN1590
      IF((E-SX)*(SN-S))5ED,5E0,490                                      BODN1600
  490 S=1.D/SQRT(S)                                                     BODN1610
      NAT=NOMB(J)/64                                                    BODN1620
      IK=J+1                                                            BODN1630
C               SECOND LOOP THROUGH LIST                              BODN1640
                                DO 570 K=IK,LIST                        BODN1650
      S2=0.0                                                            BODN1660
                                DO 500 N=1,3                            BODN1670
  50D TP(N)=XPR(N,K)-XYZ(N,I)                                           BODN16B0
                                DO 520 N=1,3                            BODN1690
      XZP=0.0                                                           BODN1700
                                DO 510 M=1,3                            BODN1710
  510 XZP=XZP+YP(M)*G(M,N)                                              BODN1720
  520 S2=S2+XZP*TP(N)                                                   BODN1730
      IF((S2-SX)*(SN-S2))570,570,530                                    BODN1740
  530 S2=S/SQRT(S2)                                                     BODN1750
      NTA=NUMB(K)/64                                                    BODN1760
      CTH=0.0                                                           BODN1770
                                DO 540 M=1,3                            BODN17B0
  540 CTH=CTH+XXP(M)*YP(M)                                              BODN1790
      CTH=CTH*S2                                                        BODN1B00
      TEMP=ABS(CTH)                                                     BODN1B10
      IF(TEMP-1.0)560,560,550                                           BODN1B20
  550 CTH=CTH/TEMP                                                      BODN1B30
  560 CTH=180.0*ACOS(CTH)/PI                                            BODN1B40
      WRITE(IOUT,14)(TAG(M,NAT),M=1,2),J,(TAG(M,NTA),M=1,2),K,CTH       BODN1B50
  570 CONTINUE                                                          BODN1B60
  5B0 CONTINUE                                                          BODN1B70
      RETURN                                                            BODN1B80
      END                                                               BODN1B90
```

```
      SUBROUTINE ELVIB1(MELVIB,MHOOAN)                                  ELVB  10
                                                                        ELVB  20
C         THIS ROUTINE REVISED  JULY 28, 1970                           ELVB  30
                                                                        ELVB  40
C               ELLIPSOIDS OF VIBRATION CALCULATION                     ELVB  50
                                                                        ELVB  60
      INCLUDE ACOM,LIST                                                 ELVB  70
      COMMON/C/CELLP(3,4),ICENT,JCELL,NSYM,RMAT(3,3,24),TTRANS(3,24)    ELVB  80
      COMMON/D/B(3,3),O(3,3),F(6)  ,X(3,3),XP(3),ATEMP(65),HUMB(300),   ELVB  90
     1XPR(3,300),LIST,SH,SX,COEP(3),EIG(3),ROOT(3),YP(3),Y(3),ZTEMP(381)ELVB 100
      COMMON/E/OMAX,OMIN,RECELL(6)                                      ELVB 110
      INCLUDE TAPE,LIST                                                 ELVB 120
                                                                        ELVB 130
    2 FORMAT(1H1,18A4//10X,23HELLIPSOIDS OF VIBRATION)                  ELVB 140
    4 FORMAT(///10X,14HELLIPSOID FOR 2A3//15X,24HEQUIVALENT ISOTROPIC B ELVB 150
     1= F8.4//20X,4HAXIS,10X,13HRMS AMPLITUDE,10X,12HANGLE WITH A,5X,12HELVB 160
     2ANGLE WITH B,5X,12HANGLE WITH C/)                                 ELVB 170
    6 FORMAT(///10X,2A3,25H IS NON-POSITIVE DEFINITE)                   ELVB 180
    8 FORMAT(20X,I3,12X,F11.6,3X,3F17.2)                                ELVB 190
   10 FORMAT(/12X,56HRMS DISPLACEMENTS AND THERMAL CORRECTIONS TO BONDS  ELVB 200
     1PROM ,2A3,4H AT ,3P10.5      /65X,9HDISTANCES,20X,12NDISPLACEMENT/ ELVB 210
     2 15X,4HATOM,7X,1HX,8X,1HY,8X,1HZ,6X,3HRAW,4X,5HLSWER,5X,6HRIDING,  ELVB 220
     3 2X,8HBOND-CORR,2X,5HUPPER,3X,10NALONG BOND/)                     ELVB 230
   12 FORMAT(14X,2A3,9P9.5)                                             ELVB 240
                                                                        ELVB 250
      WRITE(16UT,2)TITLE                                                ELVB 260
      TFC=0.5/PI**2                                                     ELVB 270
                              DO 90 I=1,3                               ELVB 280
88    F(I)=G(I,1)                                                       ELVB 290
   90 Y(I)=SQRT(G(I,I))*RECELL(I)                                       ELVB 300
      F(4)=2.0*G(1,2)                                                   ELVB 310
      F(5)=2.0*G(1,3)                                                   ELVB 320
      F(6)=2.0*G(2,3)                                                   ELVB 330
C             START ATOM LOOP                                           ELVB 340
                              DO 420 I=1,NATOM                          ELVB 350
      IF(ISOT(I))100,265,100                                            ELVB 360
C             IGNORE ISOTROPIC ATOMS                                    ELVB 370
  100                         DO 110 K=1,2                            ELVB 380
      B(K,K)=BETA(K,I)                                                  ELVB 390
      IJ=K+1                                                            ELVB 400
                              DO 110 L=IJ,3                             ELVB 410
      IK=IJ*L                                                           ELVB 420
      B(K,L)=BETA(IK,I)                                                 ELVB 430
  110 B(L,K)=B(K,L)                                                     ELVB 440
      B(3,3)=BETA(3,I)                                                  ELVB 450
C             B NOW CONTAINS MATRIX OF THERMAL COEF. OF ATOM I          ELVB 460
C             NOW COMPUTE D = G B                                       ELVB 470
                              DO 120 K=1,3                              ELVB 480
                              DO 120 L=1,3                              ELVB 490
      D(K,L)=0.0                                                        ELVB 500
                              DO 120 M=1,3                              ELVB 510
  120 O(K,L)=O(K,L)+G(K,M)*B(M,L)                                       ELVB 520
C             SOLVE FOR EIGENVALUES OF O                                ELVB 530
      COEP(1)=D(1,1)*D(2,2)*O(3,3)                                      ELVB 540
      COEP(2)=D(1,1)*(O(2,2)*O(3,3))*D(2,2)*D(3,3)-D(1,2)*O(2,1)-O(1,3)*ELVB 550
     1 O(3,1)-O(2,3)*D(3,2)                                             ELVB S60
      COEP(3)=O(1,1)*(O(2,2)*D(3,3)-O(2,3)*O(3,2))-D(1,2)*(O(2,1)*D(3,3)ELVB 570
     1 -O(3,1)*O(2,3))*O(1,3)*(D(2,1)*D(3,2)-D(3,1)*D(2,2))             ELVB 5B0
C             EIGENVALUE EQUATION IS                                    ELVB 590
C             0 = -X**3 + COEP(1)*X**2-COEP(2)*X + COEP(3)              ELVB 600
      BISO=4.0/3.0*COEF(1)                                              ELVB 610
      TEMP=COEF(2)-COEP(1)**2/3.0                                       ELVB 620
      TEMP1=(-2.0*COEP(1)**3*9.0*COEF(1)*COEP(2)-27.0*COEP(3))/27.0     ELVB 630
      TEMP2=-TEMP**3/27.0                                               ELVB 640
      IF(D.25*TEMP1**2-TEMP2-1.0B-6)140,140,130                         ELVB 650
C             ATOM NON-POSITIVE DEFINITE                                ELVB 660
  130 WRITE(16UT,6)(TAG(K,I),K=1,2)                                     ELVB 670
      GO TO 420                                                         ELVB 680
  140 TEMP2=-0.5*TEMP1/SQRT(TEMP2)                                      ELVB 690
      TEMP3=ABS(TEMP2)                                                  ELVB 70D
      IF(TBMP3-1.0)160,160,150                                          ELVB 710
  150 TEMP2=TEMP2/TEMP3                                                 ELVB 720
  160 PNI=ACOS(TEMP2)/3.0                                               ELVB 730
      TEMP=2.0*SQRT(-TEMP/3.0)                                          ELVB 740
      COEF(2)=TEMP*COS(PNI+2.0*PI/3.0)*COEP(1)/3.0                      ELVB 750
      COEP(3)=TEMP*COS(PNI-2.0*PI/3.0)*COEP(1)/3.0                      ELVB 760
      COEF(1)=TEMP*COS(PNI)*COEP(1)/3.0                                 ELVB 770
C             ARRANGE ROOTS IN ORDER OF INCREASING SIZE                 ELVB 7B0
                              DO 180 N=1,2                              ELVB 790
      IK=N+1                                                            ELVB 800
                              DO 180 L=IK,3                             ELVB 810
      IF(COEF(N)-COEF(L))180,180,170                                    ELVB 820
  170 TEMP=COEF(N)                                                      ELVB 830
      COEF(N)=COEF(L)                                                   ELVB 840
      COEP(L)=TEMP                                                      ELVB 850
  180 CONTINUE                                                          ELVB 860
C             CALCULATE RMS DISPLACEMENTS                               ELVB 870
                              DO 190 K=1,3                              ELVB 880
      IF(COEF(K))130,190,190                                            ELVB 890
  190 ROOT(K)=SQRT(COEF(K)*TFC)                                         ELVB 900
      WRITE(16UT,4)(TAG(K,I),K=1,2),BISO                                ELVB 910
C             START FINDING EIGENVECTORS                                ELVB 920
                              DO 200 K=1,3                              ELVB 930
                              DO 200 L=1,3                              ELVB 940
  200 X(K,L)=D(K,L)                                                     ELVB 950
                              DO 260 N=1,3                              ELVB 960
                              DO 210 L=1,3                              ELVB 970
  210 X(L,L)=D(L,L)-COEP(N)                                             ELVB 98D
C             X IS NOW Q B - LAMBDA I                                   ELVB 990
      TEMP=1.0E-6                                                       ELVB1000
                              DO 220 ITMP=1,12                          ELVB1010
      TEMP=TEMP*0.01                                                    ELVB1020
                              DO 220 X=1,3                              ELVB1030
      M=K+1-3*(K/3)                                                     ELVB1040
      EIG(1)=RECELL(1)*(X(K,2)*X(M,3)-X(K,3)*X(M,2))                    ELVB1050
      EIG(2)=RECELL(2)*(X(K,3)*X(M,1)-X(K,1)*X(M,3))                    ELVB1060
      EIG(3)=RECELL(3)*(X(K,1)*X(M,2)-X(K,2)*X(M,1))                    ELVB1070
      PNC=EIG(1)**2+EIG(2)**2+EIG(3)**2                                 ELVB1080
C             CHECK FOR NULL VECTOR                                     ELVB1090
      IF(PNC-TEMP)220,220,230                                           ELVB1100
  220 CONTINUE                                                          ELVB1110
  230 FNC=SQRT(PNC*2.0*(EIG(1)*EIG(2)*RECELL(6)*EIG(1)*EIG(3)*RECELL(5)*ELVB1120
     1 EIG(2)*EIG(3)*RECELL(4)))                                        ELVB1130
                              DO 250 K=1,3                              ELVB1140
      TEMP=BIO(K)/(FNC*Y(K))                                            ELVB1150
```

```
      TEMP1=ABS(TEMP)                                                   ELVBI160
      IF(TEMP1-1.0)250,250,240                                          ELVBI170
  240 TEMP=TEMP/TEMP1                                                   ELVBI180
  250 EIG(K)=180.0*ACOS(TEMP)/PI                                        ELVBI190
C             OUTPUT RESULTS                                          ELVBI200
  260 WRITE(IOUT,8)N,ROOT(N),EIG                                        ELVBI210
  265 IF(MELVIB-1)270,420,270                                           ELVBI220
C             COMPUTE RMS DISP. ALONG BONDS AND THERMAL CORRECTION TO BOELVBI230
  270 IF(MBOOAN)290,280,290                                             ELVBI240
C             BOND DISTANCES NOT AVAILABLE, COMPUTE AND SUPPRESS OUTPUT ELVBI250
  280 CALL BODAN1(-1)                                                   ELVBI260
      MBOOAN=1                                                          ELVBI270
  290 WRITE(IOUT,10)(TAG(J,I),J=1,2),(XYZ(J,I),J=1,3)                   ELVBI280
C             TRANS IS MEAN SQUARE DISPLACEMENT OF ATOM 1               ELVBI290
      TRANS=0.75*TFC*BISM                                               ELVBI300
                                DO 410 J=1,LIST                         ELVBI310
      SUM=0.0                                                           ELVBI320
                                DO 300 K=1,3                            ELVBI330
  300 XP(K)=XPR(K,J)-XYZ(K,I)                                           ELVBI340
                                DO 320 N=1,3                            ELVBI350
      YP(N)=0.0                                                         ELVBI360
                                DO 310 K=1,3                            ELVBI370
  310 YP(N)=YP(N)+G(K,N)*XP(K)                                          ELVBI380
  320 SUM=SUM+YP(N)*XP(N)                                               ELVBI390
      IF((SUM-SX)*(SN-SUM))410,410,330                                  ELVBI400
  330 OIST=SQRT(SUM)                                                    ELVBI410
      IF(ISOT(I))338,332,338                                            ELVBI420
  332 DISP=SQRT(ABS(BETA(1,I)))*0.25*TFC)                               ELVBI430
      CORR1=0.50*ABS(BETA(1,I))*TFC                                     ELVBI440
      GO TO 358                                                         ELVBI450
  338 PARA=0.0                                                          ELVBI460
                                DO 350 N=1,3                            ELVBI470
      XP(N)=0.0                                                         ELVBI480
                                DO 340 K=1,3                            ELVBI490
  340 XP(N)=XP(N)+YP(K)*B(N,K)                                          ELVBI500
  350 PARA=PARA+XP(N)*YP(N)                                             ELVBI510
      PARA=PARA*TFC/SUM                                                 ELVBI520
C             PARA IS THE MEAN SQUARE DISPLACEMENT ALONG BOND           ELVBI530
      DISP=SQRT(PARA)                                                   ELVBI540
      CORR1=ABS(TRANS-PARA)                                             ELVBI550
  358 K=NUMB(J)/64                                                      ELVBI560
      IF(ISOT(K))370,360,370                                            ELVBI570
  360 CORR2=0.50*ABS(BETA(1,K))*TFC                                     ELVBI580
      GO TO 400                                                         ELVBI590
  370 TEMP=0.0                                                          ELVBI600
                                DO 380 N=1,6                            ELVBI610
  380 TEMP=TEMP+F(N)*BETA(N,K)                                          ELVBI620
      TEMP=TFC*ABS(TEMP)                                                ELVBI630
      TEMP1=0.0                                                         ELVBI640
      L=MOD(NUMB(J),64)                                                 ELVBI650
                                DO 386 M=1,3                            ELVBI660
      COEF(M)=0.0                                                       ELVBI670
                                DO 386 N=1,3                            ELVBI680
  386 COEF(M)=COEF(M)+RMAT(N,M,L)*YP(N)                                 ELVBI690
                                DO 388 M=1,3                            ELVBI700
  388 YP(M)=COEF(M)                                                     ELVBI710
                                DO 390 N=1,3                            ELVBI720
  390 TEMP1=TEMP1+YP(N)**2*BETA(N,K)                                    ELVBI730
      TEMP1=TEMP1+2.0*(BETA(4,K)*YP(1)*YP(2)+BETA(5,K)*YP(1)*YP(3)+BETA(ELVBI740
     1 6,K)*YP(2)*YP(3))                                                ELVBI750
      TEMP1=ABS(TEMP1)*TFC/SUM                                          ELVBI760
      CORR2=ABS(TEMP1-TEMP)                                             ELVBI770
  400 RIOE=OIST*0.5*ABS(CORR1-CORR2)/DIST                               ELVBI780
      UNCOR=OIST*0.5*(CORR1+CORR2)/DIST                                 ELVBI790
      BL=OIST*0.5*(SQRT(CORR1)-SQRT(CORR2))**2/OIST                     ELVBI800
      BU=OIST*0.5*(SQRT(CORR1)*SQRT(CORR2))**2/OIST                     ELVBI810
      WRITE(IOUT,12)(TAG(L,K),L=1,2),(XPR(L,J),L=1,3),OIST,BL,RIOE,UNCORELVBI820
     1,BU,OISP                                                          ELVBI830
  410 CONTINUE                                                          ELVBI840
  420 CONTINUE                                                          ELVBI850
      RETURN                                                            ELVBI860
      END                                                               ELVBI870
```

```
      SUBROUTINE CORECT(R3,R1,R2,ELT,THETA,RC,BIGR,BIGRP,RP2)           CRCT  10
      DIMENSION R3(3),R1(3),R2(3),RC(3),BIGR(3),BIGRP(3),ELT(3,3),      CRCT  20
     1RP(3),RPP(3),ACON(3,3),BCON(3)                                    CRCT  30
      RP2=0.                                                            CRCT  40
      DO 10 I=1,3                                                       CRCT  50
      RP(I)=R2(I)-R1(I)                                                 CRCT  60
10    RP2=RP2+RP(I)**2                                                  CRCT  70
      ISIG=1                                                            CRCT  80
      DO 20 I=1,3                                                       CRCT  90
      BCON(I)=RC(I)                                                     CRCT 100
      DO 20 J=1,3                                                       CRCT 110
      ACON(I,J)=FIJ(THETA,ELT,RP,RP2,I,J)                               CRCT 120
20    BCON(I)=BCON(I)*ACON(I,J)*R2(J)                                   CRCT 130
      GO TO 40                                                          CRCT 140
      ENTRY CORLIB(R3,ELT,RC)                                           CRCT 150
      DO 30 I=1,3                                                       CRCT 160
      BCON(I)=RC(I)                                                     CRCT 170
      DO 30 J=1,3                                                       CRCT 180
30    ACON(I,J)=0.                                                      CRCT 190
      ISIG=0                                                            CRCT 200
40    DO 50 I=1,3                                                       CRCT 210
      DO 50 J=1,3                                                       CRCT 220
50    ACON(I,J)=ACON(I,J)*FIJ(ELT,I,J)                                  CRCT 230
      CALL SOLVE(ACON,BCON,R3)                                          CRCT 240
      IF(ISIG),,60                                                      CRCT 250
      RETURN                                                            CRCT 260
60    DO 70 I=1,3                                                       CRCT 270
70    RPP(I)=R3(I)-R2(I)                                                CRCT 280
      CALL VXV(RP,RPP,BIGR)                                             CRCT 290
      CALL VXV(RP,BIGR,BIGRP)                                           CRCT 300
      RETURN                                                            CRCT 310
      END                                                               CRCT 320
```

```
      FUNCTION AIJ(R,I,J)                                          AIJ  10
      DIMENSION R(3)                                               AIJ  20
      GO TO (10,20,30),I                                           AIJ  30
10    GO TO (100,200,300),J                                        AIJ  40
20    GO TO (300,100,200),J                                        AIJ  50
30    GO TO (200,300,100),J                                        AIJ  60
100   AIJ=0.                                                       AIJ  70
      RETURN                                                       AIJ  80
200   MM=MOD(I+1,3)+1                                              AIJ  90
      AIJ=R(MM)                                                    AIJ 100
      RETURN                                                       AIJ 110
300   MM=MOD(I,3)+1                                                AIJ 120
      AIJ=-R(MM)                                                   AIJ 130
      RETURN                                                       AIJ 140
      END                                                          AIJ 150
```

```
      FUNCTION BIJK(R,I,JJ,KK)                                     BIJK  10
      DIMENSION R(3)                                               BIJK  20
      DIV=2.                                                       BIJK  30
      IF(JJ.EQ.KK)DIV=1.                                           BIJK  40
      J=MIN0(JJ,KK)                                                BIJK  50
      K=MAX0(JJ,KK)                                                BIJK  60
      GO TO (10,20,30),I                                           BIJK  70
10    GO TO (12,14,400),J                                          BIJK  80
12    GO TO (100,200,300),K                                        BIJK  90
14    IF(K-2)400,400,100                                           BIJK 100
20    GO TO (22,24,400),J                                          BIJK 110
22    GO TO (400,300,100),K                                        BIJK 120
24    IF(K-2)100,100,200                                           BIJK 130
30    GO TO (32,34,100),J                                          BIJK 140
32    GO TO (400,100,200),K                                        BIJK 150
34    IF(K-2)400,400,300                                           BIJK 160
100   BIJK=0.                                                      BIJK 170
      RETURN                                                       BIJK 180
200   MM=MOD(I,3)+1                                                BIJK 190
      BIJK=.5*R(MM)/DIV                                            BIJK 200
      RETURN                                                       BIJK 210
300   MM=MOD(I+1,3)+1                                              BIJK 220
      BIJK=.5*R(MM)/DIV                                            BIJK 230
      RETURN                                                       BIJK 240
  400 BIJK=-.5*R(I)/DIV                                            BIJK 250
      RETURN                                                       BIJK 260
      END                                                          BIJK 270
```

```
      FUNCTION CIJKL(R,I,JJ,KK,LL)                                 CIJK  10
      DIMENSION R(3)                                               CIJK  20
      IF(JJ.EQ.KK.AND.JJ.EQ.LL)GO TO 7                             CIJK  30
      IF(JJ.NE.KK.AND.JJ.NE.LL.AND.KK.NE.LL)GO TO 8                CIJK  40
      IF(JJ-KK)1,2,3                                               CIJK  50
    1 J=JJ                                                         CIJK  60
      L=KK                                                         CIJK  70
      K=LL                                                         CIJK  80
      GO TO 4                                                      CIJK  90
    2 J=MIN0(JJ,LL)                                                CIJK 100
      K=KK                                                         CIJK 110
      L=MAX0(JJ,LL)                                                CIJK 120
      GO TO 4                                                      CIJK 130
    3 J=KK                                                         CIJK 140
      K=LL                                                         CIJK 150
      L=JJ                                                         CIJK 160
    4 DIV=3.                                                       CIJK 170
      GO TO 9                                                      CIJK 180
    7 J=JJ                                                         CIJK 190
      K=JJ                                                         CIJK 200
      L=JJ                                                         CIJK 210
      DIV=1.                                                       CIJK 220
      GO TO 9                                                      CIJK 230
    8 J=1                                                          CIJK 240
      K=2                                                          CIJK 250
      L=3                                                          CIJK 260
      DIV=6.                                                       CIJK 270
    9 GO TO (10,20,30),I                                           CIJK 280
   10 GO TO (12,15,300),J                                          CIJK 290
   12 GOTO (13,100,100),K                                          CIJK 300
   13 GO TO (100,200,300),L                                        CIJK 310
   15 IF(K-2)16,16,200                                             CIJK 320
   16 IF(L-2)200,200,300                                           CIJK 330
   20 GO TO (22,25,200),J                                          CIJK 340
   22 GOTO (23,24,300),K                                           CIJK 350
   23 GO TO (300,100,200),L                                        CIJK 360
   24 IF(L-2)300,300,100                                           CIJK 370
   25 IF(K-2)26,26,100                                             CIJK 380
   26 IF(L-2)100,100,200                                           CIJK 390
   30 GO TO (32,35,100),J                                          CIJK 400
   32 GO TO (33,34,200),K                                          CIJK 410
   33 GO TO (200,300,100),L                                        CIJK 420
   34 IF(L-2)200,200,100                                           CIJK 430
   35 IF(K-2)36,36,300                                             CIJK 440
   36 IF(L-2)300,300,100                                           CIJK 450
  100 CIJKL=0.                                                     CIJK 460
      RETURN                                                       CIJK 470
  200 MM=MOD(I+1,3)+1                                              CIJK 480
      CIJKL=-R(MM)/(6.*DIV)                                        CIJK 490
      RETURN                                                       CIJK 500
  300 MM=MOD(I,3)+1                                                CIJK 510
      CIJKL=R(MM)/(6.*DIV)                                         CIJK 520
      RETORN                                                       CIJK 530
      END                                                          CIJK 540
```

```
      SUBROUTINE VXV(V1,V2,V3)                                          VXV   10
      DIMENSION V1(3),V2(3),V3(3)                                       VXV   20
      V3(1)=V1(2)*V2(3)-V1(3)*V2(2)                                     VXV   30
      V3(2)=V1(3)*V2(1)-V1(1)*V2(3)                                     VXV   40
      V3(3)=V1(1)*V2(2)-V1(2)*V2(1)                                     VXV   50
      RETURN                                                            VXV   60
      END                                                               VXV   70


      FUNCTION DETERM(C1,C2,C3)                                         DTRM  10
      DIMENSION C1(3),C2(3),C3(3)                                       DTRM  20
      DETERM=C1(1)*C2(2)*C3(3)+C1(2)*C2(3)*C3(1)+C1(3)*C2(1)*C3(2)      DTRM  30
     1 -C1(3)*C2(2)*C3(1)-C1(2)*C2(1)*C3(3)-C1(1)*C2(3)*C3(2)           DTRM  40
      RETURN                                                            DTRM  50
      END                                                               DTRM  60


      SUBROUTINE SOLVE(A,B,C)                                           SLVE  10
      DIMENSION A(3,3),B(3),C(3)                                        SLVE  20
      DELTA=DETERM(A(1,1),A(1,2),A(1,3))                                SLVE  30
      C(1)=DETERM(B(1),A(1,2),A(1,3))/DELTA                             SLVE  40
      C(2)=DETERM(A(1,1),B(1),A(1,3))/DELTA                             SLVE  50
      C(3)=DETERM(A(1,1),A(1,2),B(1))/DELTA                             SLVE  60
      RETURN                                                            SLVE  70
      END                                                               SLVE  80


      FUNCTION EL4(EL,I,J,K,L)                                          EL4   10
      DIMENSION EL(3,3)                                                 EL4   20
      EL4=EL(1,J)*EL(K,L)*EL(I,K)*EL(J,L)*EL(1,L)*EL(J,K)               EL4   30
      RETURN                                                            EL4   40
      END                                                               EL4   50


      FUNCTION EIJ(EL,I,J)                                              EIJ   10
      DIMENSION EL(3,3)                                                 EIJ   20
      IF(I.EQ.J)GO TO 20                                                EIJ   30
      EIJ=.5*EL(I,J)-(1./24.)*(EL4(EL,I,1,I,J)*EL4(EL,I,2,2,J)*EL4(EL, EIJ   40
     I 1,3,3,J))                                                        EIJ   50
      RETURN                                                            EIJ   60
   20 K=MOD(I,3)+1                                                      EIJ   70
      L=MOD(I+1,3)+1                                                    EIJ   80
      EIJ=1.-.5*(EL(K,K)*EL(L,L))*(1./24.)*(EL4(EL,I,I,K,K)*EL4(EL,I,I, EIJ   90
     I L,L)*3.+(EL(K,K)**2*EL(L,L)**2)*2.+EL4(EL,K,K,L,L))              EIJ  100
      RETURN                                                            EIJ  110
      END                                                               EIJ  120
```

```
      FUNCTION FIJ(THETA,EL,RP,RP2,I,J)                                 FIJ   10
      DIMENSION EL(3,3),RP(3),CON(3)                                    FIJ   20
      IF(RP2.LT.0.01.OR.THETA.LT.0.00D001)GO TO 100                     FIJ   30
      K=MOD(I,3)+1                                                      FIJ   40
      L=MOD(I+1,3)+1                                                    FIJ   50
      KK=MOD(J,3)+1                                                     FIJ   60
      LL=MOD(J+1,3)+1                                                   FIJ   70
      M2=MOD(J-I+2,3)+1                                                 FIJ   80
      M3=MOD(J-I+3,3)+1                                                 FIJ   90
      M4=MOD(J-I+4,3)+1                                                 FIJ  100
      CON1=1.-THETA/4.-.5*(EL(K,K)*EL(L,L))                             FIJ  110
      CON2=.5*EL(I,K)                                                   FIJ  120
      CON3=.5*EL(I,L)                                                   FIJ  130
      CON(M3)=-(RP(KK)**2*RP(LL)**2)                                    FIJ  140
      CON(M4)=RP(J)*RP(KK)                                              FIJ  150
      CON(M2)=RP(J)*RP(LL)                                              FIJ  160
      FIJ=(CON1+CON(1)*CON2+CON(2)*CON3+CON(3))*THETA/(2.*RP2)          FIJ  170
      RETURN                                                            FIJ  180
  100 FIJ=0.                                                            FIJ  190
      RETURN                                                            FIJ  200
      END                                                               FIJ  210


      COMPILER(FLD=Q)                                                   PDGN  10
      SUBROUTINE PDEGN(NEW7)                                            PDGN  20
      DEFINE I1(K)=FLD(0,9,K)                                           PDGN  30
      DEFINE I2(K)=FLD(9,9,K)                                           PDGN  40
      DEFINE I3(K)=FLD(18,9,K)                                          PDGN  50
      DEFINE I4(K)=FLO(27,9,K)                                          PDGN  60
C     SUBROUTINE TO PRINT OUT DEPENDENT PARAMETER INFORMATION IF A NEW  PDGN  70
C     DEPENDENCY LIST HAS BEEN READ IN OR IF THE DEPENDENCY RELATION,   PDGN  80
C     BEING NONLINEAR, ARE SUBJECT TO CHANGE FROM ONE CYCLE TO THE NEXT.PDGN  90
      INCLUDE DCON,LIST                                                 PDGN 100
      INCLUDE XPAR,LIST                                                 PDGN 110
      INCLUDE TAPE,LIST                                                 PDGN 120
      DIMENSION ITEM(4,5)                                               PDGN 130
      IF(NEW7.GT.0)GO TO 200                                            PDGN 140
      DO 100 I=1,NPAR                                                   PDGN 150
      IF(I1(IPAR(I)),NE.0)GO TO 200                                     PDGN 160
  100 CONTINUE                                                          PDGN 170
      RETURN                                                            PDGN 180
  200 WRITE(IOUT,9200)                                                  PDGN 190
 9200 FORMAT('0DEPENDENT PARAMETER INFORMATION')                        PDGN 200
      I=1                                                               PDGN 210
  210 J=I*4                                                             PDGN 220
      IF(J.GT.NDEGN)J=NDEGN                                             PDGN 230
      DO 220 K=I,J                                                      PDGN 240
      L=K-I*1                                                           PDGN 250
      M=2*K-1                                                           PDGN 260
      ITEM(1,L)=I3(KDEGN(M))                                            PDGN 270
      ITEM(2,L)=I4(KDEGN(M))                                            PDGN 280
      ITEM(3,L)=I1(KDECN(M))                                            PDGN 290
  220 ITEM(4,L)=I2(KDECN(M))                                            PDGN 300
      WRITE(IOUT,9220)((ITEM(K,L-I*I),K=1,4),AA(2*L),L=I,J)             PDGN 310
```

```
 9220 FORMAT(1X,5(4I3,F10.4))                                          POGN 320
      I=I+5                                                            POGN 330
      IF(I.LE.NDEGN)GO TO 210                                          POGN 340
      RETURN                                                           POGN 350
      END                                                              POGN 360
```

```
      SUBROUTINE WEIGHT                                                DUMY  1D
                                                                       DUMY  2D
C THIS ROUTINE REVISED AND/OR RESEQUENCED ON OCT. 28, 1968             DUMY  3D
                                                                       DUMY  4D
C        DUMMY ROUTINE                                                 DUMY  5D
      RETURN                                                           DUMY  6D
      ENTRY REJECT                                                     DUMY  7D
C        DUMMY ROUTINE                                                 DUMY  8D
      RETURN                                                           DUMY  9D
      END                                                              DUMY 10D
```

```
      XPAR    PROG
              COMMON/XPAR/NPAR,ATOX(3,3),XTOA(3,3),PAR(60),PARNAM(2,60),IPAR(60)
             1,IRPAR(60),OXYZ(3),SXYZ(3),OXP(3),T(3,3),EL(3,3),S(3,3),TRNS(3,3)
             2  ,TTRNS(3,3),TRANSI(3,3),TTRNSI(3,3),RTEM(3),DTEM(6),GTEM(10)
       END
      TAPE    PROG
              COMMON/TAPE/IN,INN,IOUT,ISC1,ISC2,ISC3,ISC4,ISC5,IPUN
       END
      LSMAT   PROG
              COMMON/LSMAT/H(200),IEXT,IPARA(1000),MRANK,KCYCL,LABEL(3),JSCRE(10
             1),NRSF,SIGMA,NOEGN,A(200)
       END
      ACOM    PROG
              COMMON/A/BETA(6,60),G(3,3),ISOT(60),NATOM,TAG(2,60),TITLE(18),XYZ(
             13,60),PI,CR(1D,6D),OR(15,60)
       END
      DCOM    PROG
              COMMON/D/ACALC,AIND(60),ANOM(2,20),BCALC,BIND(6D),DADB(6,6D),DADM(
             1 60),DALX(3,6D),DBOB(6,60),DBDO(60),DBDX(3,6D),IW(3),IH(3),ISCAT(2,
             2 60),ISCL,MFSEL,OCCA(60),OCCUP(60),SCALE(10),SCAT(2D),SITE(60)
             3,DADC(10,60),DBDC(10,6D),DADD(15,6D),DBDD(15,60),AMINUS,BMINUS
             4,DRDE(6),NOEGN
              DIMENSION SAVE(6D),KDEGN(2640),AA(2640)
              EQUIVALENCE (SAVE,AIND),(KDEGN,AA,DADC(1,37))
       END
      IND23   PROG
              COMMON/IND23/IND2(2,6),IND3(3,1D)
       END
```

```
. ROUTINE TO ADJUST THE SIZE OF THE MEMORY FIELD OF AN ACTIVE PROGRAM
. CALLING SEQUENCE
.      CALL CORSZ(ARRAY(N))
.      'ARRAY' MUST BE AT THE END OF THE 'I' OR 'D' BANK OF THE PROGRAM.
.      N IS THE LARGEST INDEX REQUIRED BY THE PROGRAM.
       AXRS
$(I)
CORSZ*  L   A0,0,X11   . GET FINAL ADDRESS
        TNE   A0,LAST              . SAME SIZE AS BEFORE?
        J     2,X11                . YES - RETURN.
        L     A1,LAST
        S     A0,LAST
        TG    A0,A1                . NEED MORE CORE?
        J     CRSZ1                . YES
        ER    LCORES               . NO. REDUCE NEEDS.
        J     2,X11                . RETURN.
CRSZ1   ER    MCORES               . GET MORE.
        J     2,X11
LAST    *0
        END
```

```
      SOSROOTINE SYMINV (A,NRANK,SCR,NCS,E,EPS,OET,JRANK)                SYMV  10
                                                                         SYMV  20
C THIS ROUTINE REVISED AND/OR RESEQUENCED OCT. 2E, 196E                  SYMV  30
C                                                                        SYMV  40
C       ARGUMENT LIST                                                    SYMV  50
C         A = INPOT ANO OUTPUT ARRAY - STOREO SY ROWS WITH ONLY THE       SYMV  60
C             UNIQUE PORTION IN STORAGE                                  SYMV  70
C         NRANK = INPUT SIZE OF SYSTEM OF EQUATIONS                      SYMV  E0
C         SCR = A SCRATCH ARRAY OF DIMENSION NRANK                       SYMV  90
C         E = MATRIX TO BE PREMULTIPLIED SY THE INVERSE OF A             SYMV 100
C         NCS = NOMBER OF COLUMNS OF S                                   SYMV 110
C         EPS = SIZE OF SMALLEST ELEMENT TO BS OSED AS A PIVOT           SYMV 120
C         OET = OUTPUT VALUE OF THE OSTERMINANT OF A                     SYMV 130
C         JRANK = ACTUAL RANK OF A ( THIS IS AN OUTPUT VALUE.)           SYMV 140
                                                                         SYMV 150
      OIMENSION A(1),B(NRANK,1),SCR(1)                                   SYMV 160
C       INITIALIZE                                                       SYMV 170
      OET=1.0                                                            SYMV 180
      DO 100 I=1,NRANK                                                   SYMV 190
  100 SCR(I)=1.0                                                         SYMV 200
C       START MASTER LOOP                                                SYMV 210
      DO 2E0 I=1,NRANK                                                   SYMV 220
C       SEARCH FOR PIVOT TESTING ONLY THE OIAGONAL ELEMENTS              SYMV 230
      BIG=0.0                                                            SYMV 240
C       PRESET INOEX ANO STEP FOR OIAGONAL ELEMENTS                      SYMV 250
      J=-NRANK                                                           SYMV 260
      KJ=NRANK+1                                                         SYMV 270
      DO 130 K=1,NRANK                                                   SYMV 2E0
C       AOGMENT INOEX ANO REOOCE STEP                                    SYMV 290
      J=J+KJ                                                             SYMV 300
      KJ=KJ-1                                                            SYMV 310
C       CHECK IF THIS ELEMENT ALREAOY USEO AS PIVOT                      SYMV 320
      IF(ECR(K))130,110,110                                              SYMV 330
C       J INOEXES A(K,K)                                                 SYMV 340
  110 TEMP=AES(A(J))                                                     SYMV 350
      IF(TEMP-BIG)130,130,120                                            SYMV 360
  120 BIG=TEMP                                                           SYMV 370
      KK=K                                                               SYMV 3E0
      KL=J                                                               SYMV 390
  130 CONTINUE                                                           SYMV 400
C       COMPARE LARGEST OIAGONAL ELEMENT WITH EPS                        SYMV 410
      IF(BIG-EPS)140,150,150                                             SYMV 420
C       SINGOLARITY RETURN                                               SYMV 430
  140 JRANK=I-1                                                          SYMV 440
      RETORN                                                             SYMV 450
  150 SCR(KK)=-I.0                                                       SYMV 460
C       KL INOEXES A(KK,KK)                                              SYMV 470
      OET=OET*A(KL)                                                      SYMV 4E0
      Q=1.0/A(KL)                                                        SYMV 490
C       SET UP INOEXING FOR ELIMINATION STEPS                            SYMV 500
      J=KK-NRANK                                                         SYMV 510
      NJ=NRANK+1                                                         SYMV 020
      K=0                                                                SYMV 530
      IK=KK-1                                                            SYMV 540
      JK=KK+1                                                            SYMV 550
      IF(IK)210,210,160                                                  SYMV 560
  160 DO 200 N=1,IK                                                      SYMV 570
      MJ=NJ-I                                                            SYMV 080
      J=J+NJ                                                             SYMV 090
C       J INOEXES A(KK,N)                                                SYMV 600
      TEMP=Q*A(J)                                                        SYMV 610
      NI=J                                                               SYMV 620
      MJ=NJ                                                              SYMV 630
      OO 170 M=N,IK                                                      SYMV 640
      K=K+1                                                              SYMV 600
C       J INOEXES A(KK,M)                                                SYMV 660
C       K INOEXES A(M,N)                                                 SYMV 670
      A(K)=A(K)-TEMP*SCR(M)*A(J)                                         SYMV 600
      MJ=MJ-I                                                            SYMV 690
  170 J=J+MJ                                                             SYMV 700
      J=NI                                                               SYMV 710
C       J INOEXES A(KK,N)                                                SYMV 720
      A(J)=TEMP                                                          SYMV 730
      K=K+1                                                              SYMV 740
      L=KL                                                               SYMV 700
      IF(JK-NRANK)1EO,IE0,200                                            SYMV 760
  180 DO 190 M=JK,NRANK                                                  SYMV 770
      L=L+1                                                              SYMV 780
      K=K+1                                                              SYMV 790
C       K INDEXES A(M,N)                                                 SYMV 800
C       L INDEXES A(M,KK)                                                SYMV 010
  190 A(K)=A(K)-TEMP*A(L)                                                SYMV 020
  200 CONTINOE                                                           SYMV 830
  210 K=KL+NRANK-KK                                                      SYMV 040
      J=KL                                                               SYMV 000
      IF(JK-NRANK)220,220,200                                            SYMV 060
  220 DO 240 N=JK,NRANK                                                  SYMV 870
      J=J+1                                                              SYMV 880
C       J INDEXES A(M,KK)                                                SYMV 89
      TEMP=SCR(N)*A(J)*Q                                                 SYMV 900
      MI=J                                                               SYMV 910
      DO 230 M=N,NRANK                                                   SYMV 920
      K=K+I                                                              SYMV 930
C       K INDEXES A(M,N)                                                 SYMV 940
C       J INDEXES A(M,KK)                                                SYMV 950
      A(K)=A(K)-TEMP*A(J)                                                SYMV 960
  230 J=J+1                                                              SYMV 970
  240 J=MI                                                               SYMV 980
  200 K=KL                                                               SYMV 990
C       K INOEXES A(KK,KK)                                               SYMV1000
      A(K)=Q                                                             SYMV1010
      IF(JK-NRANK)260,260,2E0                                            SYMV1020
  260 DO 270 M=JK,NRANK                                                  SYMV1030
      K=K+1                                                              SYMV1040
C       K INDEXES A(M,KK)                                                SYMV1000
  270 A(K)=-Q*A(K)                                                       SYMV1060
  280 CONTINUE                                                           SYMV1070
C       INVERSION COMPLETE, EXIT IF INVERSION ONLY                       SYMV1080
      IF(MCB)360,360,290                                                 SYMV1090
C       PREMULTIPLY THE MCS COLOMNS OF S BY A INVERSE                    SYMV1100
  290 DO 300 K=1,NCE                                                     SYMV111
      DO 330 I=1,NRANK                                                   SYMV1120
      Q=0.0                                                              SYMV1130
      L=I-NRANK                                                          SYMV1140
      KK=NRANK+I                                                         SYMV1150
```

