NBSIR 76-1102

Computer Programs for Structural Chemistry: Status. A Fortran Program for Statistical Analysis of Crystallographic Quantities

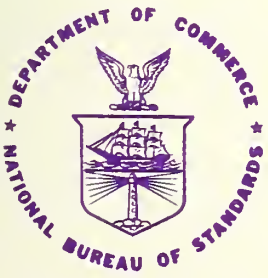
LeRoy W. Schroeder* and Brian Dickens

*Research Associate of the American Dental Assoc.
Health Foundation Research Unit at the National
Bureau of Standards

January 1976

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

Issued July, 1976



U. S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

AUTHENTICATED
U.S. GOVERNMENT
INFORMATION
GPO

NBSIR 76-1102

COMPUTER PROGRAMS FOR STRUCTURAL CHEMISTRY: STATUS. A FORTRAN PROGRAM FOR STATISTICAL ANALYSIS OF CRYSTALLOGRAPHIC QUANTITIES

LeRoy W. Schroeder* and Brian Dickens

*Research Associate of the American Dental Assoc. Health Foundation Research Unit at the National Bureau of Standards

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

January 1976

Issued July, 1976

U.S. DEPARTMENT OF COMMERCE, Elliot L. Richardson, *Secretary*

Edward O. Vetter, *Under Secretary*

Dr. Betsy Ancker-Johnson, *Assistant Secretary for Science and Technology*

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Acting Director*

This investigation was supported in part by Grant DE 00572 to the American Dental Assoc. Health Foundation from the National Institute of Dental Research and is part of the dental research program conducted by the National Bureau of Standards in cooperation with the American Dental Association Health Foundation.

Copies of this report and NBS Magnetic Tape 11 containing card images of the program described herein can both be obtained from:

National Technical Information Service
Department of Commerce
5285 Park Royal Road
Springfield, Virginia 22151

Abstract

This report describes a FORTRAN computer program for evaluation of (i) the results of crystallographic least-squares refinements by examination of the residuals $\delta R_i = (Fo-Fc)/\sigma(Fo)$, (ii) the differences in sets of data collected by different methods from the same crystal and the appropriateness of the assigned standard errors by examination of the statistics

$$\delta m_i = [Fo_i(\text{set 1}) - kFo_i(\text{set 2})] / [\sigma_i^2(\text{set 1}) + k^2\sigma_i^2(\text{set 2})]^{\frac{1}{2}},$$

(iii) differences in data sets collected by the same method from different crystals of the same material again using the δm statistics, and (iv) the significance of differences in parameters in different models representing the crystal structure of the same material by examination of the statistics

$$\delta p_i = [p_i(\text{set 1}) - p_i(\text{set 2})] / [\sigma^2 p_i(\text{set 1}) + \sigma^2 p_i(\text{set 2})]^{\frac{1}{2}}.$$

Procedure (i) provides diagnostic tests of the overall fit between observed and calculated crystallographic quantities, procedure (ii) focuses attention on experimental methods, procedure (iii) focuses attention on sample homogeneity, and procedure (iv) examines the sensitivity of refinement models to sample characteristics and experimental techniques. Procedures (iii) and (iv) have been discussed by Abrahams and Keve (1971), who suggested that the examination be accomplished by means of plots of residuals or statistics against the expected normal distribution quantiles. Here we have programmed their procedure and have extended the treatment to two additional kinds of plots. One kind of plot compares residuals with the independent variable d* and the other kind compares residuals with the calculated variable $F_c^2/\sin 2\theta$. In addition, we have used the Miller indices to divide the data into classes and octants so that the possibility of anisotropic effects can be examined. The program will handle up to 7200 data points in each of the two sets compared and is oriented specifically toward examining diffraction data.

The FORTRAN coding has been designed to minimize changes necessary to adapt this program to other computers. All plots are made on the line printer for convenience and speed. Only four input cards for program direction are required in most cases. This report includes a description of the general procedure, data specifications, program logic, a listing of the FORTRAN code, and samples of input and output.

Key words: Crystallographic data; errors; probability plots; residuals; statistics; uncertainties

Table of Contents

Computer Programs for Structural Chemistry: STATUS. A FORTRAN Program for Statistical Analysis of Crystallographic Quantities

LeRoy W. Schroeder and Brian Dickens

I. General

I.A. Introduction, Disclaimers and Acknowledgments

The text of this report describes the main features of the FORTRAN program STATUS identified in Line 50 of the main routine as the January 1975 version. The program elements of STATUS as given in this report may be consulted for other details. STATUS was written by LeRoy W. Schroeder and Brian Dickens at the National Bureau of Standards, Washington, D. C. 20234. This report is also intended as a program manual to aid in the use and implementation of STATUS. Questions concerning STATUS not covered in this report may be addressed to the authors at the above address.

The authors have extensively tested the program on the UNIVAC 1108 Exec 8 system but, of course, cannot guarantee its performance.

LWS would like to acknowledge several helpful discussions with Dr. James J. Filliben of the Statistical Engineering Laboratory at NBS. Acknowledgment is due Dr. Filliben as author of subroutines SORT and UNIMED, Dr. W. V. Loebenstein as author of SIMLEQ, R. J. Arms for subroutine PINV and S. Peavy as author of PLOTS. The authors would also like to acknowledge the able assistance of Pamela Kingsbury in preparing this report.

I.B. Raison d'être for STATUS

A recent editorial (Science, 1972) by J. R. MacDonald, Chairman of the Numerical Data Advisory Board of the National Research Council, was concerned with the question of the trustworthiness of experimental data. MacDonald points out that measures of uncertainty either are generally not given or are unaccompanied by any estimate of their reliability.

Crystallographic structural studies provide details of the average atomic arrangements, bond distances and angles, impurity distributions and site populations, and can even be used as a means, albeit expensive, of chemical analysis. From this, one obtains basic structural information about molecular and ionic geometries, ion packings, structural relationships and so on. In addition, the fine details in the parameters resulting from a structure determination provide information about the extent of positional disorder in the structure, ranges of impurity substitution in a given atomic arrangement, and the means by which such substitution can affect the physical and chemical properties of the host structure. Along more interpretive lines, structural parameters have been used as "data" in such areas of chemistry as crystal energy calculations, investigations of epitactical relationships and elucidation of hydrogen bonding by combining diffraction and spectroscopic results.

The structural parameters comprising the "model" of the crystal structure are quantities estimated by non-linear least-squares procedures. These procedures also provide estimates of the uncertainties associated with the structural parameters. However, the actual values of the derived parameters and their associated uncertainties depend on the magnitudes and uncertainties in the primary crystallographic data, i.e., the structural amplitudes derived from measurements of diffraction intensities from the specimen, assumed here to be a single crystal. Two considerations therefore arise:

(1) The observed amplitudes should be made as free of systematic error as possible and the uncertainties should be derived from a variance that combines the mean-square random error and any remaining mean-square systematic error, and

(2) The derived model should be as free of systematic effects as possible, and uncertainties in the parameters of the model should include both random error and any bias resulting from systematic effects. An example of such an effect is thermal diffuse scattering, for which corrections are difficult to apply.

Consideration (1) has generally been dealt with by applying corrections and checking agreement between equivalent reflections. Crystallographically equivalent reflections provide "repeat points" if anisotropic effects are neglected. Abrahams (1964) has proposed a method for evaluating the independent variances entering the measurement. Some of these are apparatus dependent and not easily evaluated.

Consideration (2) has been partially dealt with by referring to the standard and weighted factors, defined as

$$R = \Sigma||Fo|-|Fc||/\Sigma|Fo| \text{ and } R_w = (\Sigma w(|Fo| - |Fc|)^2/\Sigma w\ Fo^2)^{\frac{1}{2}}$$

respectively, as criteria by which to judge the fit to the diffraction data of quantities calculated from the model. The R factors are insufficient for assessing (i) estimates of errors in the model parameters and (ii) to what degree the model has compensated for systematic errors in the measured structural amplitudes. An attempt at (i) could be made by comparing supposedly identical chemical entities such as a C-C bond length as calculated in different structures after refinement. Such an attempt must assume that crystal packing effects are negligible.

The statistical procedures outlined in the Abstract and described further in Section I.C provide a means of dealing with these considerations. The statistical analysis follows standard procedures and might be handled by a general program such as OMNITAB II (Hogben, Peavy and Varner, 1971), designed for such analyses. However, the generality of OMNITAB makes for awkward handling of large quantities of data such as are found in crystallography. Analysis of experimental crystallographic data requires at least four numbers, h, k, ℓ and a statistic, for each reflection. A typical data set of 3000 reflections would therefore require handling of about 12,000 numbers.

OMNITAB II was developed around the "worksheet" concept in that columns of data (i.e., large arrays for this application) are entered and various operations performed on these columns. Difficulties arise in cataloging statistics by Miller indices using OMNITAB because a separate array is required for each Miller index, i.e., 210 belongs to the h=2 layer, the k=1 layer, the ℓ=0 layer and the +++octant.

A program such as STATUS written specifically for crystallographic applications removes these difficulties. It also can make use of crystallographic data files produced by other crystallographic programs such as the X-RAY

SYSTEM (Stewart et al., 1972). It minimizes the amount of manipulation of the data required and facilitates statistical analysis during such stages of structure refinement as in comparing two data sets where the reflection lists must be searched for matches. It calculates relevant crystallographic quantities and breaks the data into appropriate subsets with minimum user intervention.

I.C. Application of Statistical Analysis to Crystallography

Abrahams and Keve (1971) point out that normal probability plots overcome many of the shortcomings of the conventional R factors because they make use of the individual residuals, $(Fo-Fc)/\sigma(F_o)$ and are useful in comparing sets of observed data (Fo) from different crystals of the same material. Structural parameters obtained from different samples of the same material can also be compared. We have extended the procedures of Abrahams and Keve to cover (i) data collected on the same sample by different experimental procedures to test experimental methods, (ii) data collected by the same experimental method on different samples of the same material to test sample characteristics, and (iii) sensitivity of models to experimental methods by examining parameters obtained using data from case (i).

The statistics or residuals can be examined without assumptions about their distribution. The residuals, $(Fo-Fc)/\sigma(F_o)$, contain all available information about the manner in which the fitted model fails to properly explain the observed variation in Fo. Generally, residuals are plotted against time sequence, the fitted variable Fc, and independent variables such as the Miller indices and the magnitude of the reciprocal lattice vector. Examination of trends among the residuals provides a detailed basis for judging how well the calculated quantities derived from the assumed model fit the observed data. Plots of various subsets of residuals focus on specific parts of the assumed model.

Although the indices are the only independent variables free from error (by virtue of their being integers) here we take the magnitude of the reciprocal lattice vector as being free from systematic error because random and systematic errors in the cell parameters are usually insignificant when compared with errors in Fo. Crystallographic least-squares refinements generally make the same assumption.

Three types of statistics are of use in crystallography. They are:

(i) $$\delta m_i = [Fo(A)_i - kFo(B)_i] \,/\, [\sigma^2(A)_i + k^2\sigma^2(B)_i]^{\frac{1}{2}}$$

which involves two sets of observed quantities, $Fo(A)_i$ and $Fo(B)_i$, together with their associated standard deviations, $\sigma(A)_i$ and $\sigma(B)_i$. $Fo(A)_i$ and $Fo(B)_i$ are related by the scale factor k, chosen so that $\sum_i \delta m_i^2$ is a minimum. This statistic is naturally independent of any model of the crystal structure and serves to focus attention on experimental procedures and sample characteristics such as homogeneity.

(ii) $\delta R_i = [|Fo|_i - |Fc|_i] / \sigma(Fo)_i$

where the observed and calculated quantities, Fo and Fc, are related by a scale factor which is usually determined by the crystallographic least-squares procedure by minimizing

$$\sum_i \delta R_i^2 = \sum_i [(|Fo|_i - |Fc|_i) / \sigma(Fo)_i]^2 .$$

$\sigma(Fo)_i$ is the standard deviation of Fo. The δR_i statistic, which is actually a residual, measures the discrepancy between observed structural amplitudes and those calculated from the model parameters.

(iii) $\delta p_i = |p(A)_i - p(B)_i| / [\sigma_p^2(A)_i + \sigma_p^2(B)_i]^{\frac{1}{2}}$

where the model parameters $p(A)_i$, $p(B)_i$, and associated standa deviations $\sigma_p(A)_i$ and $\sigma_p(B)_i$ are derived from least-squares refinements using two data sets obtained from (a) different crystals of the same material or slightly different materials (one sample being more impure than the other, for example) or (b) the same crystal but by different experimental procedures. This statistic measures the differences between models and hence examines their sensitivity to differences in samples and experimental procedures.

The information and trends contained in sets of these statistics is revealed best by plots. Principal types of plots and their diagnostic capabilities will be discussed for each of the three statistics in turn.

1. Use of the δm_i Statistic

The statistic δm_i can be used in three major types of plots, as discussed below. In addition, a subset $\delta_{hk\ell}$ of the δm_i, chosen on the basis of reflection class, i.e., h00, hk0, etc., or layer, or octant can be used in each type of plot. This enables both isotropic and anisotropic effects to be examined.

(a) Order statistics can be used to relate the δm_i (ordered in increasing magnitude) to the quantiles X_i expected for a normal distribution. Meaningful conclusions that can be drawn from plots of this type have been discussed by Abrahams and Keve (1971) and a summary based on their discussion follows. The distribution of δm_i, calculated from two Fo values and their associated σ values, is normal (Gaussian) if the δm_i values contain only random error. In that case a linear δm_i vs X_i plot arises and the deviation of its slope from unity indicates the scale factor to be applied to the pooled σ value used in obtaining the δm_i. A markedly non-linear plot or a plot with non-zero intercept indicates that the two data sets differ systematically or that one or both sets of σ(Fo) systematically mis-estimated. Thus, the two data sets cannot be related by a single scale factor and cannot come from the same normal population. A search for the source of discrepancy must then be made.

If the distribution of δm_i is identical to the normal distribution the slope will be 1.0. A slope greater than 1.0 indicates that the average value of σ(Fo) is too small. This situation may arise in two ways: (1) The estimate of the random error as derived from counting statistics in Fo is too small, for example, if significant coincidence losses in counting occur, or (2) a more likely effect is that the Fo values contain systematic differences which are comparable or larger than the random error. Hence, Fo(A)-Fo(B) is greater than ΔF_o based on random error only. Situations (1) and (2) may be distinguished by such plots as those involving δp_i if models have been refined for the two sets, or by plots of δm_i against d* and $\log(Fo^2/\sin 2\theta)$. Trends in the latter plots indicate systematic differences in the two data sets.

A slope less than 1.0 when the δm_i are plotted against the normal distribution indicates that the $\sigma(F_o)$ are on average too large.

An appropriate way to adjust the scale of σ(Fo) is to apply the observed slope in inverse proportion to the average sigma for each data set. Systematic differences may exist in the two data sets in addition to the random error. Although δm_i plots used alone cannot give an idea of the relative magnitudes of these systematic differences(unless σ(Fo) is known to be correct), a combination of δm_i and δp_i plots can. Diagnosis using this combination is amply discussed in the Appendix of the paper by Abrahams and Keve.

(b) The δm_i statistics can be plotted against d* for all the data or for various reflection classes. Such plots should show no trends if there are no systematic differences between the two data sets. A horn-shaped plot symmetrical about the abscissa would indicate mis-estimation of $\sigma(Fo)$ varying in a systematic way with d* and sin θ. Plots involving different reflection classes or layers can be used to check on the anisotropy of error in an effect such as correction for absorption, which is a function of $\frac{\lambda}{2}$ d*, i.e., sin θ.

(c) Plots of δm_i against $\log(F_o^2/\sin 2\theta)$ for all the data and various reflection classes permit some estimate of the importance of extinction effects. The variable $F_o^2/\sin 2\theta$ was chosen because it is proportional to crystallographic Q, (i.e., $\lambda^3 F^2/V^2 \sin 2\theta$), scaled by V^2/λ^3. This scaling is of no consequence here because it enters into the scale factor when the two data sets are scaled together. Both primary and secondary extinction effects are proportional to Q and so these plots provide tests for differences in extinction between the two data sets. Differences in slope for reflection classes would be indicative of differences in anisotropic extinction. Horn-shaped plots are indicative of mis-estimation of $\sigma(Fo)$ in a systematic manner with F_o^2. One might suspect simultaneous diffraction if large δm_i values tended to be associated with small values of F_o^2.

Systematic differences in the observed data sets result from such effects as:

(i) errors in the measurement of Fo_i (set 1) and Fo_i (set 2),
(ii) physical differences in the samples used, e.g., differences in the degree of primary and secondary extinction or in absorption corrections because differences in crystal shapes were not fully taken into account, and
(iii) chemical differences between the samples used for collecting data. These chemical differences may also lead to physical differences, such as in extinction or diffuse scattering, depending on the degree of disorder.

2. Use of the δR_i Statistic

The δR_i statistic, which is the residual obtained in least-squares analysis, can be plotted in the same manner as the δm_i statistic.

(a) The δR_i can be formed into order statistics and plotted against the expected normal distribution. A linear plot with zero intercept (within limits corresponding to the error in the least-squares scale factor) indicates that the Fo contain no systematic error or trends that cannot be accounted for in some manner by the refinement model. Conversely, a non-linear plot indicates the presence of effects not taken account of by the model. A linear plot with a slope less than 1.0 would indicate overestimation of $\sigma(Fo)$, or if the $\sigma(Fo)$ values are known to be correct, which is rarely the case, that the model has been able to partially account for some systematic effect in the Fo set. A slope greater than one would indicate an inadequate model or underestimated $\sigma(Fo)$ values. These various situations may be clarified by further plots as described below.

(b) The residuals δR_i may be plotted against d* for all the data and for various reflection classes. If the model is adequate, a linear plot of uniform scatter and with a slope approximately zero should be obtained. A curved or sloped plot indicates an effect in the observed data not adequately accounted for by the model. In addition, if the scatter is not uniform, i.e., large $|\delta R_i|$ tend to be at small d* while small $|\delta R_i|$ tend to be at large d*, giving a horn-shaped plot about the abscissa, a mis-estimation of $\sigma(Fo)$ in a systematic manner with d* is indicated. Differences between plots involving reflection classes such as layers may suggest experimental or sample orientation effects. Plots showing uniform scatter with a non-zero slope would indicate a model with inadequate occupancy factors or thermal parameters.

A linear plot indicates that the residuals belong to a normal distribution and justifies the use of variance tests based on the F-distribution, such as Hamilton's R-factor ratio test (Hamilton, 1964).

(c) The δR_i statistics may be plotted against log $(Fc^2/\sin 2\theta)$, an appropriate function of the "fitted variable". As indicated previously in section 1 where this type plot was discussed in connection with δm_i, these plots can be used to test for extinction effects, which are proportional to $Fc^2/\sin 2\theta$. Plots involving different reflection classes and octants permit tests for anisotropy when using unmerged data. Again, plots should show uniform scatter and no trends. A regular horn-shaped linear plot symmetrically placed about the abscissa would indicate that the model adequately fits the data, but that the values of $\sigma(Fo)$ have been mis-estimated in some systematic way. If the largest positive δR_i values tend to be associated with small values of Fc, one might check for simultaneous diffraction, which would serve to increase Fo considerably for small values of Fo.

3. Use of the δp_i Statistic

The third type of statistic, δp_i, is plotted against the corresponding half-normal distribution quantiles because the order of the parameters $p(A)_i$ and $p(B)_i$ is not significant and the sign of δp_i is meaningless. No scaling is involved because the parameters are on the same scale. A linear plot with a slope of 1.0 and intercept approximately 0 should result. A non-linear plot indicates the presence in data sets of effects which cause the parameter values derived from the two sets to systematically differ from one another. A linear plot with a slope less than 1.0 indicates the presence in the data sets of some systematic errors or effects comparable in size to the random errors that the refinement model cannot account for, a situation which results in large values for $\sigma(p)_i$. Such effects may also be indicated by trends in the residuals, e.g., in various δR_i plots.

A slope greater than 1.0 indicates that the average $\sigma(p)_i$ is too small. This may happen if the set of Fo contains systematic trends which the model parameters have been able to absorb, and should be confirmed by the absence of noticeable trends in δR_i plots. Any systematic bias should ideally be removed from the model parameters. Alternatively, $\sigma(p)_i$ for each parameter set may be multiplied by the slope so that the resulting $\sigma(p)_i$ values will provide an estimate of the variance due to both random errors and the mean-square errors associated with the systematic bias. Abrahams and Keve provide in the Appendix and figure 6 of their paper a detailed explanation of how a combination of δm_i and δp_i plots may be used to estimate the relative magnitude of random and systematic errors. δR_i plots help to formulate corrections and improve the model.

I.D. General Description of Program STATUS

1. Overall Procedure

The uses of the STATUS program have been described in Section I.C. Because statistical analyses have been applied specifically to crystallographic quantities in STATUS, we have taken care to write the program in a manner which minimizes data manipulation and sorting while allowing maximum flexibility. This is accomplished by keeping the four arrays DM, X, LOGFC and LOGHKL in core at all times. Array DM holds the statistics under analysis. Array X contains the normal quantiles, d* values, or $\log(Fc^2/\sin 2\theta)$ values. Cataloging of the I-th statistic into its reflection classes and octant is accomplished by non-zero bits in the associated computer word LOGHKL(I). For a 36 bit word, as available on the

UNIVAC 1108, this allows up to 33 classes (3 bits are required to specify the octant) while requiring one word per statistic. The LOGFC(I) word holds the address on mass storage of the reflection information associated with the I-th statistic. Other routines such as DMDQ, DMCALC, and PRYNT use this mass storage address to locate the reflection information (stored on disk or drum).

Analysis of residuals from least-squares refinement is straightforward. Reflection information consisting of the Miller indices, Fo, Fc, sigma and, as an option, the mean path length or the extinction coefficient are read in. The statistic is calculated and stored. The reflection information is written on word-addressable mass storage and its address is stored in the LOGFC array. Fc must be available if plots involving the fitted variable, $Fc^2/\sin 2\theta$ are to be made. Note that most files produced by crystallographic programs [e.g., files used as input for a Fourier calculation and the X-RAY SYSTEM (Stewart, et al. 1972; hereafter referred to as XRAY) binary file] contain the Miller indices, Fo and Fc, rather than the statistics $(Fo-Fc)/\sigma(Fo)$ even though least-squares programs calculate this quantity or an equivalent residual.

Treatment of two sets of Fo data is a little more complex in that the two values of Fo and their associated $\sigma(Fo)$ must be matched up unless the input files are the same length and in the same order. Also the scale factor relating the two data sets must be found before the statistics can be calculated. Subroutine SCALE matches up the two sets of Fo and optionally writes the Miller indices and the two Fo and $\sigma(Fo)$ values on unit NR. SCALE also finds the value of the scale factor. Optionally, subroutine SPEC can be called to refine the value of the scale factor to ensure that the sum of the squared-statistics is a minimum. Once generated on unit NR by STATUS, the file of matched values can be read by subroutine REREAD allowing one to make subsequent runs using different values of the scale factor without requiring the initial time-consuming sort. This feature would not be available if only the indices and statistic were saved on an output file for subsequent runs.

The order of data treatment was developed so that at most three sorts are required. In order to produce normal probability plots the incoming statistics must be sorted in order of increasing magnitude to form order statistics. The corresponding X, LOGFC, and LOGHKL arrays are also rearranged to preserve relative indexing. Generation of d^* and $Fc^2/\sin 2\theta$ values is accomplished by subroutine

DMDQ using the addresses stored in array LOGFC to locate on mass storage the indices and Fc values that correspond to a given statistic. Subroutine CATLOG is also called at that time to classify the statistic because the indices are currently available. The d* values stored in the X array are then sorted in order of increasing magnitude and the corresponding statistics in DM and the catalog information in LOGFC and LOGHKL are similarly rearranged. Various subsets of the statistics, such as those belonging to the hk0 layer, can then be selected and plotted. In this manner the variation of the residuals with d*, for example, may be examined as a function of reflection class, layer, or octant with only one sorting of the data.

STATUS can also be used for statistical analysis of structural parameters. They are read in by subroutine INFING if the Fourier file written by least-squares program RFINE4 is available or in other cases by subroutine USER which must be written by the user. Subroutine MTCHEK finds all atoms with two sets of parameters and STATUS calls subroutine DPCALC to calculate the statistic. Only a half-normal probability plot is required in this case since the order of equivalent parameters is irrelevant. Subroutine UNIMED obtains the medians for the I-th order statistics and these are used together with the percentage points of the normal distribution to obtain half-normal quantiles. This avoids the errors associated with small samples (Hamilton and Abrahams, 1972). All structural parameters, statistics and half-normal quantiles are printed out.

2. Crystallographic Data Required as Input

In keeping with the design objective, STATUS makes use of crystallographic data files generated by most programs used for crystallographic calculations. A comparison of two sets of observed data requires the Miller indices; h, k, ℓ, Fo(set 1), Fo(set 2), σ(Fo)(set 1), and σ(Fo)(set 2), quantiles available on most crystallographic data files. Analysis of least-squares residuals requires Fc instead of the second set of Fo. Fc is available on files used as input to Fourier analysis.

Input of two sets of observed data is accomplished by subroutine FREAD which reads a BCD file. Optionally, STATUS will produce a binary file of matched Fo and σ values which can be used as input for subsequent runs. Quantities for analysis of least-squares residuals may be read from the XRAY binary file by subroutine INFOFC or from the Fourier file written by RFINE4 (Finger and Prince, 1975) by subroutine FINGFO. The specifications of these input files will be discussed in the section on Input Files.

Input of atomic parameters is treated somewhat differently because of the wide variety of sources possible. Presently, they may be read from the Fourier file written by RFINE4 or from cards or another file by subroutine USER which must be written by the user. Subroutine USER allows for structural parameter input from many different sources including published works.

Only a small amount of information needs to be input on cards. This includes a title to identify the job, the unit cell parameters, the wavelength at which the data were collected, and the form of the input data (Fo or Fo^2). If the optional spherical absorption correction is requested the values of μr for the two data sets are required. Formats for card input will be discussed in the section on Card Input.

3. Format of the Output Plots

All plots are output on the line printer so that no special plotting devices are required. Thus the user has a visual representation of the statistical analysis as soon as it is completed. A detailed description including examples of actual output will be discussed in the section on Output Plots.

The title of each output plot indicates the type of plot, amount of data in the set or subset, number of points in the plot and the octant or the reflection class the statistics belong to if a subset is being plotted. The actual quantities plotted are listed under each plot so information corresponding to the residuals plotted can be referred to.

In producing normal probability plots, the program ignores those statistics which lie on the extremes of the plot (i.e., with $|X_i| > 2.0$, where X_i is the normal probability plot quantile) so that they will not influence the calculated slope and intercept out of proportion to their importance. The slope, intercept and goodness of fit are calculated from the functions (i) $y = mx + c$, (ii) $y = mx^2 + nx + c$, and (iii) $y = mx^3 + nx^2 + px + c$. If the plot is non-linear (somewhat kinked) the fit of the cubic-containing form is appreciably better as revealed by the goodness of fit. However, in this case the slope, given as the coefficient of x in the equation, is that of the central portion, and is probably smaller than it will be when the scaling error has been corrected. Thus, the non-linear functions serve mainly as diagnostic services to aid in judgment of linearity rather than to obtain estimates

of slopes. The δm_i quantities, the least-squares line and the cubic-containing function are plotted on the same plot using the symbols · + and *, respectively. Multiple points are indicated by a digit signifying the number of superpositions. If there are more than 50 points in the data set, every (N/50 + 1) th point is plotted starting with the point halfway through the first range. When residuals are being analyzed the variation of scatter of the residuals with the independent variable is important. If (N/50 + 1) is six or more (i.e., each plotted point represents an interval of six or more data points) the average positive and negative deviation from the values calculated by the linear function for the interval is plotted. These values are indicated by the symbols U and L, respectively, and aid in judging the variation of the scatter with respect to the independent variable.

4. Program Configuration and Requirements

The program STATUS consists of a control routine (also called STATUS) and 35 subroutines which perform the actual functions. This modular form allows use of standard documented subroutines when possible and makes it more convenient to modify for specific situations if necessary. It also allows for overlays and mapping for conservation of core if needed. The present configuration requires about 10,000 (decimal) words (9,000 if mapped) for the code and 40,000 for data. This allotment enables treatment of 7200 statistics or 1000 structural parameters (100 atoms in the asymmetric unit) which is more than adequate for most crystallographic situations where the use of STATUS is warranted. The amount of storage reserved for data can easily be changed as described in the section on the FORTRAN code.

STATUS requires two special functions in addition to the standard library functions. They are NTRAN, a UNIVAC routine handling transfers of information between central memory and mass storage and FLD for manipulation of specified bits in a computer word. The section on Features Specific to UNIVAC 1108 FORTRAN V describes these functions and where they are called in more detail. Users can substitute appropriate routines for their computing systems. There are a few FORTRAN statements which are not American Standard FORTRAN and they are discussed in the section on Features Specific to UNIVAC 1108 FORTRAN V.

A maximum of six logical units including the card reader (or remote terminal) and line printer is required for I/O. One unit (NDRUM) refers to word-addressable random access mass storage and is used by the program for temporary storage. Two units (NTAPEA, NTAPEB) can refer to any convenient devices holding the input data files. An optional unit (NR) may also refer to any convenient mass storage device (tape, drum or disk) but not a card unit since information in that file is buffered to save I/O time.

The next section presents the subroutine hierarchy and call sequence to aid those who wish to map the program. The present mapping scheme is as follows:

Main segment (always in core)
Routines: STATUS, DRUMRD, DRUMRT, POSITN, MAXHKL, HKLGEN

First sub-segment-starts at the end of the main segment
Routines: PRIME, CELL, FREAD, SCALE, ABSORB

Second sub-segment-overlays the first.
Routines: REREAD, SPEC, DMCALC, NFACT

Third sub-segment-overlays the first and second
Routines: INFOFC

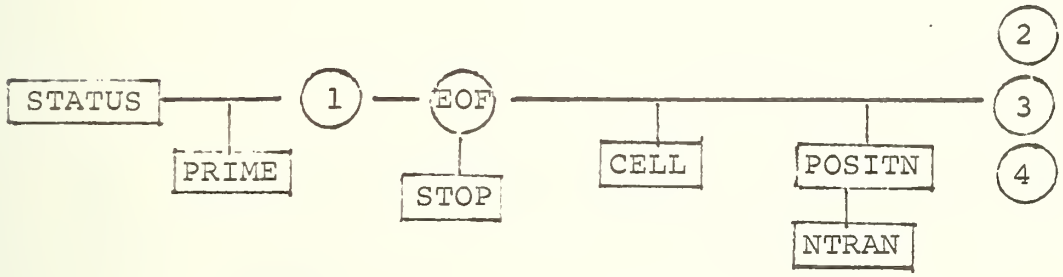
Fourth sub-segment-overlays the previous segments
Routines: SORT, PROB, PINV, UNIMED, PLOTEM, LSFIT, FITCHK, PLOTS, SIMLEQ, FINGFO

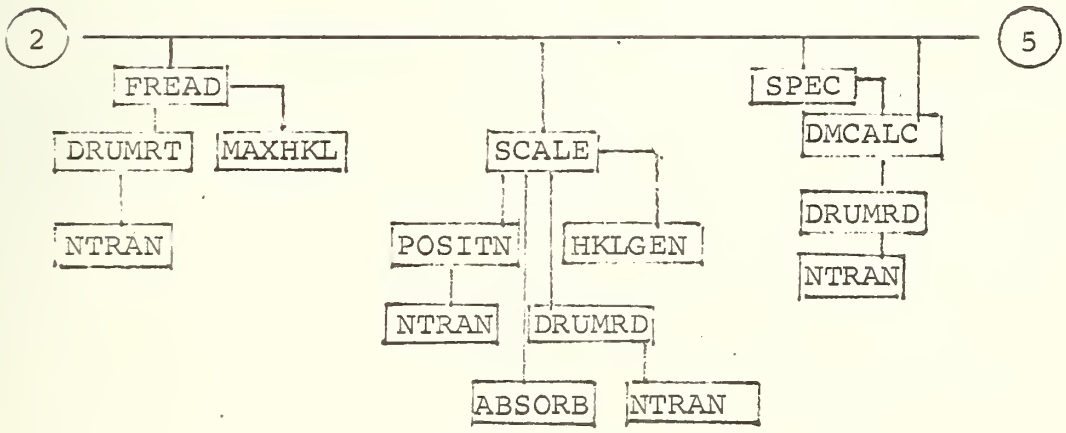
Fifth sub-segment-starts at end of the fourth
Routines: DMDQ, ANISO, PRYNT, CATLOG

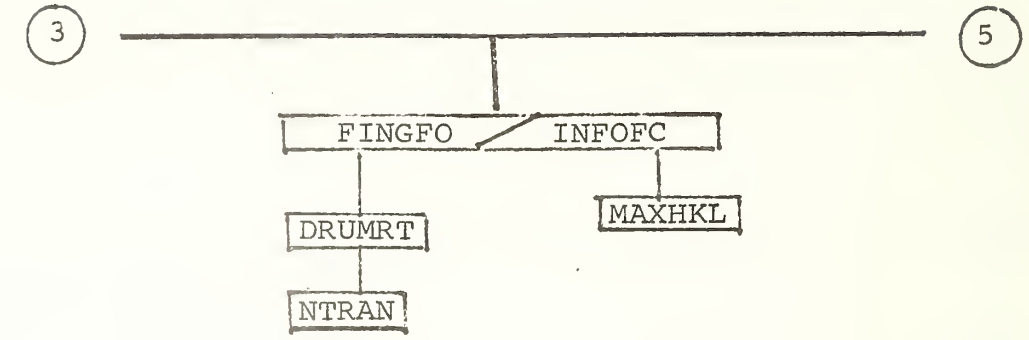
Sixth sub-segment-overlays the fifth.
Routines: DPCALC, INFING, MTCHEK, PRYNTT, USER

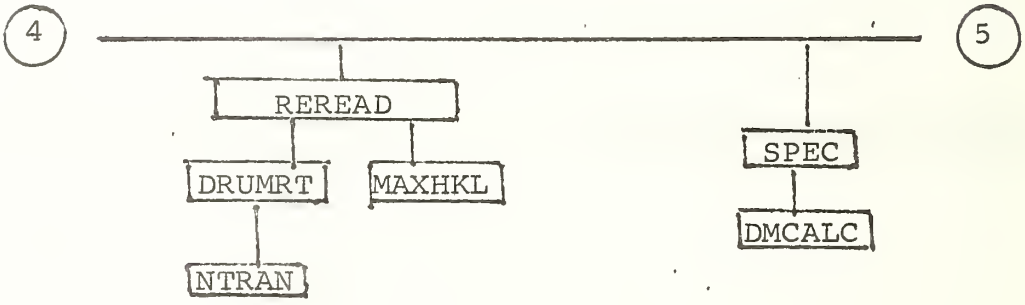
5. Subroutine Hierarchy and Call Sequence

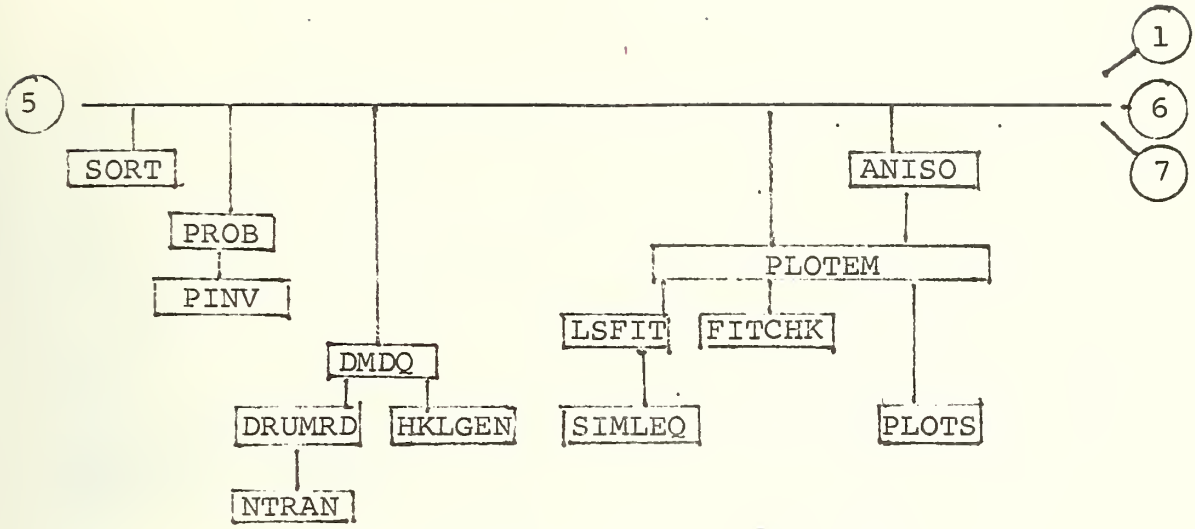
As an aid to program modification by users who may need to replace machine specific routines or to reduce the size of the program the order in which subroutines are called is given below in diagramatic form.



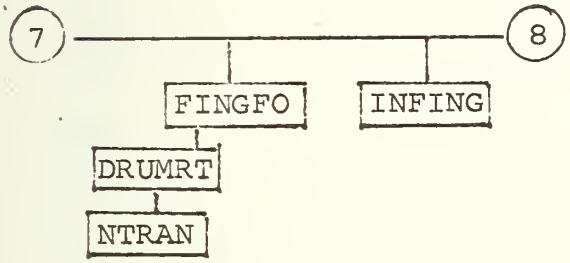


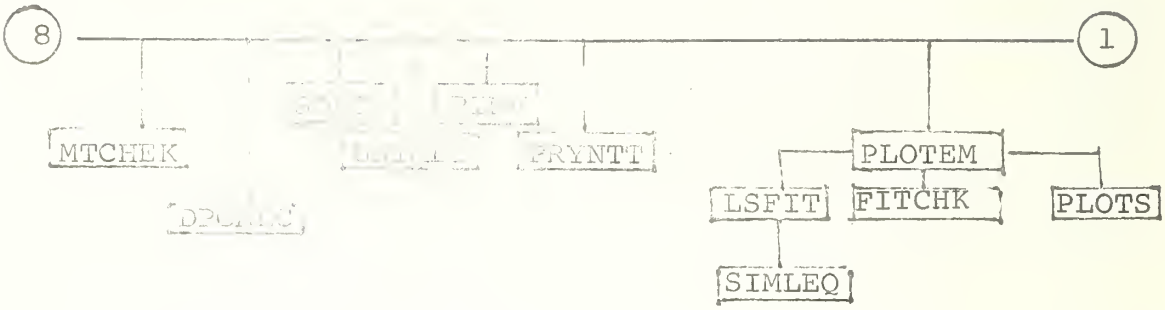

3
5
FINGFO
INFOFC
DRUMRT
MAXHKL
NTRAN


4
5
REREAD
DRUMRT
MAXHKL
NTRAN
SPEC
DMCALC


1
5
6
7
SORT
PROB
PINV
DMDQ
DRUMRD
HKLGEN
NTRAN
ANISO
PLOTEM
LSFIT
FITCHK
SIMLEQ
PLOTS


6
8
USER


7
8
FINGFO
INFING
DRUMRT
NTRAN


8
1
MTCHEK
PRYNTT
PLOTEM
LSFIT
FITCHK
PLOTS
SIMLEQ

6. Description of Subroutines

ABSORB - Calculates the transmission factors for a spherical crystal given the Miller indices and reciprocal cell constants.

ANISO - Calls PLOTEM to set up plots for various reflection classes according to variable NCL. Sets flag II so that LSFIT, FITCHK and PLOTEM take only those δm_i values and reflections whose catalog word LOGHKL has the II-th bit non-zero.

CATLOG - Catalogs a reflection according to class (h00, 0k0, etc.), layer, or octant, by setting equal to 1 the corresponding bit in word in the LOGHKL array which is associated with that reflection. Requires the indices and reflection number.

CELL - Reads in real cell, calculates reciprocal cell and stores results in common block G.

CHANGE - non-existent at present but will contain cards which will allow user to keep track of any changes made to present program.

CHANGE - Contains comment cards which allow user to keep track of changes made to program. Also contains general update information and version date.

DMCALC - Reads off mass storage using the addresses stored in LOGFC the Fo and σ values for two corresponding reflections. Calculates δm_i (unit deviate) which is stored in array DM, $\langle \delta m \rangle$, e.s.d. of δm_i, and a weighted R factor over the two data sets. Prints out these quantities to give a characterization of the δm distribution.

DMDQ - Reads from mass storage using the addresses stored in array LOGFC the indices and other information pertaining to the reflection associated with a given δm_i value. Depending on the variables ND and NQ in its call argument list, DMDQ calculates d* or crystallographic Q = $(Fo^2/\sin 2\theta)$ and stores the result in the X array so that δm can be plotted against whatever is in X. DMDQ also calls subroutine CATLOG to catalog the reflections by classes if desired.