```
      DO 300 J=1,I                                                      SYMV1160
      KK=KK-I                                                           SYMV1170
      L=L+KK                                                            SYMV1180
C         L INDEXES A(I,J)                                              SYMV1190
  300 Q=Q*A(L)*B(J,K)                                                   SYMV1200
      IF(I-NRANK)310,330,330                                            SYMV1210
  310 KJ=I+I                                                            SYMV1220
      DO 320 J=KJ,NRANK                                                 SYMV1230
      L=L+I                                                             SYMV1240
C         L INDEXES A(J,I)                                              SYMV1250
  320 Q=Q*A(L)*B(J,K)                                                   SYMV1260
  330 SCR(I)=Q                                                          SYMV1270
      DO 340 J=I,NRANK                                                  SYMV1280
  340 B(J,K)=SCR(J)                                                     SYMV1290
  350 CONTINUE                                                          SYMV1300
C         SET NON-SINGULAR FLAG AND EXIT                                SYMV1310
  360 JRANK=NRANK                                                       SYMV1320
      RETURN                                                            SYMV1330
      END                                                               SYMV1340
```

```
      READ(5,1)NAME                                                     BNDN  10
    1 FORMAT(12)                                                        BNDN  20
      IF(NAMB-2)3,4,5                                                   BNDN  30
    3 CALL BADTBA                                                       BNDN  40
      GO TO 6                                                           BNDN  50
    4 CALL BER                                                          BNDN  60
      GO TO 6                                                           BNDN  70
    5 WRITE(6,2)NAME                                                    BNDN  80
    2 FORMAT(' WRONG NUMBER ON CALLING CARD, ACTUAL NUMBER IS',I5)      BNDN  90
    6 STOP                                                              BNDN 100
      END                                                               BNDN 110
```

```
C  PROGRAM BADTEA                                                      BADT  10
      SURROUTINE BADTEA                                                 BADT  20
C THIS ROUTINE REVISED AND/OR RESEQUENCED ON OCT. 28, 1968              BADT  30
C                                                                       BADT  40
C        UNIVERSITY OF MINNESOTA   BOND ANGLES, BOND DISTANCES, THERMAL BADT  50
C           ELLIPSOIDS AND ERRORS                                       BADT  60
C                     BY  LARRY W. PINGER          NOV.1965             BADT  70
C        CONVERTED TO FORTRAN IV      MAY 1967                          BADT  80
C                                                                       BADT  90
                                                                        BADT 100
      DIMENSION S(3),IT(3),IFOS(2,3),FPR(8),IFB(10),IPA(10),            BADT 110
     1 IIR(8),SIG(180),JCELL(3,6),JPOS(3),JBETA(6),SIGG(18D)            BADT 120
                                                                        BADT 130
      COMMON/ABLE/XYZ(3,60),BETA(6,6D),A(3),ANG(3),AS(3),CS(3),ATOM(2,  BADT 140
     1 60),SIGA(6,6),SIGB(6,6,60),RS(3,3,48),TS(3,48),TP(4,3),ITF(6D),  BADT 150
     2 NA,NSS,VOL,DLIMIT(3),FI,G(3,3),NLP,IERR,TITLE(18),CORR(1)        BADT 160
      COMMON/RAZER/XYZP(3,1000),IATP(1000)                              BADT 170
                                                                        BADT 180
      EQUIVALENCE (XYZP,SIG),(XTZP(181),SIGG)                           BADT 190
                                                                        BADT 200
      DATA FPB/D.0,0.S,2*0.0,0.25,0.7S,2*D.0/                           BADT 210
      DATA JPOS/1HX,1HY,1HZ/                                            BADT 220
      DATA IIB/3H   ,3H1/2,3BI/3,3H2/3,3H1/4,3B3/4,3B1/6,3HS/6/         BADT 230
      DATA IFB/3H   ,3H  X,3B -X,3H -X,3H  V,3H -Y,3R -Y,3H  Z,3N -Z,3R BADT 240
     1-Z/                                                               BADT 250
      DATA IPA/D,1,1,-1,2,2,-2,3,3,-3/                                  BADT 260
      DATA IN,IOUT,INAUX,JCELL/5,6,4,1BA,2*1B ,1HB,2*1B ,1BC,2*1H ,4NCOSBADT 270
     1 ,4RALPR,IBA,4RCOS ,4RBETA,1H ,4HCOB ,4BOAMM,1BA/                 BADT 280
      DATA JBETA/2H11,2H22,2H33,2H12,2N13,2B23/                         BADT 290
      DATA IP1,IA1,IB1,IC1,II1,IP1/1RP,1HA,1HB,1HC,1BI,1HF/             BADT 300
                                                                        BADT 310
    1 FORMAT(1BA4)                                                      BADT 320
    2 FORMAT(1H1IOX,1BA4)                                               BADT 330
    3 FORMAT(6F10.5)                                                    BADT 340
    5 FORMAT(5I4,3XA1,6X,3F10.5)                                        BADT 350
    8 FORMAT(3(3A3,1X)I4)                                               BADT 360
    9 FORMAT(5X3(3A3,4X))                                               BADT 370
   34 FORMAT(10FB.3)                                                    BADT 380
   50 FORMAT(8F10.8)                                                    BADT 390
   56 FORMAT(2A3,7X3F7.5,I2/6FB.5)                                      BADT 400
   57 FORMAT(/24X,BA4,A1,9A4/IOB BEAL CELL,7X,6F12.5/                   HADT 410
     1 17H RECIPROCAL CELL ,6F12.5//13B CELL VOLUME ,F16.5)             BADT 420
   5B FORMAT(//19B SYMMETRY OPERATORS,5X,A1,5B CELL,5X,A1,7BCENTRIC)    BADT 430
  106 FORMAT(/29HSTANDARD DEVIATIONS FOR CELL1DX8A4,A1,9A4              BADT 440
     1    /31X,6P12.6//24B CELL CORRELATION MATRIX/)                    BADT 450
  109 FORMAT(//50X,4HB OB/5X,4HATOM,13X,2(A4,6X),A1,6X,6(4BBETAA2,4X)/) BADT 460
  11D FORMAT(4X,2A3,4X9F1D.5)                                           BADT 47D
  111 FORMAT(4X,2A3,4X3F10.S,F10.2)                                     BADT 480
  201 FORMAT(1X,3A4,2X,6P10.6)                                          BADT 490
  202 FORMAT(//44H TEMPERATURE FACTOR STANDARD DEVIATIONS FOR ,2A3/9X,  BADT 500
     16(4RBETA,A2,4X)/5X,6P10.6//19B CORRELATION MATRIX /)              BADT 510
  204 FORMAT(5X,4RBETA,A2,6F10.6)                                       BADT S20
  206 FORMAT(//34B ATOM POSITION STANDARD DEVIATIONS/10X,4BATOM,13X,3(A1BADT 530
     1 ,9X)/)                                                           BADT 540
  207 FORMAT(9X,2A3,4X,3F10.S)                                          BADT 550
  208 FORMAT(//5X,32HATOM POSITION CORRELATION MATRIX/)                 BADT 560
  209 FORMAT(/4X,2A3,2X,1BX,7X,10F10.5)                                 BADT 570
```

```
  210 FORMAT(/12XA1,7X10P10.5)                                          BADT 580
  211 FORMAT(20X10P10.5)                                                BADT 590
                                                                        BADT 600
      FPB(3)=1./3.                                                      BADT 610
      FPB(4)=2./3.                                                      BADT 620
      FPB(7)=1./6.                                                      BADT 630
      FPB(8)=5./6.                                                      BADT 640
      PI=ACOS(-1.0)                                                     BADT 650
      READ(IN,1)T1TLE                                                   BADT 660
      WRITE(IOUT,2)T1TLE                                                BADT 670
      READ(IN,3)A,ANG                                                   BADT 680
C        CALCULATE SINES AND COSINES OF DIRECT SPACE ANGLES             BADT 690
      DO 4 I=1,3                                                        BADT 700
      IF(ABS(ANG(I)-90.0).GT.1.0E-3)GO TO 1000                          BADT 710
      S(I)=1.0                                                          BADT 720
      ANG(I)=0.0                                                        BADT 730
      GO TO 4                                                           BADT 740
 1000 ANG(I)=ANG(I)*PI/180.0                                            BADT 750
      S(I)=SIN(ANG(I))                                                  BADT 760
      ANG(I)=COS(ANG(I))                                                BADT 770
C        MATRIX G IS THE REAL SPACE METRIC TENSOR                       BADT 780
    4 G(I,I)=A(I)**2                                                    BADT 790
      DO 144 I=1,2                                                      BADT 800
      IJ=I+1                                                            BADT 810
      DO 144 J=IJ,3                                                     BADT 820
      IK=6-I-J                                                          BADT 830
      G(I,J)=A(I)*A(J)*ANG(IK)                                          BADT 840
  144 G(J,I)=G(I,J)                                                     BADT 850
C        COMPUTE REAL SPACE CELL VOLUME                                 BADT 860
      VOL=A(1)*A(2)*A(3)*SQRT (1.0-ANG(1)**2-ANG(2)**2-ANG(3)**2        BADT 870
     1 +2.0*ANG(1)*ANG(2)*ANG(3))                                       BADT 880
C        COMPUTE RECIPROCAL SPACE CELL PARAMETERS                       BADT 890
      AS(1)=A(2)*A(3)*S(1)/VOL                                          BADT 900
      AS(2)=A(1)*A(3)*S(2)/VOL                                          BADT 910
      AS(3)=A(1)*A(2)*S(3)/VOL                                          BADT 920
      CS(1)=(ANG(2)*ANG(3)-ANG(1))/(S(2)*S(3))                          BADT 930
      CS(2)=(ANG(1)*ANG(3)-ANG(2))/(S(1)*S(3))                          BADT 940
      CS(3)=(ANG(1)*ANG(2)-ANG(3))/(S(1)*S(2))                          BADT 950
      WRITE(IOUT,57)JCELL,A,ANG,AS,CS,VOL                               BADT 960
C        READ CONTROL CARD                                              BADT 970
      READ(IN,5)NA,ICNT,IBOD,IELV,IERR,LTYPE,DLIMIT                     BADT 980
      IF(DLIMIT(3).LE.0.0) DLIMIT(3)=DLIMIT(2)                          BADT 990
      I=2-ICNT                                                          BADT1000
      WRITE(IOUT,58)LTYPE,JCELL(1,1)                                    BADT1010
C        READ SYMMETRY CARDS                                            BADT1020
      DO 7 I=1,48                                                       BADT1030
      READ(IN,8)(IT(K),(IPOS(L,K),L=1,2),K=1,3),IEF                     BADT1040
      WRITE(IOUT,9)(IT(K),(IPOS(L,K),L=1,2),K=1,3)                      BADT1050
C        CLEAR ROTATIONAL MATRICES                                      BADT1060
      DO 12 J=1,3                                                       BADT1070
      DO 10 K=1,3                                                       BADT1080
   10 RS(K,J,I)=0                                                       BADT1090
C        CONVERT TRANSLATIONAL OPERATOR                                 BADT1100
      DO 11 K=1,8                                                       BADT1110
      IF(IT(J)-IIH(K))11,13,11                                          BADT1120
   11 CONTINUE                                                          BADT1130
C        CARD CHARACTERS ARE NOT IN THE ACCEPTABLE LIST                 BADT1140
      STOP 11                                                           BADT1150
C        SET TRANSLATIONAL VALUE                                        BADT116D
   13 TS(J,I)=PPB(K)                                                    BADT117D
      DO 14 K=1,3                                                       BADT118O
      DO 14 L=1,2                                                       BADT119O
C        COMPARE POSITIONAL INFORMATION WITH ACCEPTABLE VALUES          BADT120D
      DO 15 N=1,10                                                      BADT121D
      IP(IPB(N)-IPOS(L,K))15,16,15                                      BADT122D
   15 CONTINUE                                                          BADT123D
C        IMPROPER CODE ON CARD                                          BADT124O
      STOP 11                                                           BADT125O
   16 IP(J-IABS (IPA(N)))14,17,14                                       BADT126O
C        STORE ROTATIONAL MATRIX ELEMENT                                BADT127D
   17 RS(K,J,1)=ISIGN (1,IPA(N))                                        BADT128D
   14 CONTINUE                                                          BADT129O
   12 CONTINUE                                                          BADT130O
C        CHECK IF THIS IS THE LAST SYMMETRY CARD                        BADT131O
      IF(IEF)18,7,18                                                    BADT132O
    7 CONTINUE                                                          BADT133O
C        TOO MANY SYMMETRY CARDS                                        BADT134D
      STOP 22                                                           BADT135O
   18 NSS=I                                                             BADT136O
      IF(ICNT)38,39,38                                                  BADT137O
C        EXPAND LIST FOR CENTRIC SPACE GROUP                            BADT138D
   39 DO 40 I=1,NSS                                                     BADT139O
      N=I+NSS                                                           BADT140O
      DO 40 J=1,3                                                       BADT141O
      TS(J,N)=-TS(J,I)                                                  BADT142O
      DO 40 K=1,3                                                       BADT143O
   40 RS(K,J,N)=-RS(K,J,I)                                              BADT144O
      NSS=2*NSS                                                         BADT145O
C        FORM MATRICES FOR LATTICE POINTS                               BADT146O
   38 NLP=1                                                             BADT147O
      DO 19 I=1,3                                                       BADT148O
   19 TP(I,I)=0                                                         BADT149O
      IF(LTYPE-1P1)21,20,21                                             BADT150O
C        LATTICE IS NOT PRIMITIVE                                       BADT151D
   21 DO 22 I=1,3                                                       BADT152O
   22 TP(2,I)=0.5                                                       BADT153O
      NLP=2                                                             BADT154O
      IP(LTYPE-1A1)23,24,23                                             BADT155O
C        A CENTERED LATTICE                                             BADT156D
   24 TP(2,1)=0                                                         BADT157D
      GO TO 20                                                          BADT158D
   23 IF(LTYPE-1B1)25,26,25                                             BADT159D
C        B CENTERED LATTICE                                             BADT160O
   26 TP(2,2)=0                                                         BADT161O
      GO TO 20                                                          BADT162O
   25 IF(LTYPE-IC1)28,27,28                                             BADT163O
C        C CENTERED LATTICE                                             BADT164O
   27 TP(2,3)=0                                                         BADT165O
      GO TO 20                                                          BADT166O
   28 IF(LTYPE-1I1)29,20,29                                             BADT167O
   29 IP(LTYPE-IP1)31,30,31                                             BADT16BD
C        P CENTERED LATTICE                                             BADT169D
   30 NLP=4                                                             BADT170O
      DO 32 I=2,4                                                       BADT171O
      DO 33 J=1,3                                                       BADT172O
   33 TP(I,J)=0.5                                                       BADT173O
```

```
   32 TP(I,I-1)=0                                                    BADT1740
      GO TO 20                                                       BADT1750
C        R CENTERED LATTICE                                         BADT1760
   31 TP(2,1)=1./3.                                                  BADT1770
      TP(3,2)=TP(2,1)                                               BADT1780
      TP(3,3)=TP(2,1)                                               BAOT1790
      NLP=3                                                         BADT1800
      TP(2,2)=2./3.                                                 BAOT1810
      TP(3,1)=TP(2,2)                                               BAOT1820
   20 IF(IERR.EQ.0)GO TO 101                                        BADT1830
C        READ CELL SIGMAS AND CORRELATION MATRIX. CHANGE TO VAR-COV  BADT1840
  100 READ(IN,50)(SIG(I),I=1,6)                                     BADT1850
C        CONVERT SIGMA OF ANGLES TO SIGMA OF COSINE                 BAOT1860
      DO 102 I=4,6                                                  BADT1870
  102 SIG(I)=S(I-3)*SIG(I)*PI/180.0                                 BADT1880
      WRITE(IOUT,106)JCELL,(SIG(I),I=1,6)                           BAOT1890
      DO 200 I=1,6                                                  BADT1900
      READ(IN,34)(SIGA(I,J),J=1,I)                                  BADT191U
      SIGA(I,1)=1.0                                                 BADT1920
      WRITE(IOUT,201)(JCELL(J,I),J=1,3),(SIGA(I,J),J=1,I)           BADT1930
      DO 200 J=1,I                                                  BADT1940
      SIGA(I,J)=SIGA(I,J)*SIG(I)*SIG(J)                             BADT1950
  200 SIGA(J,I)=SIGA(I,J)                                           BADT1960
  101 WRITE(IOUT,109)JPOS,JBETA                                     BAOT1970
      IO=IN                                                         BAOT1980
      IF(IERR.NE.2) GO TO 300                                       BADT1990
C        SET TO READ AUXILLARY INPUT UNIT                           BADT2000
      IO=INAUX                                                      BADT2010
      REWIND IO                                                     BAOT2020
C        READ STRUCTURE PARAMETERS                                  BADT2030
  300 DO 6 I=1,NA                                                   BADT2040
      READ(IO,56)(ATOM(J,I),J=1,2),(XYZ(J,I),J=1,3),ITP(I),(BETA(J,I),J=BAOT2050
     11,6)                                                          BADT2060
      IP(ITP(I))112,113,112                                         BAOT2070
  112 WRITE(IOUT,110)(ATOM(J,I),J=1,2),(XYZ(J,I),J=1,3),(BETA(J,I),J=1,6BAOT2080
     1)                                                             BAUT2090
      GO TO 6                                                       BAOT2100
  113 WRITE(IOUT,111)(ATOM(J,I),J=1,2),(XYZ(J,I),J=1,3),BETA(1,I)   BADT2110
    6 CONTINUE                                                      BADT2120
      IP(IERR.EQ.0.OR.(IELV.EQ.U.AND.IERR.NE.2)) GO TO 103          BAOT2130
C        READ TEMPERATURE FACTOR SIGMAS AND CORR. MATRIX            BAOT2140
      I=0                                                           BAOT2150
      DO 41 II=1,NA                                                 BAOT2160
      I=I+1                                                         BADT2170
      IP(ITP(I))42,41,42                                            BAOT2180
   42 READ(IO,50)(SIG(J),J=1,6)                                     BAOT2190
      WRITE(IOUT,202)(ATOM(J,I),J=1,2),BEETA,(SIG(J),J=1,6)         BADT2200
      DO 203 J=1,6                                                  BADT2210
      READ(IO,34)(SIGE(J,K,I),K=1,J)                                BADT2220
      SIGE(J,J,I)=1.0                                               BAOT2230
      WRITE(IOUT,204)JEETA(J),(SIGE(J,K,I),K=1,J)                   BADT2240
C        CONVERT CORRELATION MATRIX TO VARIANCE-COVARIANCE MATRIX   BAOT2250
      DO 203 K=1,J                                                  BADT2260
      EIGE(J,K,I)=SIGE(J,K,I)*EIO(J)*SIG(K)                         BADT2270
  203 EIGE(K,J,I)=SIGE(J,K,I)                                       BAOT2280
   41 CONTINUE                                                      BAOT2290
  103 IP(IERR.EQ.0.OR.(IBGO.EQ.0.AND.IERR.NE.2)) GO TO 105          BADT2300
C        READ ATOM POSITION SIGMAS AND CORRELATION MATRIX           BADT2310
  104 WRITE(IOUT,206)JPOS                                           BADT2320
      NAA=3*NA                                                      BADT2330
      DO 205 I=1,NA                                                 BADT2340
      IJ=3*(I-1)+1                                                  BADT2350
      JI=IJ+2                                                       BADT2360
      READ(IO,50)(SIG(J),J=IJ,JI)                                   BADT2370
  205 WRITE(IOUT,207)(ATOM(J,I),J=1,2),(BIG(J),J=IJ,JI)             BADT23E0
      WRITE(IOUT,208)                                               BADT239U
      L=0.                                                          BADT2400
      DO 223 I=1,NA                                                 BADT2410
      DO 223 J=1,3                                                  BADT2420
      IJ=3*(I-1)*J                                                  BADT2430
      READ(IO,34)(SIGG(K),K=1,IJ)                                   BADT244U
      SIGG(IJ)=1.0                                                  BADT2450
      IP(J.NE.1.OR.I.GT.4)GO TO 215                                 BADT2460
      WRITE(IOUT,209)(ATOM(K,I),K=1,2),(SIGG(K),K=I,IJ)             BADT2470
      GO TO 216                                                     BADT2480
  215 IP(J.NE.I)GO TO 21E                                           BADT2490
      WRITE(IOUT,209)(ATOM(K,I),K=1,2),(SIGG(K),K=I,IO)             BADT2500
      GO TO 221                                                     BADT2510
  21E IF(I.GE.4)GO TO 220                                           BADT2520
      WRITE(IOUT,210)JPOB(J),(EIGG(K),K=1,IJ)                       BADT2530
      GO TO 216                                                     BADT2540
  220 WRITE(IOUT,210)JPOS(J),(SIGG(K),K=1,10)                       BADT2550
  221 WRITE(IOUT,211)(SIGG(K),K=11,IJ)                              BADT2560
C        CONVERT CORRELATION MATRIX TO VARIANCE-COVARIANCE MATRIX   BADT2570
  216 DO 222 K=1,IJ                                                 BADT25E0
  222 SIGG(K)=SIGG(K)*BIG(K)*EIG(IJ)                                BADT259U
      EIGG(IJ)=SIG(IJ)**2                                           BADT2600
      DO 223 K=1,IJ                                                 BADT2610
      L=L+1                                                         BADT2620
  223 CORR(L)=SIGG(K)                                               BADT2630
  105 IP(IBGO.EQ.0)GO TO 44                                         BADT2640
C        COMPUTE BOND DISTANCES AND ANGLES                          BADT265U
   43 CALL EGOAN(IBGO)                                              BADT2660
   44 IP(IBLV)45,46,45                                              BADT267U
C        COMPUTE THERMAL ELLIPSOIDS                                 BADT2680
   45 CALL ELVIE                                                    BADT2690
C        END OF CALCULATION, TERMINATE PROGRAM                      BADT2700
   46 CALL EXIT                                                     BADT2710
      END                                                           BADT2720
```

```
      COMPILER(PLD*G)                                                   ERR  10
      SUBROUTINE ERR                                                    ERR  20
C        PROGRAM ERROR TO READ THE FILE OF REFINE MODIFIED FOR GENERAL  ERR  30
C        CONSTRAINTS.     MODIFIED JAN. 15, 1974                       ERR  40
C                                                                       ERR  50
C     PROGRAM TO READ THE DATA FILE PREPARED BY THE FINGER LEAST-SQUARES ERR  60
C        PROGRAM, REFORMAT THE PARAMETER VARIANCE-COVARIANCE MATRIX     ERR  70
C        AND THEN CALL THE BOND DISTANCE AND ANGLE SUBROUTINE OR THE    ERR  80
C        THERMAL ELLIPSOID SUBROUTINE TO COMPUTE THESE FUNCTIONS        ERR  90
C        AND THEIR ERRORS.                                              ERR 100
C                                                                       ERR 110
      DEFINE I2(C)=PLD(9,9,C)                                           ERR 120
      DEFINE I3(C)=PLD(15,9,C)                                          ERR 130
      DEFINE I4(C)=PLD(27,9,C)                                          ERR 140
      DIMENSION IPARA(1000),XPAR(60),XEPAR(60),XLAB(3),LABEL(3),S(3),   ERR 150
     1 TEMP(20),TEMP1(9,20),TEMP2(2,20),TENP(3),TEEP(5),ZT(5,60),JSP(10 ERR 160
     2,CR(10,60),DR(15,60),PAR(60),PARNAM(2,60),TEMP4(6),ITEMP4(6),NCNT( ERR 170
     34),EXTIN(6),IPNT(6,60),JPNT(1E0),CNSTR(2),CV(2),CNSDGN(2)         ERR 180
C                                                                       ERR 190
      COMMON/AELE/XYZ(3,60),BETA(6,60),A(3),ANG(3),AE(3),CS(3),ATOM(2,  ERR 200
     1 60),BIGA(6,6),BIGE(6,6,60),BS(3,3,4E),TE(3,4E),TP(4,3),ITF(60),  ERR 210
     2 NA,NSS,VOL,DLIMIT(3),PI,G(3,3),NLP,IERR,TITLE(1E),EIGX(5000)     ERR 220
      EQUIVALENCE(SIGX,IPARA),(SIGX(1001),XRPAR),(BIGX(1061),IPNT,JPNT),ERR 230
     1(SIGX(1421),PAR),(BIGX(14E1),KPAR),(BIGX(1541),CR,DR,PARNAM,      ERR 240
     2TEMP,TEMP1,TEMP2,TENP,TEOP,ZT,TEMP4,CNETR,CNSDGN,CV,XNSTR)        ERR 250
C                                                                       ERR 260
      DATA IN,IGOT,ITAPE,NCNT/5,6,10,1,6,16,31/                         ERR 270
C                                                                       ERR 280
C        DEFINE TWO INDEXING FUNCTIONS PER TRIANGULAR MATRICES          ERR 290
      INDEXL(I,J)=((MAXO(I,J)-1)*MAXO(I,J))/2+MINO(I,J)*NMXL           ERR 300
      INDEXO(I,J)=(MRANK*1)*(MINO(I,J)-1)-((MINO(I,J)-1)*MINO(I,J))/2  ERR 310
     1*MAXO(I,J)*1-MINO(I,J)*MDEGN                                      ERR 320
C                                                                       ERR 330
    2 FORMAT(1EA4/I2,3A4)                                               ERR 340
    4 FORMAT(2I1,3PE.0)                                                 ERR 350
    6 FORMAT(6E12.6)                                                    ERR 360
    8 FORMAT(' ACTUAL LABEL ON THE FILE WAS '3A4,' THE SEARCH LABEL WAS ERR 370
     8'3A4/'0JOB TERMINATED.')                                          ERR 380
C                                                                       ERR 390
      READ(IN,2)TITLE,KEY,LABEL                                         ERR 400
      PI= ACOS(-1.0)                                                    ERR 410
      CALL CORSZ(BIGX(5000))                                            ERR 420
C        IF KBT=0, A DATA SET SEARCH WILL BE MADE                       ERR 430
  100 READ(ITAPE)XCTL,XLAB,AS,CS,G,VOL,NSS,NA,NSCAT,NLP,MRANK,NPARA,NEP,ERR 440
     1TEMP,ICENT,(((BE(I,J,K),I=1,3),J=1,3),K=1,NSS),((TE(I,J),I=1,3),J=ERR 450
     21,NSS),((TP(I,J),J=1,3),I=1,NLP),(IPARA(I),I=1,NPARA),((TEMP1(I,J)ERR 460
     3,I=1,9),J=1,NSCAT),(TEMP(I),I=1,NEP),TEOP,(JEP(I),I=1,NSP),NREF  ERR 470
     4,NPAR,(PAR(I),(PARNAM(J,I),J=1,2),KPAR(I),KRPAR(I),I=1,NPAR),PISGNERR 480
     5,IPSGN,TEMP4,ITEMP4                                               ERR 490
      READ(ITAPE)(((TEMP2(I,J),I=1,2),J=1,NSCAT),IEXT,EXTIN,((ATOM(J,I),JERR 500
     1=1,2),ITP(I),(ZT(J,I),J=1,5),(XYZ(J,I),J=1,3),(BETA(J,I),J=1,6),  ERR 510
     2(CR(J,I),J=1,10),(DR(J,I),J=1,15),I=1,NA),MDEGN                   ERR 520
     3,((CNETR(I),I=1,MDEGN)                                            ERR 530
      IF(NPAR.EQ.0)GO TO 110                                            ERR 540
C     FOR RIGID BODY REFINEMENTS APPLY LIBRATION CORRECTIONS TO XYZS.    ERR 550
      CALL LIBCOR(XYZ,PAR,KPAR,NPAR,G,NA,MDEGN,XNSTR)                   ERR 560
      J=NPAR                                                            ERR 570
      NPAR=0                                                            ERR 580
      DO 105 I=1,J                                                      ERR 590
      IF(KRPAR(I).GT.0)NPAR=NPAR+1                                      ERR 600
  105 CONTINUE                                                          ERR 610
C        READ AND DISCARD STRUCTURE FACTOR RECORDS                      ERR 620
  110 READ(ITAPE)ITEMP                                                  ERR 630
      IF(ITEMP)120,110,120                                              ERR 640
C        READ VARIANCE-COVARIANCE MATRIX                                ERR 650
  120 READ(ITAPE)NDEGN,(CNSTR(I),I=1,MDEGN)                             ERR 660
      NMXL=(MRANK*(MRANK+1))/2*MDEGN                                    ERR 670
      CALL CERSZ(CV(NMXL))                                              ERR 680
      J=MDEGN+1                                                         ERR 690
      READ(ITAPE)(CV(I),I=J,NMXL)                                       ERR 700
C        CHECK IF DESIRED DATA SET READ                                 ERR 710
      IF(KEY)140,130,140                                                ERR 720
  130 DO 135 I=1,3                                                      ERR 730
      IF(XLAB(I)-LABEL(I))136,135,136                                   ERR 740
  135 CONTINUE                                                          ERR 750
      GO TO 140                                                         ERR 760
C        INCORRECT OR UNEXPECTED LABEL                                  ERR 770
  136 WRITE(IOUT,8)XLAB,LABEL                                           ERR 780
      CALL EXIT                                                         ERR 790
  140 READ(IN,4)MEEDAN,MELVIE,DLIMIT                                   ERR 800
      IF(FISGN.GT.0)GO TO 142                                           ERR 810
C        GET NEGATIVE OF POSITIONS                                      ERR 820
      DO 141 I=1,NA                                                     ERR 830
      DO 141 J=1,3                                                      ERR 840
  141 XYZ(J,I)=-XYZ(J,I)                                                ERR 850
C        CALCULATE NUMBER OF MATRIX ROWS FOR EXTINCTION CORR.           ERR 860
  142 IEXT=NPAR                                                         ERR 870
      DO 143 I=1,6                                                      ERR 880
      IF(ITEMP4(I).GT.0)IEXT=IEXT+1                                     ERR 890
  143 CONTINUE                                                          ERR 900
C        SKIP NEXT SECTION IF NE ELLIPSOIDS CALCULATED                  ERR 910
      IF(MELVIE.EQ.0) GO TO 239                                         ERR 920
C        CONVERT PARAMETER SELECTION INFORMATION TO MATRIX ROW POINTER  ERR 930
      IJ=NESF+IEXT                                                      ERR 940
      DO 160 I=1,NPARA                                                  ERR 950
      IF(IPARA(I))150,160,150                                           ERR 960
  150 IJ=IJ+1                                                           ERR 970
      IPARA(I)=IJ                                                       ERR 980
  160 CONTINUE                                                          ERR 990
C        CRACK OUT ANISOTROPIC TEMPERATURE FACTOR VAR-COV MATRICES      ERR 1000
      IL=5                                                              ERR 1010
      DO 232 I=1,NA                                                     ERR 1020
      IF(ITF(I))170,230,170                                             ERR 1030
C        CLEAR ARRAY                                                    ERR 1040
  170 DO 180 J=1,6                                                      ERR 1050
      IPNT(J,I)=0                                                       ERR 1060
      DO 180 K=1,6                                                      ERR 1070
  180 SIGE(K,J,I)=0.0                                                   ERR 1080
      IK=IL-1                                                           ERR 1090
      DO 220 J=1,6                                                      ERR 1100
C        COMPUTE NEW FOR THIS PARAMETER                                 ERR 1110
      IK=IK+1                                                           ERR 1120
      JA=IPARA(IK)                                                      ERR 1130
      IPNT(J,I)=JA                                                      ERR 1140
      IF(JA)190,220,190                                                 ERR 1150
```

```
  190 IJ=1K-1                                                          EER 1160
      KA=INDEXU(JA,JA)                                                 RRR 1170
      DO 210 K=J,6                                                     ERR 1180
      IJ=IJ+1                                                          EER 1190
      JB=IPARA(IJ)                                                     ERR 1200
      IP(JR)200,210,200                                                ERR 1210
  200 KB=KA+JR-JA                                                      ERR 1220
      SIGB(K,J,I)=CV(KB)                                               ERB 1230
      SIGH(J,K,1)=CV(KB)                                               ERR 1240
  210 CONTINUE                                                         ERR 1250
  220 CONTINUE                                                         ERR 1260
  230 IJ=ITP(I)                                                        ERR 1270
  232 IL=IL+NCNT(1J+1)+4                                               ERR 12R0
C        MODIPY SIGR IP NECESSARY                                      EER 1290
      IP(MDRGN.LE.0) GO TO 239                                         ERR 1300
C        LOOP THROUGH DEPENDENCIES                                     ERR 1310
      DO 235 1=1,MDEGN,2                                               ERR 1320
      IPAR=I4(CNSTR(I))                                                ERR 1330
      IP(IPAR.LT.5.OR.IPAR.GT.10)GO TO 235                             RRR 1340
C        DEP. PARAMETER 1S AN ANISO. COEP.                             RRR 1350
      IPAR=IPAR-4                                                      ERB 1360
      IAT=I3(CNSTR(I))                                                 RRR 1370
      J=I2(CNSTR(I))                                                   ERR 13R0
C     IS THE INORPENDENT PARAMETER A MEMBER.OF THE ANISOTROPIC         RRR 1390
C     TEMPERATURE FACTOR MATRIX FOR THR SAME ATOM? IP SO, THE          ERR 1400
C     COVARIANCE OF TRE INORPENDENT PARAMETER WITH THE DRPENDENT       ERR 1410
C     ONE MUST BE COMPUTED.                                            ERR 1420
      DO 1010 JJ=1,6                                                   ERR 1430
      JI=IPNT(JJ,IAT)                                                  ERR 1440
      IP(JI.EQ.0)GO TO 1010                                            ERR 1450
      1K=INDEXU(J,JI)                                                  ERR 1460
      SIGR(JJ,IPAR,IAT)=SIGR(JJ,IPAR,IAT)+CNSTR(I+1)*CV(IK)           EER 1470
      SIGR(IPAR,JJ,IAT)=SIGH(JJ,IPAR,IAT)                             ERR 14R0
 1010 CONTINUE                                                         ERR 1490
C     COMPUTE COVARIANCES OP DEPENDENT PARAMETERS WITH EACR OTHER.     ERR 1500
 1011 DO 1500 KK=1,MDEGN,2                                            EER 1510
      IP(I3(CNSTR(KK)).NE.IAT.OR.I4(CNSTR(XK)).LT.5.OR.I4(CNSTR(KK)).GT. ERR 1520
     110)GO TO 1500                                                    ERR 1530
      IJPAR=I4(CNSTR(KK))-4                                            ERR 1540
      L=I2(CNSTR(KK))                                                  ERR 1550
      1K=INDEXU(J,L)                                                   ERR 1560
 1200 EIGR(IPAR,IJPAR,IAT)=SIGB(IPAR,IJPAR,IAT)*CNSTR(I+1)*CNSTR(KK+1)* ERR 1570
     1                      CV(IK)                                     ERE 15R0
      IP(IPAR.NE.IJPAR)SIGR(IJPAR,IPAR,IAT)=SIGE(IPAR,IJPAR,IAT)       ERR 1590
 1500 CONTINOE                                                         RRR 1600
  235 CONTINUE                                                         ERE 1610
  239 LXYZ=0                                                           ERE 1620
C        SKIP NEXT SECTION IF NO BOND PUNCTIONS                        ERR 1630
      IP(MEGDAN.EQ.0) GO TO 790                                        ERR 1640
C        START UNPACKING POSITION MATRIX                               RRR 1650
      IPAR=RRSP+IEXT+1                                                 ERR 1660
C        KA WILL RR TEE INDEX OF THR NEXT LOCATION IN THE UNPACKED ARRAYERR 1670
C        INCR IS THE STEP NEEDRO TO AOVANCE KA AN ENTIRE ROW           ERR 16R0
      KA=INOEXD(IPAR,IPAR)                                             ERR 1690
      INCR=MRANK*1-IPAR                                                EER 1700
C        KE WILL INDEX THE NEXT LOCATION IN THE PACKED MATRIX          ERR 1710
      KE=NMIL+1                                                        ERR 1720
      LXYZ=(3+NA*(3+NA+1))/2                                           ERR 1730
      I=LXYZ+NMXL                                                      ERR 1740
C        REQDEST SUPPICIENT CORE FOR XYZ VAR-COV, THEN CLEAR           ERR 1750
      CALL CORSZ(CV(I))                                                ERR 1760
      DO 240 J=ER,I                                                    ERR 1770
  240 CV(J)=0.0                                                        ERR 1780
      NP=1                                                             ERR 1790
      KB=KE-1                                                          ERR 1E00
      IPAR=0                                                           ERR 1810
C        LOOP THROUGH ATOMS                                            ERR 1820
      DO 540 I=1,NA                                                    ERR 1E30
C        ADVANCE PARAMETRR COUNTER FOR OCCUPANCY                       ERR 1840
      IPAR=IPAR+1                                                      ERR 1E50
      J=ITP(I)                                                         ERR 1E60
      NBI=NCNT(J+1)                                                    ERR 1870
      IP(IPARA(IPAR),EQ.0)GO TO 250                                    ERR 1EE0
C        STEP OVER VAR-COV ROW FOR OCCUPANCY                           ERR 1890
      KA=KA+INCR                                                       ERR 1900
      INCR=INCR-1                                                      ERR 1910
C        LOOP THROUGH POSITIONS                                        ERR 1920
  250 DO 4E0 J=1,3                                                     ERR 1930
      KB=KB+NP                                                         ERR 1940
      IP(PISGN.LT.0.0)XYZ(J,I)=-XYZ(J,I)                               ERR 1950
      IPAR=IPAR+1                                                      ERR 1960
      JPNT(NP)=IPARA(IPAR)                                             ERR 1970
C        TEST IF POSITION REFINED                                      ERR 19E0
      IP(IPARA(IPAR))260,4E0,260                                       ERR 1990
C        COPY VARIANCE OF POSITION                                     ERR 2000
  260 CV(KE)=CV(KA)                                                    ERR 2010
      INCR=INCR-1                                                      ERR 2020
      KA=KA+I                                                          ERR 2030
      KKPAR=IPAR                                                       ERR 2040
      JP=NP                                                            ERR 2050
      KC=KE                                                            ERR 2060
      IP(J-3)2R0,340,2E0                                               ERR 2070
  2E0 ITEMP=J+1                                                        ERR 20E0
C        LOOP THROUGH OTHER POSITIONS FOR ATOM I                       ERR 2090
      DO 300 K=ITEMP,3                                                 ERR 2100
      KKPAR=KKPAR+1                                                    ERR 2110
      KC=KC+JP                                                         ERR 2120
      JP=JP+1                                                          ERR 2130
      IP(IPARA(KKPAR))290,300,290                                      ERR 2140
C        COPY COVARIANCE ELRMENT                                       ERR 2150
  290 CV(KC)=CV(KA)                                                    ERR 2160
      KA=KA+1                                                          ERR 2170
  300 CONTINDE                                                         ERR 2180
C        LOOP TERODGH TEMPERATURE FACTOR(E) FOR ATOM I                 ERR 2190
  340 DO 360 K=1,NBI                                                   ERR 2200
      KKPAR=KKPAR+1                                                    ERR 2210
      IP(IPARA(KKPAR))350,360,350                                      ERR 2220
  350 KA=KA+1                                                          ERR 2230
  360 CONTINDE                                                         ERR 2240
      IP(I-NA)370,4E0,370                                              ERR 2250
  370 ITEMP=I+1                                                        ERR 2260
C        LOOP THROOGR TEE OTHER ATOMS                                  ERR 2270
      DO 470 K=ITEMP,NA                                                ERR 2280
      KKPAR=KKPAR+1                                                    ERR 2290
      IP(IPARA(KKPAR))3E0,390,3E0                                      ERR 2300
C        OCCUP REFINED                                                 ERR 2310
```