DPCALC - Calculates δp_i for two corresponding atomic occupancy, positional or thermal parameters.

DRUMRD - Reads a record containing the indices, F and σ for one reflection from word-accessible mass storage (drum or disk).

DRUMRT - Writes one reflection record on word-accessible mass storage.

FINGFO - Specialized subroutine for reading indices, Fo, σ(Fo) and Fc for each reflection from end of a Fourier file written by program RFINE4 (Finger and Prince, 1975). Calculates the corresponding δR_i value and stores it in array DM. Can also be used to obtain two sets of Fo and σ(Fo) from different Fourier files containing different data sets obtained from the same compound.

FITCHK - Calculates points from the least-squares line and cubic-containing functions obtained by LSFIT and stores them, together with the original δM_i values, for combined plotting against abscissa values stored in array X. Also calculates the goodness-of-fit for the various least-squares functions.

FREAD - Reads reflection records containing indices, Fo and σ(Fo) from input units NTAPEA and NTAPEB. Stores the indices, Fo and σ on disk or drum and keeps the value of the slowest varying index in array ID for use in the matching procedure in subroutine SCALE. In this way SCALE is able to read in only those reflections with the same value of that index, conserve memory space and match efficiently using small batches. The position of each reflection in the mass storage file is known from the position of its slowest varying index in the ID array, since both the mass storage file and the ID array were written incrementally as the reflections were read in.

HKLGEN - Unpacks the Miller indices from word JKL according to the order specified by JJ, KK and LL.

INFING - Specialized routine to read parameters from the end of the Fourier file written by program RFINE4.

INFOFC - Reads h,k,ℓ, Fo, Fc, and σ(Fo) from XRAY binary file. Calculates δm, writes h,k,ℓ, Fo, and Fc on mass storage, stores address in array LOGFC.

LSFIT - Calculates least-squares line and quadratic- and cubic-containing functions relating variables X and Y. Y is the complete set of statistics, δM_i, or a subset thereof. X may be the normal quantiles, d* values, Q values or subsets if reflection classes are considered.

MAXHKL - Finds maximum and minimum values of Miller indices and stores values in common block D for further use.

MTCHEK - Matches up parameters for two parameter sets, checks for unequal numbers of atoms (e.g., one set may include hydrogens, etc.). Atom matching is based only on the alphanumerical name given for each atom, so that corresponding atoms must have the same name, e.g. Ca 1, with an identical number of embedded blanks.

NFACT - Computes N factoral which is used in routine SPEC.

PINV - Inverse probability function (percentage-point) used to obtain the normal quantile corresponding to i-th value of P (X) for the j-ordered statistics.

PLOTEM - Sets up plot titles, calls LSFIT for least-squares fits, writes out coefficients of fitted functions (lines and curves), calls FITCHK to calculate goodness-of-fits, prints out titles for plots, and calls PLOTS to do the actual plotting.

PLOTS - Routine to plot up to five curves stored in X and Y arrays, using the line printer. Keeps track of multiple points and finds its own limits for the axes. Does not call a new page and does not label the axes.

POSITN - Dummy routine at present. User may add statements to position logical unit NTAPE so that NFILE is ready for I/O.

PRIME - Initializes the logical unit variables: IN - card reader, NOUT - printer, NDRUM - mass storage. Separate routine allows user flexibility and easy modification.

PROB - Calculates i-th value of P (X) for the j-ordered DM values so that PINV can be used to obtain the corresponding normal quantile.

PRYNT - Prints out all δM_i values and corresponding reflection information or only those greater than 2.0 if desired. Reads values from mass storage and requires LOGFC word which tells where reflection record corresponding to δM_i value is located. Calls HKLGEN to obtain the indices.

PRYNTT - Prints out δp_i values for parameters and corresponding atom names, etc. Requires LOG word which tells what atom and type of parameter (occupancy, positional, etc.) go with a given δp_i value.

REREAD - Reads the quantities Fo and σ(Fo) for data sets one and two from a file written on unit NR by routine SCALE. Thus, the scale factor may be adjusted without re-sorting the original lists of data to find the two Fo and σ values having common indices. New δm_i values are calculated.

SCALE - Reads the indices, Fo and σ(Fo) for the two data sets from mass storage and finds reflection information having common indices. Calls MAXHKL to get range of indices, applies absorption corrections if needed. Also buffers up the matched reflection list and outputs this list on unit NR if desired. Calculates the scale factor required to put the two data sets on a common basis by forming $\sum_i Fo_i$ (set 1) / $\sum_i Fo_i$ (set 2).

SIMLEQ - solves simultaneous equations in matrix form AX=B.

SORT - Sorts the δM_i values in ascending order so that they form order statistics and keeps the LOGFC words in the same order as the sorted δM_i values. The LOGFC word tells where reflection information corresponding to the i-th δM_i value is located on mass storage and enables reflection information associated with the i-th δM_i value to be accessed easily by routine PRYNT. For atomic parameters, the δP_i values are stored in array DM and the ordinal numbers of corresponding atoms are stored in LOG. The SORT subroutine orders the δP_i values and keeps the LOG values in correspondence with the δP_i values. Parameter information is printed via PRYNTT.

SPEC - Obtains the scale factor needed to make $\sum_i \delta m_i^2$ a minimum. This may be different from that obtained in a routine SCALE.

UNIMED - Computes an approximation to the median of the i-th order statistic. Used here to generate quantities needed to obtain quantiles for the half-normal plots.

USER - Routine to allow user to input atomic parameters from various devices and formats. Must be user written.

I.E. References

Abrahams, S. C. (1964) Acta Cryst. 17, 1327.

Abrahams, S. C. and Keve, E. T. (1971) Acta Cryst. A27, 157.

Hamilton, W. C. (1964) Statistics in Physical Science, Estimation, Hypotheses Testing, and Least Squares. New York: Ronald Press.

Hamilton, W. C. and Abrahams, S. C. (1972). Acta Cryst. A28, 215.

Hogben, D., Peavy, S. T. and Varner, R. N. (1971) OMNITAB II User's Reference Manual, NBS Technical Note 552.

MacDonald, J. R. (1972) Science 176.

Finger, L. W. and Prince, E. (1975) A System of FORTRAN IV Computer Programs for Crystal Structure Computations, NBS Technical Note 854.

Stewart, J. M., Kruger, G. J., Ammon, H. L., Dickinson, C., and Hall, S. R. (1972) The X-RAY System of Crystallographic Programs. Technical Report TR-192, University of Maryland, College Park, Maryland.

II. Program Manual

II.A. Card Input

Although this section is called Card Input, STATUS could be run from a remote terminal if the printer output can be directed elsewhere. Such details depend on the particular operating system and are not discussed here. The following description refers to cards (or card images) that are read by the program STATUS itself. Card input is referenced by logical unit IN and is presently set to 5 by subroutine PRIME.

Card 1 Title Card - read by main routine STATUS
FORMAT (12A6)
Col 1-72 any title

Card 2 Cell Card - read by subroutine CELL
FORMAT (13X,3F8.3,3F9.5) (Compatible with XRAY FORMAT)
Col 14-21 A cell dimension
22-29 B cell dimension
30-37 C cell dimension
38-46 Alpha or Cos Alpha
47-55 Beta or Cos Beta
56-64 Gamma or Cos Gamma

Card 3 Control Parameters - read by main routine STATUS
FORMAT (A3,12I5,I3,16I2)
Col 1-3 Punch CTL - program checks third card read for this label
5 NTAPEA - first source of input data-reflections and/or atomic parameters.
10 JJA - sort order of indices of reflections
15 KKA
20 LLA
The sort order governs the order in which H,K and L are packed together in the "JKL" word, (i.e., - JJA=2, KKA=1, LLA=3 would pack the indices in the order K,H,L.) The reflections are broken into batches with the first index constant in the packed word (i.e.-in the example above, one batch will have all K=0, another will have all K=1, etc.). The sort will run fastest for a given number of reflections when there are the most batches, so the sort order should give 1 to the index associated with the largest (real space) dimension of the unit cell.

25 NTAPEB - second source of input data.

Col 30	NFTEST	0	no plotting of reflection data.
		1	read input from RFINE4 Fourier tape and compare Fo and Fc.
		2	read in two different reflection sets, each from a different RFINE4 Fourier tape.
		3	read reflections from two different tapes in XRAY BCD input FORMAT e.g., Col 15 less than indicator 19 H 23 K 27 L 28-37 Fo 38-47 σ(Fo).
		4	read tape previously written by subroutine SCALE. The tape is designated NTAPEA (Col 5), the input file on this tape is designated by NFILEA (Col 60).
		5	read Fo and Fc from XRAY binary file assigned to unit NTAPEA.
35	NPTEST	0	no atomic or group parameters.
		1	atomic parameters from RFINE4 Fourier tape.
		2	parameters read in by special subroutine USER supplied by the user.
40	NFILEA	No. of input file on NTAPEA, 1 for first file, 2 for second file, etc.	
45	NFILEB	Same information for unit NTAPEB	

Following section selects output options.

48	NPRINT	0	no printing of statistics (DM) and associated values.
		1	print all DM and associated values.
		2	print only those with absolute value greater than 2. This is about 5 percent of the data. About 55 DM will be printed per page of output.
50	NO	1	for plots of statistics vs normal quantiles
		0	no such plots
52	NOC	Governs reflection type for normal quantile plot.	
		0	all data regardless of class or octant.
		1	standard classes H00, 0K0, 00L, HK0, H0L, 0KL, HHH, HHL, HKK, HKH, and HKL (non-zero indices).

2 H layers, up to 10
3 K layers, up to 10
4 L layers, up to 10
5 does NOC = 1 and NOC = 2
6 does NOC = 1 and NOC = 3
7 dces NOC = 1 and NOC = 4

54 NLAYO Number of layers for normal plots if NOC.GT.1.

56 NOCTO Option specifying octants to plot.
0 all octants plotted together for each class or layer. Use NOCTO = 0 for merged data.
1 HKL and -H,-K,-L, i.e., +++ and --- (check triclinic)
2 +++, --- and +-+ (useful for monoclinic Y axis unique).
3 +++, ---, +-+, and ++- (useful for monoclinic Z axis unique).
4 +++, ---, ++-, +-+, -++ (useful orthorhombic equivalents).
7 all octants plotted separately for each class or layer.

Note the following concerning pages of output.
Number of plots generated - 1 if NOC = 0
plus 11 * (NOCTO + 1) if NOC = 1
plus (NLAYO) (NOCTO + 1) if NOC = 2,3,4
plus (11+NLAYO)(NOCTO+1) if NOC = 5,6,7
with two pages output generated per plot.

58 ND 1 for plots of statistics vs d* (2sin(θ)/wavelength)
0 no such plots

60 NDC Governs reflection type for d* plots values as above for NOC.

62 NLAYD Number of layers for d* plots if NDC.GT.1.

64 NOCTD As for NOCTO above.

Estimate number of plots by replacing variables in above formulae with NDC, NLAYD, NOCTD.

66 NQ 1 for plots of statistics vs $Q(Fo^2/\sin(2\theta))$
0 no such plots

68 NQC Values as above for NOC

70 NLAYQ Number of layers for Q plots if layers being done.

72 NOCTQ As for NOCTO above.

Calculate number of plots as above using NQC, NLAYQ, NOCTQ.

It is possible to produce a maximum of 588 plots or approximately 1200 pages. Therefore do not abuse the flexibility of this program. Program checks to see how many plots will be generated and stops if more than 120 would be produced in one run.

74 NR Output of matched reflections (pair of Fo, etc.) on unit NR allows for SCALE adjustments without resorting.

76 NFILER File to be used on unit NR.

78 NABS Greater than zero for spherical absorption corrections.

80 NINDM Greater than zero for adjustment of scale factor to ensure that $\Sigma\delta m^2$ is a minimum.

Card 4 Read by main routine STATUS only if NQ or NABS $\neq$ 0.
FORMAT (2F10.7,I1,4F5.2)

Col 1-10 Wavelength--needed for Q plots and absorption correction.

11-20 Scale factor to be applied to second set of structure factors (needed only if matched reflection being read by subroutine REREAD from file written by SCALE).

21 0 for Fo data, 1 for Fo^2 data input.

Following 2 quantities needed only if NABS.GT.1

22-25 μR for first data set.

26-30 μR for second data set.

31-35 Delta μR (max μR - min μR) for first data set (may be left blank).

36-40 Delta μR for second data set (may be left blank).

II.B. Data Files

1. Input Files

A new run will require at least one and more likely two input files which are referenced by logical units NTAPEA and NTAPEB. These input files may have different formats depending on their source. Control parameter NFTEST allows for selection of input type (setting NFTEST = 0 means no input is expected). Different types will be discussed in turn.

a. Comparison of two sets of observed structure amplitudes.

If NFTEST = 2 subroutine FINGFO will read the Fourier file written by program RFINE4. FINGFO was specifically written for use with RFINE4.

If NFTEST = 3, subroutine FREAD will be called to read a file having values of h, k, ℓ, Fo and σ for each reflection. The present form of the read statements allows for a BCD file of card images compatible with the format (13X, I2, 3I4, F10.2, F10.4), which is compatible with the XRAY reflection data format. However, a different format or a binary file may be read after the read statement in FREAD has been changed. A description of the procedure and flow chart for FREAD has been included in Section III.E to aid in modification.

If NFTEST = 4, subroutine REREAD is called to read a file of two sets of corresponding Fo's and their σ values. This file must have been previously written by subroutine SCALE and is a binary data file consisting of blocks of 490 words each (this is a logical record for a FORTRAN write operation). Each block consists of 70 reflection records. A reflection record consists of the seven quantities h, k, ℓ, Fo(set 1), Fo(set 2), σ(set 1) and σ(set 2). Logical unit NR refers to this file. No modification should be necessary unless the user wishes to optimize storage (such optimization requires a detailed knowledge of the local computing system).

b. Comparison of a set of observed and calculated structure amplitudes. If NFTEST = 1, subroutine FINGFO will read Fo and Fc values from the Fourier file written by program RFINE4.

If NFTEST = 5, subroutine INFOFC will read in indices, Fo, Fc, and σ from the XRAY system binary data file. A complete description of this file is given in the report describing the XRAY system. Subroutine INFOFC can easily be modified by users wishing to read from a different type of file.

A description of the procedure and flow chart for INFOFC is given in Section III.F. This description taken together with the listing should aid in modification.

2. Output Files

All output except the optional file of matched reflections already discussed is on the line printer. Logical unit NOUT references the printer and is set to 6 in subroutine PRIME.

3. Temporary File

During execution, STATUS refers to a random-access word addressable file which is written by subroutine DRUMRT and read by DRUMRD. Presently, this is a binary file and is written and read in blocks of four words: the packed index word, JKL; the reflection flag, NT; Fo, and σ. It is addressed relative to the beginning of the file. If Fo and Fc are being considered, the file structure has JKL, NT, Fo, σ followed immediately by JKL, NT, Fc, absorption path length or extinction coefficient. If two sets of observed data are being treated the corresponding values may not follow one another but both their addresses on the file are stored in one LOGFC word.

4. File Assignments

A description of actual file assignments to the physical devices is not given here since this depends on the particular computing system in use. STATUS allows the user to specify the actual values of the logical units either as control parameters on input card 3 or by simple modification of subroutine PRIME. This should provide sufficient flexibility regardless of the particular form of file assignment.

II.C. Examples of Output Plots

Several plots from an actual production run have been selected to provide examples of the types of plots possible and their uses. The purpose of the production run was to compare agreement among intensities equivalent by symmetry. The unit cell of a tin phosphate appeared to be monoclinic, but the intensities of some of the equivalent reflections were not consistent with monoclinic symmetry even after an absorption correction was applied (with some difficulty because the crystal was an irregularly shaped fragment). The statistics,

$$\delta m_{hk\ell} = (F_{hk\ell} - F_{h\bar{k}\ell}) / (\sigma^2_{hk\ell} + \sigma^2_{h\bar{k}\ell})^{\frac{1}{2}}$$

and $\delta m_{hk\bar{\ell}} = (F_{hk\bar{\ell}} - F_{\bar{h}k\ell}) / (\sigma^2_{\bar{h}k\ell} + \sigma^2_{hk\bar{\ell}})^{\frac{1}{2}}$ were formed by reading in $hk\ell$ and $h\bar{k}\ell$ reflections as two separate files for the first case and $hk\bar{\ell}$ and $hk\ell$ reflections for the second case. The results were analyzed in various ways in hopes of detecting systematic effects. A re-establishment of the scale factor by minimizing $\Sigma\delta m^2_{hk\ell}$ gave a value of 1.019. This is satisfactorily close to the expected value of 1.000 because our scaling of the data through remeasurement of standard reflections has a precision of 1% on intensities.

1. Normal Probability Plot

The first line of the plot title (Fig. 1) indicates that all statistics regardless of reflection class or octant were plotted against the expected normal quantiles. The designation DM(VERT) reminds us that the ordinate axis refers to the statistics. The line printer does not allow resolution of every point if there are more than 100. Plotting 50 or less eliminates this problem and allows for easier reading. Thus, the second line states that every 106th point of the total 5275 in the data set was plotted. The next line gives the equation of the least-squares line and quadratic that best relate the statistics to the normal quantiles. The third line gives the cubic equation that relates the statistics to the normal quantiles. Finally, the fourth line gives the goodness-of-fit,

$$S = (\Sigma(\delta_i - \delta_{calc})^2 / N\text{-}P)^{\frac{1}{2}},$$ for the various curves. These quantities are intended to aid in judging the variation. The definitions of the plot symbols are given below the plot. An accompanying table shows the values corresponding to the 50 points that were plotted.

The cubic function curves downward more rapidly than the line for negative statistics and upward more rapidly for positive statistics. This is also revealed directly by the lower value of the goodness-of-fit parameter for the cubic function when compared with those for the quadratic function line. Thus, the distribution of statistics has "thicker" tails than the normal, (i.e., there are more extreme values than expected) and the distribution tends to be symmetrical about its mean.