```
  380 KA=KA+1                                                          88R 2320
C         LOOP THROUGH POSITIONS FOR ATOM K                          ERR 2330
  390 DO 410 L=1,3                                                     8RR 2340
      KC=KC+JP                                                         8RR 2350
      JP=JP+1                                                          ERR 2360
      KKPAR=KKPAR+1                                                    8RR 2370
      IF(IPARA(KKPAR))400,410,400                                      8RR 2380
C         COPY COVARIANCE BLOCK                                      8RR 2390
  400 CV(KC)=CV(KA)                                                    ERR 2400
      KA=KA+1                                                          8RR 2410
  410 CONTINUE                                                         8RR 2420
      L=ITP(K)                                                         ERR 2430
      N8=NCNT(L+1)                                                     ERR 2440
C         LOOP THROUGH TEMPERATURE FACTOR(S)                         8RR 2450
  440 DO 460 L=1,N8                                                    8RR 2460
      KKPAR=KKPAR+1                                                    8RR 2470
      IF(IPARA(KKPAR))450,460,450                                      8RR 2480
C         ADVANCE COUNTER                                            8RR 2490
  450 KA=KA+1                                                          ERR 2500
  460 CONTINUE                                                         8RR 2510
  470 CONTINUE                                                         ERR 2520
  480 NP=NP+1                                                          8RR 2530
C         LOOP THROUGH VAR-COV ROWS FOR ATOM I TEMP FACTORS          ERR 2540
      DO 530 J=1,N8I                                                   8RR 2550
      IPAR=IPAR+1                                                      ERR 2560
      IF(IPARA(IPAR))520,530,520                                       ERR 2570
C       ADVANCE INDEX AND REDUCE INCREMENT                           8RR 2580
  520 KA=KA+INCR                                                       8RR 2590
      INCR=INCR-1                                                      8RR 2600
  530 CONTINUE                                                         ERR 2610
  540 CONTINUE                                                         ERR 2620
C         ATOM VARIANCE-COVARIANCE MATRIX IS EXTRACTED. NOW MODIFY FOR ERR 2630
C           CONSTRAINTS                                              8RR 2640
      IF(MDEGN.EQ.0) GO TO 570                                         8RR 2650
      N8=3*NA                                                          8RR 2660
      DO 565 I=1,MDEGN,2                                               8RR 2670
      IPAR=I4(CNSTR(I))                                                ERR 2680
      IF(IPAR.LT.2.OR.IPAR.GT.4)GO TO 565                              ERR 2690
      IAT=I3(CNSTR(I))                                                 ERR 2700
      J=I2(CNSTR(I))                                                   8RR 2710
      IPAR=3*(IAT-1)+IPAR-1                                            8RR 2720
      DO 560 K=1,N8                                                    ERR 2730
      IF(JPNT(K).EQ.0)GO TO 560                                        ERR 2740
C     COMPUTE COVARIANCE OF POSITION WITH A REFINED POSITION.          ERR 2750
      KB=INDEXL(K,IPAR)                                                ERR 2760
      KA=INDEXD(J,JPNT(K))                                             8RR 2770
      CV(KB)=CV(KB)+CV(KA)*CNSDON(I+1)                                 ERR 2780
  560 CONTINUE                                                         ERR 2790
C     COMPUTE COVARIANCES OF UNREFINED POSITIONS WITH EACH OTHER.      ERR 2800
      DO 563 KK=1,MDEGN,2                                              ERR 2810
      IJPAR=I4(CNSTR(KK))                                              ERR 2820
      IF(IJPAR.LT.2.OR.IJPAR.GT.4)GO TO 563                            ERR 2830
      JAT=I3(CNSTR(KK))                                                ERR 2840
      L=I2(CNSTR(KK))                                                  ERR 2850
      IJPAR=3*(JAT-1)+IJPAR-1                                          ERR 2860
      KA=INDEXO(J,L)                                                   ERR 2870
      KB=INDEXL(IPAR,IJPAR)                                            ERR 2880
      CV(KB)=CV(KB)+CNSDON(I+1)*CNSDGN(KK+1)*CV(KA)                    ERR 2890
  563 CONTINUE                                                         ERR 2900
  565 CONTINUE                                                         ERR 2910
C         MATRIX NOW TRANSFORMED, COPY TO LOW CORE                   ERR 2920
  570 KB=NMXL+1                                                        ERR 2930
      DO 580 I=1,LKYZ                                                  ERR 2940
      SIGX(I)=CV(KB)                                                   ERR 2950
  580 KB=KB+1                                                          ERR 2950
C         CONTRACT MATRIX TO TRUE MINIMUM REQUIRED                   ERR 2970
  790 CALL CORSZ(SIGX(LIYZ))                                           ERR 2980
C         READ AND CONVERT CELL ERROR MATRIX                         ERR 2990
      READ(IN,6)((SIGA(J,I),J=I,6),I=1,6)                              ERR 3000
C         GET REAL SPACE CELL QUANTITIES                             ERR 3010
      IERR=1                                                           ERR 3020
      DO 800 I=1,3                                                     ERR 3030
  800 A(I)=SQRT(G(I,I))                                                ERR 3040
      ANG(1)=G(2,3)/(A(2)*A(3))                                        ERR 3050
      ANG(2)=G(1,3)/(A(1)*A(3))                                        ERR 3060
      ANG(3)=G(1,2)/(A(1)*A(2))                                        ERR 3070
      DO 810 I=1,3                                                     ERR 3080
  810 S(I)=SQRT(1.0-ANG(I)**2)                                         ERR 3090
      DO 840 I=1,6                                                     ERR 3100
      DO 840 J=1,6                                                     ERR 3110
      IF(I.LE.3.AND.J.LE.3) GO TO 840                                  ERR 3120
      IF(I-3)820,820,830                                               ERR 3130
  820 SIGA(J,I)=SIGA(J,I)*S(I-3)                                       ERR 3140
      GO TO 840                                                        ERR 3150
  830 SIGA(J,I)=SIGA(J,I)*S(I-3)*S(J-3)                                ERR 3160
  840 SIGA(I,J)=SIGA(J,I)                                              ERR 3170
  850 IF(ICENT)880,860,880                                             ERR 3180
C         EXPAND SYMMETRY OPERATORS FOR A CENTRIC CRYSTAL            ERR 3190
  860 ITEMP=NSS                                                        ERR 3200
      DO 870 I=1,N8S                                                   ERR 3210
      ITEMP=ITEMP+1                                                    ERR 3220
      DO 870 J=1,3                                                     ERR 3230
      TS(J,ITEMP)=-TS(J,I)                                             ERR 3240
      DO 870 K=1,3                                                     ERR 3250
  870 RS(K,J,ITEMP)=-RS(K,J,I)                                         ERR 3260
      N8S=ITEMP                                                        ERR 3270
  880 IF(MBODAN)890,900,890                                            ERR 3280
C         COMPUTE SOME FUNCTIONS                                     ERR 3290
  890 CALL SODAN(MBODAN)                                               ERR 3300
  900 IF(MSLVI8)910,920,910                                            ERR 3310
C         COMPUTE TRIGONAL FUNCTIONS                                 ERR 3320
  910 CALL SLVI8                                                       ERR 3330
C         END OF JOB                                                 ERR 3340
  920 CALL EXIT                                                        ERR 3350
      END                                                              ERR 3360
```

```
      SUBROUTINE BODAN(IBOD)                                            BODN  10
                                                                        BODN  20
C THIS ROUTINE REVISED AND/OR RESEQUENCED ON OCT. 20, 1966              BODN  30
                                                                        BODN  40
C         BOND DISTANCE AND ANGLE SUBPROGRAM WITH ERRORS                BODN  50
                                                                        BODN  60
      COMMON/ABLE/XYZ(3,60),BETA(6,60),A(3),ANC(3),AS(3),CB(3),ATOM(2,  BODN  70
     1 60),SIGA(6,6),SIGB(6,6,60),RS(3,3,40),T8(3,48),TP(4,3),ITP(60),  BODN  80
     2 NA,NBB,VOL,DLIMIT(3),PI,G(3,3),NLP,IERR,TITLE(10),CORR(1)        BODN  90
      COMMON/BAKER/XYZP(3,1000),IATP(1000)                              BOON 100
                                                                        BODN 110
      DIMENSION XP(3),YYP(3),XXP(3),SIGX(6,6),DLDX(6),DSDX(9,2),YP(3),  BODN 120
     1 DLDA(6,2),BIGY(9,9),DNDA(6),DNDX(9),DTHDA(6),DTHDX(9),XZP(3),    BOON 130
     2 X(3,3),NXX(6),DXX(3),DXY(3),DXZ(3),DBD(3)                        BODN 140
                                                                        BOON 150
      EQUIVALENCE (DLDX,DSDX),(NXX(1),NXL),(NXX(2),NYL),(NXX(3),NZL)    BODN 160
                                                                        BOON 170
      DATA IOUT/6/                                                      BODN 180
                                                                        BODN 190
 1002 FORMAT(46X,2A3,4P10.5,10X,I8)                                     BODN 200
 8999 FORMAT(31H THE ATOM ARRAY IS OVERFLOWING. ///15X,54HREDUCE THE LIMBODN 210
     1ITS OF THE BOND DISTANCES AND TRY AGAIN. )                        BOON 220
 9000 FORMAT(1H1,18A4/25H  BOND DISTANCES BETWEEN ,F5.2,5H AND ,F5.2,10HBODN 230
     1 ANGSTROMS//10X,6HATOM 1,7XIHX,9X,1HY,9X,1HZ,2X,6HATOM 2,7X,1HX,9XBODN 240
     2,1HY,9X,1HZ,4X,8HDISTANCE,3X5HERROR,4X,6HNUMBER//10X2A3,3P10.5,63XBODN 250
     3,I8)                                                              BODN 260
 9001 FORMAT(10X2A3,3P10.5,56XI8)                                       BODN 270
 9002 FORMAT(1H1,1BA4//69H BOND ANGLES AND ERRORS FOR ATOMS WITH INTERATBODN 280
     1OMIC DISTANCES BETWEEN F6.2,5H AND F6.2,11H ANGSTROMS./102H  THE ABODN 290
     2NGLE IS ATOM I=CENTRAL ATOM-ATOM 2.  THE NUMBER POINTS TO AN ENTRYBODN 300
     3 IN THE BOND DISTANCE OUTPUT.)                                    BODN 310
 9003 FORMAT(15X,2A3,I6,I2X,2A3,I6,F17.2,F10.2)                         BODN 320
 9005 FORMAT(46X2A3,5F10.5,I6)                                          BODN 330
 9010 FORMAT(///10X,12HCENTRAL ATOM,5X2A3,3PI0.5//15X,6HATOM 1,2X,6HNUMBBODN 340
     1ER,10X6HATOM 2,2X,6HNUMBER,10X,5HANCLE,5X,5HERROR/)               BODN 350
                                                                        BODN 360
      WRITE(IOUT,9000)TITLE,DLIMIT(1),DLIMIT(3),((ATOM(J,I),J=1,2),(XYZ(JBODN 370
     1,I),J=1,3)                                                        BODN 380
C         BET UP PRELIMINARY INFORMATION FOR SEARCHING FOR ATOMS        BODN 390
      FROD=A(1)*A(2)*A(3)                                               BODN 400
      DO 61 J=1,3                                                       BODN 410
      XYZP(J,1)=XYZ(J,I)                                                BODN 420
      DBD(J)=PROD/A(J)                                                  BODN 430
   61 DXX(J)=ABS(J)*DLIMIT(3)                                           BODN 440
C           INITIALIZE AND PINISH STORING ATOM 1 IN THE ASYMMETRIC UNIT BODN 450
C                 INTO THE GENERATED ATOM LIST                          BODN 460
C         THE ELEMENTS OF THE ARRAY IATP ARE 64*ATOM NUMBER + SYMMETRY NOBODN 470
      IATP(1)=65                                                        BODN 480
      NAA=3*NA                                                          BODN 490
      DO 37 I=1,9                                                       BODN 500
      DO 37 J=1,2                                                       BODN 510
   37 DBDX(I,J)=0                                                       BODN 520
      NATOM=1                                                           BODN 530
      SN=DLIMIT(1)**2                                                   BODN 540
      SX=DLIMIT(3)**2                                                   BODN S50
C           LOOP THROUGH THE ATOMS IN THE ASYMMETRIC UNIT               BODN 560
      DO 1 II=1,NA                                                      BODN 570
C           SET UP SEARCH ARRAYS FOR THIS ATOM                          BOON 580
      DO 62 J=1,3                                                       BODN 590
      DXY(J)=XYZ(J,II)+DXX(J)                                           BODN 600
   62 OXZ(J)=XYZ(J,II)-DXX(J)                                           BODN 610
C           EXTRACT SELP VARIANCE-COVARIANCE MATRIX                     BODN 620
      IJ=3*II-2                                                         BODN 630
      JI=((IJ+1)*IJ)/2                                                  BOON 640
      DO 3 N=1,3                                                        BOON 650
      L=JI                                                              BOON 660
      SIGX(N,1)=CORR(L)                                                 BOON 670
      IF(N.EQ.1)GO TO 3                                                 BOON 680
      DO 2 J=2,N                                                        BODN 690
      L=L+1                                                             BODN 700
    2 SIGX(N,J)=CORR(L)                                                 BODN 710
    3 JI=JI+IJ+N-1                                                      BOON 720
C           IF FIRST ATOM, JUMP AROUND NEXT COOR                        BODN 730
      IF(II.EQ.1)GO TO 67                                               BODN 740
C           SEARCH FOR PREVIOUS GENERATION OF THIS ATOM                 BODN 750
   68 DO 115 I=1,NATOM                                                  BOON 760
      IF(IATP(I)/64.EQ.II)GO TO 116                                     BOON 770
      GO TO 115                                                         BODN 780
  116 DO 117 J=1,3                                                      BODN 790
      IF(ABS (XYZP(J,I)-XYZ(J,II))-0.00002)117,117,115                  BOON 800
  117 CONTINUE                                                          BODN 810
C           ABOVE LOOP COMPLETED IF NOT NEW POSITION                    BODN 820
      IJ=I                                                              BODN 830
      GO TO 118                                                         BOON 840
  115 CONTINUE                                                          BODN 850
C           ADD ATOM TO LIST                                            BOON 860
      IJ=NATOM+1                                                        BODN 870
      NATOM=IJ                                                          BOON 880
      IATP(NATOM)=II*64+1                                               BODN 890
      DO 119 J=1,3                                                      BOON 900
  119 XYZP(J,NATOM)=XYZ(J,II)                                           BODN 910
C           PRINT READING FOR ATOM IN ASYMMETRIC UNIT                   BODN 920
  118 WRITE(IOUT,9001)(ATOM(J,II),J=1,2),(XYZ(J,II),J=1,3),IJ           BOON 930
C           LOOP THROUGH GENERATED LIST TO FIND DISTANCES               BODN 940
   70 DO 71 J=1,NATOM                                                   BODN 950
      B=0                                                               BODN 960
      DO 72 K=1,3                                                       BODN 970
   72 XP(K)=XYZ(K,II)-XYZP(K,J)                                         BODN 980
      DO 73 N=1,3                                                       BODN 990
      XZP(N)=0                                                          BODN1000
      DO 74 K=1,3                                                       BODN1010
   74 XZP(N)=XZP(N)+XP(K)*G(K,N)                                        BODN1020
   73 S=S+XZP(N)*XP(N)                                                  BODN1030
      IP((S-SX)*(SN-S))71,71,75                                         BODN1040
   75 S=SQRT (S)                                                        BODN1050
      IP(IERR.EQ.0)GO TO 1001                                           BODN1060
C           COMPUTE DERIVATIVES                                         BODN1070
      L=IATP(J)-64*(IATP(J)/64)                                         BODN1080
      DLDA(4,1)=DBD(1)*XP(2)*XP(3)                                      BODN1090
      DLDA(5,1)=DBD(2)*XP(1)*XP(3)                                      BODN1100
      DLDA(6,1)=DBD(3)*XP(1)*XP(2)                                      BODN1110
      DO 76 K=1,3                                                       BODN1120
      DLDA(K,1)=XP(K)*XZP(K)/A(K)                                       BODN1130
      DLDX(K)=XZP(K)                                                    BODN1140
      DLDX(K+3)=0                                                       BODN1150
```

```
      DO 76 N=1,3                                                       BODN1160
   76 OLDX(K*3)=DLDX(K*3)-XZP(N)*RS(N,K,L)                              BODN1170
C          FILL IN SELP VARIANCE-COVARIANCE MATRIX                      BODN1180
      IJ=3*(IATP(J)/64)-2                                               BODN1190
      JI=((IJ+1)*IJ)/2                                                  BODN1200
      DO 77 K=1,3                                                       BODN1210
      L=JI                                                              BODN1220
      SIGX(K*3,4)=CORR(L)                                               BODN1230
      IP(K.EQ.1)GO TO 77                                                BODN1240
      DO 76 N=2,K                                                       BODN1250
      L=L+1                                                             BODN1260
   76 SIGX(K*3,N*3)=CORR(L)                                             BODN1270
   77 JI=JI+IJ*K-1                                                      BODN1280
C          GET CROSS TERMS FOR VARIANCE-COVARIANCE MATRIX               BODN1290
      CALL CORRI(SIGX,6,II,IATP(J)/64,CORR,1,4,NA)                      BODN1300
C          COMPUTE VARIANCE OF DISTANCE                                 BODN1310
      CALL VARIAN(DS,DLDX,DLDA,SIGX,SIGA,1)                             BODN1320
      DS=DS/S                                                           BODN1330
      IJ=IATP(J)/64                                                     BODN1340
      WRITE(IOUT,9005)(ATOM(K,IJ),K=1,2),(XYZP(K,J),K=1,3),S,DS,J       BODN1350
      GO TO 71                                                          BODN1360
C          PRINT DISTANCE FOR CALCULATION WITHOUT ERRORS                BODN1370
 1001 IJ=IATP(J)/64                                                     BODN1380
      WRITE(IOUT,1002)(ATOM(K,IJ),K=1,2),(XYZP(K,J),K=1,3),S,J          BODN1390
   71 CONTINUE                                                          BODN1400
                                                                        BODN1410
                                                                        BODN1420
C          LOOP THROUGH THE ASYMMETRIC UNIT TO GENERATE ATOMS           BODN1430
   67 DO I JJ= 1,NA                                                     BODN1440
C          LOOP THROUGH THE SYMMETRY OPERATIONS                         BODN1450
      DO 12 J=1,NSS                                                     BODN1460
C          GENERATE TRANSFORMED POSITION                                BODN1470
      DO 13 L=1,3                                                       BODN1480
      XP(L)=TS(L,J)                                                     BODN1490
      DO 13 M=1,3                                                       BODN1500
   13 XP(L)=XP(L)+RS(L,M,J)*XYZ(M,JJ)                                   BODN1510
C          LOOP THROUGH LATTICE POINTS                                  BODN1520
      DO 12 M=1,NLP                                                     BODN1530
      DO 14 L=1,3                                                       BODN1540
   14 XXP(L)=XP(L)+TP(M,L)                                              BODN1550
C          COMPUTE TRANSLATION LIMITS                                   BODN1560
      DO 63 I=1,3                                                       BODN1570
      NXX(I*3)=3.0*DXY(I)-XXP(I)                                        BODN1580
   63 NXX(I)=3-IFIX (XXP(I)-DXZ(I))                                     BODN1590
      NKU=NXX(4)                                                        BODN1600
      NYU=NXX(5)                                                        BODN1610
      NZO=NXX(6)                                                        BODN1620
      DO 12 IA=NXL,NKU                                                  BODN1630
      XYP(1)=XXP(1)+FLOAT (IA-3)                                        BODN1640
      DO 12 IB=NYL,NYU                                                  BODN1650
      XYP(2)=XXP(2)+FLOAT (IB-3)                                        BODN1660
      DO 12 IC=NZL,NZU                                                  BODN1670
      XYP(3)=XXP(3)+FLOAT (IC-3)                                        BODN1680
C          CHECK FOR ATOM DUPLICATION                                   BODN1690
      DO 113 I=1,NATOM                                                  BODN1700
      IP(IATP(I)/64.EQ.JJ)GO TO 24                                      BODN1710
      GO TO 113                                                         BODN1720
   24 DO 114 L=1,3                                                      BODN1730
      IP(ABS (XTP(L)-XTZP(L,I))-0.00002)114,114,113                     BODN1740
  114 CONTINUE                                                          BODN1750
C          THROUGH HERE IF THIS ATOM HAS ALREADY BEEN GENERATED         BODN1760
      GO TO 12                                                          BODN1770
  113 CONTINUE                                                          BODN1780
C          COMPUTE DISTANCE                                             BODN1790
      S=0                                                               BODN1800
      DO 15 K=1,3                                                       BODN1810
   15 YP(K)=XTZ(K,II)-XTP(K)                                            BODN1820
      DO 17 M=1,3                                                       BODN1830
      XZP(N)=0                                                          BODN1840
      DO 16 K=1,3                                                       BODN1850
   16 XZP(N)=XZP(N)+YP(K)*O(K,N)                                        BODN1860
   17 S=S+XZP(N)*TP(N)                                                  BODN1870
C          CHECK IF DISTANCE WITHIN LIMITS                              BODN1880
      IP((E-EK)*(EN-S))12,12,16                                         BODN1890
C          DISTANCE OK, CHECK IF FROM IN GENERATED LIST                 BODN1900
   16 IP(NATOM-1000)19,20,20                                            BODN1910
C          ARRAYS OVERFLOWING,  PRINT MESSAGE AND TERMINATE             BODN1920
   20 WRITE(IOUT,6999)                                                  BODN1930
      RETURN                                                            BODN1940
C          ADD GENERATED ATOM TO LIST AND COMPUTE DERIVATIVES           BODN1950
   19 NATOM=NATOM+1                                                     BODN1960
      IATP(NATOM)=JJ*64+J                                               BODN1970
      DLDA(4,1)=DBD(1)*TP(2)*YP(3)                                      BODN1980
      DLDA(5,1)=DBD(2)*TP(1)*YP(3)                                      BODN1990
      DLDA(6,1)=DBD(3)*TP(1)*YP(2)                                      BODN2000
      DO 21 K=1,3                                                       BODN2010
      XTZP(K,NATOM)=XTP(K)                                              BODN2020
      DLDX(K)=XZP(K)                                                    BODN2030
      DLDX(K*3)=0                                                       BODN2040
      DLDA(K,1)=TP(K)*XZP(K)/A(K)                                       BODN2050
      DO 21 N=1,3                                                       BODN2060
   21 DLDX(K*3)=DLDX(K*3)-XZP(N)*RS(N,K,J)                              BODN2070
      IJ=3*JJ-2                                                         BODN2080
      IP(IGRR.EQ.0)GO TO 1004                                           BODN2090
 1003 JI=((IJ+1)*IJ)/2                                                  BODN2100
      DO 23 K=1,3                                                       BODN2110
      L=JI                                                              BODN2120
      EIGX(K*3,4)=CORR(L)                                               BODN2130
      IP(K.EQ.1)GO TO 23                                                BODN2140
      DO 22 N=2,K                                                       BODN2150
      L=L+1                                                             BODN2160
   22 EIGX(K*3,N*3)=CORR(L)                                             BODN2170
   23 JI=JI+IJ*K-1                                                      BODN2180
C          GENERATE CROSS MATRIX ELEMENTS                               BODN2190
      CALL CORRI(EIGX,6,II,JJ,CORR,1,4,NA)                              BODN2200
      S=SQRT (E)                                                        BODN2210
      CALL VARIAN(DG,DLDX,DLDA,EIGX,EIGA,I)                             BODN2220
      DE=DS/G                                                           BODN2230
      WRITE(IOUT,9005)(ATOM(M,JJ),M=1,2),XYP,S,DS,NATOM                 BODN2240
      GO TO 12                                                          BODN2250
 1004 G=SQRT (E)                                                        BODN2260
C          PRINT DISTANCE FOR NO ERRORS CALCULATED                      BODN2270
      WRITE(IOUT,1002)(ATOM(M,JJ),M=1,2),XYP,S,NATOM                    BODN2280
   12 CONTINUE                                                          BODN2290
    1 CONTINUE                                                          BODN2300
C          END OF BOND DISTANCE CALCULATION, RETURN IF NO ANGLES        BODN2310
```

```
      IF(IB60.EQ.1) RETURN                                              B0DN2320
      WRITE(IOUT,9002)TITLE,(DLIMIT(I),I=1,2)                           B0DN2330
C        SET UPPER LIMIT OF DISTANCE TO CONTACT DISTANCE                B0DN2340
      SX=DLIMIT(2)**2                                                   B0DN2350
C        LOOP THROUGH ASYMMETRIC UNIT                                   B0DN2360
      DO 28 I=1,NA                                                      B0DN2370
      IJ=3*I-2                                                          B0DN2380
      JI=((IJ+1)*IJ)/2                                                  B0DN2390
      WRITE(IOUT,9010)(ATOM(K,I),K=1,2),(XYZ(K,I),K=1,3)                B0DN2400
      IF(IBRR.ED.0)GO TO 1006                                           B0DN2410
C        GET SELF VAR-COV MATRIX                                        B0DN2420
      DO 124 N=1,3                                                      B0DN2430
      L=JI                                                              B0DN2440
      SIDY(N+3,4)=CORR(L)                                               B0DN2450
      IF(N.EQ.1)GO TO 124                                               B0DN2460
      DO 25 K=2,N                                                       B0DN2470
      L=L+1                                                             B0DN2480
   25 SIGY(N+3,K+3)=CORR(L)                                             B0DN2490
  124 JI=JI+IJ+N-1                                                      B0DN2500
C        OUTSIDE LOOP THROUGH GENERATED ATOM LIST                       B0DN2510
 1006 ITEMP=NATOM-1                                                     B0DN2520
      DO 29 J=1,ITEMP                                                   B0DN2530
      DO 30 N=1,3                                                       B0DN2540
   30 XP(N)=XYZP(N,J)-XYZ(N,I)                                          B0DN2550
C        COMPUTE AND CHECK DISTANCE                                     B0DN2560
      S=0                                                               B0DN2570
      DO 31 N=1,3                                                       B0DN2580
      XXP(N)=0                                                          B0DN2590
      DO 32 K=1,3                                                       B0DN2600
   32 XXP(N)=XXP(N)+XP(K)*G(K,N)                                        B0DN2610
   31 S=S+XXP(N)*XP(N)                                                  B0DN2620
      IF((S-SX)*(SN-S))29,29,33                                         B0DN2630
   33 S=SQRT (S)                                                        B0DN2640
      NAT=IATP(J)/64                                                    B0DN2650
      L    =IATP(J)-64*NAT                                              B0DN2660
      LI=L                                                              B0DN2670
      IF(IERR.EQ.0)GO TO 1008                                           B0DN2680
C        COMPUTE DERIVATIVES                                            B0DN2690
      DLDA(4,1)=DBO(1)*XP(2)*XP(3)/S                                    B0DN2700
      DLDA(5,I)=DBO(2)*XP(I)*XP(3)/S                                    B0DN2710
      DLDA(6,I)=DBO(3)*XP(1)*XP(2)/S                                    B0DN2720
      DO 34 N=1,3                                                       B0DN2730
      DSDX(N+3,1)=-XXP(N)/S                                             B0DN2740
      DSDX(N,1)=0.0                                                     B0DN2750
      DLDA(N,I)=XP(N)*XXP(N)/(S*A(N))                                   B0DN2760
      DO 34 K=1,3                                                       B0DN2770
   34 DSDX(N,1)=DSDX(N,I)+XXP(K)*RS(K,N,L)/S                            B0DN2780
      IJ=3*NAT-2                                                        B0DN2790
      JI=((IJ+1)*IJ)/2                                                  B0DN2800
      DO 26 X=1,3                                                       B0DN2810
      L=JI                                                              B0DN2820
      SIGY(X,1)=CORR(L)                                                 B0DN2830
      IF(X.EQ.1)DO TO 26                                                B0DN2840
      DO 27 N=2,X                                                       B0DN2850
      L=L+1                                                             B0DN2860
   27 SIGY(K,N)=CORR(L)                                                 B0DN2870
   26 JI=JI+IJ+K-1                                                      B0DN2880
      CALL CORRI(SIGY,9,I,NAT,CORR,1,4,NA)                              B0DN2890
C        INNER LOOP THROUGH GENERATED ATOM LIST                         B0DN2900
 1008 IK=J+1                                                            B0DN291D
      DO 40 K=IK,NATOM                                                  B0DN2920
C        COMPUTE AND CHECK DISTANCE                                     B0DN2930
      S2=0                                                              B0DN2940
      DO 41 N=1,3                                                       B0DN295D
   41 YP(N)=XYZP(N,K)-XYZ(N,I)                                          B0DN2960
      DO 42 N=1,3                                                       B0DN297D
      XYP(N)=0                                                          B0DN2980
      DO 43M=1,3                                                        B0DN2990
   43 XYP(N)=XYP(N)+YP(M)*G(M,N)                                        B0DN3000
   42 S2=S2+XYP(N)*YP(N)                                                B0DN3010
      IP((S2-SX)*(5N-S2))40,40,44                                       B0DN3020
   44 S2=SQRT (S2)                                                      B0DN303D
      NTA=IATP(K)/64                                                    B0DN3040
      L2=  IATP(K)-NTA*64                                               B0DN3050
      L=L2                                                              B0DN3060
      IP(IERR.EQ.0)GO TO I010                                           B0DN3D7D
C        COMPDTE DERIVATIVBS                                            B0DN306D
      DLDA(4,2)=DBO(I)*YP(2)*YP(3)/S2                                   B0DN3090
      DLDA(5,2)=DBD(2)*YP(I)*YP(3)/S2                                   B0DN3I00
      DLDA(6,2)=DBD(3)*YP(1)*YP(2)/S2                                   B0DN3I10
      0O 45 M=1,3                                                       B0DN3I20
      05DX(M*3,2)=-XYP(M)/S2                                            B0DN3I30
      DSDX(M*6,2)=0                                                     B0DN3140
      DLDA(M,2)=YP(M)*XYP(M)/(S2*A(M))                                  B0DN3I50
      0O 45 N=1,3                                                       B0DN3I60
   45 DSDX(M*6,2)=DSOX(M*6,2)+XYP(N)*RS(N,M,L)/S2                       B0DN3170
C        COMPUTE COSINB TABTA                                           B0DN3180
 10I0 CTB=0                                                             B0DN3190
      DO 46 M=1,3                                                       B0DN3200
   46 CTH=CTH+XXP(M)*YP(M)                                              B0DN32I0
      CTH=CTB/(S*S2)                                                    B0DN3220
C        CHECK FOR COSINB OUT OF RANGB                                  B0DN323D
      IP(ABS (CTB)-I.0)47,47,48                                         B0DN3240
   48 CTH=SIGN (I.0,CTH)                                                B0DN3250
   47 STN=SQRT (1.0-CTB**2)                                             B0DN3260
      CHT=ACOS (CTH)*180.0/PI                                           B0DN327D
      IP(IBRR.NE.0) GO TO I0I2                                          B0DN3280
C        PRINT ANGLB FOR NO ERRORS                                      B0DN329D
      WRITE(IOUT,9003)(ATOM(M,NAT),M=I,2),J,(ATOM(M,NTA),M=I,2),K,CHT   B0DN3300
      GO TO 40                                                          B0DN3310
C        IP ANGLE CLOSE TO N*PI/2, SKIP BRROR CALCDLATION               B0DN3320
 1012 IP(ABS (BTH)-I.0B-6)64,64,65                                      B0DN333D
   64 DTH=0                                                             B0DN3340
      GO TO 66                                                          B0DN335D
C        EXTRACT VAR-COV MATRICBS AND COMPUTE DHRIVATIVES               B0DN336D
   65 IJ=3*NTA-2                                                        B0DN337D
      JI=((IJ+I)*IJ)/2                                                  B0DN338D
      DO 50 M=I,3                                                       B0DN339D
      L=JI                                                              B0DN34DD
      SIGY(M*6,7)=CORR(L)                                               B0DN341D
      IP(M.BO.1)DO TO 5D                                                B0DN3420
      DO 49 N=2,M                                                       B0DN343D
      L=L+1                                                             B0DN3440
   49 SIGY(M*6,N*6)=CORR(L)                                             B0DN3450
   50 JI=JI+IJ*N-1                                                      B0DN3460
      CALL CORRI(SIGY,9,I,NTA,CORR,4,7,NA)                              B0DN3470
```

```
      CALL CORR1( BIGT,9,NAT,NTA,CORR,1,7,NA )                          BODN3480
      DO 51 M=1,3                                                       BODN3490
      DO 51 N=1,3                                                       BODN3500
   51 X(M,N)=XP(M)*YP(M)                                                BODN3510
      DO 52 N=1,3                                                       BODN3520
      NN=MOD(N,3)+1                                                     BODN3530
      MN=MOD(N+1,3)+1                                                   BODN3540
C        DNDA IS THE DERIVATIVE OF THE NUMERATOR WRT THE CELL PARAMS.   BODN3550
      DNDA(N)=2.0*A(N)*X(N,N)+A(NN)*AMG(MN)*(X(N,NN)+X(NN,N))           BODN3560
     1 A(MN)*ANG(NN)*(X(M,MN)+X(MN,N))                                  BODN3570
      DNDA(N+3)=A(NN)*A(MN)*(X(NN,MN)+X(MN,NN))                         BODN3580
      DNDX(N+3)=-A(N)*2.*(XP(N)*YP(N))-G(N,NN)*(YP(NN)*XP(NN))-G(N,MN)* BODN3590
     1 (XP(MN)*YP(MN))                                                  BODN3600
      DNDX(N)=0.                                                        BODN3610
      DNDX(N+6)=DNDX(N)                                                 BODN3620
      DO 52 M=1,3                                                       BODN3630
      DO 52 L=1,3                                                       BODN3640
      DNDX(N)=DNDX(N)+G(M,L)*RS(L,M,L1)*YP(M)                           BODN3650
   52 DNDX(N+6)=DNDX(N+6)+G(M,L)*RS(L,N,L2)*XP(M)                       BODN3660
      BH=1.0/(S*H2)                                                     BODN3670
      DO 56 N=1,6                                                       BODN3680
      DTNDA(N)=BH*DNDA(N)-CTH*(DLDA(M,2)/S2*DLDA(N,1)/S)                BODN3690
   56 DTBDX(N)=SS*DNDX(M)-CTH*(DBDX(N,2)/B2*DBDX(N,1)/S)                BODN3700
      DTH=DTHDA(1)**2*SIGA(1,1)*DTHDX(1)**2*SIGY(1,1)                   BODN3710
      DO 79 N=7,9                                                       BODN3720
   79 DTHDX(N)=BS*DNDX(N)-CTH*(DSDX(N,2)/H2*D6DX(N,1)/B)                BODN3730
      DO 57 N=2,6                                                       BODN3740
      IJ=M-1                                                            BODN3750
      DTH=DTH*DTHDA(N)**2*BIGA(N,N)                                     BODN3760
      DO 57 M=1,IJ                                                      BODN3770
   57 DTH=DTH*2.0*DTHDA(N)*DTHDA(M)*BIGA(N,M)                           BODN3780
      DO 5H M=2,9                                                       BODN3790
      IJ=N-1                                                            BODN3800
      DTH=DTB*DTBDX(N)**2*SIGY(M,N)                                     BODN3810
      DO 5H M=1,IJ                                                      BODN3820
   5H DTH=DTB*DTBDX(N)*DTBDX(M)*SIGY(M,M)*2.0                           BODN3830
      IF(DTH)59,60,60                                                   BODN3840
   59 DTH=0                                                             BODN3850
   60 DTH=SQRT (DTH)/STH*180.0/PI                                       BODN3860
   66 WRITE( IOUT,9003 )(ATOM(M,MAT),M=1,2),J,(ATOM(M,NTA),M=1,2),K,CBT,DTBODN3870
     1H                                                                 BODN3880
   40 CONTINUE                                                          BODN3890
   29 CONTINUE                                                          BODN3900
   28 CONTINUE                                                          BODN3910
      RETURN                                                            BODN3920
      END                                                               BODN3930
```

```
      SUBROUTINE ELVIB                                                  ELVB  10
                                                                        ELVB  20
C THIS ROUTINE HAVISBD AND/OR HASEQUENCED ON OCT. 28, 1968              ELVB  30
                                                                        ELVB  40
C        THIS HOUTINE COMPUTES ELLIPSOIDS OF THERMAL MOTION AND THE     ELVB  50
C         ERRORS ASSOCIATED WITH THE PARAMETERS OF THESE ELLIPSOIDS     ELVB  60
                                                                        ELVB  70
      COMMON/ABLE/XYZ(3,60),BETA(6,60),A(3),ANG(3),AS(3),CS(3),ATOM(2,  ELVB  80
     1 60),SIGA(6,6),SIGB(6,6,60),RS(3,3,48),TS(3,48),TP(4,3),ITP(60),  ELVB  90
     2 NA,NSS,V6L,DLIMIT(3),PI,G(3,3),NLP,IBRH,TITLE(18),CORR(1)        ELVB 100
                                                                        ELVB 110
      DIMENSION DDDC(6,3,3),D00B(6,3,3),DTDA(6),DTDB(6),DSDA(6),DH0B(6),ELVB 120
     1 DDETOA(6),DOETDB(6),0L0A(6),0LDB(6),DQOA(6,3),0QDB(6,3),         ELVB 130
     2 DBQDA(6),B(3,3),D(3,3),R66T(3),DBQ0B(6),DASDA(6,6)               ELVB 140
     3 ,THETA(3),STH(3),DTH0B(6,3),DTHDA(6,3),DXDA(6,3,3),DXDB(6,3,3),  ELVB 150
     4 X(3,3),IAXIS(3),QI(3)                                            ELVB 160
                                                                        ELVB 170
      DATA IAXIS,IOUT/1HA,1HB,1HC,6/                                    ELVB 180
                                                                        ELVB 190
 1013 FORMAT( 1H1,18A4/11H ELLIPSOIDS)                                  ELVB 200
 1020 FORMAT(/10X,5HAXIS ,I1/18X,13HRMS AMPLITUDE,P10.6,16X,9HANGSTROMS)ELVB 210
 1025 FORMAT( 14X,15HANGLE WITH THE ,A1,5H AXIS,P14.2,16X,7HDEGREES)    ELVB 220
 9002 FORMAT(///10X2A3,25H IS NON POSITIVE DEFINITE)                    ELVB 230
 9003 FORMAT(///5X,27HELLIPSOID OF VIBRATION FOR 2A3//20X,9HPARAMETER,15ELVB 240
     1X,5HVALUE,10X,5HERROR//13X22HEQUIVALENT ISOTROPIC B,P14.3,P15.3)  ELVB 250
 9004 FORMAT(/10X,5HAXIS ,I1/18X,13HRMS AMPLITUDE,P10.6,P15.6,10H ANGSTRELVB 260
     1OMS )                                                             ELVB 270
 9005 FORMAT(14X,15HANGLE WITH THE ,A1,5H AXIS,P14.2,P15.2,6H DEGREES)  ELVB 280
                                                                        ELVB 290
      TWOPI=2.0*PI                                                      ELVB 300
      TPC=0.5/PI**2                                                     ELVB 310
      IF(IERR.NE.0)GO TO 1001                                           ELVB 320
      WRITE(IOUT,1013)TITLE                                             ELVB 330
      GO TO 1012                                                        ELVB 340
C        GET DERIVATIVES OF RECIPROCAL CELL PARAMS WRT DIRECT PARAMS    ELVB 350
 1001 CALL CELLD(DASDA,A,ANG,AS,CS,V6L)                                 ELVB 360
      WRITE(IOUT,9001)TITLE                                             ELVB 370
 9001 FORMAT(1H1,18A4//22H ELLIPSOIDS AND ERRORS)                       ELVB 380
C        MATRIX D IS THE PRODUCT GB                                     ELVB 390
C         B IS THE MATRIX OF ANISOTROPIC COEFFICIENTS                   ELVB 400
C         G IS THE REAL SPACE METRIC TENSOR                             ELVB 410
C         0DDB(I,J,K) IS DERIVATIVE OF D(I,J) WRT BETA(K)               ELVB 420
C         DDDC(I,J,K) IS DERIVATIVE OF D(I,J) WRT CELL(K)               ELVB 430
      DO 120 I=1,3                                                      ELVB 440
      DO 120 J=1,3                                                      ELVB 450
      DO 120 K=1,6                                                      ELVB 460
      DD0B(K,J,I)=0.                                                    ELVB 470
  120 DDDC(K,J,I)=0.0                                                   ELVB 480
      DO 121 I=1,3                                                      ELVB 490
      DO 122 J=1,3                                                      ELVB 500
  122 DDDB(I,J,I)=G(I,J)                                                ELVB 510
      DDDB(4,I,1)=G(I,2)                                                ELVB 520
      DDDB(6,I,3)=DDDB(4,I,1)                                           ELVB 530
      DDDB(5,I,1)=G(I,3)                                                ELVB 540
      DDDB(6,I,2)=DDDB(5,I,1)                                           ELVB 550
      DDDB(4,I,2)=G(I,1)                                                ELVB 560
  121 DDDB(5,I,3)=DDDB(4,I,2)                                           ELVB 570
```

```
C         START LOOP THROUGH ATOM LIST                                  ELVB 580
 1012 DO 2 I=1,NA                                                       BLVB 590
      II=1                                                              ELVB 600
C         SKIP OUT IF ATOM IS OTHER TYPE                                ELVB 610
      IF(ITF(I))3,2,3                                                   ELVB 620
C         EXPAND BETA LIST INTO SYMMETRIC MATRIX                        ELVB 630
C         B IS THE MATRIX OF THERMAL COEF.                              ELVB 640
    3 DO 4 K=1,2                                                        ELVB 650
      B(K,K)=BETA(K,I)                                                  BLVB 660
      L=K+1                                                             ELVB 670
      DO 4 J=L,3                                                        BLVB 680
      JX2=K*J+1                                                         BLVB 690
      B(J,K)=BETA(JX2,I)                                                BLVB 700
    4 B(K,J)=B(J,K)                                                     BLVB 710
      B(3,3)=BBTA(3,I)                                                  ELVB 720
C         COMPUTE D MATRIX                                              ELVB 730
C         D IS THE MATRIX PRODUCT GE                                    BLVB 740
      DO 5 K=1,3                                                        BLVB 750
      DO 5 L=1,3                                                        BLVB 760
      D(K,L)=0                                                          BLVB 770
      DO 5 M=1,3                                                        BLVB 780
    5 D(K,L)=D(K,L)+G(K,M)*B(M,L)                                       BLVB 790
C         SOLVE THIRD ORDER EQUATION TO GET RMS VALUES                  BLVB 800
C          T = TRACE OF MATRIX D                                        BLVB 810
C          S = SUM OF PRINCIPAL MINORS OF ORDER 2                       BLVB 820
C          DET = DETERMINANT OF D                                       BLVB 830
      T=D(1,1)+D(2,2)+D(3,3)                                            BLVB 840
      E=D(1,1)*(D(2,2)*D(3,3))+D(2,2)*D(3,3)-D(1,2)*(D(2,1))-D(1,3)*D(3,BLVB 850
     1 1)-D(2,3)*D(3,2)                                                 BLVB 860
      DET=D(1,1)*(D(2,2)*D(3,3)-D(2,3)*D(3,2))-D(1,2)*(D(3,3)*D(2,1)-   BLVB 870
     1 D(2,3)*D(3,1))+D(1,3)*(D(3,2)*D(2,1)-D(2,2)*D(3,1))              BLVB 880
C         COMPUTE EQUIVALENT ISOTROPIC B                                BLVB 890
      BISO=4.0/3.0*T                                                    BLVB 900
      DTDB(1)=B-T**2/3.0                                                BLVB 910
      DTDB(2)=(-2.0*T**3+9.0*T*S-27.0*DET)/27.0                         BLVB 920
      DTDB(3)=-DTDB(1)**3/27.0                                          BLVB 930
C         CHECK IF ATOM POSITIVE-DEFINITE                               BLVB 940
      IF(0.25*DTDB(2)**2-DTDB(3)-1.0E-6)1002,1002,7                     BLVB 950
 1002 DTDB(3)=-0.5*DTDB(2)/SQRT (DTDB(3))                               BLVB 960
      IF(ABS (DTDB(3))-1.0)1003,1003,6                                  BLVB 970
    6 DTDB(3)=SIGN (1.0,DTDB(3))                                        BLVB 980
 1003 PHI=ACOS (DTDB(3))                                                BLVB 990
      C=PHI/3.0                                                         BLVB1000
      DTDB(1)=2.0*SQRT (-DTDB(1)/3.0)                                   BLVB1010
      ROOT(1)=DTDB(1)*COS (C)+T/3.0                                     BLVB1020
      ROOT(2)=DTDB(1)*COS (C+TWOPI/3.0)+T/3.0                           BLVB1030
      ROOT(3)=DTDB(1)*COS (C-TWOPI/3.0)+T/3.0                           BLVB1040
C         PLACE ROOTS IN INCREASING ORDER                               BLVB1050
      DO 1004 J=1,2                                                     BLVB1060
      IJ=J+1                                                            BLVB1070
      DO 1004 K=IJ,3                                                    BLVB1080
      IF(ROOT(J),LE,ROOT(K))GO TO 1004                                  BLVB1090
 1005 HX=ROOT(J)                                                        BLVB1100
      ROOT(J)=ROOT(K)                                                   BLVB1110
      ROOT(K)=HX                                                        BLVB1120
 1004 CONTINUE                                                          BLVB1130
      IF(IERR)8,1006,8                                                  BLVB1140
C         PRINT EQUIVALENT B FOR NO ERRORS                              BLVB11S0
 1006 WRITE( 16UT,9003)(ATOM(K,I),K=1,2),BISO                           BLVB1160
      GO TO 1007                                                        BLVB1170
C         PRINT NON-POSITIVE DEFINITE MESSAGE                           BLVB1180
    7 WRITE( 16UT,9002)(ATOM(K,I),K=1,2)                                BLVB1190
      GO TO 2                                                           BLVB1200
    8 DO 10 J=1,3                                                       BLVB1210
      DO 10 K=1,3                                                       ELVB1220
      M=MOD(K,3)+1                                                      BLVB1230
      N=MOD(K+1,3)+1                                                    BLVB1240
      DDDC(K,K,J)=D(K,J)/A(K)+A(K)*B(J,K)                               ELVB1250
      DDDC(K,M,J)=A(M)*B(K,J)*ANG(N)                                    ELVB1260
      DDDC(K,N,J)=A(N)*B(K,J)*ANG(M)                                    BLVB1270
      DDDC(K+3,M,J)=A(M)*A(N)*B(J,M)                                    BLVB1280
      DDDC(K+3,N,J)=A(N)*A(M)*B(J,M)                                    BLVB1290
   10 CONTINUE                                                          ELVB1300
C         DTDA = DERIV OF TRACE WRT CELL PARAMS                         BLVB1310
C         DTDA = DERIV OF TRACE WRT TEMP FACTORS                        BLVB1320
      DO 20 K=1,6                                                       ELVB1330
      DTDA(K)=0.                                                        ELVB1340
      DTDB(K)=0.0                                                       ELVB1350
      DO 18 J=1,3                                                       ELVB1360
      DTDB(K)=DTDB(K)+DDDB(K,J,J)                                       ELVB1370
   18 DTDA(K)=DTDA(K)+DDDC(K,J,J)                                       ELVB1380
C         DSDA AND DSDB ARE DERIVS OF S WRT CELL PARAMS AND TEMP FACTORS ELVB1390
      DSDA(K)=0.                                                        ELVB1400
      DSDB(K)=0.0                                                       ELVB1410
      DO 20 J=1,3                                                       ELVB1420
      DO 20 L=1,3                                                       ELVB1430
      IF(L.EQ.J)GO TO 20                                                ELVB1440
   22 DSDA(K)=DSDA(K)+DJ(J,J)*DDDC(K,L,L)-DJ(J,L)*DDDC(K,L,J)           ELVB14S0
   21 DSDB(K)=DSDB(K)+DJ(J,J)*DDDB(K,L,L)-DJ(J,L)*DDDB(K,L,J)           ELVB1460
   20 CONTINUE                                                          ELVB1470
C         COMPUTE DERIVATIVES OF DETERMINANT WRT CELL AND THERMAL PARAMS ELVB1480
C         DERDET EVALUATES THE DERIVATIVE OF THE DETERMINANT            ELVB1490
      CALL DERDBT( DDBTDA,D,DDDC )                                      BLVB1S00
      CALL DERDET( DDETDB,D,DDDB )                                      ELVB1510
C         COMPUTE SIGMA FOR EQ IS B                                     ELVB1S20
      CALL VARIAN(DELTA,DTDA,DTDB,SIGA,SIGB,I1)                         BLVB1530
      DELTA=4.0/3.0*DELTA                                               ELVB1540
      WRITE( 16UT,9003)(ATOM(K,I),K=1,2),BISO,DELTA                     ELVB1550
C         START FINDING EIGENVECTORS AND ERRORS                         ELVB1560
 1007 DO 98 K=1,3                                                       ELVB1570
      DO 98 J=1,3                                                       ELVB1580
      X(K,J)=D(K,J)                                                     BLVB1590
      DO 98 L=1,6                                                       BLVB1600
      DXDA(L,J,K)=DDDC(L,J,K)                                           BLVB1610
   98 DXDB(L,J,K)=DDDB(L,J,K)                                           ELVB1620
      DO 19 N=1,3                                                       ELVB1630
      DEN=    (2.0*T*ROOT(N)-S-3.0*ROOT(N)**2)                          ELVB1640
      IF(ABS(DEN),GT.0.0)DEN=1./DEN                                     ELVB1650
      IF(IERR.EQ.0)GO TO 1018                                           BLVB1660
      DO 23 K=1,6                                                       ELVB1670
      DLDA(K)=DEN*(-DDETDA(K)+ROOT(N)*DSDA(K)-ROOT(N)**2*DTDA(K))       ELVB1680
   23 DLDB(K)=DEN*(-DDETDB(K)+ROOT(N)*DSDB(K)-ROOT(N)**2*DTDB(K))       ELVB1690
      CALL VARIAN(DELTA,DLDA,DLDB,SIOA,SIGB,11)                         ELVB1700
C         DELTA IS THE SIGMA OF THE ROOT                                BLVB1710
 1018 IF(ROOT(N))7,2S,25                                                ELVB1720
   25 ROOT1=SQRT (ROOT(N)*0.5)/PI                                       ELVB1730
```

```
      IF( IERR.EQ.0)GO TO 1016                                          ELVE1740
      DRGGT=DELTA/(4.0*PI**2*RGGT1)                                     ELVE1750
      WRITE( IGUT,9004)N,REET1,DRGET                                    ELVE1760
      GO TO 1030                                                        ELVE1770
 1016 WRITE( IGUT,1020)N,RGET1                                          ELVE1780
 1030       DO 100 K=1,3                                               ELVE1790
      K(K,K)=D(K,K)-RGGT(N)                                             ELVE1800
      DO 100 L=1,6                                                      ELVE1810
      DKDA(L,K,K)=DDDC(L,K,K)-DLDA(L)                                   ELVE1820
  100 DXDB(L,K,K)=DDDB(L,K,K)-DLDB(L)                                   ELVE1830
      QMAG=0.                                                           ELVE1840
      DO 104 L=1,3                                                      ELVE1850
      M=MGD(L,3)+1                                                      ELVE1860
      STB(1)=(X(L,2)*X(M,3)-X(L,3)*X(M,2))*AS(1)                        ELVE1870
      STB(2)=(X(L,3)*X(M,1)-X(L,1)*X(M,3))*AS(2)                        ELVE1880
      STB(3)=(X(L,1)*X(M,2)-X(L,2)*X(M,1))*AS(3)                        ELVE1890
      TEMP=STB(1)**2+STB(2)**2+STB(3)**2                                ELVE1900
      IF(TEMP-QMAG)104,102,102                                          ELVE1910
  102 QMAG=TEMP                                                         ELVE1920
      K=L                                                               ELVE1930
      DO 103 M=1,3                                                      ELVE194(
  103 QI(M)=STB(M)                                                      ELVE195(
  104 CONTINDB                                                          ELVE196(
      QMAG=(QMAG+2.0*(QI(1)*QI(2)*CS(3)+QI(1)*QI(3)*CS(2)+QI(2)*QI(3)*  ELVE1970
     $CS(1)))                                                           ELVE198(
      DO 7531 M=1,3                                                     ELVE1990
 7531 QI(M)=QI(M)/AS(M)                                                 ELVE2000
      IF(QMAG.GT.0.0)GO TO 2001                                         ELVE2010
      QMAG=1.0E-3                                                       ELVE2020
 2001 QMAG=SQRT(QMAG)                                                   ELVE2030
      IF( IERR.EQ.0) GO TO 1021                                         ELVE2040
      M=MGD(K,3)+1                                                      ELVE2050
      DO 107 L=1,6                                                      ELVE2060
      DO 107 NN=1,3                                                     ELVE2070
      NM=MBD(NN,3)+1                                                    ELVE208(
      MN=MGD(NM,3)+1                                                    ELVE2090
  110 DQDA(L,NN)=X(K,NM)*DKDA(L,M,MN)+K(M,MN)*DXDA(L,K,NM)-X(K,MN)*DKDA( ELVE2100
     1L,M,NM)-K(M,NM)*DKDA(L,K,MN)                                      ELVE2110
  107 DQDB(L,NN)=K(K,NM)*DXDB(L,M,MN)+K(M,MN)*DXDB(L,K,NM)-X(K,MN)*DXDB( ELVE2120
     1L,M,NM)-K(M,NM)*DKDB(L,K,MN)                                      ELVE2130
      K=1                                                               ELVE2140
   27 DBQDB(K)=0.                                                       ELVE2150
      DBQDA(K)=0.                                                       ELVE2160
      DO 26 J=1,3                                                       ELVE2170
      DBQDB(K)=DBQDB(K)+2.0*QI(J)*AS(J)**2*DQDB(K,J)                    ELVE218(
      DBQDA(K)=DBQDA(K)+2.0*QI(J)*AS(J)**2*DQDA(K,J)+2.0*AS(J)*QI(J)**2 ELVE2190
     1*DASDA(J,K)                                                       ELVE2200
      DO 26 L=1,3                                                       ELVE2210
      IF(L.EQ.J) GO TO 26                                               ELVE2220
      JX2=6-L-J                                                         ELVE2230
      DBQDB(K)=DBQDB(K)+AB(J)*AS(L)*( QI(L)*DQDS(K,J)+QI(J)*DQDB(K,L))  ELVE2240
     1 *CB(JX2)                                                         ELVE2250
      DBQDA(K)=DBQDA(K)+AS(J)*AE(L)*( QI(J)*QI(L)*DAEDA(JX2+3,K)+CS(JX2)*ELVE2260
     1(QI(J)*DQDA(K,L)+QI(L)*DQDA(K,J)))+QI(J)*QI(L)*CS(JX2)*( AS(J)*   ELVE2270
     2 DAEDA(L,K)+AE(L)*DASDA(J,K))                                     ELVE228(
   26 CONTINDE                                                          ELVE2290
      DBQDA(K)=0.5*DBQDA(K)/QMAG                                        ELVE2300
      DBQDB(K)=0.5*DEQDB(K)/QMAG                                        ELVE2310
      K=K+1                                                             ELVE2320
      IF(K.LE.6) GO TO 27                                               ELVE2330
 1021 DO 30 K=1,3                                                       ELVE2340
      THETA(K)=QI(K)/(A(K)*QMAQ)                                        ELVE2350
      IF(ABS (THETA(K))-1.0)32,32,31                                    ELVE2360
   31 THETA(K)=SIGN (1.0,THETA(K))                                      ELVE2370
   32 STB(K)=SQRT (1.0-THETA(K)**2)                                     ELVE2380
   30 THETA(K)=180.*ACOS (THETA(K))/PI                                  ELVE2390
      IF( IERR.EQ.0)GO TO 1023                                          ELVE2400
 1024 DO 33 K=1,3                                                       ELVE2410
      DO 34 J=1,6                                                       ELVE2420
C        THETA IS THE ANGLE WITH THE I TH REAL AXIS                     ELVE2430
      DTEDB(J,K)=DQDB(J,K)/(A(K)*QMAG)-QI(K)*DBQDB(J)/(A(K)*QMAG**2)    ELVE2440
   34 DTHDA(J,K)=DQDA(J,K)/(A(K)*QMAG)-QI(K)*DEQDA(J)/(A(K)*QMAG**2)    ELVE2450
      DTHDA(K,K)=DTHDA(K,K)-QI(K)/(A(K)**2*QMAG)                        ELVE2460
      DO 33 J=1,6                                                       ELVE2470
      IF(ABS (ETB(K))-1.0E-3)150,150,151                                ELVE2480
  151 DTHDA(J,K)=-DTEDA(J,K)/ETB(K)                                     ELVE2490
      DTEDB(J,K)=-DTHDB(J,K)/ETB(K)                                     ELVE2500
      GO TO 33                                                          ELVE2510
  150 DTEDA(J,K)=0.                                                     ELVE2520
      DTEDB(J,K)=0.                                                     ELVE2530
   33 CONTINDE                                                          ELVE2540
      DO 35 K=1,3                                                       ELVE2550
      CALL VARIAN( STE(K),DTHDA(1,K),DTHDB(1,K),BIGA,BIGB,II)          ELVE2560
   35 ETH(K)=ETH(K)*180./PI                                             ELVE2570
      WRITE( IGDT,9005)( IAXIB(K),THETA(K),ETH(K),K=1,3)                ELVE2580
      GO TO 19                                                          ELVE2590
 1023 WRITE( IGDT,1025)( IAXIB(K),THETA(K),K=1,3)                       ELVE2600
   19 CONTINDE                                                          ELVE2610
    2 CONTINUE                                                          ELVE2620
      RETURN                                                            ELVE2630
      END                                                               ELVE2640
```

```
      SUBROUTINE CORW1(SIGZ,NN,II,JJ,CORR,KK,LL,MM)                    COR1  10
                                                                       COR1  20
C THIS ROUTINE REVISED ANO/OR RESEQUENCED ON OCT. 2E, 1968              COR1  30
                                                                       COR1  40
C        THIS ROUTINE EXTRACTS THE COVARIANCE ELEMENTS BETWEEN TWO ATOMSCOR1  50
C         SIGZ - OUTPUT AREAY OF SIZE NN BY NN                         COR1  60
C         II = ATOM NUMBER OF FIRST ATOM                               COR1  70
C          JJ = ATOM NUMBER OF SECOND NUMBER                           COE1  80
C        CORR IS THE WHOLE POSITION VAR-COV MATRIX                     COR1  90
C        KK AND LL DESCRIBE THE BLOCK OF SIGZ TO BE FILLED             COR1 100
C        MM = THE NUMBER OF ATOMS (NEEDED TO COMPUTE THE RANK OF CORR) COR1 110
                                                                       COE1 120
      DIMENSION SIGZ(NN,NN),CORR(7260)                                 COR1 130
                                                                       COE1 140
      IK=MM                                                            COR1 150
      IL=3*II-2                                                        COR1 160
      IJ=3*JJ-2                                                        COR1 170
      KI=KK-1                                                          COR1 180
      LI=LL-1                                                          COR1 190
      IF(II-JJ)1,2,3                                                   COR1 200
    1 JI=((IJ-1)*IJ)/2+IL                                              COR1 210
      IX=1J                                                            COE1 220
      GO TO 4                                                          COR1 230
    3 JI=((IL-1)*IL)/2+IJ                                              COR1 240
      IX=IL                                                            COR1 250
    4 DO 5 I=1,3                                                       COR1 260
      L=JI                                                             COR1 270
      I1=1+LI                                                          COR1 280
      DO 6 J=1,3                                                       COR1 290
      J1=J+KI                                                          COR1 300
      SIGZ(I1,J1)=CORR(L)                                              COR1 310
    6 L=L+1                                                            COR1 320
    5 JI=J1+IX-I-1                                                     COR1 330
      RETURN                                                           COE1 340
    2 SIGZ(LI+1,KI+1)=SIGZ(KI+1,KI+1)                                  COR1 350
      DO 7 I=2,3                                                       COR1 360
      I1=I+LI                                                          COR1 370
      J1=1+KI                                                          COR1 3E0
      SIGZ(I1,J1)=SIGZ(J1,J1)                                          COE1 390
      IX=I-1                                                           COR1 400
      DO 7 J=1,IX                                                      COR1 410
      J2=J+KI                                                          COR1 420
      J3=J+LI                                                          COR1 430
      SIGZ(I1,J2)=SIGZ(J1,J2)                                          COR1 440
    7 SIGZ(J3,J1)=SIGZ(I1,J2)                                          COR1 450
      RETURN                                                           COR1 460
      END                                                              COR1 470
```

```
      SUBROUTINE CELLD(DASDA,A,ANG,AS,CS,VOL)                          CELD  10
                                                                       CELD  20
C THIS ROUTINE REVISED AND/OR RESEQUENCED ON OCT. 28, 196E              CELD  30
                                                                       CELD  40
      DIMENSION A(3),ANG(3),AS(3),CS(3),DASDA(6,6),SN(3)               CELD  50
                                                                       CELD  60
C        ROUTINE TO COMPUTE THE DERIVATIVES OF THE RECIPROCAL LATTICS   CELD  70
C         PARAMETERS WITH RESPECT TO THE REAL CELL PARAMETERS.         CELD  80
C        OASDA(I,J)= DERIVATIVE OF RECIPROCAL PARAMETER I WITH RESPSCT CELD  90
C         TO REAL CELL PARAMETER J.                                    CELD 100
      TEMP=(A(1)*A(2)*A(3)/VOL)**2                                     CELD 110
      SNPROD=1.0                                                       CELD 120
C        COMPUTE SINES OF REAL SPACE ANGLES AND PRODUCT OF SINES       CELD 130
      DO 100 I=1,3                                                     CELD 140
      SN(I)=SQRT(1.0-ANG(I)**2)                                        CELD 150
  100 SNPROD=SNPROD*SN(I)                                              CELD 160
C        CLEAR DERIVATIVE ARRAY                                        CELD 170
      DO 110 I=1,6                                                     CELD 180
      DO 110 J=1,6                                                     CELD 190
  110 DASDA(I,J)=0.0                                                   CELD 200
C        COMPUTE DERIVATIVES                                           CELD 210
      DO 120 I=1,3                                                     CELD 220
      TEMP1=TEMP*AS(I)                                                 CELD 230
      J=MOD(I,3)+1                                                     CELD 240
      K=MOD(I+1,3)+1                                                   CELD 250
      L=I+3                                                            CELD 260
      DASDA(I,I)=-AS(I)/A(I)                                           CELD 270
      DASDA(I,L)=-TEMP1*SN(J)*SN(K)*CS(I)-AS(I)*ANG(I)/SN(I)**2        CELD 280
      DASDA(I+3,I+3)=-SN(I)/SNPROD                                     CELD 290
      DASDA(I,J+3)=-TEMP1*SN(I)*SN(K)*CS(J)                            CELD 300
      DASDA(I,K+3)=-TEMP1*SN(I)*SN(J)*CS(K)                            CELD 310
      DASDA(L,J+3)=SN(I)*ANG(K)/SNPROD*CS(I)*ANG(J)/SN(J)**2           CELD 320
  120 DASDA(L,K+3)=SN(I)*ANG(J)/SNPROD*CS(I)*ANG(K)/SN(K)**2           CELD 330
      RETURN                                                           CELD 340
      END                                                              CELD 350
```

```
      SDBRØDTINE DERDBT(A,B,C)                                          DERD  10
                                                                        DERD  20
C THIS RØUTINE REVISRD AND/ØR RRSBQUENCED ØN ØCT. 2R, 1968              DERD  30
                                                                        DERD  40
C         THIS RØUTINE CØMPUTES THE DERIVATIVE ØF A DETERMINANT         DERD  50
                                                                        DBRD  60
      DIMENSIØN A(6),B(3,3),C(6,3,3)                                    DBRD  70
                                                                        DERD  R0
      DØ 4 I=1,6                                                        DERD  90
      A(I)=0                                                            DBRD 100
      DØ 4 J=I,3                                                        DERD 110
      M=MØD(J,3)+1                                                      DERD 120
      N=MØD(J+1,3)+1                                                    DERD 130
    4 A(I)=A(I)+C(I,J,J)*(B(N,N)*B(M,M)-B(N,M)*B(M,N))                  DERD 140
      1   +C(I,J,N)*(B(M,J)*B(N,M)-B(M,M)*B(N,J))                       DBRD 150
      2   +C(I,J,M)*(B(N,J)*B(M,N)-B(N,N)*B(M,J))                       DBRD 160
      RETURN                                                            DRRD 170
      END                                                               DERD 1R0


      BUBRØDTINE VARIAN(A,B,C,D,B,I1)                                   VARN  10
                                                                        VARN  20
C THIS RØUTINE RBVISRD AND/ØR RBSEQUENCED ØN ØCT. 2R, 196R              VARN  30
      DIMENSIØN B(6),C(6),D(6,6)   ,B(6,6,40)                           VARN  40
                                                                        VARN  50
      N=II                                                              VARN  60
      F=B(1)**2*D(1,1)*C(1)**2*B(1,1,N)                                 VARN  70
      DØ 1 I=2,6                                                        VARN  80
      F=F+B(1)**2*D(I,1)*C(I)**2*R(I,1,N)                               VARN  90
      IJ=I-1                                                            VARN 100
      DØ 1 J=1,IJ                                                       VARN 110
    1 F=F+2.*R(I)*R(J)*D(1,J)*2.0*C(1)*C(J)*R(I,J,N)                    VARN 120
      IF(F)2,3,3                                                        VARN 130
    2 F=0                                                               VARN 140
    3 A=SQRT (F)                                                        VARN 150
      RETURN                                                            VARN 160
      END                                                               VARN 170
                                                                        VARN 1B0
```

```
      CØMPILER(FLD=Q)                                                   LRCR  10
      SUBRØDTINE LIBCØR(XYZ,PAR,KPAR,NPAR,G,NA,MDECN,KNSTR)             LRCR  20
C     SUBRØUTINE TØ APPLY THE LIBRATIØN CØRRECTIØN TØ THE PØSITIØN      LBCR  30
C     PARAMETERS, IN PREPARATIØN PØR CALCDLATIØN ØP BØND DISTANCES AND  LBCR  40
C     ANCLES. IT IS ASSØMED THAT TRE CØNTRIBUTIØNS TØ THE ERRØRS IN     LBCR  50
C     DISTANCES AND ANCLBS DOE TØ THE BRRØRS IN DETBRMINATIØN ØF TBR    LRCR  60
C     LIBRATIØN PARAMETBRS IS NBGLBGIBLR CØMPARED WITR TRB ERRØRS IN    LRCR  70
C     TBB DRTBRMINATIØN ØF TBB PØS1TIØN PARAMRTERS.                     LRCR  R0
      DEFINR I1(K)=FLD(0,9,K)                                           LBCR  90
      0EPINE 12(K)=FLD(9,9,K)                                           LBCR 100
      DBFINE I3(K)=FLD(18,9,K)                                          LBCR 110
      DRFINR I4(K)=PLD(27,9,K)                                          LBCR 120
      DIMENSIØN XYZ(3,60),PAR(60),KPAR(60),C(3,3),KNSTR(MDBGN),ATØX(6)  LRCR 130
     1,XTØA(6),RC(3),R(3),BL(3,3),ØXYZ(3),SXYZ(3),L(6)                  LRCR 140
C     CØMPUTE TRANS PØRMATIØN FRØM CRYSTAL TØ ØRTRØNØRMAL ANO BACK.     LBCR 150
      ATØX(1)=SQRT(G(1,1))                                              LBCR 160
      ATØX(2)=G(1,2)/ATØX(1)                                            LBCR 170
      ATØX(3)=C(1,3)/ATØX(1)                                            LRCR 1R0
      ATØX(4)=SQRT(G(2,2)-ATØX(2)**2)                                   LBCR 190
      ATØX(5)=(C(2,3)-ATØX(2)*ATØX(3))/ATØX(4)                          LBCR 200
      ATØX(6)=SQRT(G(3,3)-ATØX(3)**2-ATØX(5)**2)                        LBCR 210
      XTØA(1)=I./ATØX(1)                                                LBCR 220
      XTØA(4)=1./ATØX(4)                                                LBCR 230
      XTØA(6)=1./ATØX(6)                                                LBCR 240
      XTØA(2)=-ATØX(2)/(ATØX(1)*ATØX(4))                                LRCR 250
      XTØA(5)=-ATØX(5)/(ATØX(4)*ATØX(6))                                LRCR 260
      XTØA(3)=(ATØX(2)*ATØX(5)/ATØX(4)-ATØX(3))/(ATØX(1)*ATØX(6))       LBCR 270
C     DETERMINE WBETHER ANY SPECIAL PARAMETERS ARR L TENSØR ELEMENTS.   LBCR 2R0
      IBBT=-99                                                          LBCR 290
      IST=1                                                             LBCR 300
   50 DØ 100 I=IST,NPAR                                                 LBCR 310
      IF(11(KPAR(1)).RQ.1.AND.(14(KPAR(1)).GB.16.AND.I4(KFAR(I)).LE.21) LBCR 320
     1 .AND.I2(KPAR(I)).NE.ISBT) GØ TØ 110                              LBCR 330
  100 CØNTINUB                                                          LBCR 340
      RBTDRN                                                            LBCR 350
C     THERB IS AT LBAST 1 BET ØP L TRNBØR ELEMENTS. FIND THE MBMRERS ØF LRCR 360
C     THE SRT AND FILL TBEM IN.                                         LBCR 370
  110 ISBT=I2(KPAR(I))                                                  LBCR 3R0
      DØ 120 J=1,3                                                      LBCR 390
      ØXYZ(J)=0.                                                        LBCR 400
      DØ 120 K=1,3                                                      LBCR 410
  120 EL(J,K)=0.                                                        LBCR 420
      IL=0                                                              LBCR 430
      DØ 130 J=1,NPAR                                                   LBCR 440
      IF(12(KPAR(J)).NE.ISET.ØR.I1(KPAR(J)).NB.1) GØ TØ 130             LBCR 450
      IDRN=14(KPAR(J))                                                  LBCR 460
      IF(IDEN.GE.1.AND.IDEN.LB.3)GØ TØ 124                              LBCR 470
      IF(IORN.CE.16.AND.IDEN.LB.21)GØ TØ 126                            LRCR 4R0
      GØ TØ 130                                                         LBCR 490
  124 ØXYZ(IDEN)=FAR(J)                                                 LBCR 500
      GØ TØ 130                                                         LBCR 510
  126 IDRN=IDEN-15                                                      LBCR 520
      IF(IDBN-3),,128                                                   LBCR 530
      EL(IDRN,IDEN)=FAR(J)                                              LRCR 540
      GØ TØ 129                                                         LBCR 550
  128 JJ=1                                                              LBCR 560
      KK=3                                                              LBCR 570
```

```
      IF(IDBN.EQ.6)JJ=2                                                 LBCR 580
      IF(IDRN.EQ.4)KK=2                                                 LBCR 590
      EL(JJ,KK)=FAR(J)                                                  LBCR 600
      EL(KK,JJ)=FAR(J)                                                  LBCR 610
  129 IL=IL+1                                                           LBCR 620
      L(IL)=J                                                           LRCR 630
  130 CONTINUB                                                          LRCR 640
C     WB MUST NOW FIND WBICR ATOMS DBFBND ON L IN THIS SET. ARRAY L     LRCR 650
C     CONTAINS A LIBT OF TBB PARAMETERS WBICB ARE MEMRERS OF THR L      LBCR 660
C     TENSOR IN TRIS BET.                                               LRCR 670
      DO 1000 I=1,NA                                                    LBCR 680
      DO 140 J=1,IL                                                     LBCR 690
      JJ=L(J)                                                           LRCR 700
      DO 140 K=1,MDBON,2                                                LBCR 710
      IF(I1(ENSTR(K)).NE.0.OR.I2(ENSTR(K)),GT,L(IL))GO TO 1000          LBCR 720
      IF(I2(ENSTR(K)).BQ.JJ.AND.I3(ENSTR(K)).RQ.I)GO TO 150             LBCR 730
  140 CONTINOE                                                          LBCR 740
C     NO MATCE FOOND FOR TBIS ATOM                                      LRCR 750
      GO TO 1000                                                        LBCR 760
C     ATOM I DBFBNDS ON THB CURRRNT SET, APPLT LIBRATION CORRRCTION.    LBCR 770
  150 DO 160 J=1,3                                                      LBCR 780
  160 SXTZ(J)=XYZ(J,I)-OXTZ(J)                                          LBCR 790
      CALL UTMOL(ATOX,SXYZ,3,RC)                                        LBCR 800
      CALL CORLIR(R,RL,RC)                                              LBCR 810
      CALL UTMUL(XTOA,R,3,SXYZ)                                         LRCR 820
      DO 170 J=1,3                                                      LBCR R30
  170 XYZ(J,I)=SXTZ(J)+OXTZ(J)                                          LBCR R40
1000  CONTINOE                                                          LBCR R50
      IBT=L(IL)+1                                                       LBCR R60
      IF(IST.GT.NFAR)RETUEN                                             LBCR R70
      GO TO 50                                                          LBCR 8R0
      END                                                               LBCR R90
```

```
      SUBROUTINB UTMUL(A,B,M,C)                                         OTML  10
C     MULTIPLIBS VRCTOR B RY UPPRR TRIANGOLAR MATRIX A, STORING THE     UTML  20
C     RBBULT IN C. A IS BTORED BY ROWB.                                 UTML  30
      DIMRNSION A(1),R(1),C(1)                                          UTML  40
      K=0                                                               UTML  50
      DO 100 I=1,N                                                      UTML  60
      C(I)=0.                                                           UTML  70
      DO 100 J=1,N                                                      UTML  80
      K=K+1                                                             UTML  90
  100 C(I)=C(I)+A(K)*R(J)                                               UTML 100
      RETURN                                                            UTML 110
      END                                                               UTML 120
```