ALL REFLECTIONS INCLUDED PLOT OF DM(VERT) AGAINST NORMAL QUANTILE

EVERY106 TH POINT IN PLOTS5275 POINTS IN THIS SET OF REFLECTIONS 50 POINTS IN THIS PLOT

LEAST SQUARES LINE IS DM= .093 + 2.190*X -- LEAST SQUARES QUADRATIC IS DM= .239 + 2.190 *X+ -.188 *X**2

LEAST SQUARES CUBIC IS DM= .239 + 1.654 *X + -.188* X**2 + .293*X**3

GOODNESS OF FIT IS .353 FOR LS LINE, .309 FOR LS QUADRATIC AND .146 FOR CUBIC FIT IF MORE THAN 4 PTS IN PLOT

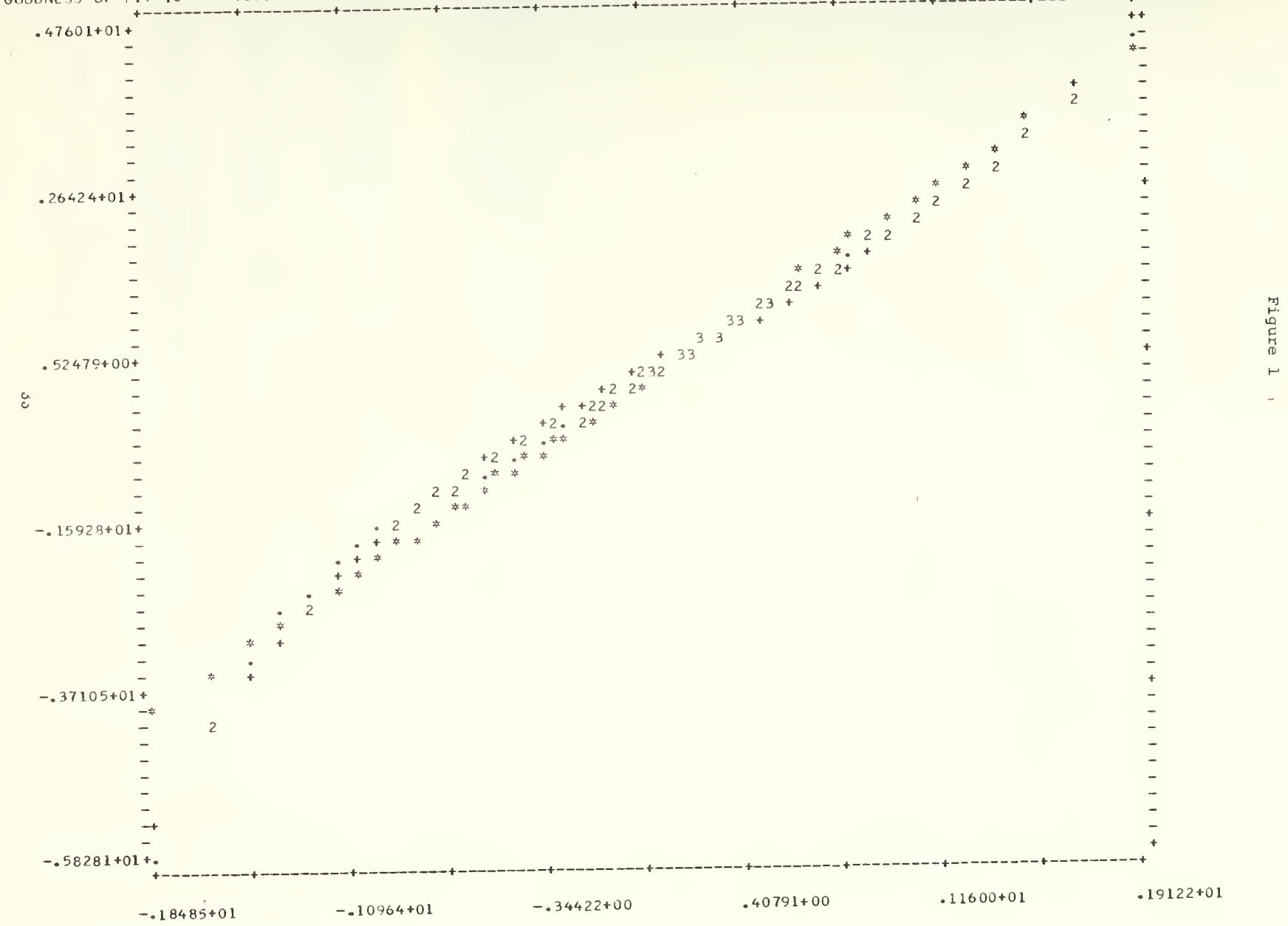


Figure 1

A = DM(I) (ORDINATE), SYMBOL IS . B = DM(I)- L. S. LINE, SYMBOL IS * C = DM(I) - L. S. CURVE, SYMBOL IS +
D = ABSCISSA VALUES (DSTAR, LOG F**2/SIN(2*THETA), OR QUANTILE

POINT	A	B	C	D
1	-5.828	-3.954	-5.309	-.185+01
2	-4.111	-3.478	-4.229	-.163+01
3	-3.213	-3.128	-3.533	-.147+01
4	-2.721	-2.845	-3.027	-.134+01
5	-2.400	-2.604	-2.631	-.123+01
6	-2.111	-2.392	-2.309	-.114+01
7	-1.885	-2.201	-2.037	-.105+01
8	-1.691	-2.026	-1.803	-.968+00
9	-1.520	-1.863	-1.598	-.894+00
10	-1.385	-1.711	-1.416	-.824+00
11	-1.247	-1.567	-1.251	-.758+00
12	-1.139	-1.430	-1.101	-.696+00
13	-1.032	-1.298	-.963	-.636+00
14	-.923	-1.171	-.836	-.578+00
15	-.816	-1.049	-.717	-.522+00
16	-.709	-.930	-.605	-.467+00
17	-.617	-.814	-.500	-.414+00
18	-.519	-.700	-.399	-.362+00
19	-.420	-.589	-.304	-.312+00
20	-.326	-.479	-.212	-.262+00
21	-.242	-.371	-.123	-.212+00
22	-.156	-.264	-.037	-.163+00
23	-.081	-.158	.046	-.115+00
24	.010	-.052	.128	-.665-01
25	.101	.053	.208	-.185-01
26	.214	.158	.287	.294-01
27	.308	.263	.366	.774-01
28	.400	.369	.444	.126+00
29	.495	.475	.523	.174+00
30	.578	.582	.602	.223+00
31	.673	.691	.682	.273+00
32	.770	.801	.764	.323+00
33	.863	.913	.847	.374+00
34	.961	1.027	.932	.426+00
35	1.065	1.143	1.021	.480+00
36	1.155	1.263	1.114	.534+00
37	1.269	1.387	1.211	.591+00
38	1.401	1.514	1.313	.649+00
39	1.532	1.647	1.422	.710+00
40	1.659	1.786	1.540	.773+00
41	1.805	1.932	1.668	.839+00
42	1.929	2.086	1.809	.910+00
43	2.063	2.251	1.966	.985+00
44	2.219	2.430	2.145	.107+01
45	2.425	2.625	2.352	.116+01
46	2.651	2.843	2.599	.126+01
47	2.909	3.092	2.903	.137+01
48	3.270	3.387	3.297	.150+01
49	3.714	3.758	3.852	.167+01
50	4.564	4.281	4.760	.191+01

Figure 1
contd.

The non-linear character of the plot indicates the presence of non-random effects. This inference is confirmed by plots of the deviations against various independent variables such as d*. The coefficients of the linear equation describe the mean and standard deviation of a normal distribution that is closest (in the least-squares sense) to the actual distribution.

The slope indicates that the average $\Delta F/\sigma$ is 2.2 rather than 1. This may result from ΔF being greater than ΔR, the deviation due to random errors, or from σ being underestimated, or from a combination of these effects. Further progress can be made if one has independent or prior knowledge about the correctness of the assigned σ. Our experience with our normal data collection procedure suggests that the assigned σ based on counting statistics tends to represent certain errors satisfactorily. These errors we associate with short-term (time required to record a data point) instrument and environmental fluctuations and are taken as random. Since $\Delta F/\sigma = \Delta R/\sigma + \Delta S/\sigma = 2.2$ and $\Delta R/\sigma \sim 1$, $\Delta S/\sigma \sim 1.2$ (ΔS is the deviation resulting from systematic effects in the two F values). $\Delta S = \langle \underset{\sim}{S}(1) - \underset{\sim}{S}(2) \rangle$, where $\underset{\sim}{S}(1)$ and $\underset{\sim}{S}(2)$ are vectors since they have both a magnitude and direction. Thus $0 \leq \Delta S/\sigma \leq |\underset{\sim}{S}(1)|/\sigma + \underset{\sim}{S}(2)/\sigma$, and $|\underset{\sim}{S}(1)|/\sigma + |\underset{\sim}{S}(2)|/\sigma \geq 1.2$. The systematic effects, $|\underset{\sim}{S}(1)|$ and $|\underset{\sim}{S}(2)|$ are at least 0.6σ for the two equivalent sets of reflections and may be much larger if significant cancellation of systematic effects has occurred. However, we note that the slope for the cubic near a normal quantile of zero is 1.65 which shows that the actual distribution envelopes the normal distribution and suggests that systematic effects have not completely cancelled each other out. A slope of less than one would have indicated significant cancellation of systematic effects.

2. d* Plot

The second plot (Fig. 2) shows the variation of statistics from reflections belonging to the (1kℓ) layer with the independent variable d*. The "+++OCTANT(S)" indicates that all indices are positive which in this special case (see section II.C., Introduction) means statistics $\delta hk\ell$ formed from $F_{1k\ell}$ and $F_{1\bar{k}\ell}$. The plot heading has already been explained. The plot shows that the deviations vary with d* and in particular, that negative deviates occur below $0.50 Å^{-1}$ (Bragg angles <15 degrees) indicating that $F_{h\bar{k}\ell}$ is larger than the equivalent $F_{hk\ell}$. At higher angles the trend of

the deviates is toward zero as it should be. The plot symbols U and L represent the positive and negative deviations of the statistics from the values given by the line as averaged over the 14 statistics that each interval represents. Thus U and L show the scatter of the statistics about their linear trend. We see that the scatter is not uniform as it should be, but that the largest $\Delta F/\sigma$ values occur for d* values below 0.50Å^{-1} (the lower third of the range). This implies that the σ values for the observed F values in this region are underestimated and that systematic effects are significant because σ_T is proportional to $F_T + S$ where S is the systematic effect and F_T, σ_T are the "true" values of F and σ. There will be cases for $F_T + S < F_T$ and thus $\sigma < \sigma_T$ especially when $S > R$.

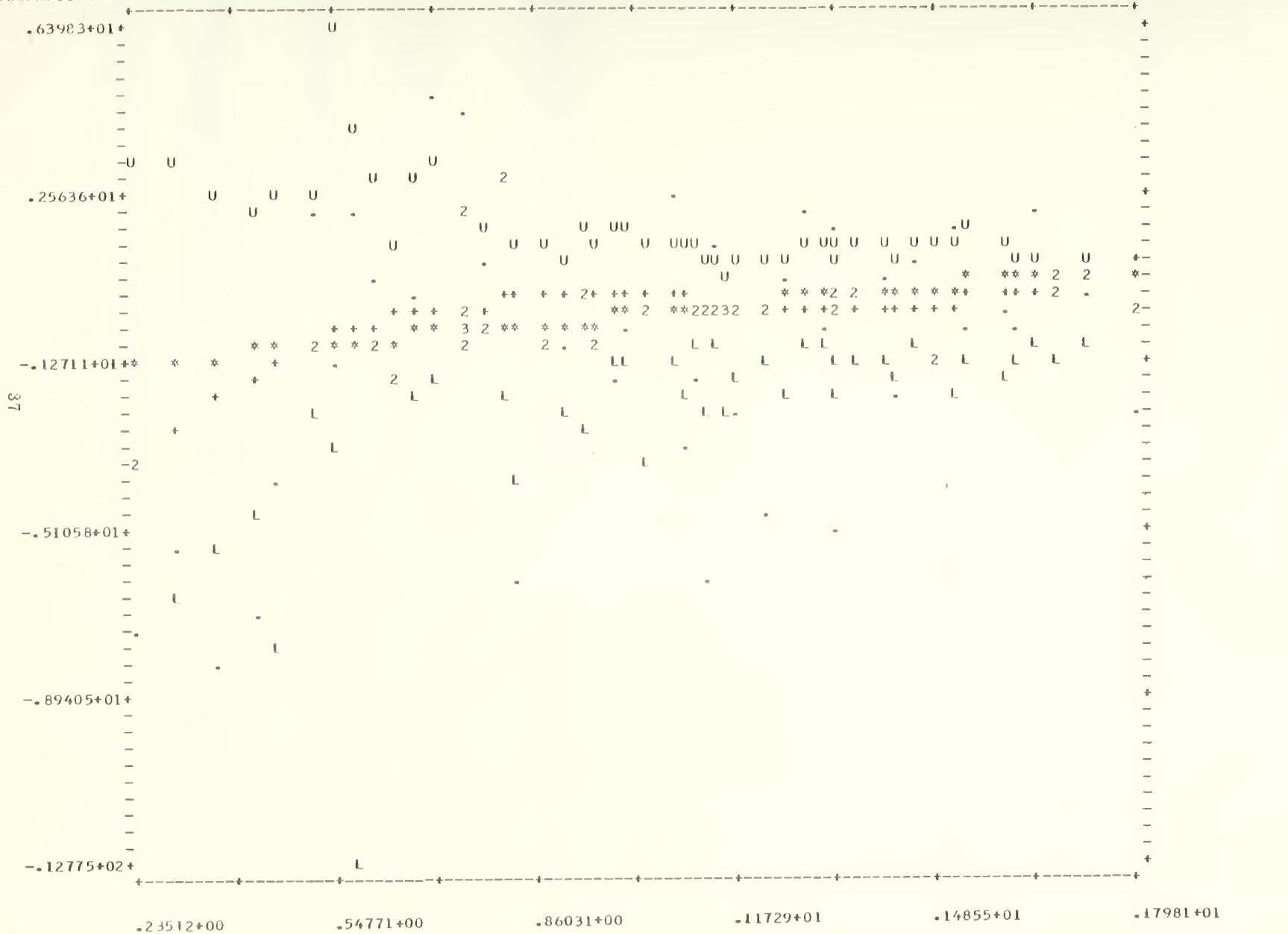

+++ OCTANT(S) 1 K L REFLECTIONS PLOT OF DM(VERT) AGAINST DSTAR
EVERY 14 TH POINT IN PLOTS 695 POINTS IN THIS SET OF REFLECTIONS 50 POINTS IN THIS PLOT
LEAST SQUARES LINE IS DM= -1.585 + 1.331*X -- LEAST SQUARES QUADRATIC IS DM= -3.553 + 5.893 *X+ -2.237 *X**2
LEAST SQUARES CUBIC IS DM= -7.469 + 21.305 *X + -19.364* X**2 + 5.663*X**3
GOODNESS OF FIT IS 3.031 FOR LS LINE, 3.011 FOR LS QUADRATIC AND 2.990 FOR CUBIC FIT IF MORE THAN 4 PTS IN PLOT
.63983+01
.25636+01
-.12711+01
-.51058+01
-.89405+01
-.12775+02
.23512+00
.54771+00
.86031+00
.11729+01
.14855+01
.17981+01

Figure 2

A = DM(I) (ORDINATE), SYMBOL IS . B = DM(I)- L. S. LINE, SYMBOL IS * C = DM(I) - L. S. CUBIC, SYMBOL IS +
D = ABSCISSA VALUES (DSTAR, LOG F**2/SIN(2*THETA), OR QUANTILE
SYMBOLS L, U ARE FOR AVERAGE - AND + DEVIATIONS (A-B VALUES FOR 14 POINTS IN THE INTERVAL.)

POINT	A	B	C	D
1	-7.405	-1.272	-3.457	.235+00
2	-5.496	-1.191	-2.710	.296+00
3	-8.218	-1.096	-1.975	.367+00
4	-6.932	-1.024	-1.503	.422+00
5	-4.065	-.983	-1.267	.453+00
6	2.354	-.900	-.861	.515+00
7	-1.135	-.859	-.691	.545+00
8	2.074	-.808	-.506	.584+00
9	.730	-.767	-.376	.615+00
10	-1.635	-.738	-.294	.636+00
11	.202	-.682	-.161	.678+00
12	4.989	-.644	-.084	.707+00
13	-.355	-.584	.010	.752+00
14	4.332	-.575	.021	.759+00
15	1.127	-.550	.052	.778+00
16	2.919	-.508	.091	.809+00
17	-6.341	-.473	.116	.835+00
18	-1.017	-.419	.141	.876+00
19	-.912	-.374	.150	.910+00
20	.215	-.336	.150	.938+00
21	-.948	-.324	.149	.948+00
22	-1.656	-.269	.138	.989+00
23	-.390	-.262	.136	.994+00
24	-.283	-.203	.114	.104+01
25	2.732	-.159	.093	.107+01
26	-3.085	-.124	.074	.110+01
27	-1.469	-.112	.067	.111+01
28	-6.113	-.093	.056	.112+01
29	1.302	-.074	.046	.113+01
30	-.276	-.046	.029	.116+01
31	-2.522	-.018	.013	.118+01
32	-4.596	.038	-.016	.122+01
33	.645	.088	-.039	.126+01
34	2.271	.124	-.052	.128+01
35	-.498	.153	-.060	.131+01
36	-5.118	.186	-.065	.133+01
37	1.925	.190	-.066	.133+01
38	.102	.234	-.066	.137+01
39	.665	.279	-.058	.140+01
40	-2.158	.311	-.046	.142+01
41	1.019	.357	-.019	.146+01
42	-1.146	.396	.014	.149+01
43	1.611	.433	.055	.152+01
44	-.542	.460	.090	.154+01
45	-.131	.536	.222	.159+01
46	-.317	.568	.291	.162+01
47	2.076	.595	.359	.164+01
48	.415	.643	.493	.167+01
49	.230	.706	.707	.172+01
50	-2.357	.808	1.151	.180+01

Figure 2
contd.

Figure 3 is a plot of statistics from the h = 2 layer against d*. The "++- OCTANT(S)" and the reflection class indicates the statistics are those of the form $\delta m_{2k\ell}$. Again the statistics show a variation with d*, and the linear slope of 3.9 indicates a more severe trend than that for the $1k\ell$ statistics. Absorption could well be the cause since it is known to have a larger effect on those reflections occurring at larger d* values (higher 2θ angles). Non-uniform scatter of the statistics is also indicated.

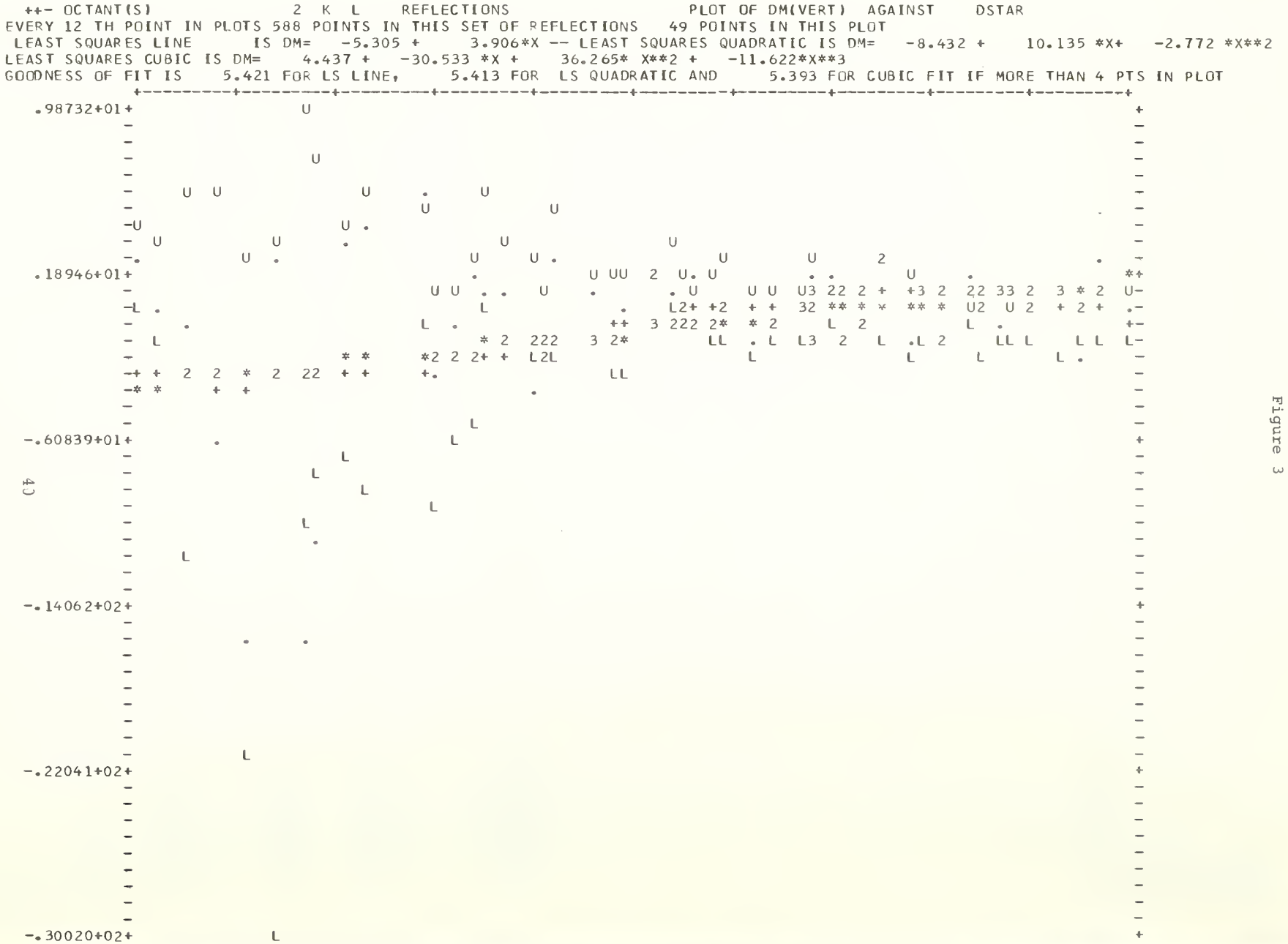

++- OCTANT(S) 2 K L REFLECTIONS PLOT OF DM(VERT) AGAINST DSTAR
EVERY 12 TH POINT IN PLOTS 588 POINTS IN THIS SET OF REFLECTIONS 49 POINTS IN THIS PLOT
LEAST SQUARES LINE IS DM= -5.305 + 3.906*X -- LEAST SQUARES QUADRATIC IS DM= -8.432 + 10.135 *X+ -2.772 *X**2
LEAST SQUARES CUBIC IS DM= 4.437 + -30.533 *X + 36.265* X**2 + -11.622*X**3
GOODNESS OF FIT IS 5.421 FOR LS LINE, 5.413 FOR LS QUADRATIC AND 5.393 FOR CUBIC FIT IF MORE THAN 4 PTS IN PLOT
.98732+01
.18946+01
-.60839+01
-.14062+02
-.22041+02
-.30020+02

Figure 3

A = DM(I) (ORDINATE), SYMBOL IS . B = DM(I)- L. S. LINE, SYMBOL IS * C = DM(I) - L. S. CUBIC, SYMBOL IS +
D = ABSCISSA VALUES (DSTAR, LOG F**2/SIN(2*THETA), OR QUANTILE
SYMBOLS L, U ARE FOR AVERAGE - AND + DEVIATIONS (A-B VALUES FOR 12 POINTS IN THE INTERVAL.)