```
C  GENERALIZED FOURIER PROGRAM   MAY 1974   8Y L. W. F(NGER             FOUR  10
       COMMON/A8LE/F8UF(504)                                            FOUR  20
       01MENSION TITLE(20),JH(3),IH(3),1P(43),RMAT(3,3,24),1RMAT(3,     FOUR  30
      13,24),TRANS(3,24),IEXTR(6),FX(21),FMULT(12)                      FOUR  40
       (NTEGER FRMT(50)/4H(A1,,4HF4.0,3H,1X,47*1H /                     FOUR  50
       1NTEGER LA8EL(20)/1H ,4HO(FF,3HXYZ,3HXZY,3HYXZ,3HYZX,3HZYX,3HZXY, FOUR  60
      1 3HHOO,3HOOL,3HHKO,3HHOL,3HOKL,3HHHO,3HHOH,3HOKK,3HHHH,3HHHL,3HHKKFOUR  70
      2,3HHKL/                                                          FOUR  80
       OIMENSION IF8UF(505),ARRAY(1),IARRAY(1)                          FOUR  90
       EOU(VALENCE ((EXTR(1),(8X),((EXTR(2),(8Y),((EXTR(3),(8Z)        FOUR 100
       EOU(VALENCE ((EXTR(4),1MAX),((EXTR(5),KMAX),((EXTR(6),LMAX)     FOUR 110
       EOU(VALENCE (F8UF,1F8UF),(IF8UF(505),ARRAY(1),(ARRAY(1))        FOUR 120
       EOU(VALENCE (RMAT,IRMAT),(FX,(P)                                FOUR 130
       (NTEGER OUT,S1,ZRO,ENDO,S2                                       FOUR 140
       OATA IN,OUT,S1,ZRO,NZRU,ENOO,S2/5,6,12,3H,4X,3H,I4,1H),11/       FOUR 150
       OATA TWOP(,(SPC,I8NK/6.28318531,1HO,1H /                         FOUR 160
C                                                                       FOUR 170
    2 FORMAT(20A4)                                                      FOUR 180
    4 FORMAT(6I1,2X,9(4,3F8.0/12F6.0)                                   FOUR 190
    6 FORMAT(A1,F4.0)                                                   FOUR 200
    8 FORMAT(1H1,20A4,5X,7HSECTION,I4,1H/,(4//F10.0,14F8.0)             FOUR 210
   12 FORMAT(/5X,24HALL REFLECTIONS (NCLUOED )                          FOUR 220
   13 FORMAT(/5X,43HREFLECTIONS WITH REJ. FLAG SET NOT INCLUOED )       FOUR 230
   14 FORMAT(1H1,20A4//4X,A4,8H FOUR(ER,10X,A3,9H SECTIONS//5X,9HF(000) FOUR 240
     1= F10.3//5X,19HMAP EXAGGERATION = F10.3//5X,31HMULTIPL(CITY OF GENFOUR 250
     2ERAL REFL = F10.3)                                                FOUR 260
   15 FORMAT(/40X,14HMULTIPL(CIT(ES/5X,11(A3,5X),A3//12F8.2//           FOUR 270
     2//9X,50HINIT(AL   FINAL    1NTERVAL   MIN INOEX  MAX INOEX/)      FOUR 280
   16 FORMAT(10H POINTS   I4,18,I12,I10,(11/8H LINES  ,16,18,I12,110,I11FOUR 290
     1/9H SECTIONS,(5,I8,I12,I10,I11)                                   FOUR 300
   17 FORMAT(/5X,34HSECTIONS SAVED FOR LATER PLOTTING.)                 FOUR 310
   18 FORMAT(///10X,22HSTORAGE REOUIREMENTS ARE (6,11H LOCATIONS.)      FOUR 320
   20 FORMAT(/15X,18HFRACTIONAL COOROS.,11X,11HMAP COOROS./5H ATOM,8X,1HFOUR 330
     1X,8X,1HY,8X,1HZ,9X,5HPOINT,4X,4HLINE,3X,7HSECT(ON/)               FOUR 340
   22 FORMAT(1X,2A3,1X,3F9.5,3X,3F8.2)                                  FOUR 350
C                                                                       FOUR 360
  100 READ((N,2,ENO=900)TITLE                                           FOUR 370
       REW(NO S1                                                        FOUR 380
       READ(S1)NSYM,VOL,(((RMAT(1,J,K),1=1,3),J=1,3),K=1,NSYM),(((TRANS(J,FOUR 390
      1 1),J=1,3),(=1,NSYM)                                             FOUR 400
       CALL CORSZ(ARRAY(1))                                             FOUR 410
       NF8R=504                                                         FOUR 420
       READ(IN,4)ITYPE,ISECT,(CENT,KREJ,1PLT,LSPC,IX,IX(,IXF,IY,IYI,IYF,(FOUR 430
      1Z,1ZI,(ZF,EXXAG,FOOU,TMULT,FMULT                                 FOUR 440
       WRITE(OUT,14)TITLE,LA8EL((TYPE+1),LA8EL((SECT+3),FOOO,EXXAG,TMULT FOUR 450
       IF((PLT.GT.0)WRITE(OUT,17)                                       FOUR 460
       1F(KREJ)101,102,101                                              FOUR 470
  101 WRITE(OUT,13)                                                     FOUR 480
       GO TO 103                                                        FOUR 490
  102 WRITE(OUT,12)                                                     FOUR 500
  103 WRITE(OUT,15)(LA8EL(I),I=9,20),FMULT                              FOUR 510
       FOOO=FOOO*EXXAG/VOL                                              FOUR 520
       EXXAG=2.0*EXXAG/(VOL*TMULT)                                      FOUR 530
C         TRANSFORM AND CONVERT SYMMETRY OPERATORS                      FOUR 540
                                 OO 113 1=1,NSYM                        FOUR 550
                                 OO 113 J=1,3                           FOUR 560
       TRANS(J,I)=TWOP1*TRANS(J,I)                                      FOUR 570
```

```
                              00 112 K=1,3                                  FOUR 580
  112 IH(K)=RMAT(J,K,I)                                                     FOUR 590
      CALL SECTI)SECT,IH)                                                   FOUR 600
                              00 113 K=1,3                                  FOUR 610
  113 (RMAT(J,K,()=IH(K)                                                    FOUR 620
      MAXRFL=0                                                              FOUR 630
      00 120 I=1,6                                                          FOUR 640
  120 )EXTRII)=0                                                            FOUR 650
      INOEX=1                                                               FOUR 660
      NREFL=0                                                               FOUR 670
      NF8R=1000                                                             FOUR 680
      IF(LSPC.NE.0)GO TO 125                                                FOUR 690
C        00U8LE SPACE THE MAP                                               FOUR 700
      LSPC=ISPC                                                             FOUR 710
      GO TO 130                                                             FOUR 720
C        S)NGLE SPACE THE MAP                                               FOUR 730
  125 LSPC=I8NK                                                             FOUR 740
C        REAO REFLECTION OATA                                               FOUR 750
  130 NF8R=NF8R+10                                                          FOUR 760
      )F(NF8R-500)150,140,140                                               FOUR 770
C        GET NEW 8LOCK                                                      FOUR 780
  140 REAOIS1)IF8UF(I),I=1,504)                                             FOUR 790
      NF8R=0                                                                FOUR 800
  150 IH(1)=IF8UF(NF8R+1)                                                   FOUR 810
      )H(2)=IF8UF(NF8R+2)                                                   FOUR 820
      IH(3)=(F8UF(NF8R+3)                                                   FOUR 830
      F08S=F8UF(NF8R+4)                                                     FOUR 840
      )FIIHI1)-99)160,600,160                                               FOUR 850
C        NOT ENO, CHECK 1F REJECTEO                                         FOUR 860
  160 )F(IF8UF(NF8R+8))170,180,170                                          FOUR 870
  170 1F(KREJ)130,180,130                                                   FOUR 880
C        REFLECTION NOT REJECTEO                                            FOUR 890
  180 NREFL=NREFL+1                                                         FOUR 900
      )F(NREFL-MAXRFL)200,200,190                                           FOUR 910
C        EXPAND MAXIMUM SIZE                                                FOUR 920
  190 MAXRFL=MAXRFL+400                                                     FOUR 930
      CALL CORSZ(ARRAYI3*MAXRFL))                                           FOUR 940
  200 )NDEX=INOEX+3                                                         FOUR 950
      IF()TYPE)210,220,210                                                  FOUR 960
  210 F08S=F08S-F8UF(NF8R+5)                                                FOUR 970
C        CALCULATE PHASE ANGLE                                              FOUR 980
  220 IF(ICENT)230,225,230                                                  FOUR 990
  225 F8UF(NF8R+7)=0.0                                                      FOUR1000
  230 ARRAYIINOEX-1)=ATAN2IF8UF(NF8R+7),F8UFINF8R+6))                       FOUR1010
C           OETERMINE REFLECTION TYPE                                      FOUR1020
      IF(IH(1))350,280,350                                                  FOUR1030
  280 IF(IH(2))300,290,300                                                  FOUR1040
C           OOL REFLECTION                                                 FOUR1050
  290 IC=2                                                                  FOUR1060
      GO TO 530                                                             FOUR1070
  300 )F(IH(3))320,310,320                                                  FOUR1080
C           OKO REFLECTION                                                 FOUR1090
  310 IC=1                                                                  FOUR1100
      GO TO 530                                                             FOUR1110
  320 IFIIH(2)-)H(3))340,330,340                                            FOUR1120
C           OKK REFLECTION                                                 FOUR1130
  330 IC=8                                                                  FOUR1140
      GO TO 530                                                             FOUR1150
C           OKL REFLECTION                                                 FOUR1160
  340 IC=5                                                                  FOUR1170
      GO TO 530                                                             FOUR1180
  350 IF(IH(2))410,360,410                                                  FOUR1190
  360 IF(IH(3)380,370,380                                                   FOUR1200
C           HOO REFLECTION                                                 FOUR1210
  370 (C=1                                                                  FOUR1220
      GO TO 530                                                             FOUR1230
  380 )F(IH(1)-1H(3))400,390,400                                            FOUR1240
C           HOH REFLECTION                                                 FOUR1250
  390 (C=7                                                                  FOUR1260
      GO TO 530                                                             FOUR1270
C           HOL REFLECTION                                                 FOUR1280
  400 IC=4                                                                  FOUR1290
      GO TO 530                                                             FOUR1300
  410 IF(IH(3))450,420,450                                                  FOUR1310
  420 (F()H(1)-IH(2))440,430,440                                            FOUR1320
C           HHO REFLECTION                                                 FOUR1330
  430 (C=6                                                                  FOUR1340
      GO TO 530                                                             FOUR1350
C           HKO REFLECTION                                                 FOUR1360
  440 (C=3                                                                  FOUR1370
      GO TO 530                                                             FOUR1380
  450 (F(IHI1)-(H(2))490,460,490                                            FOUR1390
  460 )F(IH(1)-)H(3))480,470,480                                            FOUR1400
C           HHH REFLECTION                                                 FOUR1410
  470 (C=9                                                                  FOUR1420
      GO TO 530                                                             FOUR1430
C           HHL REFLECTION                                                 FOUR1440
  480 (C=10                                                                 FOUR1450
      GO TO 530                                                             FOUR1460
  490 IF(IH(2)-(H(3))510,500,510                                            FOUR1470
C           HKK OR HKH REFLECTION                                          FOUR1480
  500 IC=11                                                                 FOUR1490
      GO TO 530                                                             FOUR1500
  510 IF(IH(1)-IH(3))520,500,520                                            FOUR1510
C           HKL REFLECTION                                                 FOUR1520
  520 (C=12                                                                 FOUR1530
  530 ARRAY(INOEX-2)=EXXAG*FMULT()C)*F08S                                   FOUR1540
      (ARRAY(INOEX-3)=((H(I)*256+IH(2))*256+(H(3)+6579300                   FOUR1550
C        EXPAND INOEX LIST                                                  FOUR1560
      00 590 I=1,NSYM                                                       FOUR1570
      00 540 J=1,3                                                          FOUR1580
      JH(J)=0.0                                                             FOUR1590
      00 540 K=1,3                                                          FOUR1600
  540 JH(J)=JH(J)+)RMAT(K,J,I)*IH(K)                                        FOUR1610
      00 590 J=1,3                                                          FOUR1620
      IF(JH(J)-IEXTR(J))570,570,560                                         FOUR1630
C        SET NEW MAXIMUM                                                    FOUR1640
  560 (EXTR(J)=JH(J)                                                        FOUR1650
      GO TO 590                                                             FOUR1660
  570 )F(JH(J)-(EXTR(J+3))580,590,590                                       FOUR1670
C        SET NEW M)NIMUM                                                    FOUR1680
  580 )EXTR(J+3)=JH(J)                                                      FOUR1690
  590 CONTINUE                                                              FOUR1700
      GO TU 130                                                             FOUR1710
C        ENO OF REFLECT)ON L)ST                                             FOUR1720
  600 )ARRAY(INOEX)=0                                                       FOUR1730
```

```
C        MOVE REFLECTION DATA TO START OF BUFFER AREA                    FOUR1740
      DO 610 I=1,INOEX                                                   FOUR1750
  610 F8UF(I)=ARRAY(I)                                                   FOUR1760
      IPN1=INOEX                                                         FOUR1770
      WRITE(OUT,16)IX(,IXF,(X,IEXTR(4),I8X,IYI,IYF,IY,IEXTR(5),I8Y,     FOUR1780
     $IZI,IZF,IZ,IEXTR(6),I8Z                                           FOUR1790
      XX=TWOPI/FLOAT(IX)                                                 FOUR1800
      YY=TWOPI/FLOAT(IY)                                                 FOUR1810
      ZZ=TWOPI/FLOAT(IZ)                                                 FOUR1820
      IXA=(XF-IX1+1                                                      FOUR1830
      IYA=IYF-IYI+1                                                      FOUR1840
      IZA=IZF-IZI+1                                                      FOUR1850
      DO 620 I=1,3                                                       FOUR1860
      IEXTR(I+3)=MAXO(IEXTRII),-IEXTR(I+3))                              FOUR1870
  620 IEXTR(I)=2*IEXTR(I+3)+1                                            FOUR1880
      I8X=IMAX+1                                                         FOUR1890
      IPN2=IPN1+I8X*(8Y                                                  FOUR1900
      IPN3=IPN2+I8X*I8Y                                                  FOUR1910
      IPN4=IPN3+IMAX*IXA                                                 FOUR1920
      IPN5=IPN4+IMAX*IXA                                                 FOUR1930
      I=MAXO(KMAX,I8Z)                                                   FOUR1940
      IPN6=IPN5+I                                                        FOUR1950
      IPN7=IPN6+I                                                        FOUR1960
      IPN8=(PN7+I8X                                                      FOUR1970
      IPN9=IPN8+I8X                                                      FOUR1980
      IPN10=IPN9                                                         FOUR1990
      IF(IPLT.EQ.0)GO TO 621                                             FOUR2000
C        SECTIONS TO BE SAVED FOR PLOTTING                               FOUR2010
      IPN10=IPN9+IXA*IYA                                                 FOUR2020
  621 IPN10=MAXO(IPN10,IPN1+1020)                                        FOUR2030
      WRITE(OUT,18)IPN10                                                 FOUR2040
      CALL CORSZ(F8UF(IPN10))                                            FOUR2050
C  READ ATOM PARAMETERS FROM THE FILE                                     FOUR2060
      READ(S1)NATUM,((F8UF(IPN1+17*I+J-17),J=1,17),I=1,NATOM)            FOUR2070
      WRITE(OUT,20)                                                      FOUR2080
      J=IPN1-17                                                          FOUR2090
      DO 623 I=1,NATOM                                                   FOUR2100
      J=J+17                                                             FOUR2110
      DO 622 K=1,3                                                       FOUR2120
  622 FX(K)=F8UF(J+K+8)                                                  FOUR2130
C        TRANSFORM ATOM COORDINATES TO MAP SYSTEM                        FOUR2140
      CALL SECT(ISECT,FX)                                                FOUR2150
      FX(1)=FX1I)*IX                                                     FOUR2160
      FX(2)=FX(2)*IY                                                     FOUR2170
      FX(3)=FX(3)*IZ                                                     FOUR2180
C        PRINT BOTH ORIGINAL AND TRANSFORMED COORDINATES                 FOUR2190
  623 WRITE(OUT,22)(F8UF(J+K),K=1,2),(F8UF(J+K),K=9,I1),(FX(K),K=1,3)    FOUR2200
C        FILL IN SIN AND COS 2 PI H X                                    FOUR2210
      K=IPN3                                                             FOUR2220
      L=IPN4                                                             FOUR2230
      DO 630 I=1,IXA                                                     FOUR2240
      HY=XX*FLOAT((+IXI-1)                                               FOUR2250
      DO 630 J=1,IMAX                                                    FOUR2260
      HX=HY*FLOATIJ)                                                     FOUR2270
      K=K+I                                                              FOUR2280
      L=L+1                                                              FOUR2290
      F8UF(K)=COS(HX)                                                    FOUR2300
  630 F8UF(L)=SIN(HX)                                                    FOUR2310

C        START SECTION LOOP                                              FOUR2320
      DO 890 IZC=1,IZA                                                   FOUR2330
      J=IZC+IZI-1                                                        FOUR2340
C        PRECALCULATE COS AND SIN 2 PI L Z .                             FOUR2350
      HY=ZZ*FLOAT(J)                                                     FOUR2360
      DO 650 I=1,I8Z                                                     FOUR2370
      HX=HY*FLOAT(I-LMAX-1)                                              FOUR2380
      F8UF(IPN5+I)=COSIHX)                                               FOUR2390
  650 F8UF(IPN6+I=SIN(HX)                                                FOUR2400
C        INITIALIZE SECTION ARRAYS                                       FOUR2410
      J=IPN1+1                                                           FOUR2420
      DO 660 I=J,IPN3                                                    FOUR2430
  660 F8UF(I)=0.0                                                        FOUR2440
      F8UF(IPNI+KMAX+1)=FUOO                                             FOUR2450
C        LOOP THROUGH REFLECTION LIST                                    FOUR2460
      INOEX=-2                                                           FOUR2470
  700 INOEX=INOEX+3                                                      FOUR2480
      IF(IF8UF(INOEX))710,770,710                                        FOUR2490
C        UNPACK INOICES                                                  FOUR2500
  710 IH(2)=IF8UF(INOEX)/256                                             FOUR2510
      IH(3)=IF8UF(INOEX)-IH(2)*256-100                                   FOUR2520
      IH(1)=IH(2)/256                                                    FOUR2530
      IH(2)=IH(2)-256*IH(1)-100                                          FOUR2540
      IH(1)=IH(1)-100                                                    FOUR2550
C        LOOP THROUGH SYMMETRY OPERATIONS                                FOUR2560
      DO 760 I=1,NSYM                                                    FOUR2570
      HX=F8UF(INOEX+2)                                                   FOUR2580
      DO 720 J=1,3                                                       FOUR2590
      HX=HX-IH(J)*TRANS(J,I                                              FOUR2600
      JH(J)=0                                                            FOUR2610
      DO 720 K=1,3                                                       FOUR2620
  720 JH(J)=JH(J)+(RMAT(K,J,I)*IH(K)                                     FOUR2630
      IF(JH(1))730,750,750                                               FOUR2640
C        TRANSFORMED H.LT.0, CHANGE REFLECTION TO FRIEDEL EQUIV.         FOUR2650
  730 DO 740 J=1,3                                                       FOUR2660
  740 JH(J)=-JH(J)                                                       FOUR2670
      HX=-HX                                                             FOUR2680
  750 J=JH(1)*I8Y+JH(2)+KMAX+1                                           FOUR2690
      K=JH(3)+LMAX+1                                                     FOUR2700
      FSS=F8UF(INOEX+1)*SIN(HX)                                          FOUR2710
      FCC=F8UF(INOEX+1)*COS(HX)                                          FOUR2720
      F8UF(IPN1+J)=F8UF(IPN1+JI+FCC*F8UF(K+IPN5I+FSS*F8UF(K+IPN6I        FOUR2730
  760 F8UF(IPN2+JJ=F8UF(IPN2+JI+FSS*F8UF(K+IPN5I-FCC*F8UF(K+IPN6I       FOUR2740
      GO TO 700                                                          FOUR2750
C        ALL REFLECTIONS SUMMED, REARRANGE SECTION SUMS                  FOUR2760
  770 DO 780 I=1,I8X                                                     FOUR2770
      K=I8Y*(I-I)+KMAX+1                                                 FOUR2780
      L=K                                                                FOUR2790
      DO 780 J=I,KMAX                                                    FOUR2800
      K=K+1                                                              FOUR2810
      L=L-1                                                              FOUR2820
      HX=F8UF(IPN1+K)                                                    FOUR2830
      HY=F8UF(IPN1+L)                                                    FOUR2840
      F8UF(IPN1+K)=HX+HY                                                 FOUR2850
      F8UF(IPN1+LI=HY-HX                                                 FOUR2860
      HX=F8UF(IPN2+K)                                                    FOUR2870
      HY=F8UF(IPN2+L)                                                    FOUR2880
      F8UF((PN2+KI=HX+HY                                                 FOUR2890
```

```
  780 F8UF(IPN2+L)=HX-HY                                                FOUR2900
C         START LOOP THROUGH STRIPS ALONG X                             FOUR2910
      DO 880 IXZ=1,IXA,30                                               FOUR2920
      IXX=IXZ+IXI-1                                                     FOUR2930
      NXA=MINO(30,IXF-IXX+1)                                            FOUR2940
      NX=0                                                              FOUR2950
      DO 782 I=1,NXA,2                                                  FOUR2960
      NX=NX+1                                                           FOUR2970
  782 FX(NX)=I+IXX-1                                                    FOUR2980
      J=IZC+IZI-1                                                       FOUR2990
      WRITE(OUT,8)TITLE,J,IZ,(FX(I),I=1,NX)                             FOUR3000
C         START LOOP FOR LINES                                          FOUR3010
      DO 880 IAJ=1,IYA                                                  FOUR3020
      IPSV=I(AJ-1)*IXA+IXX-IXI+IPN9                                     FOUR3030
      FY=IAJ+IYI-1                                                      FOUR3040
      HY=YY*FY                                                          FOUR3050
      DO 785 I=1,KMAX                                                   FOUR3060
      HX=HY*FLOAT(I)                                                    FOUR3070
      F8UF(IPN5+I)=COS(HX)                                              FOUR3080
  785 F8UF(IPN6+I)=SIN(HX)                                              FOUR3090
C         LOOP OVER VALUES OF H                                         FOUR3100
      DO 790 I=1,I8X                                                    FOUR3110
      K=I8Y*(I-1)+KMAX+1                                                FOUR3120
      L=K                                                               FOUR3130
      M=IPN7+I                                                          FOUR3140
      N=IPN8+I                                                          FOUR3150
      F8UF(M)=F8UF(K+IPN1)                                              FOUR3160
      F8UF(N)=F8UF(K+IPN2)                                              FOUR3170
C         SUM OVER K                                                    FOUR3180
      DO 790 J=1,KMAX                                                   FOUR3190
      K=K+1                                                             FOUR3200
      L=L-1                                                             FOUR3210
      F8UF(M)=F8UF(M)+F8UF(IPN1+K)*F8UF(J+IPN5)+F8UF(L+IPN2)*F8UF(J+IPN6FOUR3220
     1)                                                                 FOUR3230
  790 F8UF(N)=F8UF(N)+F8UF(IPN2+K)*F8UF(J+IPN5)+F8UF(IPN1+L)*F8UF(J+IPN6FOUR3240
     1)                                                                 FOUR3250
C         START POINTS LOOP                                             FOUR3260
      L=0                                                               FOUR3270
      K=IIXX-IXI)*IMAX                                                  FOUR3280
      DO 870 I=1,NXA                                                    FOUR3290
      HX=F8UF(IPN7+I)                                                   FOUR3300
C         SUM OVER H                                                    FOUR3310
      DO 800 J=1,IMAX                                                   FOUR3320
      K=K+1                                                             FOUR3330
  800 HX=HX+F8UF(IPN7+J+1)*F8UF(IPN3+K)+F8UF(IPN8+J+1)*F8UF(IPN4+K)     FOUR3340
C         MAP POINT CALCULATED. CONVERT FOR PRINTING AND ADD TO PLOT OUTPFOUR3350
      IF(IPLT.EQ.0)GO TO 805                                            FOUR3360
      IPSV=IPSV+1                                                       FOUR3370
      F8UF(IPSV)=HX                                                     FOUR3380
  805 J=HX+SIGN(.5,HX)                                                  FOUR3390
      IF(J) 810,860,830                                                 FOUR3400
  810 IF(J+10000)820,820,850                                            FOUR3410
  820 J=J+1000                                                          FOUR3420
      GO TO 810                                                         FOUR3430
  830 IF(J-10000)850,840,840                                            FOUR3440
  840 J=J-10000                                                         FOUR3450
      GO TO 830                                                         FOUR3460
  850 L=L+1                                                             FOUR3470
      (P(L)=J                                                           FOUR3480
      FRMT(I+3)=NZRO                                                    FOUR3490
      GO TO 870                                                         FOUR3500
  860 FRMT((+3)=ZRO                                                     FOUR3510
  870 CONTINUE                                                          FOUR3520
      IF(L)872,872,876                                                  FOUR3530
  872 WRITE(OUT,6)LSPC,FY                                               FOUR3540
      GO TO 880                                                         FOUR3550
  876 FRMT((XA+4)=ENDO                                                  FOUR3560
C         PRINT LINE                                                    FOUR3570
      WRITE(OUT,FRMT)LSPC,FY,(IP(I),I=1,L)                              FOUR3580
  880 CONTINUE                                                          FOUR3590
C         SECTION NOW COMPLETE - DUMP PLOT OUTPUT IF NEEDED             FOUR3600
      IF((PLT.EQ.0)GO TO 890                                            FOUR3610
      J=IZC+IZI-1                                                       FOUR3620
      WRITE(52)IXA,(YA,J,IZ,IX,IXI,IXF,IY,IYI,IYF,IZI,IZF               FOUR3630
      J=IPN9+1                                                          FOUR3640
      WRITE(52)(F8UF(I),I=J,(PN10)                                      FOUR3650
  890 CONTINUE                                                          FOUR3660
C         END OF MAP, GO GET DATA FOR NEXT ONE                          FOUR3670
      GO TO 100                                                         FOUR3680
  900 STOP                                                              FOUR3690
      SUBROUTINE SECT(ISECT,IH)                                         FOUR3700
C                                                                       FOUR3710
      INTEGER IH(3),(TEMP                                               FOUR3720
C                                                                       FOUR3730
      IF(ISECT)25,25,5                                                  FOUR3740
    5 GO TO (10,30,50,60,70),(SECT                                      FOUR3750
C               XZY SECTIONS                                            FOUR3760
   10 ITEMP=IH(2)                                                       FOUR3770
   20 IH(2)=IH(3)                                                       FOUR3780
      IH(3)=ITEMP                                                       FOUR3790
   25 RETURN                                                            FOUR3800
C               YXZ SECTIONS                                            FOUR3810
   30 ITEMP=IH(2)                                                       FOUR3820
   40 IH(2)=IH(1)                                                       FOUR3830
      IH(1)=ITEMP                                                       FOUR3840
      RETURN                                                            FOUR3850
C               YZX SECTIONS                                            FOUR3860
   50 ITEMP=IH(1)                                                       FOUR3870
      IH(1)=IH(2)                                                       FOUR3880
      GO TO 20                                                          FOUR3890
C               ZYX SECTIONS                                            FOUR3900
   60 ITEMP=IH(1)                                                       FOUR3910
      IH(1)=IH(3)                                                       FOUR3920
      IH(3)=ITEMP                                                       FOUR3930
      RETURN                                                            FOUR3940
C               ZXY SECTIONS                                            FOUR3950
   70 ITEMP=IH(3)                                                       FOUR3960
      IH(3)=IH(2)                                                       FOUR3970
      GO TO 40                                                          FOUR3980
      END                                                               FOUR3990
```

```
C     P R O G R A M   A R B S E C T                                     ARBS  10
C                                                                       ARBS  20
C     PROGRAM TO READ A FOURIER MAP GENERATED BY THE FINGER FOURIER     ARBS  30
C     PROGRAM AND COMPUTE AN ARBITRARY SECTION.                         ARBS  40
      DIMENSION LINE(33),POINTS(3,3),A(4),TITLE(18),FY(15),JOEXI4I      ARBS  50
     1,CELL(6),JSECT(1)                                                 ARBS  60
      REAL ILINE(1)                                                     ARBS  70
      COMMON /ABLE/BUFF(1)                                              ARBS  80
      EQUIVALENCE (BUFF,ILINE),(JOEXI1),IP1),IJOEX(2),IP2),IJOEXI3),IP3)ARBS  90
     1,(JOEX(4),IP4),(BUFF,JSECTI                                       ARBS 100
      INTEGER FRMT(2)/1H ,1HO/                                          ARBS 110
      DATA IN,IOUT,ISC,ISC1/5,6,11,12/                                  ARBS 120
    1 FORMAT(1H1,18A4)                                                  ARBS 130
    2 FORMAT(18A4I                                                      ARBS 140
    3 FORMAT(10I3)                                                      ARBS 150
    4 FORMAT(1H1,18A4,'   PAGE',I3,'  OF APPARENT SECTION',I3//F10.0,14FARBS 160
     18.0)                                                              ARBS 170
    5 FORMAT('0IMPROPER SECTION SELECTION, JOB ABORTED.')               ARBS 180
    6 FORMAT('0  SECTION SAVED FOR LATER PLOTTING')                     ARBS 190
    8 FORMAT('0  FRACTIONAL COORDINATES OF CORNERS:',IT45,3F10.5))      ARBS 200
    9 FORMAT('0  LENGTH OF TRANSFORMED X =',F13.3/'   LENGTH OF TRANSFORARBS 210
     1MED Y =',F13.3/'   ANGLE BETWEEN TRANS. X AND Y =',F8.3)          ARBS 220
   10 FORMAT('   EQUATION OF SECTION IS',F10.4,' X +',F10.4,' Y +',F10.4ARBS 230
     1,' Z +',F10.4,' = 0.0')                                           ARBS 240
   12 FORMAT(9FB.0,I2)                                                  ARBS 250
   14 FORMAT(A1,I3,'. ',30I4I                                           ARBS 260
C                                                                       ARBS 270
C        INITIALIZE CORE REQUIREMENTS                                   ARBS 280
      CALL CORSZ(BUFF(1I)                                               ARBS 290
   15 READ(IN,2,END=900)TITLE                                           ARBS 300
      READ(IN,3)IO,IZO,ISPC,IMRK,IYI,IYF,IZI,IZF                        ARBS 310
      READ(IN,I2)CELL                                                   ARBS 320
      DO 1B I=4,6                                                       ARBS 330
   1B CELL(I)=COS(CELL(I)*.017453)                                      ARBS 340
      ISPC=MOD(IABS(ISPC),2)+1                                          ARBS 350
      WRITE(IOUT,1)TITLE                                                ARBS 360
      REWIND ISC                                                        ARBS 370
      READ(IN,I2)POINTS,ITP                                             ARBS 380
      INIT=-1                                                           ARBS 390
   30 READ(ISC,END=100)JXA,JYA,KZC,IZ,IX,IXI,IXF,IY,JYI,JYF,JZI,J2F     ARBS 400
      INIT=INIT+1                                                       ARBS 410
      IF(INIT.NE.0)GO TO 35                                             ARBS 420
      IF(IMRK.NE.0)GO TO 300                                            ARBS 430
      IYI=JYI                                                           ARBS 440
      I2I=J2I                                                           ARBS 450
      IYF=JYF                                                           ARBS 460
      IZF=J2F                                                           ARBS 470
  300 IF(ITP.EQ.0)GO TO 32                                              ARBS 480
C        CONVERT FRACTIONAL TO MAP COORDINATES                          ARBS 490
      DO 31 I=1,3                                                       ARBS 500
      POINTS(1,I)=POINTSI1,I)*IX                                        ARBS 510
      POINTS(2,I)=POINTSI2,I)*IY                                        ARBS 520
   31 POINTSI3,I)=POINTSI3,I)*I2                                        ARBS 530
C        CALCULATE EQUATION OF PLANE                                    ARBS 540
   32 A(1)=(POINTS(2,2I-POINTSI2,II)*(POINTSI3,3)-POINTSI3,1))-IPOINTSI3ARBS 550
     1,2)-POINTS(3,1))*IPOINTS(2,3)-POINTSI2,1))                        ARBS 560
      AI2)=(POINTS(3,2)-POINTS(3,1))*(POINTSI1,3)-POINTS(1,1))-(POINTS(1ARBS 570
     1,2)-POINTS(1,I)I*(POINTSI3,3)-POINTSI3,1))                        ARBS 580
      A(3)=(POINTS(1,2)-POINTSI1,1)I*IPOINTSI2,3I-POINTSI2,1I)-IPOINTSI2ARBS 590
     1,2)-POINTS(2,1))*IPOINTSI1,3)-POINTSI1,1)I                        ARBS 600
      AI4I=-(A(1)*POINTSI1,1)+A(2)*POINTS(2,1)+AI3)*POINTSI3,1))        ARBS 610
      WRITE(IOUT,10)A                                                   ARBS 620
      IF(IO.NE.0)WRITEIIOUT,6I                                          ARBS 630
      IF(ABSIA(1)).GT.1.0E-10IGO TO 33                                  ARBS 640
  320 WRITEIIOUT,5)                                                     ARBS 650
      STOP                                                              ARBS 660
   33 DO 20 I=2,4                                                       ARBS 670
   20 A(I)=-AII)/A(1)                                                   ARBS 680
C        COMPUTE COORDINATES OF CORNERS OF MAP                          ARBS 690
      POINTS(1,1)=(A(2)*IYI+AI3)*I2I+AI4)I/IX                           ARBS 700
      POINTSI1,2)=(A(2)*IYF+AI3)*IZI+A(4))/IX                           ARBS 710
      POINTS(1,3)=(A(2)*IYI+A(3)*IZF+AI4))/IX                           ARBS 720
      POINTS(2,1)=FLOATIIYII/FLOAT(IY)                                  ARBS 730
      POINTSI2,3)=POINTSI2,1)                                           ARBS 740
      POINTSI3,1)=FLOATII2I)/FLOATIIZ)                                  ARBS 750
      POINTS(3,2I=POINTSI3,I)                                           ARBS 760
      POINTSI2,2)=FLOATIIYFI/FLOAT(IY)                                  ARBS 770
      POINTS(3,3)=FLOATIIZF)/FLOAT(I2)                                  ARBS 780
      WRITE(IOUT,8)POINTS                                               ARBS 790
      IF(CELL(I).LE.0.0)GO TO 342                                       ARBS 800
C        COMPUTE SIZE OF TRANSFORMED SECTION                            ARBS 810
      AA=0.0                                                            ARBS 820
      BB=0.0                                                            ARBS 830
      CC=0.0                                                            ARBS 840
      DO 340 I=1,3                                                      ARBS 850
      POINTS(I,2)=(POINTSII,2I-POINTS(I,I))*CELLII)                     ARBS 860
      AA=AA+POINTSI1,2)**2                                              ARBS 870
      POINTS(I,3)=(POINTS(I,3)-POINTSII,1))*CELLII)                     ARBS 880
      BB=BB+POINTS(I,3)**2                                              ARBS 890
  340 CC=CC+POINTS(I,2)*POINTSII,3)                                     ARBS 900
      DO 341 I=1,3                                                      ARBS 910
      J=MOD(I,3I+1                                                      ARBS 920
      K=9-I-J                                                           ARBS 930
      AA=AA+2.0*CELL(K)*POINTSII,2)*POINTSIJ,2)                         ARBS 940
      BB=BB+2.0*CELL(K)*POINTSII,3)*POINTSIJ,3)                         ARBS 950
  341 CC=CC+CELL(K)*IPOINTSII,2)*POINTSIJ,3)+POINTSIJ,2)*POINTS(I,3))   ARBS 960
      AA=SQRT(AAI                                                       ARBS 970
      BB=SQRT(BB)                                                       ARBS 980
      CC=ACOS(CC/(AA*BB))*57.2957                                       ARBS 990
      WRITEIIOUT,9)AA,BB,CC                                             ARBS1000
C        COMPUTE SIZE OF SECTION AND REQUEST SUFFICIENT CORE            ARBS1010
  342 NUM=JXA*JYA                                                       ARBS1020
      I2A=IZF-IZI+1                                                     ARBS1030
      IYA=IYF-IYI+1                                                     ARBS1040
      NUM2=IYA*IZA+NUM                                                  ARBS1050
      CALL CORSZ(BUFFINUM2+IZA))                                        ARBS1060
      DO 339 II=1,IZA                                                   ARBS1070
  339 JSECTIII+NUM2)=1                                                  ARBS1080
      IFC=1                                                             ARBS1090
      IFCY=1                                                            ARBS1100
      IFC2=1                                                            ARBS1110
C        CLEAR FLAGS IF FULL CELL AVAILABLE ALONG X, Y, AND Z           ARBS1120
      IFIIXF-IXI.GE.IX-1)IFC=0                                          ARBS1130
      IFIJYF-JYI.GE.IY-1)IFCY=0                                         ARBS1140
      IFIJ2F-JZI.GE.IZ-1)IFC2=0                                         ARBS1150
```

```
C        READ SECTION                                                    ARBS1160
   35 READ(ISC)(BUFF(I),I=1,NUM)                                         ARBS1170
      DO 90 (I=1,IZA                                                     ARBS1180
      JZC=II+IZI-1                                                       ARBS1190
      IZC=JZC                                                            ARBS1200
C        GET Z INTO ACTUAL SET COMPUTED                                  ARBS1210
  343 IF(JZC.GE.JZI.AND.JZC.LE.JZF)GO TO 3 44                            ARBS1220
      IF(IFCZ.NE.0)GO TO 320                                             ARBS1230
      JZC=JZC-ISIGN(IZ,JZC)                                              ARBS1240
      GO TO 343                                                          ARBS1250
  344 IF(JZC.NE.KZC)GO TO 90                                             ARBS1260
      JSECT(II+NUM2)=0                                                   ARBS1270
      IYC=IYI                                                            ARBS1280
   40 IF(IYC.GT.IYF)GO TO 90                                             ARBS1290
      JYC=IYC                                                            ARBS1300
C        GET Y INTO ACTUAL SET COMPUTED                                  ARBS1310
   41 IF(JYC.GE.JYI.AND.JYC.LE.JYF)GO TO 42                              ARBS1320
      IF(IFCY.NE.0)GO TO 320                                             ARBS1330
      JYC=JYC-ISIGN(IY,JYC)                                              ARBS1340
      GO TO 41                                                           ARBS1350
   42 INOEX=NUM+(IZC-IZI)*IYA+IYC-IYI+1                                  ARBS1360
      X=A(2)*IYC+A(3)*IZC+A(4)                                           ARBS1370
      J=(JYC-JYI)*JXA-IXI+1                                              ARBS1380
      I=X                                                                ARBS1390
      FX=X-I                                                             ARBS1400
      JOEX(1)=I-1                                                        ARBS1410
      JOEX(2)=I                                                          ARBS1420
      JOEX(3)=I+1                                                        ARBS1430
      JOEX(4)=I+2                                                        ARBS1440
      DO 60 I=1,4                                                        ARBS1450
   50 IF(JOEX(I).GE.IXI.AND.JOEX(I).LE.IXF)GO TO 60                      ARBS1460
      IF(IFC.NE.0)GO TO 70                                               ARBS1470
      JOEX(I)=JOEX(I)-ISIGN(IX,JOEX(I))                                  ARBS1480
      GO TO 50                                                           ARBS1490
   60 JOEX(I)=J+JOEXII)                                                  ARBS1500
      AA=-ILINE(IP1)/6.+ILINE(IP2)/2.-ILINE(IP3)/2.+ILINE(IP4)/6.        ARBS1510
      BB=ILINE(IP1)/2.-ILINE(IP2)+ILINE(IP3)/2.                          ARBS1520
      CC=-ILINE(IP1)/3.-ILINE(IP2)/2.+ILINE(IP3)-ILINE(IP4)/6.           ARBS1530
      BUFF(INOEX)=((AA*FX+BB)*FX+CC)*FX+ILINE(IP2)                       ARBS1540
      GO TO 80                                                           ARBS1550
   70 BUFFIINOEX)=11111.                                                 ARBS1560
   80 IYC=IYC+1                                                          ARBS1570
      GO TO 40                                                           ARBS1580
   90 CONTINUE                                                           ARBS1590
C        CHECK IF ANY SECTIONS LEFT TO FINO                              ARBS1600
      DO 91 I1=1,IZA                                                     ARBS1610
      IFIJSECT(II+NUM2))30,91,30                                         ARBS1620
   91 CONTINUE                                                           ARBS1630
  100 N=0                                                                ARBS1640
      DO 120 I=1,IYA,30                                                  ARBS1650
      N=N+1                                                              ARBS1660
      L=MINO130,IYA-I+1)                                                 ARBS1670
      J=I+IYI-1                                                          ARBS1680
      M=0                                                                ARBS1690
      DO 105 K=1,L,2                                                     ARBS1700
      M=M+1                                                              ARBS1710
  105 FYIM)=J+K-1                                                        ARBS1720
      WRITEIIOUT,4)TITLE,N,IZO,(FY(K),K=1,M)                             ARBS1730
      DO 120 J=1,IZA                                                     ARBS1740
      K=NUM+(J-1)*IYA+(                                                  ARBS1750
      KK=K+L-1                                                           ARBS1760
      DO 110 M=K,KK                                                      ARBS1770
  110 LINE(M-K+1)=BUFF(M)+SIGN(.5,BUFF(M))                               ARBS1780
      II=J+IZI-1                                                         ARBS1790
  120 WR(TE((OUT,14)FRMT((SPC),I(,(L(NE(M),M=1,L)                        ARBS1800
      (F(IU.EO.0)GO TO 15                                                ARBS1810
      K=NUM+1                                                            ARBS1820
      L=1000                                                             ARBS1830
      WR(TE(ISC1)IYA,IZA,IZO,L,IX,IXI,IXF,IY,IYI,IYF,IZI,IZF             ARBS1840
      WRITE((SCI)(BUFF(M),M=K,NUM2)                                      ARBS1850
      GO TO 15                                                           ARBS1860
  900 STOP                                                               ARBS1870
      END                                                                ARBS1880
```

```
      DIMENSION  SBEG(10),CSTEP(10),SLIM(10),XLINE(20),YLINE(20),BX(20),CNTP  10
     2 BY(20),I1(10),I2(10),I3(10),NSIG(10)                          CNTP  20
      DIMENSION C(1),UBUFF(2000)                                     CNTP  30
      DOUBLE PRECISION LABL                                          CNTP  40
      COMMON /BI/ A,B,ALF,NA,NB,AA,BB,AX,AY,AF,BF,X1,X2,Y1,Y2,CALF,SALF,CNTP  50
     1ATITLE(20),IC(10)                                              CNTP  60
      DATA BLNK/1H /                                                 CNTP  70
      DATA IN,IOUT,IFILE/5,6,10/                                     CNTP  80
      COMMON /PLT1/ ITOT,IN,NP,(T,NT,U1NT,LUG(6),NOIG                CNTP  90
      EQUIVALENCE (C,IC)                                             CNTP 100
      INITAL=0                                                       CNTP 110
      CALL CORSZ(IC(10))                                             CNTP 120
C                                                                    CNTP 130
C     READ PROBLEM CONTROL INFORMATION                               CNTP 140
C                                                                    CNTP 150
      READ(IN,157) ATITLE,TLEHGT,NLABL,A,B,ALP,NA,NB,NP,NCONT,NL(NE,NCROCNTP 160
     1SS,SCRUSS,NSECT,NZ                                             CNTP 170
      NRUSO=NA                                                       CNTP 180
      NCUSO=NB                                                       CNTP 190
      IF(NP.LE.0)NP=4                                                CNTP 200
      REWIND IFILE                                                   CNTP 210
      WRITE(IOUT,158) ATITLE,A,B,ALP,NA,NB,NP                        CNTP 220
      IF (NCONT.GT.0) GO TO 102                                      CNTP 230
      WRITE(IOUT,160)                                                CNTP 240
      GO TO 117                                                      CNTP 250
  102 NCONT=MINO(NCONT,10)                                           CNTP 260
      DO 103 J=1,NCONT                                               CNTP 270
  103 READ(IN,179)SBEG(J),CSTEP(J),SLIM(J),NSIG(J)                   CNTP 280
  179 FORMAT(3F5.0,I5)                                               CNTP 290
      NLINE=MINO(NLINE,20)                                           CNTP 300
      IF (NLINE.LE.0) GO TO 105                                      CNTP 310
      READ(IN,162) (XL(NE(I),YL(NE(I),I=1,NLINE)                     CNTP 320
  105 NCROSS=MINO(NCROSS,20)                                         CNTP 330
      IF (NCROSS.LE.0) GO TO 106                                     CNTP 340
      READ(IN,162) (BX(I),BY(I),(=1,NCRUSS)                          CNTP 350
  106 NSECT=MINO(NSECT,10)                                           CNTP 360
      IF (NSECT.LE.0) GO TO 107                                      CNTP 370
      READ(IN,163) ((I(I),I2(I),I3(I),I=1,NSECT)                     CNTP 380
  107 WRITE(IOUT,164) NCONT,(SBEG(I),CSTEP(I),SLIM(I),NSIG(I),I=1,NCONT)CNTP 390
      IF (NLINE.LE.0) GO TO 108                                      CNTP 400
      WRITE(IOUT,165) NLINE,(XLINE(I),YLINE(I),I=1,NLINE)            CNTP 410
  108 IF (NCROSS.LE.0) GO TO 109                                     CNTP 420
      WRITE(IOUT,166) NCROSS,SCROSS,(BX(I),BY(I),I=1,NCROSS)         CNTP 430
      SCROSS=SCROSS*.5                                               CNTP 440
  109 IF (NSECT.GT.0) GO TO 110                                      CNTP 450
      WRITE((OUT,167)                                                CNTP 460
      GO TO 117                                                      CNTP 470
  110 WRITE(IOUT,168) NSECT,NZ,((I(I),(2(I),I3(I),I=1,NSECT)         CNTP 480
      ISECT=1                                                        CNTP 490
      GO TO 120                                                      CNTP 500
  111 INITAL=1                                                       CNTP 510
C * * * * * SYSTEM DEPENDENT STATEMENT FOLLOWS **************       CNTP 520
      CALL PLOTS(OBUFF,2000,12)                                      CNTP 530
      ALF=ALP*.017453                                                CNTP 540
      CALF=COS(ALF)                                                  CNTP 550
      SALF=SIN(ALF)                                                  CNTP 560
      YMAX=10.5                                                      CNTP 570
      IF(A.LT.0)YMAX=...                                             CNTP 580
      A=ABS(A)                                                       CNTP 590
      U1NT=1./FLOAT(NP+1)                                            CNTP 600
      AA=A/FLOAT(NA-1)                                               CNTP 610
      BB=B/FLOAT(NB-1)                                               CNTP 620
      AX=0.0                                                         CNTP 630
      IF(CALF)112,113,113                                            CNTP 640
  112 AX=A*ABS(CALF)                                                 CNTP 650
  113 X1=6.0*TLEHGT                                                  CNTP 660
      XFIN=X1+B+A*ABS(CALF)                                          CNTP 670
      X1=X1+AX                                                       CNTP 680
      Y1=0.0                                                         CNTP 690
      CALL PLOT(0.,.5,-3)                                            CNTP 700
      (F(A*SALF-YMAX)124,124,119                                     CNTP 710
  116 ISECT=ISECT+1                                                  CNTP 720
      IF (ISECT.GT.NSECT) GO TO 117                                  CNTP 730
      REWIND (FILE                                                   CNTP 740
      GO TO 120                                                      CNTP 750
  117 IF(INITAL.EQ.0) GO TO 146                                      CNTP 760
      GO TO 146                                                      CNTP 770
  118 WRITE(IOUT,169) JZ,NZ                                          CNTP 780
      GO TO 117                                                      CNTP 790
  119 WRITE((UUT,170) Y2,YMAX                                        CNTP 800
      GO TO 117                                                      CNTP 810
  120 JZ=(1(ISECT)                                                   CNTP 820
      J=(13(ISECT)-JZ)*12(ISECT)                                     CNTP 830
      IF (J.LT.0) GO TO 116                                          CNTP 840
      I1(ISECT)=I1(ISECT)+(2(ISECT)                                  CNTP 850
      WRITE((OUT,171) JZ,NZ                                          CNTP 860
      NROW=NRUSO                                                     CNTP 870
      NCOL=NCUSO                                                     CNTP 880
      NTOT=NCOL*NROW                                                 CNTP 890
      CALL CORSZ(IC(NTOT))                                           CNTP 900
  121 READ(IFILE,ENO=118)IPTSLN,ILNPLN,KIZ,KNZ                       CNTP 910
      IF (KIZ.NE.JZ.OR.KNZ.NE.NZ) WRITE(IOUT,172) KIZ,KNZ,JZ,NZ      CNTP 920
      IF ((PTSLN.EQ.NRUSO.AND.ILNPLN.EQ.NCUSO) GO TO 123             CNTP 930
      WRITE(IOUT,173) IPTSLN,ILNPLN,NRUSO,NCUSO                      CNTP 940
      GO TO 117                                                      CNTP 950
  123 READ(IFILE,ENO=118)(C(I),I=1,NTOT)                             CNTP 960
      IF (KIZ.NE.JZ) GO TO 121                                       CNTP 970
      IF(INITAL.EQ.0)GO TO 111                                       CNTP 980
  124 WRITE(IOUT,174) JZ,NZ                                          CNTP 990
      CALL INTPLT                                                    CNTP1000
      BMIN=10000.                                                    CNTP1010
      BMAX=-BMIN                                                     CNTP1020
      DO 127 I=(,NTOT                                                CNTP1030
         BMIN=AMIN1(BMIN,C(I))                                       CNTP1040
         BMAX=AMAX1(BMAX,C(I))                                       CNTP1050
  127 CONTINUE                                                       CNTP1060
      IF (NLINE.LE.0) GO TO 134                                      CNTP1070
      DO 131 I=1,NLINE                                               CNTP1080
      WY=ABS(XLINE(I))                                               CNTP1090
      WX=ABS(YLINE(I))                                               CNTP1100
      WX=WX*BB+WY*AA*CALF+X1                                         CNTP1110
      WY=WY*AA*SALF+Y1                                               CNTP1120
      IPEN=3                                                         CNTP1130
      IF(I.EQ.1.OR.XLINE(I).LT.0)GO TO 131                           CNTP1140
      IF(YLINE(I).LT.0)GO TO 134                                     CNTP1150
```

```
      IPEN=2                                                            CNTP1160
  131 CALL PLOT(WX,WY,IPEN)                                             CNTP1170
  134 IF (NCROSS.LE.0) GO TO 137                                        CNTP1180
      DO 136 I=1,NCROSS                                                 CNTP1190
      IF(BX(I).LT.0.0)GO TO 137                                         CNTP1200
      YY=BX(I)*BB+BY(I)*AA*CALF+X1                                      CNTP1210
      XX=BY(I)*AA*SALF+Y1                                               CNTP1220
      CALL PLOT(XX-SCROSS,YY,3)                                         CNTP1230
      CALL PLOT(XX+SCROSS,YY,2)                                         CNTP1240
      CALL PLOT(XX,YY-SCROSS,3)                                         CNTP1250
      CALL PLOT(XX,YY+SCROSS,2)                                         CNTP1260
  136 CONTINUE                                                          CNTP1270
  137 IF(TLEHGT.LE.0.0)GO TO 1369                                       CNTP1280
      DO 1360 I=1,20                                                    CNTP1290
      IF(ATITLE(21-I).NE.BLNK) GO TO 1361                               CNTP1300
 1360 CONTINUE                                                          CNTP1310
      GO TO 1369                                                        CNTP1320
 1361 NTL=21-I                                                          CNTP1330
      DO 1362 I=1,NTL                                                   CNTP1340
      TEMP=(I-1)*24.*TLEHGT/7.                                          CNTP1350
 1362 CALL SYMBOL(TLEHGT,TEMP,TLEHGT,ATITLE(I),90.,4)                   CNTP1360
      TEMP=0.5*TLEHGT                                                   CNTP1370
      CALL SYMBOL(3.0*TLEHGT,.2,TEMP,7HSECTION,90.,7)                   CNTP1380
      CALL NUMBER(3.0*TLEHGT,10.*TEMP,TEMP,FLOAT(IZ),90.,0)             CNTP1390
 1369 IF(NLABL.EQ.0)GO TO 1371                                          CNTP1400
C       LABEL THE DESIRED VERTICES                                      CNTP1410
      DO 1370 I=1,NLABL                                                 CNTP1420
      READ(IN,178)WX,WY,OX,OY,TEMP,LABL                                 CNTP1430
      XX=WX*BB+WY*AA*CALF+X1+OX                                         CNTP1440
      YY=WY*AA*SALF+Y1+OY                                               CNTP1450
 1370 CALL SYMBOL(XX,YY,TEMP,LABL,90.,6)                                CNTP1460
 1371 I=1                                                               CNTP1470
      ICC=0                                                             CNTP1480
  138 CBEG=AMIN1(SBEG(I),SLIM(I))                                       CNTP1490
      CLIM=AMAX1(SBEG(I),SLIM(I))                                       CNTP1500
      ACSTEP=AMAX1(1.,ABS(CSTEP(I)))                                    CNTP1510
      IF (CBEG.GT.BMAX.OR.CLIM.LT.BMIN) GO TO 142                       CNTP1520
  139 IF (CBEG.GE.BMIN) GO TO 140                                       CNTP1530
      Q=CBEG+ACSTEP                                                     CNTP1540
      IF (Q.GT.CLIM) GO TO 140                                          CNTP1550
      CBEG=Q                                                            CNTP1560
      GO TO 139                                                         CNTP1570
  140 IF (CLIM.LE.BMAX) GO TO 141                                       CNTP1580
      Q=CLIM-ACSTEP                                                     CNTP1590
      IF (Q.LE.CBEG) GO TO 141                                          CNTP1600
      CLIM=Q                                                            CNTP1610
      GO TO 140                                                         CNTP1620
  141 ICC=1                                                             CNTP1630
      NOIG=NSIG(I)                                                      CNTP1640
      CALL ICONT (C,NROW,NCOL,NRUSO,NCUSO,CBEG,CLIM,ACSTEP)             CNTP1650
      GO TO 143                                                         CNTP1660
  142 WRITE(IOUT,176) CBEG,CLIM,BMIN,BMAX                               CNTP1670
  143 I=I+1                                                             CNTP1680
      IF (I.LE.NCONT) GO TO 138                                         CNTP1690
      IF (ICC.GT.0) GO TO 144                                           CNTP1700
      WRITE(IOUT,177)                                                   CNTP1710
  144 INITAL=1                                                          CNTP1720
      CALL PLOTIXFIN,0.,-3)                                             CNTP1730
      IF(INITAL)146,120,120                                             CNTP1740
  146 CALL PLOT(0.,0.,999)                                              CNTP1750
      STOP                                                              CNTP1760
C                                                                       CNTP1770
  157 FORMAT(20A4/F5.0,15,3F5.0,6I5,F5.0,2I5)                           CNTP1780
  158 FORMAT(1H1,20A4/1X,10H EDGES OF ,2F7.2, 20H LNS. WITH ANGLE OF ,F  CNTP1790
     1 7.2,  9H DEGREES,/1X,21H POINTS PER EDGE AND ,I2, 32H INTERPOLATNTP1800
     2ION POINTS PER SPLINE)                                            CNTP1810
  160 FORMAT (1H0, 22H NO CONTOURS SPECIFIED)                           CNTP1820
  162 FORMAT (16F5.0)                                                   CNTP1830
  163 FORMAT (3I5)                                                      CNTP1840
  164 FORMAT(1H0,I5,14H CONTOUR TYPES,/(1X,3F8.2,I4))                   CNTP1850
  165 FORMAT (1H0,I5,  9H VERTICES,/(1X,5(4X,2F8.2)))                   CNTP1860
  166 FORMAT (1H0,I5, 24H CROSSES WITH ARM LENGTH,F6.2,  5H INS.,/(1X,5(4XCNTP1870
     1 2F8.2)))                                                         CNTP1880
  167 FORMAT (1H0, 42H NO SECTION SELECTIONS. - - JOB ABANDONED.)       CNTP1890
  168 FORMAT (1H0,I5, 44H SECTION SELECTIONS GIVEN IN INTERVALS OF 1/,I4/CNTP1900
     1 (1X,  5HFROM ,I4, 13H IN STEPS OF ,I4,  4H TO ,I4))             CNTP1910
  169 FORMAT (1H0, 33H E O F  READ WHILE SEARCHING FOR ,2I5)            CNTP1920
  170 FORMAT (1X, 35H SPECIFIED SECTION WIDTH TOO GREAT ,2F10.4)        CNTP1930
  171 FORMAT(1H0,17H PLOT OF SECTION ,I5,1H/,I5,9H SELECTED)            CNTP1940
  172 FORMAT (1X, 41H SELECTED AND READ SECTION NUMBERS DIFFER,/1X,I4,  CNTP1950
     1 7H READ, ,I4,  9H SELECTED)                                      CNTP1960
  173 FORMAT (1H0, 58H SELECTED AND READ SECTION SIZES DIFFER. - - JOB ACNTP1970
     1 BANDONED.,/1X,I4, 12H SIZE READ, ,2I4, 14H SIZE SELECTED)        CNTP1980
  174 FORMAT (1H0, 17H PLOT OF SECTION ,I4,1H/,I4,  6H BEGUN)           CNTP1990
  176 FORMAT (1H0, 25H NO MAP DENSITY IN RANGE ,2F10.2, 20H AVAILABLE RACNTP2000
     1 NGE IS ,2F10.2)                                                  CNTP2010
  177 FORMAT (1H0, 50H NO CONTOURS FOR THIS SECTION. - - NOTHING PLOTTEDCNTP2020
     1 )                                                                CNTP2030
  178 FORMAT(5F5.0,A6)                                                  CNTP2040
C                                                                       CNTP2050
      END                                                               CNTP2060
```

```
      SUBROUTINE CPLOT (IPLOT,ROW,COL,ELEV)                              CPLT  10
      COMMON /B1/ A,B,ALF,NA,NB,AA,BB,AX,AY,AF,BF,X1,X2,Y1,Y2,CALF,SALF,CPLT  20
     1 ATITLE(20)                                                        CPLT  30
      COMMON /PLT1/ ITOT,IN,NP,IT,NT,U1NT,LOG1,LOG2,LOG3,LOG4,ICOUNT,NCOCPLT  40
     1UNT,NOIG                                                          CPLT  50
      COMMON /PLT2/ C(400),XA(400),YA(400),VX(400),VY(400)              CPLT  60
      XXX=COL+BB+ROW*AA*CALF+X1                                         CPLT  70
      YYY=ROW*AA*SALF+Y1                                                CPLT  80
      IF (IPLOT-1) 101,103,106                                          CPLT  90
  101 IF (ICOUNT.LE.1) GO TO 102                                        CPLT 100
      IRET=1                                                            CPLT 110
      GO TO 107                                                         CPLT 120
  102 IF (LOG4.EQ.1) GO TO 103                                          CPLT 130
      ICOUNT=0                                                          CPLT 140
  103 IF (ICOUNT.LT.NCOUNT) GO TO 104                                   CPLT 150
      IRET=2                                                            CPLT 160
      GO TO 107                                                         CPLT 170
  104 ICOUNT=ICOUNT+1                                                   CPLT 180
      XA(ICOUNT)=XXX                                                    CPLT 190
      YA(ICOUNT)=YYY                                                    CPLT 200
C                                                                       CPLT 210
C     NOTE INTERCHANGE OF X AND Y                                       CPLT 220
C                                                                       CPLT 230
      RETURN                                                            CPLT 240
 1050 IF(NOIG.GE.0)CALL NUMBER(XXX,YYY,.1,ELEV,90.0,NOIG)               CPLT 250
      RETURN                                                            CPLT 260
  106 IRET=3                                                            CPLT 270
      IF (LOG4.EQ.1) GO TO 118                                          CPLT 280
  107 IF (LOG4.EQ.1) GO TO 117                                          CPLT 290
      X=ABS(XA(1)-XA(ICOUNT))                                           CPLT 300
      Y=ABS(YA(1)-YA(ICOUNT))                                           CPLT 310
      IF (AMAX1(X,Y).LE.1.E-6) GO TO 112                                CPLT 320
      LOG3=0                                                            CPLT 330
      X=XA(1)                                                           CPLT 340
      Y=YA(1)                                                           CPLT 350
  108 BY=Y-Y1                                                           CPLT 360
      IF (AMIN1(ABS(BY),ABS(Y-Y2)).LE.1.E-4) GO TO 110                  CPLT 370
      BY=BY*CALF/SALF                                                   CPLT 380
      IF (AMIN1(ABS(X-X1-BY),ABS(X-X2-BY)).LT.1.E-4) GO TO 110          CPLT 390
      IF (LOG3.GT.0) GO TO 116                                          CPLT 400
      LOG3=1                                                            CPLT 410
  109 X=XA(ICOUNT)                                                      CPLT 420
      Y=YA(ICOUNT)                                                      CPLT 430
      GO TO 108                                                         CPLT 440
  110 IF (LOG3.EQ.2) GO TO 116                                          CPLT 450
      IF (LOG3.EQ.1) GO TO 111                                          CPLT 460
      LOG3=2                                                            CPLT 470
      GO TO 109                                                         CPLT 480
  111 LOG4=1                                                            CPLT 490
      ISAVE=ICOUNT                                                      CPLT 500
      GO TO (102,104,1050,123),IRET                                     CPLT 510
  112 VX(ICOUNT)=(XA(2)-XA(ICOUNT-1))*0.5*UINT                          CPLT 520
      VX(1)=VX(ICOUNT)                                                  CPLT 530
      VY(ICOUNT)=(YA(2)-YA(ICOUNT-1))*0.5*UINT                          CPLT 540
      VY(1)=VY(ICOUNT)                                                  CPLT 550
  113 UA=AA*UINT*0.7                                                    CPLT 560
      UB=BB*UINT*0.7                                                    CPLT 570
      VA=XA(1)                                                          CPLT 580
      VB=YA(1)                                                          CPLT 590
      J=1                                                               CPLT 600
      DO 115 I=2,ICOUNT                                                 CPLT 610
         IF (ABS(XA(I)-VA).GT.UA) GO TO 114                             CPLT 620
         IF (ABS(YA(I)-VB).GT.UB) GO TO 114                             CPLT 630
         GO TO 115                                                      CPLT 640
  114    J=J+1                                                          CPLT 650
         VA=XA(I)                                                       CPLT 660
         XA(J)=VA                                                       CPLT 670
         VB=YA(I)                                                       CPLT 680
         YA(J)=VB                                                       CPLT 690
  115 CONTINUE                                                          CPLT 700
      XA(J)=XA(ICOUNT)                                                  CPLT 710
      YA(J)=YA(ICOUNT)                                                  CPLT 720
      VX(J)=VX(ICOUNT)                                                  CPLT 730
      VY(J)=VY(ICOUNT)                                                  CPLT 740
      IN=J                                                              CPLT 750
      IF (IN.GT.2) CALL PREPLT                                          CPLT 760
      XA(1)=XA(ICOUNT)                                                  CPLT 770
      YA(1)=YA(ICOUNT)                                                  CPLT 780
      ICOUNT=1                                                          CPLT 790
      GO TO (102,104,1050,123),IRET                                     CPLT 800
  116 VX(1)=(XA(2)-XA(1))*UINT                                          CPLT 810
      VY(1)=(YA(2)-YA(1))*UINT                                          CPLT 820
      VX(ICOUNT)=(XA(ICOUNT)-XA(ICOUNT-1))*UINT                         CPLT 830
      VY(ICOUNT)=(YA(ICOUNT)-YA(ICOUNT-1))*UINT                         CPLT 840
      GO TO 113                                                         CPLT 850
  117 IF (ISAVE.NE.ICOUNT) GO TO 118                                    CPLT 860
      GO TO (102,104,1050,123),IRET                                     CPLT 870
  118 LOG4=0                                                            CPLT 880
      X=ABS(XA(1)-XA(ISAVE+1))                                          CPLT 890
      Y=ABS(YA(1)-YA(ISAVE+1))                                          CPLT 900
      IF (AMAX1(X,Y).GT.1.E-4) GO TO 122                                CPLT 910
      I=ISAVE/2                                                         CPLT 920
      K=ISAVE                                                           CPLT 930
      DO 119 J=1,I                                                      CPLT 940
         X=XA(J)                                                        CPLT 950
         XA(J)=XA(K)                                                    CPLT 960
         XA(K)=X                                                        CPLT 970
         Y=YA(J)                                                        CPLT 980
         YA(J)=YA(K)                                                    CPLT 990
         YA(K)=Y                                                        CPLT1000
  119 K=K-1                                                             CPLT1010
      ICOUNT=ICOUNT-I                                                   CPLT1020
      K=ISAVE+1                                                         CPLT1030
  120 DO 121 J=ISAVE,ICOUNT                                             CPLT1040
         XA(J)=XA(K)                                                    CPLT1050
         YA(J)=YA(K)                                                    CPLT1060
  121 K=K+1                                                             CPLT1070
      ISAVE=0                                                           CPLT1080
      GO TO 116                                                         CPLT1090
  122 KRET=IRET                                                         CPLT1100
      KCOUNT=ICOUNT                                                     CPLT1110
      KSAVE=ISAVE                                                       CPLT1120
      ICOUNT=KSAVE                                                      CPLT1130
      IRET=4                                                            CPLT1140
      GO TO 116                                                         CPLT1150
```

```
  123 IRET=KRET                                                          CPLT1160
      ICOUNT=KCOUNT-KSAVE                                                CPLT1170
      K=KSAVE+1                                                          CPLT1180
      ISAVE=1                                                            CPLT1190
      GO TO 120                                                          CPLT1200
C                                                                        CPLT1210
      END                                                                CPLT1220
```

```
      SUBROUTINE ICONT (DATA,NROW,NCOL,NRUSO,NCUSO,CBEG,CLIM,ACSTP)      ICNT  10
C                                                                        ICNT  20
C     CONTOURING SUBPROGRAM BY S. H. ZISK, MIT LINCOLN LAB, MARCH 1968.  ICNT  30
C     REWRITTEN IN USASI BASIC FORTRAN BY NM BRENNER MIT LL, MARCH 1968  ICNT  40
C     DO 4000 XLEV=CBEG,CLIM,CSTP  (EFFECTIVELY, NOT LITERALLY!          ICNT  50
C     CSTP IS ACSTP WITH THE SIGN CHOSEN TO INCREASE IN THE CORRECT      ICNT  60
C     DIRECTION (FROM CBEG TO CLIM).                                     ICNT  70
C     FOR EACH XLEV, PLOTS A CONTOUR OF DATA(I,J), OUTPUT BEING RETURNED ICNT  80
C     VIA SUCCESSIVE CALLS TO SUBROUTINES.  CPLOT(0 OR 1,...) FOR PEN    ICNT  90
C     UP OR PEN DOWN TRACES.  CPLOT(2,...) FOR LABELING AT THIS POSITION ICNT 100
C     IF DESIRED.  FULL CALL IS CALL CPLOT(IPLOT,ROW,COL,XLEV), WHERE    ICNT 110
C     ROW IS FROM 1. TO FLOAT(MAXX), COL IS FROM 1. TO FLOAT(MAXY).      ICNT 120
C     CONTR EXTRUDES ONE POINT AT A TIME TO CPLOT.  IF DESIRED FOR       ICNT 130
C     EFFICIENCY, CPLOT MAY SAVE UP POINTS IN A SCRATCH ARRAY.           ICNT 140
C     WHEN IPLOT IS 0, CPLOT SHOULD JUST REMEMBER THIS (ROW,COL) POINT.  ICNT 150
C     WHEN IPLOT=1, CPLOT SHOULD DRAW A LINE SEGMENT FROM THE REMEMBER-  ICNT 160
C     ED POINT TO THE NEW POINT, AND THEN MAKE THE NEW PUINT THE REMEM-  ICNT 170
C     BERED POINT.                                                       ICNT 180
C     A CONTOUR LABEL IS CALLED FROM CPLOT (IPLOT=2) ONLY ONCE PER       ICNT 190
C     COMPLETE CONTOUR SEGMENT, SO THAT IF SEVERAL DIFFERENT LOOPS ARE   ICNT 200
C     AT HEIGHT XLEV, THEY WILL EACH GET A LABEL.                        ICNT 210
C                                                                        ICNT 220
C     THIS PROGRAM IS WRITTEN IN USASI BASIC FORTRAN IN AN ATTEMPT TO BE ICNT 230
C     COMPATIBLE WITH AS MANY EXISTING FORTRAN COMPILERS AS POSSIBLE.    ICNT 240
C     AS WRITTEN, THIS PROGRAM ASSUMES DATA IS A FLOATING POINT ARRAY.   ICNT 250
C                                                                        ICNT 260
      DIMENSION DATA(1)                                                  ICNT 270
      REAL MARK                                                          ICNT 280
C                                                                        ICNT 290
C     DIMENSION DATA(NROW,NCOL), OF WHICH ONLY (MAXX,MAXY) ARE USED.     ICNT 300
C     NOTE--THE LOW ORDER BIT OF EACH ENTRY DATA(I,J) IS LIABLE TO CHANG ICNT 310
C     THIS WILL NEGLIGIBLY AFFECT A FLOATING POINT ARRAY.                ICNT 320
C                                                                        ICNT 330
C     CORRECTED AND MODIFIED  FEB. 1970.  G F BIO. SC.                   ICNT 340
C                                                                        ICNT 350
C     IRT AND IUP INDICATE THE DIRECTION OF THE LAST CONTOUR STEP.       ICNT 360
C     IRT=0,1 INDICATES THAT THE LAST STEP HAD X DECREASING,INCREASING A ICNT 370
C     IUP=0,1 INDICATES THAT THIS STEP HAD Y DECREASING, INCREASING .    ICNT 380
C     DIRECTION OF THE NEXT CONTOUR STEP IS DETERMINED BY THE VALUES AT  ICNT 390
C     A,B,C,AND D WITH THE LAST STEP ENDING ON DA OR AT D (SPECIAL CASE) ICNT 400
C     AND A,B,C,D THE VERTICES OF A SQUARE LABELLED IN ORDER.            ICNT 410
C     A TEST BASED ON THE VALUE DIE (AN ESTIMATE OF THE VALUE AT THE CEN ICNT 420
C     OF THE SQUARE) WAS ADDED TO MAKE THE CONTOURING MORE CONSISTENT.   ICNT 430
C                                                                        ICNT 440
      MAXX=NRUSO                                                         ICNT 450
      MAXY=NCUSO                                                         ICNT 460
      CSTP=SIGN(ACSTP,CLIM-CBEG)                                         ICNT 470
      CTR=CBEG                                                           ICNT 480
C       CLEAR LEAST SIGNIFICANT BIT                                      ICNT 490
  101 XLEV=UNMARK(CTR)                                                   ICNT 500
      JLOOP=1                                                            ICNT 510
      JUP=JLOOP                                                          ICNT 520
      IX=1                                                               ICNT 530
      IY=1                                                               ICNT 540
      NTOT=NROW*MAXY                                                     ICNT 550
      DO 102 I=1,NTOT                                                    ICNT 560
  102 DATA(I)=UNMARK(DATA(I))                                            ICNT 570
```

```
C                                                                       ICNT 580
C     THE LEAST SIGNIFICANT BIT OF DATA(I) IS USED TO MARK THE LEFTHAND ICNT 590
C     EDGE OF THE CONTOUR WHEN IT CROSSES THAT ROW.                     ICNT 600
C     SRCHX SCANS THE ROWS FROM LEFT TO RIGHT, AND BOTTOM TO TOP.       ICNT 610
C     LOOKING FOR THE FIRST POINT ON THE XLEV CONTOUR.                  ICNT 620
C                                                                       ICNT 630
  103 IUP=JUP                                                           ICNT 640
      ILOOP=JLOOP                                                       ICNT 650
      IRT=1                                                             ICNT 660
      IDUMP=0                                                           ICNT 670
      IF (IX-MAXX) 104,131,212                                          ICNT 680
  104 L=IX+NROW*(IY-1)                                                  ICNT 690
C                                                                       ICNT 700
C     IX,IY SAVE THE LAST POINT THAT WAS SEARCHED, TO RESTART THE       ICNT 710
C     SEARCH AFTER FINISHING THE CURRENT CONTOUR LINE.                  ICNT 720
C                                                                       ICNT 730
  105 DIA1=DATA(L)                                                      ICNT 740
      DJA1=UNMARK(DIA1)                                                 ICNT 750
      IF (DJA1-XLEV) 106,116,106                                        ICNT 760
  106 IF (IX-1) 213,107,116                                             ICNT 770
  107 IF (IY-1) 214,112,108                                             ICNT 780
  108 LMX=L-NROW                                                        ICNT 790
      DJA2=DATA(LMX)                                                    ICNT 800
      DID=UNMARK(DJA2)                                                  ICNT 810
      IF (DID-XLEV) 111,109,111                                         ICNT 820
  109 IF (DID-DJA2) 111,110,111                                         ICNT 830
  110 YPT=IY-1                                                          ICNT 840
      GO TO 115                                                         ICNT 850
  111 IF (IY-MAXY) 112,116,215                                          ICNT 860
C                                                                       ICNT 870
C     LOOKING FOR AN INTERSECTION IN THE LEFT-HAND COLUMN BEFORE        ICNT 880
C     SCANNING FOR INTERSECTIONS IN THE CURRENT ROW.                    ICNT 890
C                                                                       ICNT 900
  112 LPX=L+NROW                                                        ICNT 910
      DID=UNMARK(DATA(LPX))                                             ICNT 920
      IF (DID-XLEV) 113,116,113                                         ICNT 930
  113 IF ((DID-XLEV)*(DJA1-XLEV)) 114,114,116                           ICNT 940
C                                                                       ICNT 950
C     ARE BOTH DID AND DJA1 ON THE SAME SIDE OF THE CONTOUR             ICNT 960
C                                                                       ICNT 970
  114 DEL=DID-DJA1                                                      ICNT 980
      IF(ABS(DEL).LT.1.0E-4)DEL=SIGN(1.0E-4,DEL)                        ICNT 990
      YPT=FLOAT(IY)+(XLEV-DJA1)/DEL                                     ICNT1000
  115 XPT=1.                                                            ICNT1010
      X1=XPT                                                            ICNT1020
      Y1=YPT                                                            ICNT1030
      ILOOP=0                                                           ICNT1040
      IF(DID-DATA(LPX))130,138,130                                      ICNT1050
C                                                                       ICNT1060
C     IF TWO ROWS ARE EACH FILLED WITH HIGH AND LOW NUMBERS RESPECTIVELYICNT1070
C     WE HAVE JUST FOUND THE INTER-ROW CONTOUR.                         ICNT1080
C                                                                       ICNT1090
  116 DIA2=DATA(L+1)                                                    ICNT1100
      DJA2=UNMARK(DIA2)                                                 ICNT1110
      IF (DIA1-DJA1) 130,117,130                                        ICNT1120
  117 IF (DJA1-XLEV) 129,118,128                                        ICNT1130
  118 IF (DJA2-XLEV) 119,121,119                                        ICNT1140
  119 XPT=IX                                                            ICNT1150
      X1=XPT                                                            ICNT1160
      IF (IX-1) 216,120,137                                             ICNT1170
  120 ILOOP=0                                                           ICNT1180
      GO TO 137                                                         ICNT1190
  121 IF (IX-MAXX+1) 130,122,122                                        ICNT1200
  122 IF (IY-1) 217,130,123                                             ICNT1210
C                                                                       ICNT1220
C     FINDING THE LOWER OR UPPER CORNER OF A PLATEAU THAT HITS THE      ICNT1230
C     RIGHT EDGE.                                                       ICNT1240
C                                                                       ICNT1250
  123 IF (DIA2-DJA2) 131,124,131                                        ICNT1260
  124 L=L+1                                                             ICNT1270
      LPX=L+NROW                                                        ICNT1280
      LMX=L-NROW                                                        ICNT1290
      DJA2=UNMARK(DATA(LPX))                                            ICNT1300
      DIA2=UNMARK(DATA(LMX))                                            ICNT1310
      IF (DIA2-XLEV) 125,126,125                                        ICNT1320
  125 IUP=0                                                             ICNT1330
      GO TO 127                                                         ICNT1340
  126 IF (DJA2-XLEV) 127,131,127                                        ICNT1350
C                                                                       ICNT1360
C     IF BOTH UPPER AND LOWER POINTS ARE ALSO ON, THEN DUMP THIS ONE.   ICNT1370
C                                                                       ICNT1380
  127 IRT=0                                                             ICNT1390
      ILOOP=0                                                           ICNT1400
      XPT=MAXX                                                          ICNT1410
      X1=XPT                                                            ICNT1420
      GO TO 137                                                         ICNT1430
C                                                                       ICNT1440
C     SKIPS ANY BLOCK OF ALL XLEV                                       ICNT1450
C     FOR 1ST POINT GREATER THAN XLEV                                   ICNT1460
C                                                                       ICNT1470
  128 IF (XLEV-DJA2) 130,130,136                                        ICNT1480
C                                                                       ICNT1490
C     FOR 1ST POINT LESS THAN XLEV                                      ICNT1500
C                                                                       ICNT1510
  129 IF (XLEV-DJA2) 136,130,130                                        ICNT1520
C                                                                       ICNT1530
C     OMITS ANY RIGHT HAND EDGE COLUMN OF ALL XLEV                      ICNT1540
C                                                                       ICNT1550
  130 IF (IX-MAXX+1) 135,131,131                                        ICNT1560
C                                                                       ICNT1570
C     IF IX = MAXX-1 WE JUST SCANNED THE LAST COLUMN-GAP...MUST SWITCH  ICNT1580
C     ROWS.                                                             ICNT1590
C                                                                       ICNT1600
  131 IF (IY-MAXY) 133,132,218                                          ICNT1610
C                                                                       ICNT1620
C     IF IY = MAXY, WE ARE AT THE TOP ROW ALSO AND CONTOURING IS DONE.  ICNT1630
C                                                                       ICNT1640
  132 IDUMP=1                                                           ICNT1650
      GO TO 207                                                         ICNT1660
  133 IX=1                                                              ICNT1670
      IY=IY+1                                                           ICNT1680
      L=IX+NROW*(IY-1)                                                  ICNT1690
      IF (IY-MAXY) 105,134,219                                          ICNT1700
  134 JLOOP=0                                                           ICNT1710
      ILOOP=JLOOP                                                       ICNT1720
      JUP=ILOOP                                                         ICNT1730
```