POINT	A	B	C	D
1	2.315	-3.484	-3.093	.466+00
2	.528	-3.399	-3.177	.488+00
3	-.685	-3.243	-3.285	.528+00
4	-5.755	-3.086	-3.336	.568+00
5	-15.367	-2.909	-3.329	.613+00
6	2.992	-2.773	-3.282	.648+00
7	-15.999	-2.632	-3.199	.684+00
8	-10.558	-2.548	-3.134	.706+00
9	3.820	-2.438	-3.032	.734+00
10	4.530	-2.332	-2.918	.761+00
11	5.789	-2.025	-2.512	.840+00
12	-2.518	-1.978	-2.441	.852+00
13	-.103	-1.852	-2.244	.884+00
14	2.153	-1.753	-2.078	.909+00
15	1.095	-1.680	-1.954	.928+00
16	1.277	-1.583	-1.784	.953+00
17	-3.606	-1.458	-1.559	.985+00
18	-1.732	-1.382	-1.421	.100+01
19	2.759	-1.329	-1.325	.102+01
20	1.439	-1.169	-1.030	.106+01
21	-1.225	-1.042	-.798	.109+01
22	.175	-.984	-.694	.111+01
23	2.193	-.864	-.480	.114+01
24	.881	-.758	-.296	.116+01
25	.553	-.694	-.188	.118+01
26	2.055	-.626	-.077	.120+01
27	-.802	-.547	.049	.122+01
28	.540	-.489	.138	.123+01
29	-1.343	-.355	.332	.127+01
30	-.593	-.220	.508	.130+01
31	.088	-.097	.650	.133+01
32	1.507	-.046	.703	.135+01
33	-1.051	-.008	.740	.136+01
34	2.094	.087	.824	.138+01
35	-1.361	.111	.842	.139+01
36	-.288	.224	.919	.142+01
37	2.814	.331	.971	.144+01
38	-1.044	.459	1.006	.148+01
39	1.433	.509	1.011	.149+01
40	-1.219	.655	.995	.153+01
41	2.045	.778	.945	.156+01
42	-.083	.827	.915	.157+01
43	-.620	.956	.808	.160+01
44	1.299	.985	.779	.161+01
45	1.079	1.111	.624	.164+01
46	1.390	1.268	.370	.168+01
47	-1.770	1.332	.245	.170+01
48	2.609	1.455	-.030	.173+01
49	.331	1.598	-.409	.177+01

Figure 3
contd.

3. Log($Fo^2/\sin 2\theta$) Plot

Figure 4 shows a plot of all statistics against log ($Fo^2/\sin 2\theta$). In this particular case we are testing two equivalent sets of Fo values for anisotropy in extinction. This type of plot could also show absolute effects of extinction (i.e., between Fo and Fc) if least-squares residuals were used, because in this case the abscissa values would be log($Fc^2/\sin 2\theta$). The plot shows that negative statistics tend to occur at large abscissa values. The plotted U and L symbols show that the largest $|\Delta F|/\sigma(Fo)$ values tend to occur at large values of $Fo^2/\sin 2\theta$. However, $\sin 2\theta = \cos\theta \sin\theta$ and thus $1/\sin 2\theta = 2/(\lambda d^* \cos\theta)$. Thus the variations of the statistics with d* and $Fo^2/\sin 2\theta$ are not independent. We need the $\partial \log[2/(\lambda d^* \cos\theta)] / \partial d^*$ in order to relate the variations as $\partial \delta m_{hk\ell} \partial \log[(2/\lambda d^* \cos\theta)] = \partial \delta m_{hk\ell}/\partial d^* \cdot \partial d^*/\partial \log[2/(\lambda d^* \cos\theta)]$. Then $\partial \log(2/\lambda d^* \cos\theta)/\partial d^* = -1/2.3\ d^*$ and its inverse is $-2.3\ d^*$, showing that the inverse relation between the two variations varies with d*. Thus the slopes of plots against d* should be of different sign from slopes of plots against log($Fo^2/\sin 2\theta$). Here the d* plot shows a positive slope (negative statistics tend to occur at small d* values) while the log($Fo^2/\sin 2\theta$) plots shows a negative trend as expected. However, if the statistics vary in a systematic manner with Fo^2 we would expect a more pronounced trend in the log($Fo^2/\sin 2\theta$) plot because the range of Fo^2 values is 50-100 times greater than the range of $1/\sin 2\theta$. The ordering in $Fo^2/\sin 2\theta$ plots is primarily due to Fo^2. A comparison of the d* plot (Fig. 2) with this one shows that the cubic curve reaches an extreme of -4 for low d* values while the corresponding value here is -2.6. This indicates that the disagreement between intense equivalent reflections occurring at low d* values is not worse than the disagreement between equivalent reflections occurring at low d* regardless of their intensity.

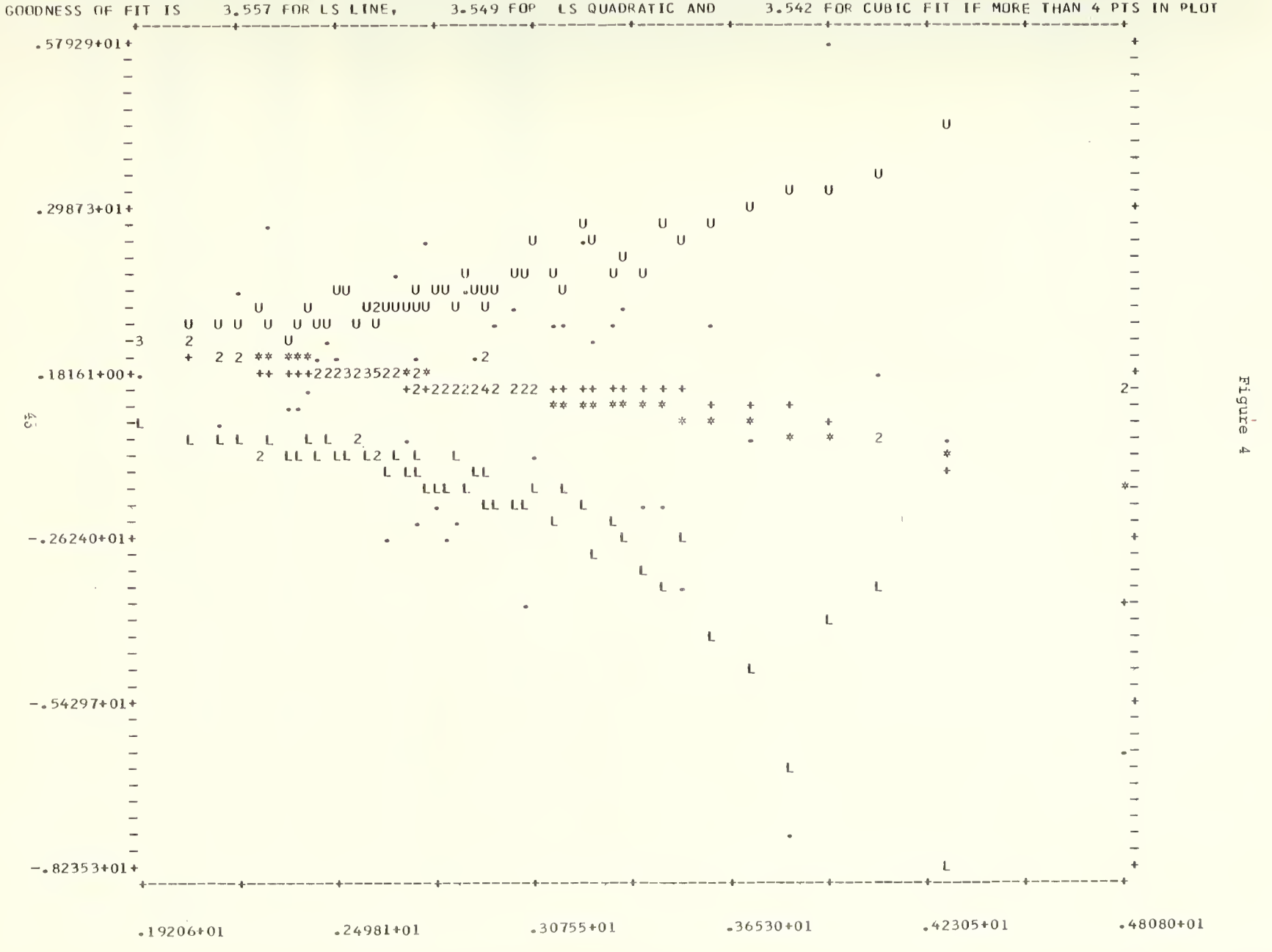

ALL REFLECTIONS INCLUDED
PLOT OF DM(VERT) AGAINST LOGF**2/SIN(2T)
EVERY111 TH POINT IN PLOTS5527 POINTS IN THIS SET OF REFLECTIONS 50 POINTS IN THIS PLOT
LEAST SQUARES LINE IS DM= 2.350 + -.831*X -- LEAST SQUARES QUADRATIC IS DM= -2.559 + 2.387 *X+ -.507 *X**2
LEAST SQUARES CUBIC IS DM= 15.222 + -15.048 *X + 5.004* X**2 + -.561*X**3
GOODNESS OF FIT IS 3.557 FOR LS LINE, 3.549 FOR LS QUADRATIC AND 3.542 FOR CUBIC FIT IF MORE THAN 4 PTS IN PLOT
.57929+01
.29873+01
.18161+00
-.26240+01
-.54297+01
-.82353+01
.19206+01
.24981+01
.30755+01
.36530+01
.42305+01
.48080+01

Figure 4

A = DM(I) (ORDINATE), SYMBOL IS . B = DM(I)- L. S. LINE, SYMBOL IS * C = DM(I) - L. S. CUBIC, SYMBOL IS +
D = ABSCISSA VALUES (DSTAR, LOG F**2/SIN(2*THETA), OR QUANTILE
SYMBOLS L, U ARE FOR AVERAGE - AND + DEVIATIONS (A-B VALUES FOR 111 POINTS IN THE INTERVAL.)

POINT	A	B	C	D
1	.175	.755	.807	.192+01
2	.758	.637	.553	.206+01
3	-.660	.568	.433	.215+01
4	1.620	.518	.358	.221+01
5	-1.219	.475	.302	.226+01
6	2.809	.433	.253	.231+01
7	-.460	.406	.224	.234+01
8	-.367	.373	.193	.238+01
9	-.223	.343	.166	.242+01
10	.367	.318	.147	.245+01
11	.783	.290	.126	.248+01
12	.456	.270	.113	.250+01
13	.247	.245	.098	.253+01
14	-.952	.230	.088	.255+01
15	.119	.212	.079	.257+01
16	.286	.187	.066	.260+01
17	1.275	.172	.060	.262+01
18	-2.618	.154	.052	.264+01
19	1.991	.131	.042	.267+01
20	-.988	.110	.035	.270+01
21	.521	.091	.028	.272+01
22	-2.357	.071	.022	.274+01
23	2.499	.051	.016	.277+01
24	-2.044	.031	.010	.279+01
25	-2.714	.011	.005	.282+01
26	-2.280	-.006	.001	.284+01
27	1.693	-.028	-.004	.286+01
28	.393	-.053	-.009	.289+01
29	.503	-.075	-.014	.292+01
30	.560	-.094	-.018	.294+01
31	1.102	-.121	-.023	.297+01
32	1.292	-.155	-.030	.302+01
33	-3.862	-.185	-.036	.305+01
34	-1.122	-.216	-.043	.309+01
35	1.065	-.243	-.049	.312+01
36	.928	-.276	-.057	.316+01
37	2.549	-.319	-.070	.321+01
38	.627	-.354	-.082	.325+01
39	.934	-.389	-.095	.330+01
40	1.336	-.432	-.115	.335+01
41	-2.189	-.464	-.131	.339+01
42	-2.160	-.514	-.161	.345+01
43	-3.548	-.569	-.200	.351+01
44	.980	-.646	-.269	.361+01
45	-.893	-.731	-.367	.371+01
46	-7.672	-.817	-.492	.381+01
47	5.793	-.923	-.687	.394+01
48	.046	-1.052	-.999	.410+01
49	-.953	-1.204	-1.482	.428+01
50	-6.268	-1.644	-3.784	.481+01

Figure 4
contd.

Figure 5 shows the variation of statistics from the $h2\ell$ layer with $\log(Fo^2/\sin 2\theta)$. The statistics, $\delta m_{h2\ell}$, are from the $F_{h2\ell}$ and $F_{\bar{h}2\ell}$ layers so we are comparing the two octants of the $+h$, $+\ell$ quadrant. The statistics tend towards negative values for large $Fo^2/\sin 2\theta$ values with the cubic curve showing that the trend is most evident for the 63 largest statistics (7 plot points x 9 reflections per point = 63). This is about 15% of the data. The overall trend is very similar to that observed in the overall plot except for the very large and very small $Fo^2/\sin 2\theta$ values where it is more severe.

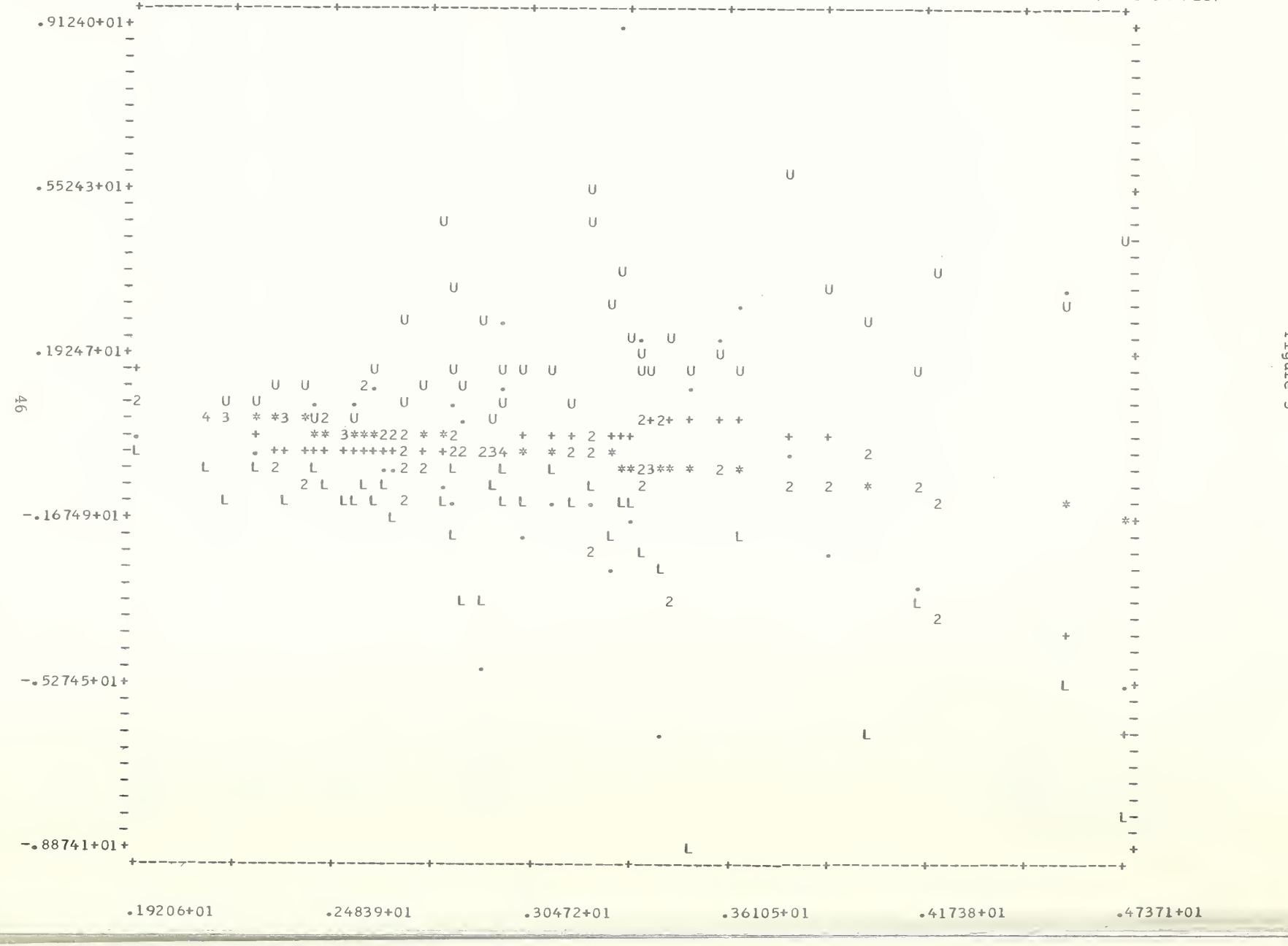

+++ OCTANT(S) H 2 L REFLECTIONS PLOT OF DM(VERT) AGAINST LOGF**2/SIN(2T)
EVERY 9 TH POINT IN PLOTS 438 POINTS IN THIS SET OF REFLECTIONS 49 POINTS IN THIS PLOT
LEAST SQUARES LINE IS DM= 2.309 + -.829*X -- LEAST SQUARES QUADRATIC IS DM= -9.459 + 6.754 *X+ -1.174 *X**2
LEAST SQUARES CUBIC IS DM= 49.253 + -50.463 *X + 16.815* X**2 + -1.825*X**3
GOODNESS OF FIT IS 3.006 FOR LS LINE, 2.955 FOR LS QUADRATIC AND 2.881 FOR CUBIC FIT IF MORE THAN 4 PTS IN PLOT
.91240+01
.55243+01
.19247+01
-.16749+01
-.52745+01
-.88741+01
.19206+01
.24839+01
.30472+01
.36105+01
.41738+01
.47371+01

Figure 5

A = DM(I) (ORDINATE), SYMBOL IS . B = DM(I)- L. S. LINE, SYMBOL IS * C = DM(I) - L. S. CUBIC, SYMBOL IS +
D = ABSCISSA VALUES (DSTAR, LOG F**2/SIN(2*THETA), OR QUANTILE
 SYMBOLS L, U ARE FOR AVERAGE - AND + DEVIATIONS (A-B VALUES FOR 9 POINTS IN THE INTERVAL.)

POINT	A	B	C	D
1	.175	.716	1.432	.192+01
2	.420	.554	.473	.212+01
3	.646	.516	.310	.216+01
4	-.333	.444	.056	.225+01
5	-.617	.393	-.082	.231+01
6	.590	.368	-.138	.234+01
7	-.865	.318	-.229	.240+01
8	.813	.288	-.272	.244+01
9	.571	.267	-.295	.246+01
10	.085	.229	-.327	.251+01
11	.894	.209	-.339	.253+01
12	1.083	.167	-.352	.258+01
13	1.276	.153	-.353	.260+01
14	-.766	.140	-.353	.262+01
15	-.770	.119	-.351	.264+01
16	-.458	.092	-.343	.267+01
17	-.677	.085	-.340	.268+01
18	-.607	.037	-.313	.274+01
19	-1.020	-.004	-.280	.279+01
20	-1.385	-.025	-.260	.281+01
21	.804	-.037	-.248	.283+01
22	.486	-.066	-.216	.286+01
23	-4.748	-.093	-.184	.290+01
24	-.083	-.126	-.142	.294+01
25	1.303	-.144	-.119	.296+01
26	2.603	-.155	-.104	.297+01
27	-1.873	-.195	-.048	.302+01
28	-1.135	-.268	.055	.311+01
29	-.299	-.319	.127	.317+01
30	-1.304	-.352	.171	.321+01
31	-2.522	-.368	.192	.323+01
32	-2.651	-.400	.231	.327+01
33	9.124	-.419	.253	.329+01
34	-1.561	-.450	.286	.333+01
35	2.321	-.471	.306	.335+01
36	-1.117	-.481	.315	.336+01
37	-.588	-.491	.324	.338+01
38	-6.338	-.520	.345	.341+01
39	-3.603	-.538	.356	.343+01
40	1.361	-.599	.380	.351+01
41	2.322	-.669	.374	.359+01
42	2.956	-.701	.359	.363+01
43	-.227	-.825	.207	.378+01
44	-2.519	-.915	-.009	.389+01
45	-.234	-1.014	-.369	.401+01
46	-3.193	-1.120	-.915	.413+01
47	-3.932	-1.174	-1.263	.420+01
48	3.279	-1.473	-4.230	.456+01
49	-5.452	-1.620	-6.444	.474+01

Figure 5
contd.