```
      IUP=JUP                                                          ICNT1740
C                                                                      ICNT1750
C     FOR LAST ROW ONLY                                                ICNT1760
C                                                                      ICNT1770
      GO TO 105                                                        ICNT1780
  135 IX=IX+1                                                          ICNT1790
      L=L+1                                                            ICNT1800
      DJA1=DJA2                                                        ICNT1810
      DIA1=DIA2                                                        ICNT1820
      GO TO 116                                                        ICNT1830
  136 DEL=DJA2-DJA1                                                    ICNT1840
      IF(ABS(DEL).LT.1.0E-4)DEL=SIGN(1.0E-4,DEL)                       ICNT1850
      XPT=FLOAT(IX)+IXLEV-DJA1)/DEL                                    ICNT1860
      X1=XPT                                                           ICNT1870
C       SET LEAST SIGNIFICANT BIT                                      ICNT1880
  137 DATA(L)=MARK(DATA(L))                                            ICNT1890
C                                                                      ICNT1900
C     MARK DATA(L)                                                     ICNT1910
C                                                                      ICNT1920
      YPT=IY                                                           ICNT1930
      Y1=YPT                                                           ICNT1940
  138 CALL CPLOT IO,XPT,YPT,XLEV)                                      ICNT1950
      IX=IX+1                                                          ICNT1960
      IF (IY-1) 220,139,140                                            ICNT1970
  139 ILOOP=0                                                          ICNT1980
C                                                                      ICNT1990
C     TO START THE NEXT SEARCH AT A NEW POINT.                         ICNT2000
C                                                                      ICNT2010
  140 IF (IOUMP) 207,141,207                                           ICNT2020
C                                                                      ICNT2030
C     X1, Y1 ARE TO SAVE THE STARTING POINT OF THIS CONTOUR LINE.      ICNT2040
C     ILOOP=0 IF WE START ON AN EDGE, AND 1 IF OTHERWISE.              ICNT2050
C     CONPT FINDS THE NEXT POINT ON THE CONTOUR, WORKING FROM THE LAST ICNT2060
C     POINT THAT WAS FOUND AND THE GENERAL DIRECTION IUP OR DOWN, RIGHT ICNT2070
C     OR LEFT) THAT THE CONTOUR IS PROGRESSING.  IT ALSO NOTES ANY     ICNT2080
C     CHANGE IN THIS DIRECTION AND STORES IT FOR THE NEXT CALCULATION. ICNT2090
C                                                                      ICNT2100
  141 IOENT=-1                                                         ICNT2110
      IXPT=XPT                                                         ICNT2120
      IYPT=YPT                                                         ICNT2130
      IF (YPT-FLOAT(IYPT)) 142,143,142                                 ICNT2140
C                                                                      ICNT2150
C     GO TO 2030 IF IT'S ON A ROW.                                     ICNT2160
C                                                                      ICNT2170
  142 IYA=IYPT+IUP                                                     ICNT2180
      IYB=IYA                                                          ICNT2190
      IYC=IYPT+1-IUP                                                   ICNT2200
      IXA=IXPT                                                         ICNT2210
      IXB=IXPT-1+IRT+IRT                                               ICNT2220
      IXC=IXB                                                          ICNT2230
      IINO=I                                                           ICNT2240
      IO=IXA+NROW*IIYC-1)                                              ICNT2250
      GO TO 148                                                        ICNT2260
  143 IF (XPT-FLOAT(IXPT)) 144,145,144                                 ICNT2270
C                                                                      ICNT2280
C     GO TO 2050 IF IT'S ON A ROW AND ON A COLUMN.                     ICNT2290
C                                                                      ICNT2300
  144 IXA=IXPT+IRT                                                     ICNT2310
      GO TO 147                                                        ICNT2320
  145 IOENT=0                                                          ICNT2330
  146 IXA=IXPT-1+IRT+IRT                                               ICNT2340
  147 IXB=IXA                                                          ICNT2350
      IXC=IXA+1-IRT-IRT                                                ICNT2360
      IYA=IYPT                                                         ICNT2370
      IYB=IYPT-1+IUP+IUP                                               ICNT2380
      IYC=IYB                                                          ICNT2390
      IINO=2                                                           ICNT2400
      IO=IXC+NROW*IIYA-1)                                              ICNT2410
  148 IA=IXA+NROW*IIYA-1)                                              ICNT2420
      IB=IXB+NROW*IIYB-1)                                              ICNT2430
      IC=IXC+NROW*IIYC-1)                                              ICNT2440
      OIA=UNMARK(DATA(IA))                                             ICNT2450
      OIB=UNMARK(DATA(IB))                                             ICNT2460
      OIC=UNMARK(DATA(IC))                                             ICNT2470
      OID=UNMARK(DATA(IO))                                             ICNT2480
      DIE=DIA+DIB+DIC+OID)*0.25                                        ICNT2490
      IF (DIA-XLEV) 149,156,161                                        ICNT2500
C                                                                      ICNT2510
C     WE TEST IN ORDER DIA,DIC,DIB AND (POSSIBLY) DIE TO DETERMINE THE ICNT2520
C     NEXT POINT ON THE CONTOUR.  DIE TEST IS TO TRY TO PREVENT THE    ICNT2530
C     CONTOUR CROSSING A SADDLE OR TROUGH.                             ICNT2540
C                                                                      ICNT2550
  149 IF (DIC-XLEV) 150,153,155                                        ICNT2560
  150 IF (DIB-XLEV) 168,151,152                                        ICNT2570
  151 IF (DIE-XLEV) 190,180,180                                        ICNT2580
  152 IF (DIE-XLEV) 190,184,184                                        ICNT2590
  153 IF (DIB-XLEV) 181,180,154                                        ICNT2600
  154 IF (DIE-XLEV) 181,184,184                                        ICNT2610
  155 IF (DIB-XLEV) 187,180,184                                        ICNT2620
  156 IF (DIC-XLEV) 157,159,160                                        ICNT2630
  157 IF (DIB-XLEV) 179,180,158                                        ICNT2640
  158 IF (DIE-XLEV) 179,187,187                                        ICNT2650
  159 IF (DIB-XLEV) 179,168,179                                        ICNT2660
  160 IF (DIB-XLEV) 158,180,179                                        ICNT2670
  161 IF (DIC-XLEV) 162,163,165                                        ICNT2680
  162 IF (DIB-XLEV) 184,180,187                                        ICNT2690
  163 IF (DIB-XLEV) 164,181,181                                        ICNT2700
  164 IF (DIE-XLEV) 184,181,181                                        ICNT2710
  165 IF (DIB-XLEV) 166,167,168                                        ICNT2720
  166 IF (DIE-XLEV) 184,190,190                                        ICNT2730
  167 IF (DIE-XLEV) 180,190,190                                        ICNT2740
  168 IF (IDENT) 190,169,170                                           ICNT2750
  169 IRT=1-IRT                                                        ICNT2760
      GO TO 174                                                        ICNT2770
  170 IF (IOENT-2) 171,173,173                                         ICNT2780
  171 IRT=1-IRT                                                        ICNT2790
      IUP=1-IUP                                                        ICNT2800
      GO TO 177                                                        ICNT2810
  172 IDENT=IDENT+1                                                    ICNT2820
      GO TO 146                                                        ICNT2830
  173 IOUMP=3                                                          ICNT2840
      GO TO 194                                                        ICNT2850
C                                                                      ICNT2860
C     X-LIMIT CHECK                                                    ICNT2870
C                                                                      ICNT2880
  174 IF (IXPT+IRT-2) 176,172,175                                      ICNT2890
```

```
  175 IF (IXPT+IRT-MAXX) 172,172,176                                ICNT2900
  176 IDENT=IDENT+2                                                 ICNT2910
      GO TO 171                                                     ICNT2920
C                                                                   ICNT2930
C     Y-LIMIT CHECK                                                 ICNT2940
C                                                                   ICNT2950
  177 IF (IYPT+IUP-2) 173,172,178                                   ICNT2960
  178 IF (IYPT+IUP-MAXY) 172,172,173                                ICNT2970
  179 XPT=IXA                                                       ICNT2980
      YPT=IYA                                                       ICNT2990
      GO TO 194                                                     ICNT3000
  180 XPT=IX8                                                       ICNT3010
      YPT=IY8                                                       ICNT3020
      GO TO 194                                                     ICNT3030
  181 XPT=IXC                                                       ICNT3040
      YPT=IYC                                                       ICNT3050
      IF (IDENT) 182,194,194                                        ICNT3060
C                                                                   ICNT3070
C     DON'T CHANGE DIRECTION IF IT'S AN ..ON-TO-ON..                ICNT3080
C                                                                   ICNT3090
  182 IF (IIND-1) 183,192,183                                       ICNT3100
  183 IRT=1-IRT                                                     ICNT3110
      GO TO 194                                                     ICNT3120
  184 DEL=DIB-DIA                                                   ICNT3130
      IF(ABS(DEL).LT.1.0E-4)DEL=SIGN(1.0E-4,DEL)                    ICNT3140
      DEL=(XLEV-DIA)/DEL                                            ICNT3150
      IF (IIND-2) 185,186,185                                       ICNT3160
  185 XPT=FLOAT(IXA)+DEL*FLOAT(IRT+IRT-1)                           ICNT3170
      YPT=IYA                                                       ICNT3180
      GO TO 194                                                     ICNT3190
  186 YPT=FLOAT(IYA)+DEL*FLOAT(IUP+IUP-1)                           ICNT3200
      XPT=IXA                                                       ICNT3210
      GO TO 194                                                     ICNT3220
  187 DEL=DI8-DIC                                                   ICNT3230
      IF(ABS(DEL).LT.1.0E-4)DEL=SIGN(1.0E-4,DEL)                    ICNT3240
      DEL=(XLEV-DIC)/DEL                                            ICNT3250
      IF (IIND-2) 188,189,188                                       ICNT3260
  188 YPT=FLOAT(IYC)+DEL*FLOAT(IUP+IUP-1)                           ICNT3270
      XPT=IX8                                                       ICNT3280
      GO TO 194                                                     ICNT3290
  189 XPT=FLOAT(IXC)+DEL*FLOAT(IRT+IRT-1)                           ICNT3300
      YPT=IY8                                                       ICNT3310
      GO TO 194                                                     ICNT3320
  190 DEL=DID-DIC                                                   ICNT3330
      IF(ABS(DEL).LT.1.0E-4)DEL=SIGN(1.0E-4,DEL)                    ICNT3340
      DEL=(XLEV-DIC)/DEL                                            ICNT3350
      IF (IIND-2) 191,193,191                                       ICNT3360
  191 XPT=FLOAT(IXC)-DEL*FLOAT(IRT+IRT-1)                           ICNT3370
      YPT=IYC                                                       ICNT3380
  192 IUP=1-IUP                                                     ICNT3390
      GO TO 194                                                     ICNT3400
  193 YPT=FLOAT(IYC)-DEL*FLOAT(IUP+IUP-1)                           ICNT3410
      XPT=IXC                                                       ICNT3420
      GO TO 183                                                     ICNT3430
  194 CALL CPLOT (1,XPT,YPT,XLEV)                                   ICNT3440
C                                                                   ICNT3450
C     PLOTS A LINE TO THIS LOCATION.                                ICNT3460
C                                                                   ICNT3470
      IF (IOUMP-3) 195,206,206                                      ICNT3480
  195 IYPT=YPT                                                      ICNT3490
      I=IF(X(XPT)+NROW*(IYPT-1)                                     ICNT3500
     (F(ABS(YPT-FLOAT(IYPT))-0.01)196,196,198                       ICNT3510
C                                                                   ICNT3520
C     IF YPT IS AN INTEGER, MARK  THE POINT IMMEDI-                 ICNT3530
C     ATELY TO THE LEFT.  IF IT IS ALREADY MARKED, THIS             ICNT3540
C     HAS BEEN PLOTTED BEFORE.                                      ICNT3550
C                                                                   ICNT3560
  196 IFIDATA(I)-UNMARK(DATAI(I)))202,197,202                       ICNT3570
C                                                                   ICNT3580
C     IF THIS PUINT HAS 8EEN TRAVERSED 8EFURE...                    ICNT3590
C                                                                   ICNT3600
  197 DATA(I)=MARK(DATA(I))                                         ICNT3610
C                                                                   ICNT3620
C     IF NOT, KEYS THIS POINT FOR FUTURE REFERENCE.                 ICNT3630
C                                                                   ICNT3640
  198 IF(XPT-1.0)221,2040,199                                       ICNT3650
  199 IF (XPT-FLOAT(MAXX)) 200,204,222                              ICNT3660
  200 IF (YPT-1.) 223,204,201                                       ICNT3670
  201 IF IYPT-FLOAT(MAXY)) 141,204,224                              ICNT3680
  202 IF (XPT-X1) 204,203,204                                       ICNT3690
  203 IF IYPT-Y1) 204,206,204                                       ICNT3700
C                                                                   ICNT3710
C     IF CONTOUR HAS COME FULL CIRCLE, THAT'S ALL...                ICNT3720
C                                                                   ICNT3730
 2040 DATA(I)=MARK(DATA(I))                                         ICNT3740
  204 IF ((LOOP-1) 206,205,206                                      ICNT3750
C                                                                   ICNT3760
C     IF THE CONTOUR HITS EITHER IANY EDGE) OR (ITSELF IN THE MIDDLE),  ICNT3770
C     IT RETURNS ONCE TO THE STARTING POINT AND BEGINS CONTUURING IN    ICNT3780
C     THE OTHER DIRECTION.  IF THE STARTING PUINT IS ON AN EDGE, IT'S T ICNT3790
C     THE END OF THE CONTOUR LINE.                                  ICNT3800
C                                                                   ICNT3810
  205 IRT=0                                                         ICNT3820
      IUP=0                                                         ICNT3830
      ILOOP=2                                                       ICNT3840
      XPT=XI                                                        ICNT3850
      YPT=YI                                                        ICNT3860
      CALL CPLOT (0,XPT,YPT,XLEV)                                   ICNT3870
C                                                                   ICNT3880
C     MOVES THE PEN 8ACK TO STARTING POINT WITHOUT PLOTTING.        ICNT3890
C                                                                   ICNT3900
      GO TO 141                                                     ICNT3910
  206 CALL CPLUT I2,XPT,YPT,XLEV)                                   ICNT3920
C                                                                   ICNT3930
C     TO NUMBER EACH SECTIUN OF A CONTOUR                           ICNT3940
C                                                                   ICNT3950
      GO TO 103                                                     ICNT3960
C                                                                   ICNT3970
C     IOUMP=0, MORE TO CONTOUR...1, END OF MAP, NO MORE...          ICNT3980
C     3, ONE-POINT CONTOUR.                                         ICNT3990
C                                                                   ICNT4000
  207 IF ICSTP) 208,210,208                                         ICNT4010
  208 IF IICTR+CSTP-CLIM)*SIGNI1.,CSTP)-1.E-5*ABS(CSTP)) 209,209,210  ICNT4020
  209 CTR=CTR+CSTP                                                  ICNT4030
      GO TO 101                                                     ICNT4040
  210 CALL CPLOT I0,0.,0.,0.)                                       ICNT4050
```

```
C                                                                       (CNT4060
C     MAKE SURE ALL POINTS HAVE BEEN PLOTTED                            (CNT4070
C                                                                       ICNT4080
      RETURN                                                            (CNT4090
C                                                                       ICNT4100
C     TRAPS TO OUTPUT A WARNING IF ANY BOUNDARY IS CROSSED.             (CNT4110
C                                                                       ICNT4120
  211 WRITE(6,226)IT,(X,MAXX,IY,MAXY,XPT,YPT                            ICNT4130
      STOP                                                              ICNT4140
  212 IT=101                                                            ICNT4150
      WR(TE(6,226) IT,IX,MAXX,IY,MAXY                                   ICNT4160
      GO TO 131                                                         ICNT4170
  213 IT=104                                                            ICNT4180
      WRITE(6,226) IT,IX,MAXX,IY,MAXY                                   ICNT4190
      GO TO 116                                                         ICNT4200
  214 IT=105                                                            ICNT4210
      WR(TE(6,226) IT,IX,MAXX,IY,MAXY                                   ICNT4220
      GO TO 111                                                         ICNT4230
  215 IT=109                                                            ICNT4240
      WR(TE(6,226) IT,IX,MAXX,IY,MAXY                                   ICNT4250
      GO TO 116                                                         ICNT4260
  216 IT=116                                                            ICNT4270
      WRITE(6,226) IT,IX,MAXX,IY,MAXY                                   (CNT4280
      GO TO 137                                                         ICNT4290
  217 IT=119                                                            ICNT4300
      WRITE(6,226) IT,IX,MAXX,IY,MAXY                                   (CNT4310
      GO TO 123                                                         ICNT4320
  218 IT=128                                                            ICNT4330
      WRITE(6,226) (T,IX,MAXX,IY,MAXY                                   ICNT4340
      GO TO 132                                                         ICNT4350
  219 IT=130                                                            (CNT4360
      WRITE(6,226) IT,IX,MAXX,IY,MAXY                                   ICNT4370
      GO TO 105                                                         (CNT4380
  220 (T=135                                                            ICNT4390
      WRITE(6,226) IT,IX,MAXX,IY,MAXY                                   (CNT4400
      GO TO 140                                                         (CNT4410
  221 IT=303                                                            ICNT4420
      GO TO 211                                                         ICNT4430
  222 IT=304                                                            (CNT4440
      GO TO 211                                                         ICNT4450
  223 IT=305                                                            ICNT4460
      GO TO 211                                                         ICNT4470
  224 IT=306                                                            ICNT4480
      GO TO 211                                                         (CNT4490
C                                                                       (CNT4500
  226 FORMAT(IX,516,2F10.3)                                             ICNT4510
C                                                                       (CNT4520
      FUNCTION UNMARK(I)                                                ICNT4530
      DATA MASK/07777777777776/                                         ICNT4540
      UNMARK=AND(I,MASK)                                                ICNT4550
      RETURN                                                            ICNT4560
      REAL FUNCTION MARK(I)                                             ICNT4570
      MARK=OR(I,1)                                                      ICNT4580
      RETURN                                                            ICNT4590
      END                                                               ICNT4600
```

```
      SUBROUTINE INTPLT                                                 INTP  10
C                                                                       INTP  20
C     USING RATIONAL POLN. SPLINE INT. OF O V AHUJA  I 8 M SYST J  1968 INTP  30
C     P 208-217                                                         INTP  40
C                                                                       INTP  50
      DIMENSION X(400), Y(400)                                          INTP  60
      COMMON /PLT2/ C(400),XA(400),YA(400),VX(400),VY(400)              INTP  70
      COMMON /PLT1/ ITOT,IN,NP,IT,NT,U(NT,LOG(6)                        INTP  80
      EQUIVALENCE (X,XA), (Y,YA)                                        INTP  90
      LOG(6)=400                                                        INTP 100
      NT=LOG(6)                                                         INTP 110
      ITOT=NT                                                           INTP 120
      C(1)=.25                                                          INTP 130
      CA=C(1)                                                           INTP 140
      LOG(5)=0                                                          INTP 150
      LOG(4)=LOG(5)                                                     INTP 160
      LOG(3)=LOG(4)                                                     INTP 170
      LOG(2)=LOG(3)                                                     INTP 180
      LOG(1)=LOG(2)                                                     INTP 190
      DO 101 I=2,ITOT                                                   INTP 200
      C(I)=1./(4.-CA)                                                   INTP 210
  101 CA=C(I)                                                           INTP 220
      RETURN                                                            INTP 230
      ENTRY PREPLT                                                      INTP 240
      NC=IN-2                                                           INTP 250
      N=NC                                                              INTP 260
      NVV=IN-1                                                          INTP 270
      NV=IN                                                             INTP 280
      OX=VX(IN)                                                         INTP 290
      OY=VY(IN)                                                         INTP 300
      DO 102 I=2,N                                                      (NTP 310
         OX=C(NC)*(3.*(X(NV)-X(NV-2))-OX)                               INTP 320
         VX(NVV)=OX                                                     INTP 330
         OY=C(NC)*(3.*(Y(NV)-Y(NV-2))-OY)                               INTP 340
         VY(NVV)=OY                                                     INTP 350
         NC=NC-1                                                        INTP 360
         NVV=NVV-1                                                      INTP 370
         NV=NV-1                                                        INTP 380
  102 CONTINUE                                                          INTP 390
      VX(2)=C(1)*(3.*(X(3)-X(1))-OX-VX(1))                              INTP 400
      VY(2)=C(1)*(3.*(Y(3)-Y(1))-OY-VY(1))                              INTP 410
      XIC=X(1)                                                          INTP 420
      YIC=Y(1)                                                          INTP 430
      CALL PLOT(XIC,Y(C,3)                                              INTP 440
      IC=1                                                              INTP 450
      AA9=VX((C)                                                        INTP 460
      AA10=VY((C)                                                       INTP 470
  103 XC=XIC                                                            INTP 480
      XIC=X(IC+1)                                                       INTP 490
      YC=YIC                                                            INTP 500
      Y(C=Y((C+I)                                                       INTP 510
      A9=AA9                                                            INTP 520
      AA9=VX((C+1)                                                      INTP 530
      A10=AA10                                                          INTP 540
      AA10=VY((C+1)                                                     INTP 550
      A1=2.*(XC-XIC)+A9+AA9                                             INTP 560
      A2=2.*(YC-YIC)+A10+AA10                                           INTP 570
```

```
      A5=3.*(XIC-XC)-A9-A9-AA9                                          INTP 580
      A6=3.*(YIC-YC)-A10-A10-AA10                                       INTP 590
      I=1                                                               INTP 600
      U=0.                                                              INTP 610
  104 U=U+UINT                                                          INTP 620
      WX=((A1*U+A5)*U+A9)*U+XC                                          INTP 630
      WY=((A2*U+A6)*U+A10)*U+YC                                         INTP 640
      CALL PLOT(WX,WY,2)                                                INTP 650
      I=I+1                                                             INTP 660
      IF (I.LE.NP) GO TO 104                                            INTP 670
      IC=IC+1                                                           INTP 680
      IF (IC.LT.IN) GO TO 103                                           INTP 690
      RETURN                                                            INTP 700
C                                                                       INTP 710
      END                                                               INTP 720
```

```
C    COMMENT CARDS FOR FINGFC                                                  LSTF  10
C    NBS SEPTEMBER , 1971                                                      LSTF  2D
C THIS PROGRAM IS AN ADAPTION OF THE XRAY 67 PROGRAM LIST FC, IT HAS           LSTF  3D
C BEEN ALTERED TO*STAND ALONE*AND READ THE FOURIER OUTPUT TAPE WRITTEN BLSTF  4D
C  REINE, THE LEAST SQUARES PROGRAM WRITTEN BY L. W. FINGER AT THE            LSTF  5D
C GEOPHYSICAL LAB, CARNEGIE INSTITUTE OF WASHINGTON.                         LBTF  60
C THE OUTPUT OF THE PROGRAM HAS BEEN CHANGED SO THAT THE 'PHASE IN           LSTF  70
C MILLICYCLES' HAS BEEN REPLACED BY SIGMA(FO), SCALED UP BY A FACTOR OF LSTF  80
C 10 TO MAKE IT CONSISTENT WITH FO.                                          LBTF  90
C   CARD ONE                                                                 LSTF  1D0
C  COL 1-6 COMPOUND IDENTIFICATION, 7-7B ANY TITLE                           LSTF  11D
C   CARD TWO                                                                 LSTF  120
C 14  (1)/(2)/(3) FOR I INDEX VARIES (MOST)/(NEXT MOST)/(LEAST) RAPIDLT LSTF  130
C     (CODE TO TELL HOW INPUT DATA ARE SORTED)                               LSTF  140
C 15  SORTING FREQUENCY FOR K INDEX                                          LSTF  15D
C 16  SORTING FREQUENCY FOR L INDEX                                          LSTF  160
C     (NOTE THAT SUM AND PRODUCT OF NUMBERS IN COLUMNS 14,15, AND 16 MUSLSTF 170
C     EQUAL 6)                                                               LSTF  180
C 17-20  NUMBER OF LINES PER "LISTFC" PAGE                                   LSTF  190
C 21  (BLANK)/(A) FOR (DO)/(ABSOLUTELY DO NOT) INTERNALLT CHANGE THE VALLSTF 200
C     OF THE NUMBER OF LINES IN ORDER TO MAKE THE BOTTOM OF THE LAST PAGLSTF 210
C     AS EVEN AS POSSIBLE                                                    LBTF  22D
C  22-24  NUMBER OF LIST COLUMNS PER "LISTFC" PAGE                           LBTF  230
C    NOTE THE PRODUCT OF THE NUMBER OF LINES AND THE NUMBER OF COLUMNS FLSTF 240
C    "LISTFC" PAGE CANNOT EXCEED 3000.  THAT IS, NO MORE THAN 3000 FOR TLSTF 250
C    TOTAL OF REFLECTIONS AND HEADINGS (WITH THEIR SPACES), FOR PAGE.      LSTF  26D
C    ZERO (OR BLANK) IN ANY OF THE FIVE FOLLOWING FIELDS MEANS JUST THATLSTF 270
C  26  NUMBER OF BLANK PRINT COLUMNS BEFORE THE LISTFC COLUMN               LSTF  2BD
C 28  NUMBER OF PRINT COLUMNS FOR MOST RAPIDLT CHANGING INDEX              LBTF  290
C 30  NUMBER OF PRINT COLUMNS FOR FO TIMES 10                              LSTF  300
C 32  NUMBER OF PRINT COLUMNS FOR FC TIMES 10                              LSTF  310
C 34  NUMBER OF PRINT COLUMNS FOR SIDFO TIMES 10                           LSTF  320
C 36  SPECIAL FLAG FOR LESS THANS*(BLANK = 0)*(JCODE = 2)                  LSTF  330
C 3B  SPECIAL FLAG FOR EXTINCT*(BLANK = B)*(JCODE = 3)                     LSTF  340
C 40  SPECIAL FLAG FOR SPECIAL REFLECTION (JCODE=4)                        LSTF  350
C    THE NEXT NINE FIELDS ARE (BLANK)/(1) FOR (DO NOT)/(DO)                LSTF  360
C 42  PRINT STUBOL FOR LESS THAN OR EXTINCT REFLECTIONS                    LSTF  370
C    THIS SYMBOL SWITCH ADDS ONE MORE PRINT COLUMN)                        LSTF  3B0
C 44  FOR CENTRIC STRUCTURES, ATTACH SIGN OF A TO FC.                      LSTF  390
C 46  DOUBLE SPACE THE LINES                                               LBTF  400
C 4B  RESTORE EACH "LISTFC" PAGE TO BE THE TOP OF A PRINTER PAGE           LBTF  410
C 50  PRINT CURRENT*TITLE* AT TOP OF EACH PRINTER PAGE                     LSTF  420
C 52 PUNCH A SET OF *FCARD*CARDS (HAS H,K,L,FO,FC,A,B,JCODE, AND           LSTF  430
C    LEVEL INDICATORS)                                                     LSTF  440
C 56  PRINT MINUS SIGN ON THE FO OF THE REFLECTIONS WHICH ARE LESS THANSLSTF 450
C 5B  WRITE A SEPARATE COPY OF THE FC LIST ON TAPE A                       LSTF  460
C 62  NUMBER OF TIMES TO TRY TO GET HEADINGS AT THE TOP OF ALL COLUMNS     LSTF  470
C 64 (1)/(2)/(3) FOR HEADINGS SEPARATED BY (NO BLANK LINES)/(1 BLANK       LSTF  4B0
C    LINE ABOVE)/(1 ABOVE AND 1 BELOW)  (BLANK = 3)                        LSTF  490
C 65-72  FACTOR  F'S ARE  TO BE MULTIPLIED BY,  BLANK = 10                 LSTF  500
C 73-B0  FACTOR SIG(FO) IS TO BE MULTIPLIED BY,  BLANK = 10                LBTF  510
C    AS PRESENTLY WRITTEN, THE PROGRAM WILL NOT READ F VALUES FROM CARDSLSTF 520
C    BUT THIS CAN EASILY BE ARRANGED BY MAKING A USED SPECIFIC VERSION     LSTF  530
C    OF SUBROUTINE FING                                                    LBTF  540
C                                                                          LSTF  550
C----VERSION AND UPDATING ESTABLISHED I JUNE 1969                          LSTF  560
C      ALTERATIONS MADE AS FOLLOWS... 20  JUN 69, 23 SEP 69                LSTF  570
```

```
C      JAN 70, JULY 70                                                  LSTF 580
C                                                                       LSTP 590
C                                                                       LSTP 600
CLISTFC     FCLISTING ROUTINE                                           LSTF 610
C      VERSION OF SEPT 16, 1965                                         LSTP 620
C----TRANSLATED TO XRAY67 JULY 1968                                     LSTP 630
C----SYSTEM COMMON, DIMENSION AND EQUIVALENCE PACKAGE                   LSTP 640
C                                                                       LSTP 650
      DIMENSION IQTO(100),ICARO(21),NTIND(1)                            LSTP 660
      EQUIVALENCE (QTO(1),IQTO(1)),(NTAPEC,NTPCH),(NTCU,NTPCHM)         LSTF 670
     * EQUIVALENCE (CARO(1),ICARO(1)),(NTIND(1),NTIN)                  LSTF 680
C    .                                                                  LSTF 690
C----INPUT-OUTPUT COMMON, DIMENSION AND EQUIVALENCE PACKAGE             LSTP 700
C                                                                       LSTF 710
      COMMON /IO/ MAXREC,IOBMX,IOBUP(504)                               LSTF 720
      DIMENSION BUFIO(504)                                              LSTF 730
      EQUIVALENCE (IOBUP(1),IOBUF1),(IOBUP(2),IOBUP2),                 LSTF 740
     $(IOBUP(3),IOBUP3),(IOBUP(4),IOBUP4),(BUFIO(I),IOBUF(I))          LSTF 750
C-----ARRAY FOR CONSTANTS FOR SETTING FORMATS AND COUNTS.               LSTF 760
      DIMENSION ICA8(17)                                                LSTF 770
      DIMENSION IPL1(3),IPL2(3)                                         LSTF 780
      DIMENSION IFNT(160),IRE(6),IBL(6),IX(10),IFMT(7)                  LSTF 790
      DIMENSION IQBU(90),QBU(9)                                         LSTF 800
      DIMENSION INUM(10),II(9),IXX(9)                                   LSTF 810
      DIMENSION IIND(3),MSIQ(4)                                         LSTF 820
      DIMENSION TITLE(12),IDCBMP(2)                                     LSTF 830
      DATA ICA8 / 1H , 3HA1,, 1H=, 1BE, 4R(IX,, 4H1H*), 4H(1H , 4H(IHO, LSTF 840
     $4H2RH,, 3H1H,, 3H1H,, 4B28X,, 3H1B,, 4H2H,L, 1HA, 1H), 4H1H*, /   LSTF 850
      0ATA IFL1,IFL2/2HEN,2HFC,2HSC,4HO   ,4HARO ,4HALE /               LSTF 860
C-----FORMAT ARRAYS                                                     LSTF 870
C-----STORAGE ARRAYS FOR ONE PAGE OR MATRIX AS A STRING OR STACK        LSTF 880
      COMMON/LPC/ INDEX(3000),IP0(3001),IFC(3000),ISIG(3000),IPHS(3000) LSTF 890
C-----YES, THAT SHOULD BE 3001 OR ITOTR+1 IN ANY CASE IF STORAGE IS     LSTF 900
C-----CHANGED.  THATS THE BLANK THAT HANGS OFF THE END OF THE STACK.    LSTF 910
C-----OUTPUT BUFFER OF DATA FROM THE STACK                              LSTF 920
      EQUIVALENCE (IOBU(1),QBU(1))                                      LSTF 930
      DATA IBLANK /1H /                                                 LSTF 940
      DATA IXXX/1HX/                                                    LSTF 950
      DATA INUM/1B0,1HI,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/                LSTF 960
      DATA II/3HI1,,3HI2,,3H13,,3H14,,3H15,,3H16,,3HI7,,3HI8,,3H19,/    LSTF 970
      DATA IXX/2H1X,2H2X,2H3X,2H4X,2H5X,2H6X,2H7X,2H8X,2H9X/            LSTF 980
      DATA NTIN,NTOUT,NTAPEA,NTAPED,NTOUTM/5,6,12,8,39/                 LSTF 990
C-----CONSTANTS FOR SETTING TOTAL AVAILABLE STORAGE                     LSTP1000
      ITOTR=3000                                                        LSTFI010
C-----READ THE CONTROL CARD AND SET UP THE FORMAT ARRAYS                LSTP1020
 1    CONTINUE                                                          LSTP1030
CDC   DECODE(80,101,ICDC)ISORH,ISORX,ISORL,LINE,IAPAGE,ICOLPP,IBLNK,IIN,LSTPI040
CDC  $,IFOC,IFCC,IPHC,IPLLT,IFLEX,IPLSPC,ISIGNO,ISIGNC,ISPACE,IRESTO,ITITLSTP1050
CDC  $,IPNCH,ICAD,ISIGS,ITOTR,ITRY,ISPAZ                                LSTFI060
      READ(NTIN,2)IOCOMP,TITLE                                          LSTP1070
 2    FORMAT(A2,A4,12A6)                                                LSTP1080
      READ(NTIN  ,101) ISORH,ISORX,ISORL,LINE,IAPAGE,ICOLPP,IBLNK,IIN,  LSTFI090
     $  IFOC,IFCC,IPHC,IPLLT,IFLEX,IPLSPC,ISIGNO,ISIGNC,ISPACE,IRESTO,  LSTFI100
     $ITIT,IPNCH,ICAD,ISIGS,ITOTR,ITRY,ISPAZ,FSCAL,SGSCAL               LSTFI110
 101  FORMAT(13X311,14,A1,13,5(1X11),3(1XA1),9(1X11),2X2(1X11),2F8.0)   LSTFI120
      IF(FSCAL.LT..0001)FSCAL=10.                                       LSTPI130
      IF(SGSCAL.LT..0001)SGSCAL=10.                                     LSTPI140
      ICAO=0                                                            LSTP1150
C-----CHECK IF OUTPUT GOES ON TWO SEPARATE FILES                        LSTPI160
      IF(ITOT) 99,99,95                                                 LSTPI170
 95   CONTINUE                                                          LSTP1180
C-----MAKE SURE THEY ARE NOT THE SAME                                   LSTPI190
      IF(NTOUT-NTAPED) 96,97,96                                         LSTPI200
 96   IF(NTOUTM-NTAPED) 98,97,98                                        LSTFI210
 97   CONTINUE                                                          LSTPI220
      WRITE (NTOUT,100)                                                 LSTPI230
100   FORMAT(72HONTOUT AND NPILED ARE THE SAME, REQUEST FOR SPECIAL OUTPLSTPI240
     $UT FILE IGNORED.  )                                               LSTPI250
      ITOT=0                                                            LSTP1260
 98   CONTINUE                                                          LSTP1270
C-----REWIND THE SPECIAL OUTPUT FILE                                    LSTPI280
      REWIND NTAPED                                                     LSTPI290
 99   CONTINUE                                                          LSTP1300
C----CALCULATE THE STORAGE REQUIRED FOR THE LINES AND COLUMNS REQUESTED LSTFI310
      MATRIX=LINE*ICOLPP                                                LSTPI320
      IF(MATRIX-ITOTR) 102,401,401                                      LSTFI330
 102  IP(MATRIX) 401,401,103                                            LSTP1340
 103  CONTINUE                                                          LSTP1350
C----INITIALIZE COUNTS AND SIGNALS                                      LSTPI360
      IREPL=0                                                           LSTP1370
      ICOUNT = 0                                                        LSTPI380
      NOGO = 0                                                          LSTPI390
      WRITE (NTOUT,104)                                                 LSTPI400
104   FORMAT (120X)                                                     LSTP1410
C----CALCULATE THE NUMBER OF TYPE POSITIONS PER LIST COLUMN             LSTF1420
      IPERC=IBLNK* 1IN + IFOC * IFCC * IPHC                             LSTP1430
C-----TEST IF LESS TRANS TO BE MARKED BY SIGN ON PO                     LSTP1440
      IF(ISIGNO) 107,107,108                                            LSTP1450
C-----THEY ARE,....SET APPROPRIATE QUANTITIES                           LSTP1460
 107  CONTINUE                                                          LSTP1470
      IRE(4) = ICA8(1)                                                  LSTP1480
      GO TO 109                                                         LSTP1490
C-----A SPECIAL COLUMN IS TO BE WRITTEN                                 LSTPI500
 108  CONTINUE                                                          LSTP1510
      IPERC=IPERC + 1                                                   LSTP1520
      IRE(4) = ICA8(2)                                                  LSTP1530
C----TEST FOR SPECIAL USER SUPPLIES CHARACTERS TO MARK LESS THAN AND     LSTP1540
C      EXTINCT REFLECTIONS.                                             LSTP1550
 109  IF(IPLLT-IBLANK) 111,110,111                                      LSTP1560
 110  IFLLT = ICA8(3)                                                  LSTP1570
 111  IF(IFLEX-IBLANK) 113,112,113                                      LSTP1580
 112  IFLEX = ICA8(4)                                                   LSTP1590
 113  CONTINUE                                                          LSTP1600
      MSIG(1)=IBLANK                                                    LSTP1610
      MSIG(2)=IFLLT                                                     LSTP1620
      MSIG(3)=IFLEX                                                     LSTP1630
      MSIQ(4)=IFLSPC                                                    LSTP1640
C----TEST SIGNAL FOR HOW HEADINGS ARE TO BE SPACED IN OUTPUT I=NO BLANK LSTP1650
C      LINES, 2=1 BLANK LINE ABOVE 3=ONE ABOVE AND ONE BELOW.           LSTP1660
      IP(ISPAZ) 155,155,156                                             LSTP1670
 155  ISPAZ=3                                                           LSTP1680
 156  IP(ISPAZ-3) 157,157,155                                           LSTP1690
 157  CONTINUE                                                          LSTP1700
C-----TEST THE NUMBER OF SHEETS PER PAGE THAT WILL BE REQUIRED          LSTP1710
      NUMCPS = 120 / IPERC                                              LSTP1720
C-----CALCULATE NUMBER OF SHEETS REQUIRED                               LSTP1730
```