4. Half-Normal Probability Plot

Figure 6 is a plot of statistics, $\delta p_i = p_i(\text{set 1}) - p_i(\text{set 2}) \; / \; [\sigma^2 p_i(\text{set 1}) - \sigma^2 p_i(\text{set 2})]^{\frac{1}{2}}$ against half-normal quantiles. Half-normal quantiles are used because the order of subtraction of sets 1 and 2 is not meaningful. The p_i are fractional coordinates from structural refinements of data sets from two separate crystals of β-$Ca_3(PO_4)_2$. Each crystal contained about 5 wt% of Mg.

Thus, this plot tests the sensitivity of structural parameters to sample differences since the same experimental procedure was used in both cases. The plot shows that the actual δp_i (represented by ·) wiggle about the least-squares line (represented by *). This is mostly due to the paucity of data (43 points). Indeed the plot shows that the cubic curve fits the variation only slightly better. The δp_i represents mainly random differences and shows the samples are nearly identical structurally. The slope of 1.275 for the line suggests that errors in the parameters, $\sigma(p)_i$ are too small by 20%. These errors are estimates from the least-squares and are likely to be too small if the parameters in the model account for errors in the measured Fo.

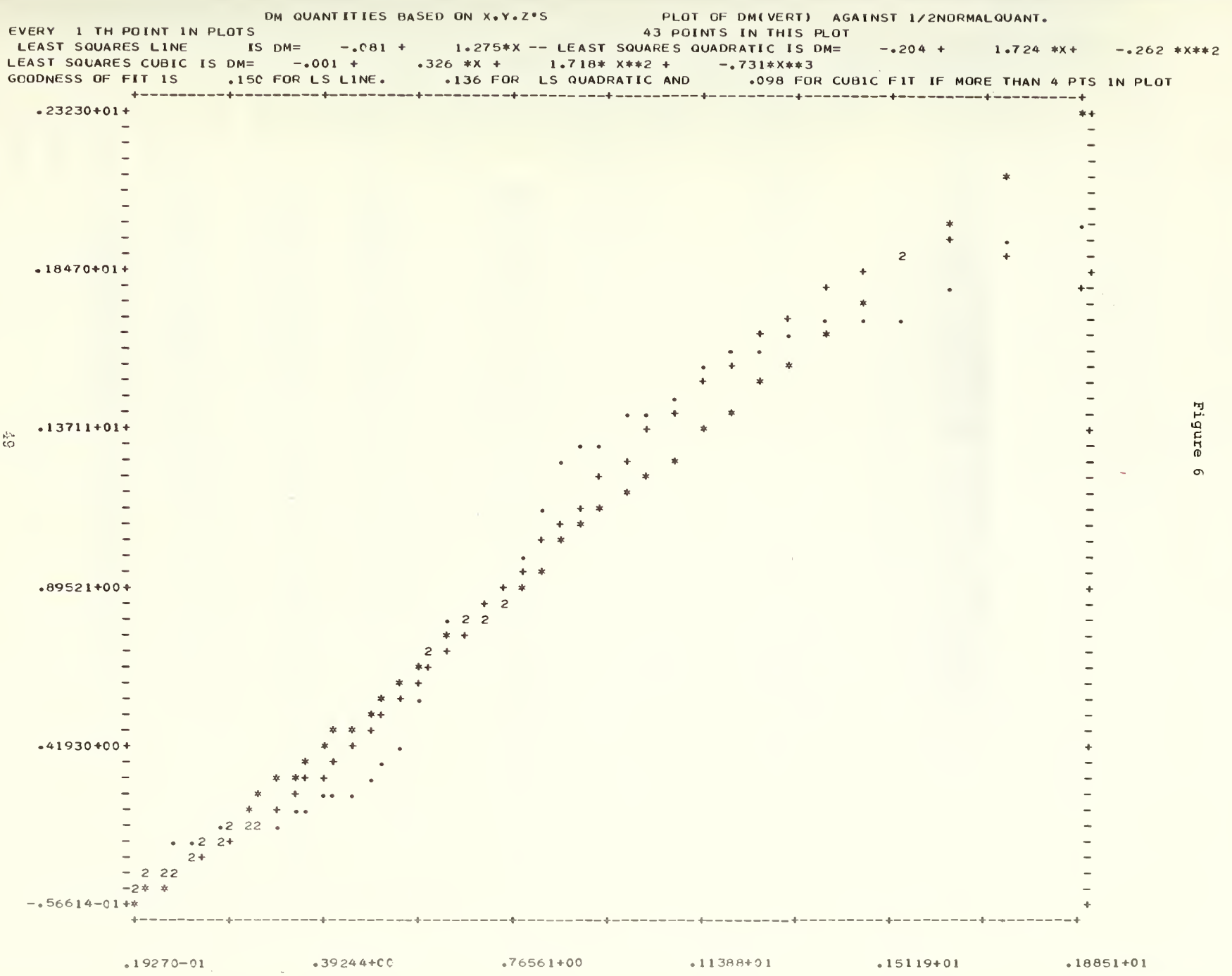

DM QUANTITIES BASED ON X,Y,Z'S
PLOT OF DM(VERT) AGAINST 1/2NORMALQUANT.
EVERY 1 TH POINT IN PLOTS
43 POINTS IN THIS PLOT
LEAST SQUARES LINE IS DM= -.081 + 1.275*X -- LEAST SQUARES QUADRATIC IS DM= -.204 + 1.724 *X+ -.262 *X**2
LEAST SQUARES CUBIC IS DM= -.001 + .326 *X + 1.718* X**2 + -.731*X**3
GOODNESS OF FIT IS .150 FOR LS LINE. .136 FOR LS QUADRATIC AND .098 FOR CUBIC FIT IF MORE THAN 4 PTS IN PLOT
.23230+01
.18470+01
.13711+01
.89521+00
.41930+00
-.56614-01
.19270-01
.39244+00
.76561+00
.11388+01
.15119+01
.18851+01

Figure 6

A = DM(I) (ORDINATE), SYMBOL IS . B = DM(I)- L. S. LINE, SYMBOL IS * C = DM(I) - L. S. CUBIC, SYMBOL IS +
D = ABSCISSA VALUES (DSTAR, LOG F**2/SIN(2*THETA), OR QUANTILE

POINT	A	B	C	D
1	.011	-.057	.006	.193-01
2	.029	-.021	.018	.468-01
3	.044	.014	.033	.744-01
4	.124	.049	.050	.102+00
5	.129	.084	.069	.130+00
6	.140	.120	.091	.158+00
7	.167	.156	.115	.186+00
8	.188	.191	.141	.214+00
9	.195	.228	.169	.242+00
10	.199	.264	.199	.271+00
11	.203	.301	.231	.299+00
12	.231	.338	.266	.328+00
13	.248	.375	.302	.358+00
14	.261	.413	.341	.387+00
15	.262	.451	.381	.417+00
16	.263	.490	.424	.448+00
17	.339	.529	.468	.478+00
18	.390	.569	.515	.509+00
19	.429	.609	.563	.541+00
20	.564	.650	.613	.573+00
21	.694	.692	.666	.606+00
22	.777	.735	.720	.640+00
23	.798	.779	.776	.674+00
24	.822	.823	.834	.709+00
25	.835	.869	.894	.745+00
26	1.012	.916	.956	.782+00
27	1.144	.965	1.019	.820+00
28	1.284	1.015	1.085	.860+00
29	1.310	1.067	1.152	.900+00
30	1.339	1.121	1.221	.943+00
31	1.415	1.177	1.291	.987+00
32	1.437	1.236	1.363	.103+01
33	1.487	1.297	1.436	.108+01
34	1.582	1.362	1.509	.113+01
35	1.589	1.431	1.583	.119+01
36	1.609	1.505	1.656	.124+01
37	1.664	1.584	1.727	.131+01
38	1.687	1.671	1.794	.137+01
39	1.720	1.766	1.854	.145+01
40	1.724	1.873	1.902	.153+01
41	1.802	1.996	1.929	.163+01
42	1.951	2.141	1.916	.174+01
43	2.000	2.323	1.822	.189+01

Figure 6
contd.

III. Detailed Description of Main Algorithms

A general description of the program STATUS and its subroutines has already been given. Because STATUS consists of a main control routine and 35 subroutines, it is not practical to describe the logic of each routine in great detail. Instead we have provided macro-flow charts for the control routines and those routines likely to be modified by a user, typically those dealing with input and output. A description of each routine includes its purpose, its procedure, a definition of the important variables and a macro-flow chart detailing the logical procedure. These aids together with the FORTRAN listing should enable a user to make any required modification.

III. A. Main Routine STATUS

Purpose:

Controls the execution of a job by reading from cards or their images the control parameters discussed in the section on Card Inputs.

Procedure:

Reads job title card (stops on EOF), a card with cell constants, then control card and finally a card with experimental parameters. Checks control parameters for consistency and stops if some are out of bounds and cannot be corrected. Calls subroutines to read in data, calculate statistics, etc. When job is finished attempts to read another job card.

Important Variables:

See section on Common Blocks and Arrays.

See section on Card Inputs for description of control parameters.

JC - total number of statistics, JA in set one, JB in set two, JC = JA + JB.

JZ - pointer for drum address of Fc or absorption path length.

M - packed word indicating class, layer and octant.

M - 1000x class type + 10x number of layers + octant option.

II - number of bit in word LOGHKL to be examined in cataloging of reflections into classes.

LA - number of atomic parameters in first set.

LB - number of atomic parameters in second set.

IP - number of parameters common to the two sets.

KIND - type of parameter; 1 - occupancy factor, 2 - coordinate, 3 - thermal.

1. Common Blocks and Arrays

a. Parameter Variables -- these set the array limits.

NN - Maximum number of δm or residuals (7200 at present).

MM - Maximum number of atoms, 100 at present (equivalent to ~1000 parameters in least-squares).

N - Maximum number of points in plots - restricted to 50 by PLOTS.

NNA - Maximum number of reflection records in storage.

NNB=NNA + 1 - Boundary for overlay by equivalence statement.

NNC=NN/2 + 1 - Boundary for overlay by equivalence statement.

NND=NN/2 + NNA + 1 - Boundary for overlay by equivalence statement.

b. Common Blocks

Block/D MAXH, MINH, MAXK, MINK, MAXL, MINL

Maximum and minimum values of the Miller indices.

Block/H/DUMMY (NN) - Used in equivalencing arrays.

Block/G/ASTAR, BSTAR, CSTAR, COSAST, COSBST, COSGST

Reciprocal cell values.

Block/IO/IN, NOUT, NDRUM, NTAPEA, NTAPEB.

IN - Logical unit for card reader.

NOUT - Logical unit for line printer.

NDRUM - Logical unit for mass storage.

NTAPEA - Logical unit for input data set 1.

NTAPEB - Logical unit for input data set 2.

Block/ORDER/JJA, KKA, LLA, NR, NFILER, NABS

JJA, KKA, and LLA give order in which Miller indices are to be packed in JKL word, i.e., 2, 1, 3 for JJA, KKA and LLA gives JKL = 1,000,000 (k+200) + 1000 (h+200) + (ℓ+200).

The indices are packed into the JKL word to save memory storage. Variable index order is an option which is helpful when reflections are being matched efficiently.

NR - Output unit from SCALE

NFILER - File number on unit NR

NABS - Flag for optional absorption correction.

Block/TITLE/ATITLE (14)

ATITLE - 14 Hollerith words set by STATUS and PLOTEM to provide headings for the various plots.

c. Arrays listed in groups requiring similar dimensions.

1. Arrays holding reflection information:

Dimension NN; IDA, IDB, NNA; FA, FB, SGA, SGB, JKLA, JKLB, KA, KB and LOGFC.

2. Arrays holding atom parameter information:

Dimension MM; NAMEA, NAMEB, OCA, OCB, XYZA, XYZB, SIGXA, SIGXB, BA, BB, SIGBA, SIGBB, SIGOA, SIGOB, MATCH, LOG.

3. Arrays holding statistic information:

Dimension NN; X, LOGHKL.

4. Arrays holding plotting information:

N; DMSMAL, XSMALL.

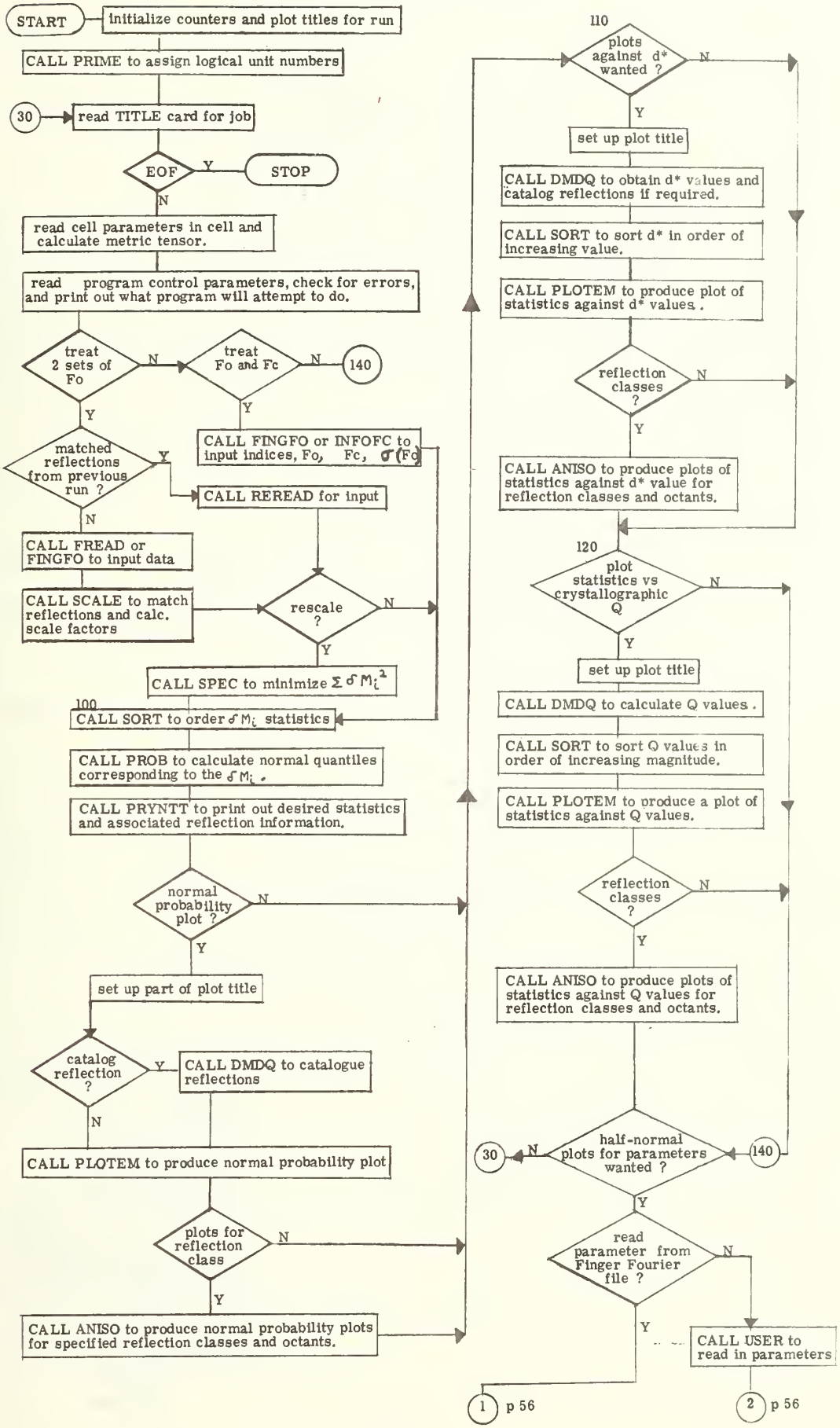
d. Equivalences to conserve storage:

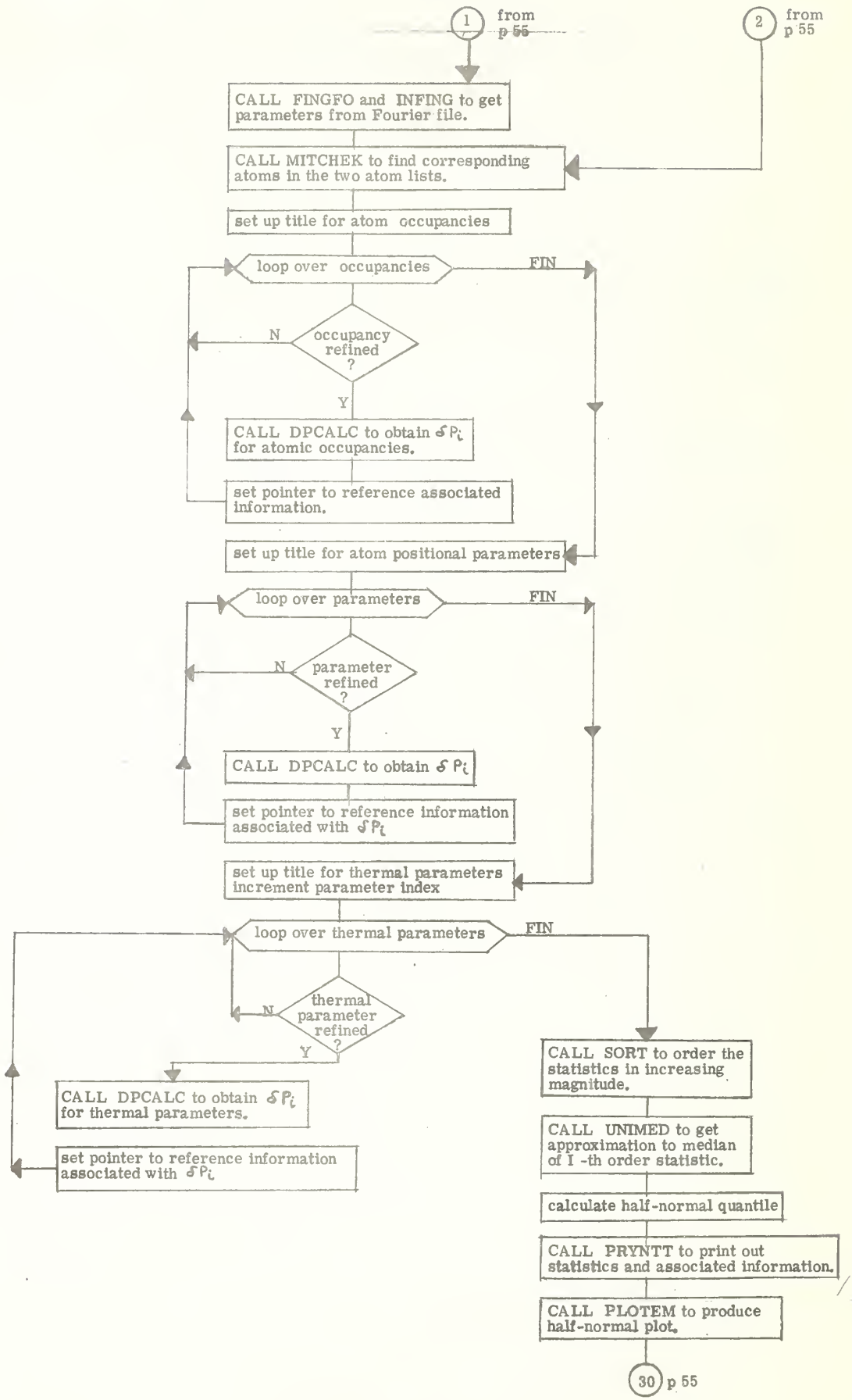
1. LOGFC, LOG: LOGFC is written over when parameters treated.

2. DM, IDA, X, IDB: DM and X are written over IDA and IDB.

3. LOGHKL and JKLA, JKLB, KA, KB.

Note: These equivalences work only if MM ≤ 100 i.e., there is a maximum of 100 atoms per asymmetric unit at present.