```
      NSBEET=1                                                          LSTF1740
  200 IF(IC0LPP-NSBEET*NUMCPS) 202,202,201                              LSTF1750
C-----MORE SHEETS NEEDED                                                LSTF1760
  201 NSBEET=NSBEET + 1                                                 LSTF1770
      G0 T0 200                                                         LSTF1780
  202 C0NTINUE                                                          LSTF1790
C-----SET UP FIDUCIARY WRITING F0RMAT                                   LSTF1800
C----IT IS 0F THE F0RM (2H*NNNX 1H*) WHERE NNN IS THE VALUE 0F IPIX IS  LSTF1810
C     THE PRODUCT 0F THE NUMBER 0F TYPE P0SITI0NS PER LIST C0LUMN TIMES LSTF1820
C     THE NUMBER 0F LIST C0LUMNS PER SHEET IS LESS 0NE.                 LSTF1830
      IFFMT(1) = ICAB(5)                                                LSTF1840
      IFFMT(2) = ICAB(17)                                               LSTF1850
      IFFMT(6) = IXXX                                                   LSTF1860
      IFFMT(7) = ICAB(6)                                                LSTF1870
      IPIX=NUMCPS*IPERC-1                                               LSTF1880
      IP(IPIX-10) 203,204,204                                           LSTF1890
  203 IFFMT(3) = IBLANK                                                 LSTF1900
      IFFMT(4) = IBLANK                                                 LSTF1910
      IFFMT(5) = INUM(IPIX*1)                                           LSTF1920
      G0 T0 207                                                         LSTF1930
  204 IP(IPIX-100) 205,206,206                                          LSTF1940
  205 C0NTINUE                                                          LSTF1950
      ITEMP=IPIX/10                                                     LSTF1960
      IFFMT(3) = IBLANK                                                 LSTF1970
      IFFMT(4) = INUM(ITEMP*1)                                          LSTF1980
      IPIX=-ITEMP*10 + IPIX                                             LSTF1990
      IFFMT(5) = INUM(IPIX *1)                                          LSTF2000
      G0 T0 207                                                         LSTF2010
  206 ITEMP=IPIX/100                                                    LSTF2020
      IFFMT(3) = INUM(ITEMP*1)                                          LSTF2030
      ITEMP1=(-ITEMP*100 + IPIX) / 10                                   LSTF2040
      IFFMT(4) = INUM(ITEMP1*1)                                         LSTF2050
      IPIX=-ITEMP*100-ITEMP1*10 + IPIX                                  LSTF2060
      IFFMT(5) = INUM(IPIX *1)                                          LSTF2070
  207 C0NTINUE                                                          LSTF2080
C-----FIND NUMBER 0F C0LUMNS F0R PHASE INF0RMATI0N                      LSTF2090
      IP(IPEC) 301,301,302                                              LSTP2100
  301 IRE(6) = IBLANK                                                   LSTP2110
      G0 T0 303                                                         LSTF2120
C-----BUILD F0RMAT F0R PHASE PRINT-0UT                                  LSTF2130
  302 IRE(6) = II(IPHC)                                                 LSTP2140
  303 C0NTINUE                                                          LSTF2150
C-----CHECK T0 SEE IF PC G0BS 0UT                                       LSTP2160
      IP(IPCC) 305,305,306                                              LSTP2170
  305 IRE(5) = IBLANK                                                   LSTP2180
      G0 T0 307                                                         LSTP2190
  306 IRE(5) = II(IPCC)                                                 LSTP2200
  307 C0NTINUB                                                          LSTP2210
C-----CBBCK T0 0EB IP P0 C0BS 0UT                                       LSTP2220
      IP(IP0C) 308,308,309                                              LSTF2230
  308 IRE(3) = IBLANK                                                   LSTP2240
      G0 T0 310                                                         LSTP2250
  309 IRE(3) = II(IP0C)                                                 LSTP2260
  310 C0NTINUE                                                          LSTP2270
C-----CHECK N0MBER 0F C0LUMNS F0R THE INDEX (FASTEST CHAN01NG)          LSTP2280
      IP(IIN) 311,311,312                                               LSTF2290
  311 IRE(2) = IBLANK                                                   LSTF2300
      G0 T0 313                                                         LSTF2310
  312 IRE(2) = II(IIN)                                                  LSTF2320
  313 C0NTINUE                                                          LSTP2330
C-----CHECK 0N REQUIRED BLANK C0LUMNS                                   LSTP2340
      IF(IBLNK) 314,314,315                                             LSTP2350
  314 IRE(1) = IBLANK                                                   LSTP2360
      G0 T0 316                                                         LSTP2370
  315 IRE(1) = IXX(IBLNK)                                               LSTP2380
  316 C0NTINUE                                                          LSTP2390
C----AT THIS P0INT THE ARRAY WHICH C0NSTITUTES THE C0RRECT 0UTPUT F0RMATLSTP2400
C     IS BEGUN.  IT IS MADE UP 0N A LINE BASIS BELOW.  THE IBL ARRAY IS LSTP2410
C     THE PR0L0GUE 0F SKIPPED SPACES. THE START 0F IFMT IS (1NO 0R (1H ,LSTF2420
C     THB IBL ARRAY IS NNX P0R LEADING BLANKS 0N EACH LISTING C0LUMN.   LSTF2430
C-----SINGLE SPACE 0R D0UBLE SPACE                                      LSTF2440
      IF(ISPACE) 321,321,322                                            LSTF2450
  321 IFMT(1) = ICAB(7)                                                 LSTF2460
      G0 T0 323                                                         LSTF2470
  322 IFMT(1) = ICAB(8)                                                 LSTF2480
C-----SET BLANKS IN LIST F0RMAT                                         LSTF2490
  323 C0NTINUE                                                          LSTF2500
      D0 325 J=2,6                                                      LSTF2510
  325 IBL(J) = IBLANK                                                   LSTF2520
      IP(IPERC-9) 327,327,326                                           LSTF2530
  326 IT0P=IPERC / 10                                                   LSTF2540
      IBL(1) = INUM(IT0P*1)                                             LSTF2550
      IB0T=IPERC-IT0P*10                                                LSTF2560
      IBL(2) = INUM(IB0T*1)                                             LSTF2570
      IBL(3) = IXXX                                                     LSTF2580
      G0 T0 328                                                         LSTF2590
  327 IFIX=IPENC                                                        LSTF2600
      IBL(1) = IXX(IFIX)                                                LSTF2610
  328 C0NTINUE                                                          LSTF2620
C-----INITIALIZE INDEX HEADING F0RMAT                                   LSTF2630
C-----FIRST CHECK 0N SUMMATI0N 0N0ER                                    LSTF2640
      IF(IS0RN * IS0RK * IS0RL - 6) 401,403,401                         LSTF2650
  401 C0NTINUE                                                          LSTF2660
      WRITE (NT6UT,402)                                                 LSTF2670
  402 F0RMAT(52H S0METHING AB0UT THE LIST SPECIFICATI0N IS N0 G660.     LSTF2680
      G0 T0 905                                                         LSTF2690
  403 IF(IS0NN*1S0RK*IS0RL - 6) 401,404,401                             LSTF2700
  404 C0NTINUE                                                          LSTF2710
C-----SEE IF B IS THE M0ST RAPIDLY CHANGING 1N0EX.                      LSTF2720
      IP(IS0RB-1) 401,405,408                                           LSTF2730
  405 C0NTINUE                                                          LSTF2740
      ISET1 = 5                                                         LSTF2750
      ISET2 = 7                                                         LSTF2760
      IS0RHP=1                                                          LSTF2770
      IS0RKP=2                                                          LSTF2780
      IS0RLP=3                                                          LSTF2790
C----ICAB(9)=2HN,,ICAB(10)=1N,                                          LSTF2800
      IX(4) = ICAB(9)                                                   LSTF2810
      IX(6) = ICAB(10)                                                  LSTF2820
      G0 T0 421                                                         LSTF2830
C-----SEE IF K IS THE M0ST RAPIDLY CHANGING INDEX.                      LSTF2840
  406 C0NTINUE                                                          LSTF2850
      IP(IS0RK-1) 401,410,412                                           LSTF2860
  410 ISET1=4                                                           LSTF2870
      ISET2 = 7                                                         LSTF2880
      IS0RHP=2                                                          LSTF2890
```

```
      ISORKP=I                                                          LSTF2900
      ISORLP=3                                                          LSTF2910
C----IN A FINAL SET IF TNE NEADING ARRAY IX MAY LOOK AS FOLLOWS         LSTF2920
C     IX(1)=2HNX                                                       LSTF2930
C     IX(2)=8LANKS                                                     LSTF2940
C     IX(3)=8LANKS                                                     LSTF2950
C     IX(4)=1H,                                                        LSTF2960
C     IX(5)=1HH                                                        LSTF2970
C     IX(6)=2HK,                                                       LSTF2980
C     IX(7)=1H,                                                        LSTF2990
C     IX(8)=8LANKS                                                     LSTF3000
C     IX(9)=NX                                                         LSTF3010
C     IX(10)=8LANKS                                                    LSTF3020
C----ICAR(I1)=1HN, ICAR(I2)=2HK,                                       LSTF3030
      IX(5) = ICA8(11)                                                 LSTF3040
      IX(6) = ICA8(12)                                                 LSTF3050
      GO TO 421                                                        LSTF3060
C-----IT MUST BE L                                                     LSTF3070
 412  ISETI=4                                                          LSTF3080
      ISET2=6                                                          LSTF3090
      ISORHP=2                                                         LSTF3100
      ISORKP=3                                                         LSTF3110
      ISORLP=I                                                         LSTF3120
C----ICA8(13)=1H,,ICAB(14)=2H,L                                        LSTF3130
      IX(5) = ICA8(13)                                                 LSTF3140
      IX(7) = ICA8(14)                                                 LSTF3150
 421  CONTINUE                                                         LSTF3160
      IF(ICA0) 451,451,425                                             LSTF3170
C-----INITIALIZE THE SCALE BAND TO I.0                                 LSTF3180
 425  CONTINUE                                                         LSTF3190
C-----INFORNATION FROM TAPE                                            LSTF3200
 451  CONTINUE                                                         LSTF3210
      IGO=0                                                            LSTF3220
      NF=I                                                             LSTF3230
      IPT=500                                                          LSTF3240
C  TOP OF PAGE                                                         LSTF3250
 4001 CONTINUE                                                         LSTF3260
      D8 4002 J=I,ITOTR                                                LSTF3270
      IFO(J)=-2                                                        LSTF3280
 4002 CONTINUE                                                         LSTF3290
      IFO(ITOTR + 1) = -2                                              LSTF3300
      IP1=9999                                                         LSTF3310
      IP2=9999                                                         LSTF3320
      ICOUNT=0                                                         LSTF3330
C-----TOP OF LOOP FOR STORING REFLECTIONS                              LSTF3340
C-----DATA ON BINARY TAPE.....READ IT .                                LSTF3350
 4501 CONTINUE                                                         LSTF3360
      IF( IPT.LT.500 )GOTO4502                                         LSTF3370
      IPT=0                                                            LSTF3380
      CALL FING(NTAPEA,8UFIO)                                          LSTF3390
 4502 CONTINUE                                                         LSTF3400
      IF( IPT.GE.500 )GOTO4501                                         LSTF3410
      IH = I08UF( IPT+1)                                               LSTF3420
      IF( IH.EQ.99 ) GO TO 4503                                        LSTF3430
      IK = I08UF( IPT+2)                                               LSTF3440
      IL = I08UF( IPT+3)                                               LSTF3450
      FREL = 8UFIO( IPT+4)                                             LSTF3460
      JCOOE = I88UF( IPT + 8)                                          LSTF3470
      JCOD8=JCOOE*I                                                    LSTF3480
C---- IGNORE ALL JUNK (I.E. SYS ABS AND IGNOREOS)                      LSTF3490
      IF( JCODE-3)4515,4515,7820                                       LSTF3500
 7820 IPT=IPT+10                                                       LSTF3510
      GOTO 4502                                                        LSTF3520
 4515 CONTINUE                                                         LSTF3530
      A = 8UFIO( IPT+ 6)                                               LSTF3540
      B = 8UFIO( IPT+ 7)                                               LSTF3550
      FC=HUFIO( IPT+ 5)                                                LSTF3560
      IF(( FC**2/( A**2+B**2)),LT.0.7)JCODE=3                          LSTF3570
      IINO( ISORHP)=IH                                                 LSTF3580
      IINO( ISORKP)=IK                                                 LSTF3590
      IINO( ISORLP)=IL                                                 LSTF3600
      FO=FREL                                                          LSTF3610
      GO TO 4505                                                       LSTF3620
 4503 CONTINUE                                                         LSTF3630
      I08UF3=0                                                         LSTF3640
      IGO=I                                                            LSTF3650
C-----TEST IF DONE.                                                    LSTF3660
 4505 IF( IGO)4511,4511,6001                                           LSTF3670
C-----TREAT DATA TO CHANGE IT TO FIXED POINT AND STORE IT IN ARRAY      LSTF3680
 4511 CONTINUE                                                         LSTF3690
      ICOUNT=ICOUNT + 1                                                LSTF3700
      IREFL=IREFL + I                                                  LSTF3710
C-----SEE IF SECOND OR THIRD INDEX HAS CHANGED SINCE LAST REFLECTION    LSTF3720
      IF( IP1-IIND(2)) 4522,4521,4522                                  LSTF3730
 4521 IF( IP2-IINO(3)) 4522,4525,4522                                  LSTF3740
 4522 CONTINUE                                                         LSTF3750
      IF( MATRIX-ICOUNT-ISPAZ) 4523,4524,4524                          LSTF3760
 4523 NOGO = I                                                         LSTF3770
      ICOUNT = ICOUNT - 1                                              LSTF3780
      IREFL = IREFL - I                                                LSTF3790
      GO TO 6001                                                       LSTF3800
 4524 ICOUNT = ICOUNT + ISPAZ                                          LSTF3810
      I = ICOUNT + ISPAZ/2 - ISPAZ                                     LSTF3820
      IP1 = IINO(2)                                                    LSTF3830
      IP2 = IIND(3)                                                    LSTF3840
      INDEX(I) = IIND(2)                                               LSTF3850
      IFO(I) = -I                                                      LSTF3860
      IFC(I) = IINO(3)                                                 LSTF3870
 4525 CONTINUE                                                         LSTF3880
C-----FIX THE DATA                                                     LSTF3890
      IFO(ICOUNT) = FO*FSCAL*.5                                        LSTF3900
      INDEX(ICOUNT) = IINO(1)                                          LSTF3910
      IF( ISIGNC) 4530,4530,4531                                       LSTF3920
C-----DO NOT PLACE SIGN OF A ON FC                                     LSTF3930
 4530 CONTINUE                                                         LSTF3940
      IFC(ICOUNT) = FC*FSCAL *.5                                       LSTF3950
      GO TO 4532                                                       LSTF3960
C-----APPLY SIGN OF A TO FC                                            LSTF3970
 4531 IFC(ICOUNT) = SIGN(FC*FSCAL*.5,A)                                LSTF3980
 4532 CONTINUE                                                         LSTF3990
      JCODE=JCOOE                                                      LSTF4000
      ISIG(ICOUNT)=MSIG(JCOOE)                                         LSTF4010
C  REPLACE PHASE BY SIGMA(FO) SCALED BY FACTOR OF I0.                  LSTF4020
      ALPHA=8UFIO( IPT + 9)                                            LSTF4030
      IPNS(ICOUNT)=ALPHA*SGSCAL  *.5                                   LSTF4040
      IPT=IPT + 10                                                     LSTF4050
```

```
C  SE IP PAGE IS ALL LeAOED IN                                          LSTF4060
      IF( ICOUNT-MATRIX)4S02,600I,6001                                  LSTF4070
C-----IP NOT IN CARO MeDE GO REAO ANGTHER CARD, IN TAPE, TAPE.          LSTF4080
C                                                                       LSTF4090
C-----PREPARE TO SHAKE THE STACK FOR PRINTING NICETIES.                 LSTF4100
 600I CONTINUE                                                          LSTF4110
C-----FIGURE OUT NUNBER OF BLANKS AT THE BOTTOM OF TNE STACK            LSTF4120
      I = MATRIX * 1                                                    LSTF4130
      IBLAT = 0                                                         LSTF4140
      OO 6003 J = 1,MATRIX                                              LSTF4150
      I = I - 1                                                         LSTF4160
      IF( IFO( I) * 1) 6002,6004,6004                                   LSTF4170
 6002 IBLAT = IBLAT * I                                                 LSTF4180
 6003 CONTINUE                                                          LSTF4190
 6004 CONTINUE                                                          LSTF4200
      IP (IGO) 6012,6012,6006                                           LSTF4210
 6006 IP( IAPAGE-ICAR( I5)) 6007,60I2,6007                              LSTF4220
C-----ALL REFLECTIONS ARE NOW IN - EVEN THE BOTTOM OF THE PAGE          LSTF4230
C------UNLESS OF COURSE APAGE FLAG SAYS NO,,                          LSTF4240
 6007 IF( IBLAT/ICOLPP) 6011,601I,6008                                  LSTF4250
 6008 IBLAT = IBLAT - ICOLPP                                            LSTF4260
      LINE = LINE - 1                                                   LSTF4270
      IF(LINE) 6009,6009,6007                                           LSTF4280
 6009 LINE = 1                                                          LSTF4290
 6011 CONTINUE                                                          LSTF4300
      MATRIX = IC8LPP*LINE                                              LSTF4310
 6012 CONTINUE                                                          LSTF4320
C-----SHAKE THE STACK TO MAKE COLUMNS AS NEAT AS POSSIBLE.              LSTF4330
      IOO = 1                                                           LSTF4340
      IC8L=1                                                            LSTF4350
C-----SEARCN THE TABLE COLUMN BY LOUSEY COLUMN.  (WHERE'S THE 'E')      LSTF4360
 609S CONTINUE                                                          LSTF4370
      IBOT = IC8L*LINE                                                  LSTF4380
      ITOP = IBOT - LINE * 1                                            LSTF4390
C-----TEST IF TOP ENTRY IN THE STACK IS A BLANK                         LSTF4400
C-----IP IT IS, SEE IF THE ONE BELOW IT IS AN INOEX                     LSTF44I0
      IF( IPO( ITOP) * I) 7001,70II,701I                                LSTF4420
 7001 IF( IPO( ITOP * 1) * 1) 701I,7002,70II                            LSTF4430
C----- POP THE WHOLE STACK FROM SEARCH POINT ONWARO UP ONE SPACE        LSTF4440
 7002 CONTINUE                                                          LSTF4450
      OO 700S J = ITOP,MATRIX                                           LSTF4460
      INDEX(J) = INOEX(J*1)                                             LSTF4470
      IPO(J) = IPO(J*1)                                                 LSTF4480
      IPC(J) = IPC(J*I)                                                 LSTF4490
      ISIG(J) = ISIG(J*1)                                               LSTF4S00
      IPNS(J) = IPNS(J*1)                                               LSTF4SI0
 7005 CONTINUE                                                          LSTF4S20
      IBLAT = IBLAT * I                                                 LSTF4S30
C-----SEE IP THE BOTTOM OF THIS COLUMN IS A HEAOING OR A BLANK          LSTF4S40
 70II IF( IPO( IBOT) * I) 7012,70I4,7031                                LSTF4S50
C----- ITS A BLANK,, SEE IF ONE ABOVE IS A NEADING,                     LSTF4560
 7012 IF( IPO( IBOT - 1) * 1) 703I,7013,703I                            LSTF4570
C----- IT IS SO SET TNE "POPER" TO 2                                   LSTF4S80
 70I3 IPOP = 2                                                          LSTF4590
      GO TO 70IS                                                        LSTF4600
C-----HEAOING IS AT THE BOTTOM OP A COLUMN,, SET POPER TO I             LSTF4610
 70I4 IPOP = I                                                          LSTF4620
 70IS IF( IBLAT - IPOP) 703I,702I,7021                                  LSTF4630
C-----THERE ARE BLANKS AT ENO .... PUSH TNE STACK DOWNWARD.             LSTF4640
 7021 CONTINUE                                                          LSTF46S0
      IBLAT = IBLAT - IPOP                                              LSTF4660
      OO 7024 K = I,IPOP                                                LSTF4670
      ISTART = IBOT - IPOP * K                                          LSTF4680
      I = MATRIX * 1                                                    LSTF4690
      OO 7023 J = ISTART,MATRIX                                         LSTF4700
      I = I - 1                                                         LSTF4710
      INDEX( I) = INOEX( I -I)                                          LSTF4720
      IFO( I) = IPO( I - 1)                                             LSTF4730
      IPC( I) = IPC( I - 1)                                             LSTF4740
      ISIG( I) = ISIG( I - 1)                                           LSTF4750
      IPNS( I) = IPES( I - I)                                           LSTF4760
 7023 CONTINUE                                                          LSTF4770
      IFO(ISTART)"-2                                                    LSTF4780
 7024 CONTINUE                                                          LSTF4790
C----- INCREMENT THE COLUMN COUNT AND TEST IF ALL DONE WITH THIS PASS.  LSTF4800
 703I CONTINUE                                                          LSTF4810
      IC8L = IC8L * I                                                   LSTF4820
      IP( IC8LPP - IC8L) 7032,609S,609S                                 LSTF4830
 7032 IOO = ID8 * I                                                     LSTF4840
      IC8L = 1                                                          LSTF4850
C----- TEST IP LIST IS SNAKEN ENOUGN TO SUIT USER.                      LSTF4860
      IF( ITRY - IOO) 7033,609S,609S                                    LSTF4870
 7033 CONTINUE                                                          LSTF4880
C-----NOW TRE STACK IS READY TO SPIT OUT . SHEET BY SHEET.              LSTF4890
C-----INITIALIZE SHEET OUTPUT COUNT                                     LSTF4900
C-----TOP OP THE SHEET LOOP                                             LSTF4910
      NSH = 0                                                           LSTF4920
C-----COUNT TOTAL COLUMNS PROCESSEO                                     LSTF4930
      NACCOL=0                                                          LSTF4940
 80I1 NSN = NSH * 1                                                     LSTF4950
      NL = 0                                                            LSTF4960
C-----TEST TO SEE IF TNIS WILL BE TRE LAST SNEET OP CURRENT PAGE        LSTF4970
      IF(NSH - NSHEET) 80I3,80I2,80I2                                   LSTF4980
 8012 NACL = IC8LPP * NUMCPS*(1-NSH)                                    LSTF4990
      G8 TO 80I4                                                        LSTF5000
 8013 NACL = NUMCPS                                                     LSTF5010
 80I4 CONTINUE                                                          LSTF5020
      IBASE = NACCOL*LINE                                               LSTF5030
C-----KEEP TRACK OF ACCUMULATED COLUMNS ON WNOLE PAGE BASIS             LSTF5040
      NACCOL=NACCOL + NACL                                              LSTF5050
C-----IS TNE PAGE TO BE RESTORED                                        LSTF5060
      IP( IRESTO) 8053,8053,80SI                                        LSTF5070
 80SI WRITE (NTOUT,8052)                                                LSTF5080
 80S2 FORMAT(IHI)                                                       LSTF5090
      IP( ITWOT) 8053,80S3,8054                                         LSTF5100
 80S4 CONTINUE                                                          LSTF5110
      WRITE (NTAPEO,8052)                                               LSTF5120
 80S3 CONTINUE                                                          LSTF5130
 8061 WRITE(NTOOT,8062) TITLE                                           LSTF5140
 8062 FORMAT( IN 12A6///)                                               LSTF5150
      IF( ITWOT) 8063,8063,8064                                         LSTF5160
 8064 CONTINUE                                                          LSTF5170
      WRITE(NTAPED,8062) TITLE                                          LSTF5180
 8063 CONTINUE                                                          LSTF5190
C-----SEE IP THIS IS A SINGLE SHEET PROBLEM                             LSTF5200
      IP(NSHEET - 1) 8I03,8I03,8I01                                     LSTF5210
```

```
C-----TBERE WILL BE MULTIPLE SHEETS.. WRITE SOME FIDUCIARIES                LSTF5220
 8101 CONTINUE                                                             LSTF5230
      WRITE (NTOUT,IFFMT)                                                  LSTF5240
      WRITE (NTOUT,8102)                                                   LSTF5250
 8102 FORMAT(180/1H0/1H0)                                                  LSTF5260
      IF(ITWOT)8103,8103,8104                                              LSTF5270
 8104 CONTINUE                                                             LSTF5280
      WRITE(NTAPED,IFFMT)                                                  LSTF5290
      WRITE (NTAPED,8102)                                                  LSTF5300
 8103 CONTINUE                                                             LSTF5310
C-----TOP OF THE LINE LOOP                                                 LSTF5320
 8201 CONTINUE                                                             LSTF5330
      NL = NL + 1                                                          LSTF5340
      IBASE = IBASE + 1                                                    LSTF5350
C----SET THE NUMBER OF ITEMS TO BE WRITTEN OUT TO ZERO                     LSTF5360
      IOUT = 0                                                             LSTF5370
C---- SET THE NUMBER OF ITEMS IN THE FORMAT STATEMENT BEING STUFFED TO     LSTF5380
      IPUT = 1                                                             LSTF5390
C-----THIS IS A REFLECTION LINE FORMATION LOOP STARTING HERE.              LSTF5400
      DO  8601  K = 1,NACL                                                 LSTF5410
      I = IBASE +(K - 1)*LINE                                              LSTF5420
      IF(IFD(I) + 1) 8301,8401,8501                                        LSTF5430
C-----THE ITEM FROM THE STACK IS A BLANK                                   LSTF5440
 8301 CONTINUE                                                             LSTF5450
      DO  8302  J = 1,6                                                    LSTF5460
      IPUT = IPUT + 1                                                      LSTF5470
C----CONTINUE BUILDING THE OUTPUT LINE FORMAT BY MOVING THE IBL(LEAVING    LSTF5480
C     BLANKS IN FOR THE CURRENT LIST COLUMN. IBL INITIALIZES AT ABOUT      LSTF5490
C     XYFL0175 ABOVE.  THE NEXT PART OF A LINE IS SET IN THE IX ARRAY      LSTF5500
C     WHICH FIXES THE HEADINGS FOR EACH GROUP OF A GIVEN PAIR OF INDI-     LSTF5510
C     CES. IT IS OF THE FORM                                               LSTF5520
      IFMT(IPUT)=IBL(J)                                                    LSTF5530
 8302 CONTINUE                                                             LSTF5540
      GO TO 8601                                                           LSTF5550
C-----THE ITEM FROM THE STACK IS AN INDEX                                  LSTF5560
 8401 CONTINUE                                                             LSTF5570
      IOUT = IOUT + 2                                                      LSTF5580
      IOHU(IOUT-1) = INDEX(I)                                              LSTF5590
      IOBU(IOUT) = IFC(I)                                                  LSTF5600
      IPIX = 1                                                             LSTF5610
C-----TEST SIZE OF INDEX FOR BUILDING OUTPUT FORMAT                        LSTF5620
      IF(INDEX(1)) 8402,8403,8403                                          LSTF5630
C-----  ITS A NEGATIVE NUMBER                                              LSTF5640
 8402 IPIX = IPIX + 1                                                      LSTF5650
C-----IS THE NUMBER GREATER THAN 9                                         LSTF5660
 8403 IF(IABS(INDEX(I)) - 10) 8405,8404,8404                               LSTF5670
 8404 IPIX = IPIX + 1                                                      LSTF5680
 8405 CONTINUE                                                             LSTF5690
C-----SET FIRST INDEX FORMAT                                               LSTF5700
      IX(ISET1) = II(IPIX)                                                 LSTF5710
      ITEMP2= IPIX + 3                                                     LSTF5720
C-----TEST THE SECOND INDEX                                                LSTF5730
      IPIX = 1                                                             LSTF5740
      IF(IFC(I)) 8408,8409,8409                                            LSTF5750
C-----IT IS MINUS WIDEN COLUMNS                                            LSTF5760
 8408 IPIX = IPIX + 1                                                      LSTF5770
 8409 IF(IABS(IFC(I)) - 10) 8411,8410,8410                                 LSTF5780
 8410 IPIX = IPIX + 1                                                      LSTF5790

 8411 CONTINUE                                                             LSTF5800
      IX(ISET2) = II(IPIX)                                                 LSTF5810
      ITEMP = (IPERC - IBLNK - ITEMP2 + 1PIX) / 2                          LSTF5820
      IPIX = IPERC - ITEMP -ITEMP2 + IPIX                                  LSTF5830
      IF(ITEMP) 8421,8431,8441                                             LSTF5840
 8421 WRITE(NTOUT,8422)NSH,NL,NP                                           LSTF5850
 8422 FORMAT(1OHO ON SHEET I4,6H, LINE I4,8H OF PAGE I4,                   LSTF5860
     $44H THERE IS A HEADING THAT COMES OUT TOO WIDE.      )               LSTF5870
      GO TO 905                                                            LSTF5880
C-----NO SPARE COLUMNS FOR CENTERING THE HEADING                           LSTF5890
 8431 CONTINUE                                                             LSTF5900
      IX(8 ) = IBLANK                                                      LSTF5910
      IX(7)=IBLANK                                                         LSTF5920
      IX(8)=IBLANK                                                         LSTF5930
      GO TO 8445                                                           LSTF5940
C-----READING NEEDS CENTERING GET OUT BCD CONTROL FOR FORMAT               LSTF5950
 8441 CONTINUE                                                             LSTF5960
      IF(ITEMP - 10) 8442,8443,8443                                        LSTF5970
 8442 CONTINUE                                                             LSTF5980
      IX(8) = IXX(ITEMP)                                                   LSTF5990
      IX(9) = IBLANK                                                       LSTF6000
      IX(10) = IBLANK                                                      LSTF6010
      GO TO 8445                                                           LSTF6020
C-----ITS TWO DECIMAL PLACES LONG                                          LSTF6030
 8443 CONTINUE                                                             LSTF6040
      IPIXX = ITEMP/10                                                     LSTF6050
      ITEMP = ITEMP - IPIXX*10                                             LSTF6060
      IX(8) = INUM(IPIXX+1)                                                LSTF6070
      IX(9) = INUM(ITEMP+1)                                                LSTF6080
      IX(10) = IXXX                                                        LSTF6090
 8445 CONTINUE                                                             LSTF6100
      IX(2)=IBLANK                                                         LSTF6110
      IX(3)=IBLANK                                                         LSTF6120
      IF(IPIX) 8446,8446,8447                                              LSTF6130
 8446 CONTINUE                                                             LSTF6140
      IX(1) = IBLANK                                                       LSTF6150
      GO TO 8450                                                           LSTF6160
 8447 CONTINUE                                                             LSTF6170
      IF(IPIX - 10) 8448,8449,8449                                         LSTF618D
 8448 CONTINUE                                                             LSTF6190
      IX(1) = IXX(1P1X)                                                    LSTF6200
      GO TO 8450                                                           LSTF6210
 8449 CONTINUE                                                             LSTF6220
      IPIXX = IPIX/10                                                      LSTF6230
      IPIX = IPIX-IPIXX*10                                                 LSTF6240
      IX(1)=INUM(IPIXX+1)                                                  LSTF6250
      IX(2)=INUM(IPIX+1)                                                   LSTF6260
      IX(3)=IXXX                                                           LSTF627D
 8450 CONTINUE                                                             LSTF6280
C-----STUFF THE FORMAT FOR A COLUMN HEADING                                LSTF6290
      DO 8451 J = 1,10                                                     LSTF6300
      IPUT = IPUT + 1                                                      LSTF6310
      IFMT(IPUT) = IX(J)                                                   LSTF6320
 8451 CONTINUE                                                             LSTF6330
      GO TO 8601                                                           LSTF6340
C-----THIS IS A GENERIC REFLECTION DATUM SHOVE IT... IN THE BUFFER         LSTF6350
 8501 CONTINUE                                                             LSTF6360
      IF(ISIGS) 8504,8504,8502                                             LSTF6370
```

```
C-----SEE IF THIS IS A LESS THAN                                        LSTF6380
 8502 IF(ISIG(I) - IFLLT) 8504,8503,8504                                LSTF6390
C-----IT IS AND HE WANTS IT MINUS                                       LSTF6400
 8503 IF8(I) = -IF8(I)                                                  LSTF6410
 8504 CONTINUE                                                          LSTF6420
C-----NOW WE MOVE THE GOODIES TO THE BUFFER                             LSTF6430
C-----MOVE THE INDEX IF NEED BE                                         LSTF6440
      IF(IIN) 8512,8512,8511                                            LSTF6450
 8511 I8UT = I8UT + 1                                                   LSTF6460
      I8BU(I8UT) = INDEX(I)                                             LSTF6470
 8512 CONTINUE                                                          LSTF6480
C-----MOVE F8 TO BUFFER IF NEEDED                                       LSTF6490
      IF(IF8C) 8522,8522,8521                                           LSTF6500
 8521 I8UT = I8UT + 1                                                   LSTF6510
      I8BU(I8UT) = IF8(I)                                               LSTF6520
 8522 CONTINUE                                                          LSTF6530
C-----MOVE LESS-THAN EXTINCT SIGNAL IF NEEDED                           LSTF6540
      IF(ISIGN8) 8532,8532,8531                                         LSTF6550
 8531 I8UT = I8UT + 1                                                   LSTF6560
      I8BU(I8UT) = ISIC(I)                                              LSTF6570
 8532 CONTINUE                                                          LSTF6580
C-----MOVE PC                                                           LSTF6590
      IF(IPCC) 8542,8542,8541                                           LSTF6600
 8541 I8UT=I8UT + I                                                     LSTF6610
      I8BU(I8UT)=IPC(I)                                                 LSTF6620
 8542 CONTINUE                                                          LSTF6630
C-----MOVE PHASE                                                        LSTF6640
      IF(IPHC) 8552,8552,8551                                           LSTF6650
 8551 I8UT = I8UT + I                                                   LSTF6660
      I8BU(I8UT) = IPHS(I)                                              LSTF6670
 8552 CONTINUE                                                          LSTF6680
C----THE IRE ARRAY IS THE ACTUAL REFLECTION IN FORMATION FORMAT.  IT IS LSTF6690
C     SET DURING INITIALIZATIONS AT FIRST OF PROGRAM.                   LSTF6700
      DO 8561 J=1,6                                                     LSTF6710
      IPUT=IPUT + I                                                     LSTF6720
      IFMT(IPUT) = IR8(J)                                               LSTF6730
 8561 CONTINUE                                                          LSTF6740
C-----BOTTOM OF THE LINE LOOP                                           LSTF6750
 8601 CONTINUE                                                          LSTF6760
C----STORE THE LAST CLOSE PARENTHESIS IN THE FORMAT ARRAY                LSTF6770
      IFMT(IPUT+1) = ICA8(16)                                           LSTF6780
C-----CHOKE OUT THE LINE NOW FORMED                                     LSTF6790
      IF(I8UT) 8651,8651,8653                                           LSTF6800
 8651 CONTINUE                                                          LSTF6810
      WRITE (NT8UT,IFMT)                                                LSTF6820
      IF(ITW8T) 8655,8655,8652                                          LSTF6830
 8652 WRITE(NTAPE0,IFMT)                                                LSTF6840
      GO TO 8655                                                        LSTF6850
 8653 CONTINUE                                                          LSTF6860
      WRITE(NT8UT,IFMT) (I8BU(J) , J=1,I8UT)                            LSTF6870
      IF(ITW8T) 8655,8655,8654                                          LSTF6880
 8654 CONTINUE                                                          LSTF6890
      WRITE(NTAPE0,IFMT) (I8BU(J),J=1,I8UT)                             LSTF6900
 8655 CONTINUE                                                          LSTF6910
C-----SEE IF ALL LINES ON THIS SHEET OF THIS PAGE ARE DONE               LSTF6920
      IF(NL-LINE) 8201,8701,8701                                        LSTF6930
 8701 IF(NSHEET-1) 8705,8705,8702                                       LSTF6940
 8702 WRITE (NT8UT,8102)                                                LSTF6950
      WRITE (NT8UT,IFFMT)                                               LSTF6960
      WRITE (NT8UT,8703) NSH,NF,IOC8MP(1),IDC8MP(2)                     LSTF6970
 8703 FORMAT(8HOSHEET  I3,  SH PAGE  I3,  58 FOR  A2,A4,1H.)            LSTF6980
      IF(ITW8T) 8705,8705,8704                                          LSTF6990
 8704 CONTINUE                                                          LSTF7000
      WRITE (NTAPE0,8102)                                               LSTF7010
      WRITE (NTAPE0,IFFMT)                                              LSTF7020
      WRITE (NTAPE0,8703) NSN,NF,IOC8MP(1),IOC8MP(2)                    LSTF7030
 8705 CONTINUE                                                          LSTF7040
C-----TEST IF ALL SHEETS OF THIS PAGE ARE DONE                          LSTF7050
      IF(NSH-NSHEET) 8011,8801,8801                                     LSTF7060
 8801 IF(IG0) 8802,8802,9001                                            LSTF7070
 8802 NF = NF + I.                                                      LSTF7080
      ICOUNT = 0                                                        LSTF7090
 8805 IF(N0GB) 8806,4001,8806                                           LSTF7100
 8806 N0C8 = 0                                                          LSTF7110
      GO TO 4001                                                        LSTF7120
C-----ALL DONE                                                          LSTF7130
 9001 CONTINUE                                                          LSTF7140
      LINCT=0                                                           LSTF7150
      WRITE (NT8UT,9002) IR8PL                                          LSTF7160
 9002 FORMAT(1H0  I5, 23H REFLECTIONS PROCESSED.  )                     LSTF7170
      IF(ITW8T) 9005,9005,9003                                          LSTF7180
 9003 CONTINUE                                                          LSTF7190
 9005 CONTINUE                                                          LSTF7200
      WRITE(NT8UT,9111)SQSCAL                                           LSTF7210
 9111 FORMAT(' SIGMA(F0)=',F4.0,' IS PRINTED INSTEAD OF PHASE ANGLS.')  LSTF7220
 905  STOP                                                              LSTF7230
      END                                                               LSTF7240
```

```
      SUBROUTINE FING(NTAPE,F8UF)                                       FING  10
      DIMENSION F8UF(504),          RMAT(3,3,24),TRANS(3,24)            FING  20
      EQUIVALENCE              (V0L,8MAT(1,1,1),TRANS(1,1))             FING  30
      IF(JA.NE.0) GO TO 10                                              FING  40
      READ(NTAPE)NSYM,V0L,(((RMAT(I,J,K),I=1,3),J=1,3),K=1,NSYM),((TRANSFING  50
     $(J,I),J=1,3),I=1,NSYM)                                            FING  60
      JA=1                                                              FING  70
 10   READ(NTAPE)F8UF                                                   FING  80
      RETURN                                                            FING  90
      END                                                               FING 100
```

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBS TN-854	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

A System of Fortran IV Computer Programs for Crystal Structure Computations

5. Publication Date February 1975

6. Performing Organization Code

7. AUTHOR(S) Larry W. Finger and E. Prince

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. Sponsoring Organization Name and Complete Address *(Street, City, State, ZIP)*

Same

13. Type of Report & Period Covered

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

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.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Computer programs; contour plotting; constrained refinement; crystallographic calculations; Fourier section; Fourier synthesis; least squares.

18. AVAILABILITY

[X] Unlimited

[ ] For Official Distribution. Do Not Release to NTIS

[X] Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13. 46:854

[ ] Order From National Technical Information Service (NTIS) Springfield, Virginia 22151

19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED

21. NO. OF PAGES 133

20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED

22. Price $2.00

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)

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 appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

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 90-396).

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.

Federal Information Processing Standards Publications (FIPS PUBS)—Publications in this series collectively constitute the Federal Information Processing Standards Register. Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

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.

Order NBS publications (except NBSIR's and Bibliographic Subscription Services) from: Superintendent of Documents, Government Printing Office, Washington, D.C. 20402.

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.

Liquefied Natural Gas. A literature survey issued quarterly. Annual subscription: $20.00.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: $20.00. Send subscription orders and remittances for the preceding bibliographic services to National Technical Information Service, Springfield, Va. 22161.

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.

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Washington, D.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, $300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE
COM-215

THIRD CLASS MAIL