START
initialize counters and plot titles for run
CALL PRIME to assign logical unit numbers
30
read TITLE card for job
EOF
Y
STOP
N
read cell parameters in cell and calculate metric tensor.
read program control parameters, check for errors, and print out what program will attempt to do.
treat 2 sets of Fo
N
treat Fo and Fc
N
140
Y
Y
CALL FINGFO or INFOFC to input indices, Fo, Fc, σ(Fo)
matched reflections from previous run ?
Y
CALL REREAD for input
N
CALL FREAD or FINGFO to input data
CALL SCALE to match reflections and calc. scale factors
rescale ?
N
Y
CALL SPEC to minimize Σ σ Mi²
100
CALL SORT to order σ Mi statistics
CALL PROB to calculate normal quantiles corresponding to the σ Mi .
CALL PRYNTT to print out desired statistics and associated reflection information.
normal probability plot ?
N
Y
set up part of plot title
catalog reflection ?
Y
CALL DMDQ to catalogue reflections
N
CALL PLOTEM to produce normal probability plot
plots for reflection class
N
Y
CALL ANISO to produce normal probability plots for specified reflection classes and octants.
110
plots against d* wanted ?
N
Y
set up plot title
CALL DMDQ to obtain d* values and catalog reflections if required.
CALL SORT to sort d* in order of increasing value.
CALL PLOTEM to produce plot of statistics against d* values.
reflection classes ?
N
Y
CALL ANISO to produce plots of statistics against d* value for reflection classes and octants.
120
plot statistics vs crystallographic Q
N
Y
set up plot title
CALL DMDQ to calculate Q values.
CALL SORT to sort Q values in order of increasing magnitude.
CALL PLOTEM to produce a plot of statistics against Q values.
reflection classes ?
N
Y
CALL ANISO to produce plots of statistics against Q values for reflection classes and octants.
30
N
half-normal plots for parameters wanted ?
140
Y
read parameter from Finger Fourier file ?
N
Y
CALL USER to read in parameters
1
p 56
2
p 56


1
from p 55
2
from p 55
CALL FINGFO and INFING to get parameters from Fourier file.
CALL MITCHEK to find corresponding atoms in the two atom lists.
set up title for atom occupancies
loop over occupancies
FIN
N
occupancy refined ?
Y
CALL DPCALC to obtain δPi for atomic occupancies.
set pointer to reference associated information.
set up title for atom positional parameters
loop over parameters
FIN
N
parameter refined ?
Y
CALL DPCALC to obtain δPi
set pointer to reference information associated with δPi
set up title for thermal parameters increment parameter index
loop over thermal parameters
FIN
N
thermal parameter refined ?
Y
CALL DPCALC to obtain δPi for thermal parameters.
set pointer to reference information associated with δPi
CALL SORT to order the statistics in increasing magnitude.
CALL UNIMED to get approximation to median of I -th order statistic.
calculate half-normal quantile
CALL PRYNTT to print out statistics and associated information.
CALL PLOTEM to produce half-normal plot.
30
p 55

III. B. Subroutine ANISO

Purpose:

Controls the selection of statistics for plotting subsets of the complete data set.

Procedure:

Each statistic has associated with it the LOGHKL word that specifies all classes, layers and octant that the statistic belongs to. These may be plotted separately so comparisons can be made. Subroutine ANISO "tells" subroutine PLOTEM what to plot in the following way. If the variable MCLT in the call to PLOTEM is positive PLOTEM will include the statistic in a plot if the IIth bit of the LOGHKL word is non-zero. If an octant is specified via MCLT, PLOTEM checks bits 33, 34 and 35 to see if the statistic belongs to the specified octant.

Variables:

M - a packed word which specifies class, number of layers and octant option.

MT=M/1000; = 1 standard classes, = 2 h00 layers, = 3 0k0 layers, = 4 00ℓ layers, = 5 standard classes plus h00 layers, etc.

MLAY - number of layers to be plotted, up to 10.

MOCT - octant options.

= 0 plot all octants together.
= 1 plot the +++ (h positive, k positive, ℓ positive) and --- octants separately.
= 2 plot the +++, ---, and +-+ octants.
= 3 plot the +++, ---, ++-, and +-+ octants.
= 4 plot the +++, ++-, +-+, -++, and --- octants.
= 7 plot all eight octants.

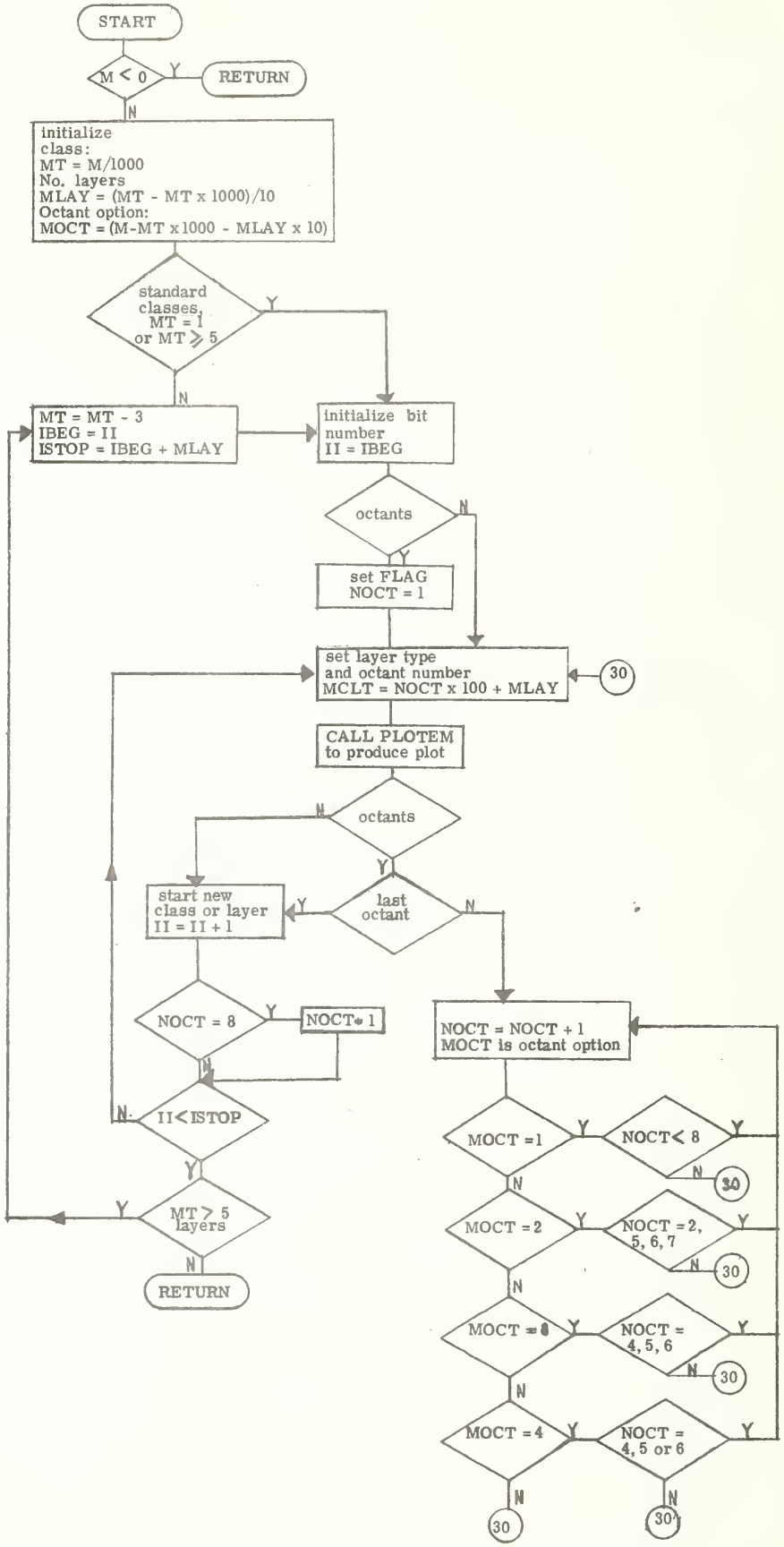
NOCT - is the octant number, one through eight, where octant 1 is the +++ octant, octant 2 is the ++- octant, etc. Octants are represented in the LOGHKL word by a 3 bit configuration with zero for positive indices and 1 for negative indices, i.e., ++- is 001.

MCLT - "hundreds" digit is the octant number, NOCT, "units" digit is the class or layer type, MT = 1, 2, 3, or 4.

II - number of the bit in the LOGHKL word to be tested, 0th through 20th.

IBEG - first bit to be examined (the 11th if layers only).

ISTOP- the last bit to be examined.


START
M < 0
Y
RETURN
N
initialize
class:
MT = M/1000
No. layers
MLAY = (MT - MT x 1000)/10
Octant option:
MOCT = (M-MT x 1000 - MLAY x 10)
standard classes, MT = 1 or MT ⩾ 5
Y
N
MT = MT - 3
IBEG = II
ISTOP = IBEG + MLAY
initialize bit number
II = IBEG
octants
N
Y
set FLAG
NOCT = 1
set layer type
and octant number
MCLT = NOCT x 100 + MLAY
30
CALL PLOTEM
to produce plot
octants
N
Y
start new class or layer
II = II + 1
Y
last octant
N
NOCT = 8
Y
NOCT = 1
N
II < ISTOP
N
Y
MT ⩾ 5 layers
Y
N
RETURN
NOCT = NOCT + 1
MOCT is octant option
MOCT = 1
Y
NOCT < 8
Y
N
30
N
MOCT = 2
Y
NOCT = 2, 5, 6, 7
Y
N
30
N
MOCT = 3
Y
NOCT = 4, 5, 6
Y
N
30
N
MOCT = 4
Y
NOCT = 4, 5 or 6
Y
N
30
N
30

III. C. Subroutine DRUMRD

Purpose:

Reads a reflection record from word-addressable drum and unpacks the four words comprising the record.

Procedure:

Sets a pointer, JPOINT, then calls UNIVAC routine NTRAN to read in the record. The pointer indexes from the beginning of the drum files and is equal to (I-1)*4, where the I-th record is wanted. The listing is simple enough for no flow chart to be needed.

Important variables:

JHKL - packed word containing the indices.

F - value of Fo

SG - value of corresponding sigma.

NT - flag indicating unobserved or some other quality indicator.

I - the reflection record address.

Subroutine DRUMRT

Purpose:

Writes a reflection record on word-addressable mass storage.

Procedure:

Packs the indices into one word and places this packed word together with NT, F and SG into the buffer FDRUM and calls UNIVAC routine NTRAN to write out buffer.

Important Variables:

N - flag indicating whether reflection is "observed" or not.

IJ - array holding the indices h, k, and ℓ.

F, S - Fo and corresponding sigma.

Modification Aids for DRUMRD and DRUMRT:

Any procedure that operates on random-access, word-or-record-addressable mass storage is desirable for these two routines. Specific procedures depend on local computing facilities.

III. D. Subroutine FITCHK

Purpose:

Calculates values corresponding to the least-squares line, quadratic, and cubic fits for statistics or selected subsets of the statistics. If the plots of the statistics against d* or Q are being made, obtain the scatter of the statistics about the least-squares line.

Procedure:

Examine input flags, if abscissa is normal quantiles leave out the tails. Select statistics according to octant as specified by NOCT and class or layer as specified by II. The plotting subroutine handles 50 points conveniently, accordingly there are K/50 + 1 points per interval where K is the number of statistics in the subset. If there are at least 6 points in the interval the positive and negative deviations of the value calculated from the least-squares line are computed. The average positive and negative deviations can be calculated and plotted with the statistic, least-squares line, quadratic, and cubic values for the midpoint of the interval.

Important Variables:

LHALF - one-half of L, the number of points in the interval.

DM - array holding JC statistics for the data set.

X - array holding JC abscissa values.

DMSMAL - array holding subset of statistics to be plotted.

XSMALL - array holding corresponding subset of abscissa values.

K - number of statistics in the subset.

M - "hundreds" digit specifies octant number, 1 through 8, "units" digit positive if classes or layers wanted (subsets of statistics).

II - number of the bit in the LOGHKL word to be examined.

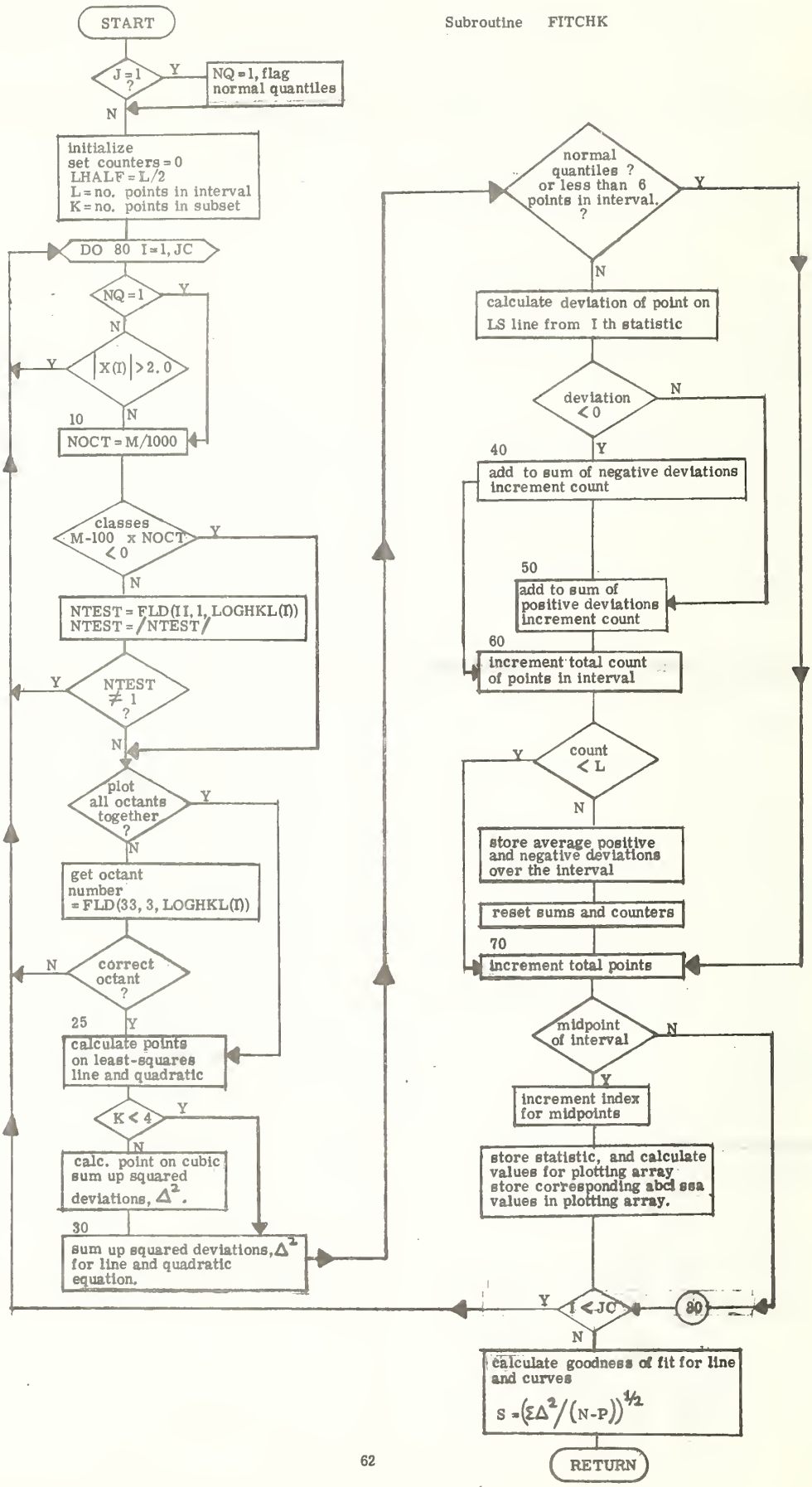
ALINE, BLINE - coefficients for linear equation.

ACURVE, BCURVE, CCURVE - coefficients for quadratic equation

ACUBIC, BCUBIC, CCUBIC, DCUBIC - coefficients for cubic equation.

SUMLN, SUMCV, SUMCUB - goodness-of-fit for line, quadratic and cubic equation.

NOCT - octant number, 1 through 8.


Subroutine FITCHK
START
J=1 ?
Y
NQ=1, flag normal quantiles
N
initialize
set counters=0
LHALF=L/2
L=no. points in interval
K=no. points in subset
DO 80 I=1, JC
NQ=1
Y
N
|X(I)|>2.0
Y
N
10
NOCT=M/1000
classes M-100 x NOCT <0
Y
N
NTEST=FLD(II, 1, LOGHKL(I))
NTEST=/NTEST/
NTEST ≠ 1 ?
Y
N
plot all octants together ?
Y
N
get octant number =FLD(33, 3, LOGHKL(I))
correct octant ?
N
Y
25
calculate points on least-squares line and quadratic
K<4
Y
N
calc. point on cubic sum up squared deviations, Δ².
30
sum up squared deviations, Δ² for line and quadratic equation.
normal quantiles ? or less than 6 points in interval. ?
Y
N
calculate deviation of point on LS line from I th statistic
deviation <0
N
Y
40
add to sum of negative deviations increment count
50
add to sum of positive deviations increment count
60
increment total count of points in interval
count <L
Y
N
store average positive and negative deviations over the interval
reset sums and counters
70
increment total points
midpoint of interval
N
Y
increment index for midpoints
store statistic, and calculate values for plotting array
store corresponding abcd ssa values in plotting array.
I<JC
Y
80
N
calculate goodness of fit for line and curves
S=(ΣΔ²/(N-P))^½
RETURN

III. E. Subroutine FREAD

Purpose:

Read in an observed data set, check for input errors, call MAXHKL to establish largest indices, store (for efficient sorting and reflection matching) slowest varying index for later identification of members of batches based on this index, establish the minimum and maximum values of the slowest varying index, and write out reflection information on drum.

Procedure:

The read-in is accomplished by looping over the maximum number of statistics allowed so that array overflow cannot occur. An end of file on the input file stops the read-in before the loop is completed. Reflections with read errors, zero values of Fo, with the unobserved indicator set to 1, and with indices all zero are skipped. A count is kept of the accepted reflections. Subroutine MAXHKL is called to establish the maximum value of the indices and the slowest varying index is stored in the ID array for each reflection.

Modification Aids:

This routine is the most likely one to be modified by users to fit their particular situation. The only part of read statement which should not be changed is the index read-in of the form IJ(JJ), IJ(KK), IJ(LL), which allows the order to be specified. IJ(1) is the slowest varying index which is stored in the ID array for each reflection.

Important Variables:

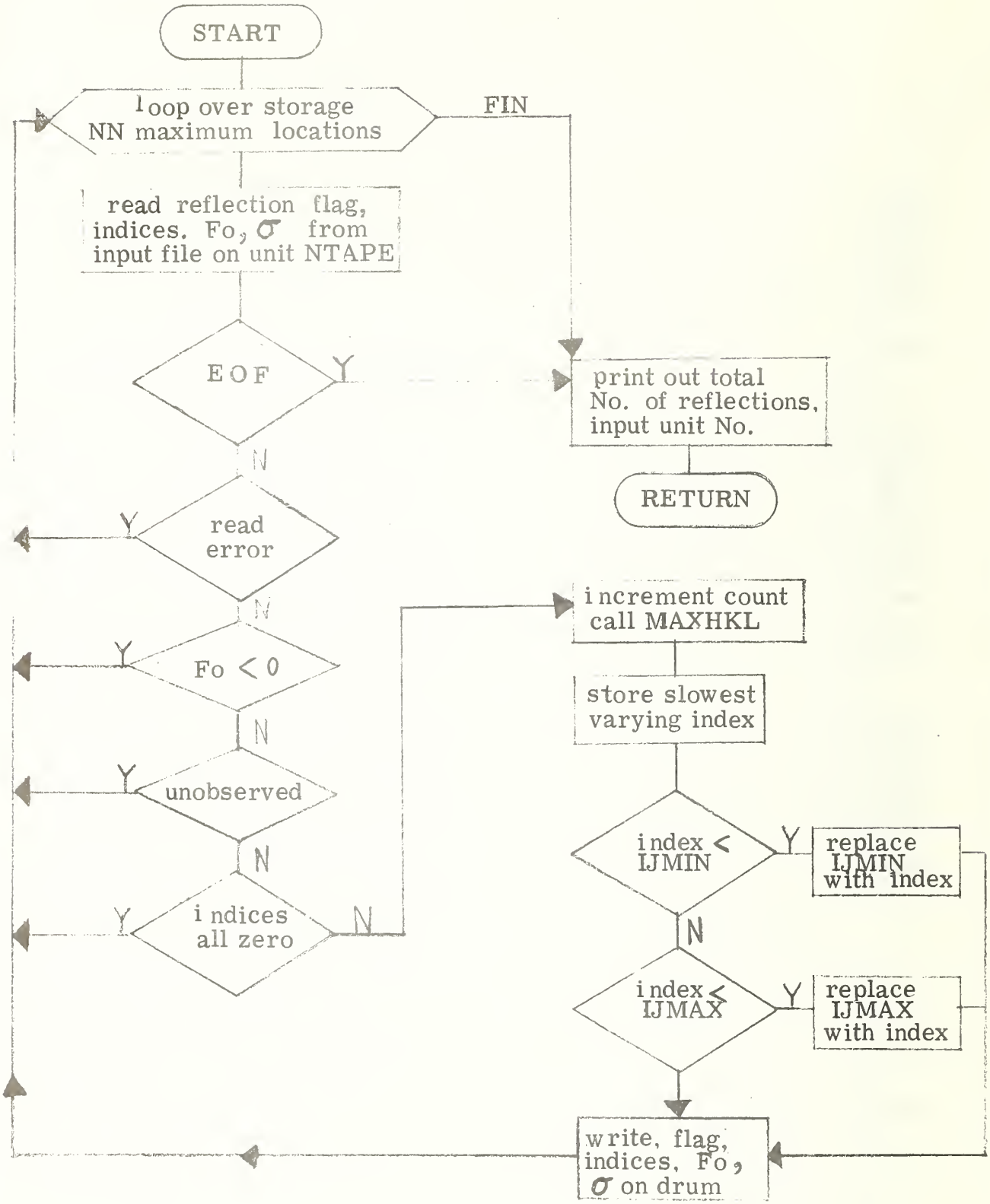
ID - array holding the value of the slowest-varying index for each reflection;

J - total number of usable reflections.

JJ, KK, LL - order of storing the indices, i.e., JJ=2, KK=1, LL=3 stores them in the order k, h, ℓ.

IJMIN, IJMAX - minimum and maximum values of the slowest varying index.

Subroutine FREAD



III. F. Subroutine INFOFC

Purpose:

Obtain reflection information from the binary data file generated by the X-RAY SYSTEM. Calculate the statistics and store corresponding reflection information on drum for further reference during the run.

Procedure:

Reads physical records from the binary data file until it finds the first one (logical record 15) containing the reflection information. Users are advised to consult the manual of the X-RAY SYSTEM for a complete description of the binary file. Each record is read into a buffer and reflection records are unblocked from this buffer. Array LOGFC contains the drum address for reflection information corresponding to each statistic. Statistics are stored in array DM. The FORTRAN listing gives further explanation of the procedure.

Important Variables:

JY - total number of usable statistics.

JZ - drum address for Fc and T-bar.

DM - array holding the JY statistics.

LOGFC - array holding drum addresses for information corresponding to the statistics for each reflection.

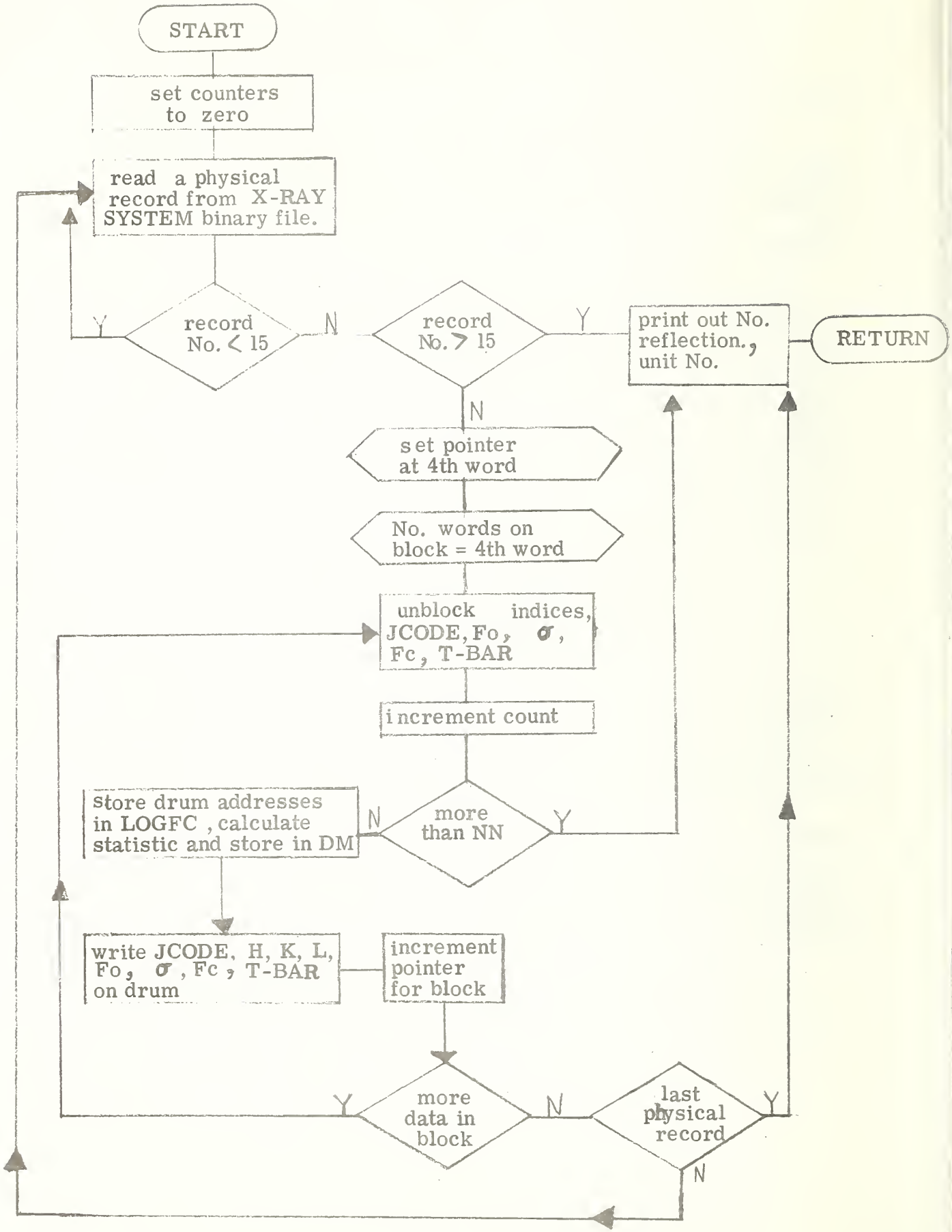
NN - maximum number of statistics allowed by size of DM array.

NFTEST- extra flag for miscellaneous purposes.

IPT - unblocking pointer-points to the last word unblocked.

NWORD - number of words in the block.

IJ - array holding the indices.


START
set counters to zero
read a physical record from X-RAY SYSTEM binary file.
record No. < 15
Y
N
record No. > 15
Y
print out No. reflection., unit No.
RETURN
N
set pointer at 4th word
No. words on block = 4th word
unblock indices, JCODE, Fo, σ, Fc, T-BAR
increment count
more than NN
N
Y
store drum addresses in LOGFC , calculate statistic and store in DM
write JCODE, H, K, L, Fo, σ, Fc, T-BAR on drum
increment pointer for block
more data in block
Y
N
last physical record
Y
N

III. G. Subroutine LSFIT

Purpose:

Select the required statistics and express their variation with whatever is selected as the abscissa (i.e., normal quantiles, d* values) in terms of linear, quadratic and cubic equations.

Procedure:

Examine input flags, if abscissa is normal quantiles leave out the tails, i.e., those values of statistics for which the absolute values of the corresponding normal quantiles are greater than 2.0. Select statistics according to specifications indicated by variables M and II. Form the required sums for a least-squares fit of linear, quadratic and cubic equations.

Important Variables:

X - array holding a total of J abscissa values.

Y - array holding a total of J statistics.

ALINE - intercept for the linear equation.

BLINE - slope for the linear equation.

ACURVE, BCURVE, CCURVE - coefficients for quadratic equation.

ACUBIC, BCUBIC, CCUBIC, DCUBIC - coefficients for cubic equation.

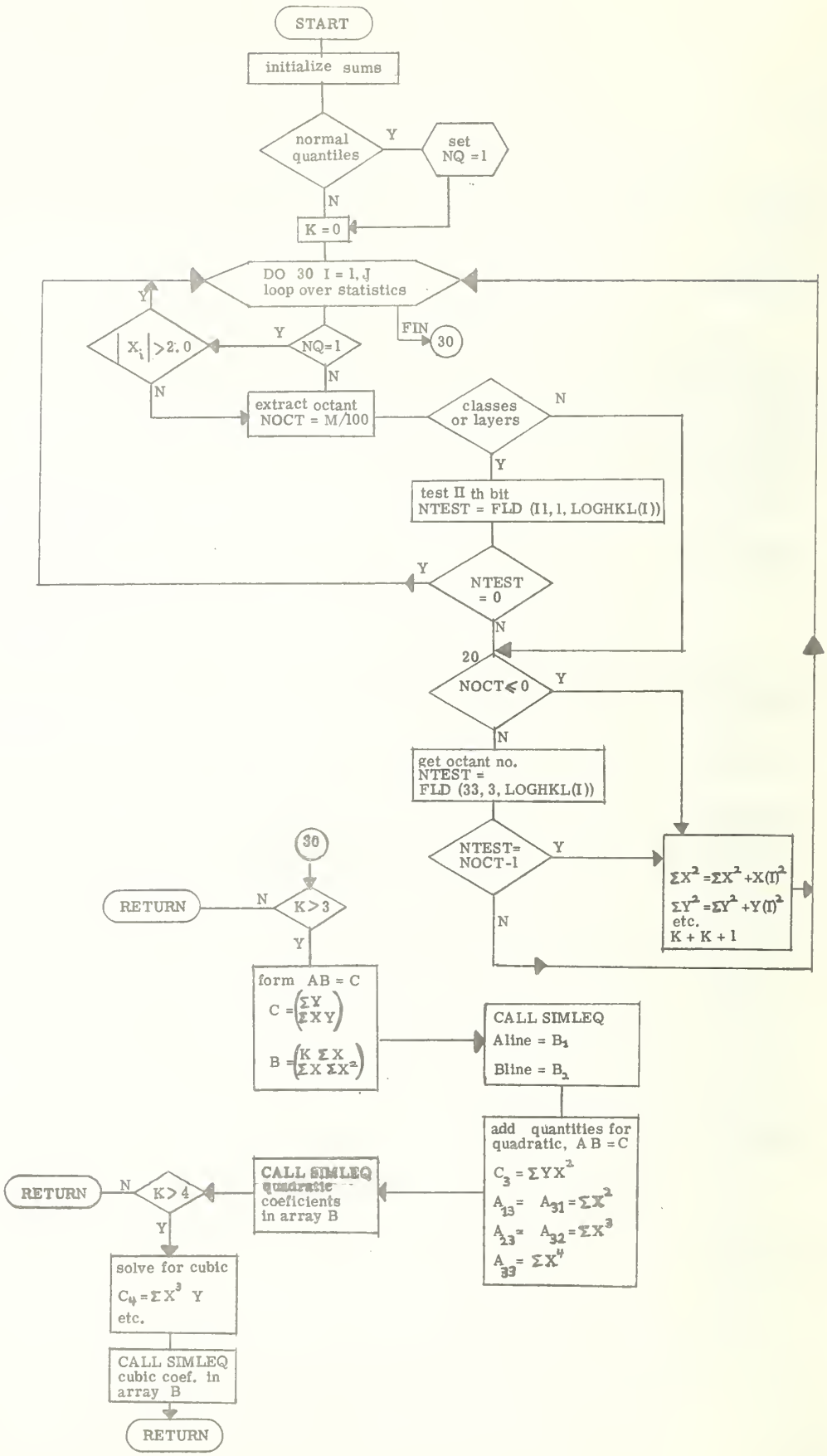
M - "hundreds" digit specified octant number, 1 through 8.
"units" digit positive if classes or layers wanted.

II - number of the bit in the LOGHKL word to be tested to divide the reflections into classes.

K - number of statistics in the subset.

NOCT - octant number, 1 through 8.

Subroutine LSFIT



III. H. Subroutine PLOTEM

Purpose:

Set up the proper title for the plot, acquire all the information required for the plot and print out the information actually plotted.

Procedure:

Examines control variables to determine what type of plot is wanted. Gets the corresponding title words and stores them in array ATITLE. Call subroutine LSFIT to obtain least-squares lines and curves for the statistics or subset. Calls subroutine FITCHK to calculate the values predicted by the least-squares lines and curves and establishes the scatter about the line if d^* or $\log (Fc^2/\sin 2\theta)$ plots are being made. Subroutine PLOTS is called to produce the actual plot on the line printer after a new page has been titled properly by PLOTEM. After the plot has been made, PLOTEM prints out the plot symbols for identification of the various curves and lists the values that have been plotted. Subroutine PLOTS sets its own axial limits from the input data.

Important Variables:

JC - total number of statistics stored in DM array.

X - array holding abscissa values, normal or half-normal quantiles, d^* values, or $\log (Fc^2/\sin 2\theta)$ values.

DMSMAL- array holding 50 values from all the statistics or subset to be plotted.

XSMALL- array holding the corresponding abscissa values.

M - control variable; "hundreds" digit is the octant number, "tens" digit indicates half-normal plot (structural parameters, etc.), "units" digit =1 indicates standard classes; =2, h layers; =3, k layers; =4, ℓ layers.

LAYT - layer type as specified by "units" digit of M.

II - number 0 through 10 for bit standard class; bit number 11 through 20 for layers 1 - 10.

J - number of points actually plotted.

K - number of points in reflection class or layer.

L - number of points in interval, i.e., L = K/50 + 1; if K/50 < 1 every point will be plotted.

ALINE, BLINE - coefficients for linear equation.

ACURVE, BCURVE, CCURVE - coefficients for quadratic.

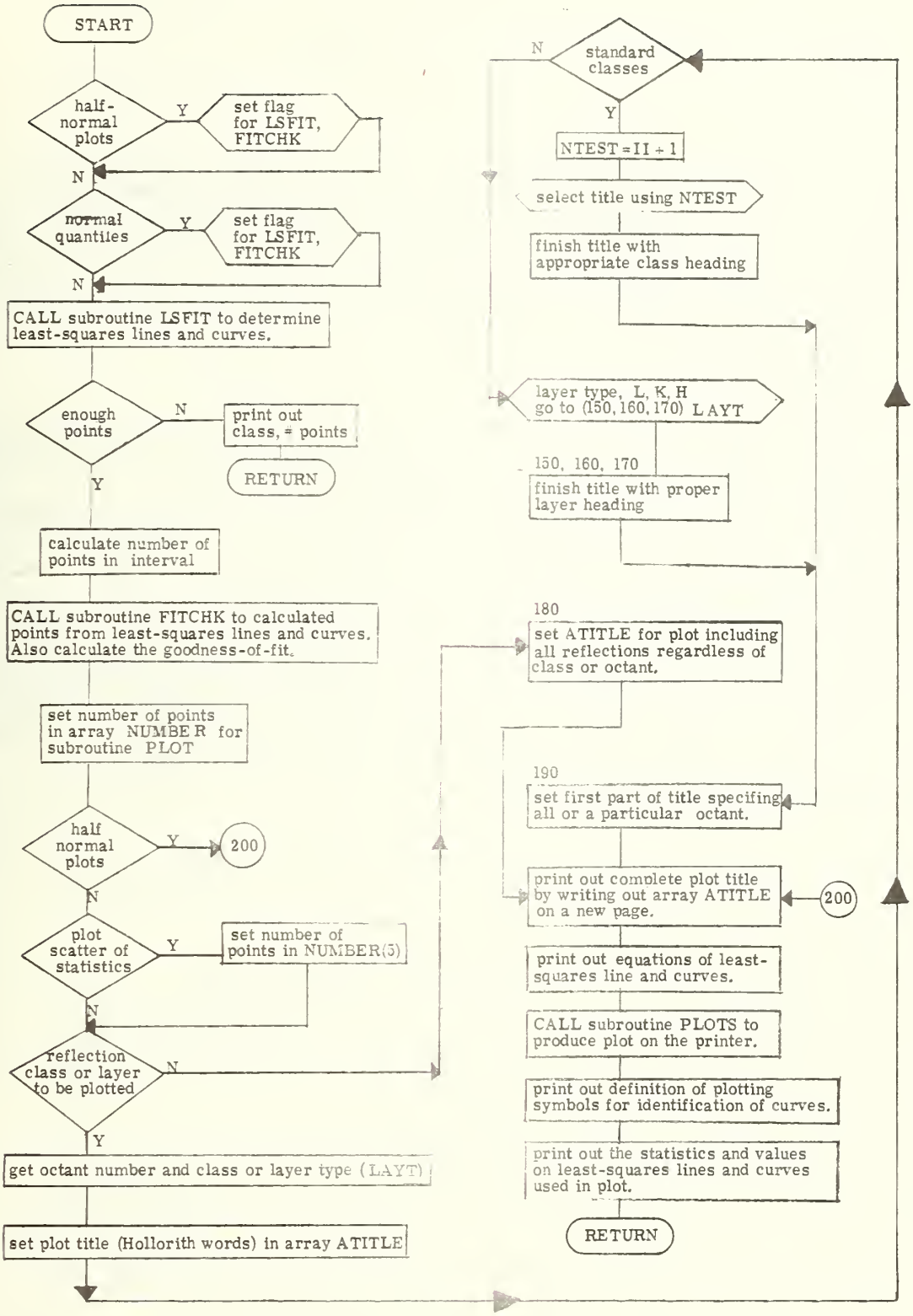
ACUBIC, BCUBIC, CCUBIC, DCUBIC - coefficients for cubic equation.

SUMLN, SUMCV, SUMCUB - goodness-of-fit for line, quadratic, and cubic equations.

NPT - number of curves to plot, 3 or 5.

NUMBER - number of points in each curve.

ATITLE - array holding the plot title.


START
half-normal plots
Y
set flag for LSFIT, FITCHK
N
normal quantiles
Y
set flag for LSFIT, FITCHK
N
CALL subroutine LSFIT to determine least-squares lines and curves.
enough points
N
print out class, # points
RETURN
Y
calculate number of points in interval
CALL subroutine FITCHK to calculated points from least-squares lines and curves. Also calculate the goodness-of-fit.
set number of points in array NUMBER for subroutine PLOT
half normal plots
Y
200
N
plot scatter of statistics
Y
set number of points in NUMBER(5)
N
reflection class or layer to be plotted
N
Y
get octant number and class or layer type (LAYT)
set plot title (Hollorith words) in array ATITLE
N
standard classes
Y
NTEST = II + 1
select title using NTEST
finish title with appropriate class heading
layer type, L, K, H
go to (150, 160, 170) LAYT
150, 160, 170
finish title with proper layer heading
180
set ATITLE for plot including all reflections regardless of class or octant.
190
set first part of title specifing all or a particular octant.
print out complete plot title by writing out array ATITLE on a new page.
200
print out equations of least-squares line and curves.
CALL subroutine PLOTS to produce plot on the printer.
print out definition of plotting symbols for identification of curves.
print out the statistics and values on least-squares lines and curves used in plot.
RETURN

III. I. Subroutine REREAD

Purpose:

Reads the file of matched reflections (two sets of observed data) previously generated by subroutine SCALE, calculates and stores the corresponding statistic using the value for the scale factor SKALE (one of REREAD input arguments). Writes the reflection pair on drum file for further reference and stores the location in catalog word LOGFC. Also characterizes the distribution of statistics.

Procedure:

The input file is buffered into a buffer of 490 words which contain information on 7 pairs of matched reflections. Each pair consists of the indices and two values of Fo and two values of σ(Fo). These are unblocked until the buffer is empty, when a new read-in takes place. The end of the data is marked by a value of 99 for the first index. After unblocking, counters JY and JZ are incremented, the statistic is calculated, and the indices and associated values for the pair are written on the drum file. The location of this information is stored in the LOGFC(I) word corresponding to the I-th statistic. Upon completion of input the mean and e.s.d. of the statistics are calculated and printed out to provide a characterization of their distribution.

Important Variables:

BUFF - array holding 7 pairs of matched reflections.

NBUFF -size of array BUFF.

JY - total number of statistics calculated.

JZ - number of Fo, σ values, used for pointing to corresponding drum locations.

DM - array holding JY statistics.

LOGFC- array holding locations of information on drum corresponding to the statistics.

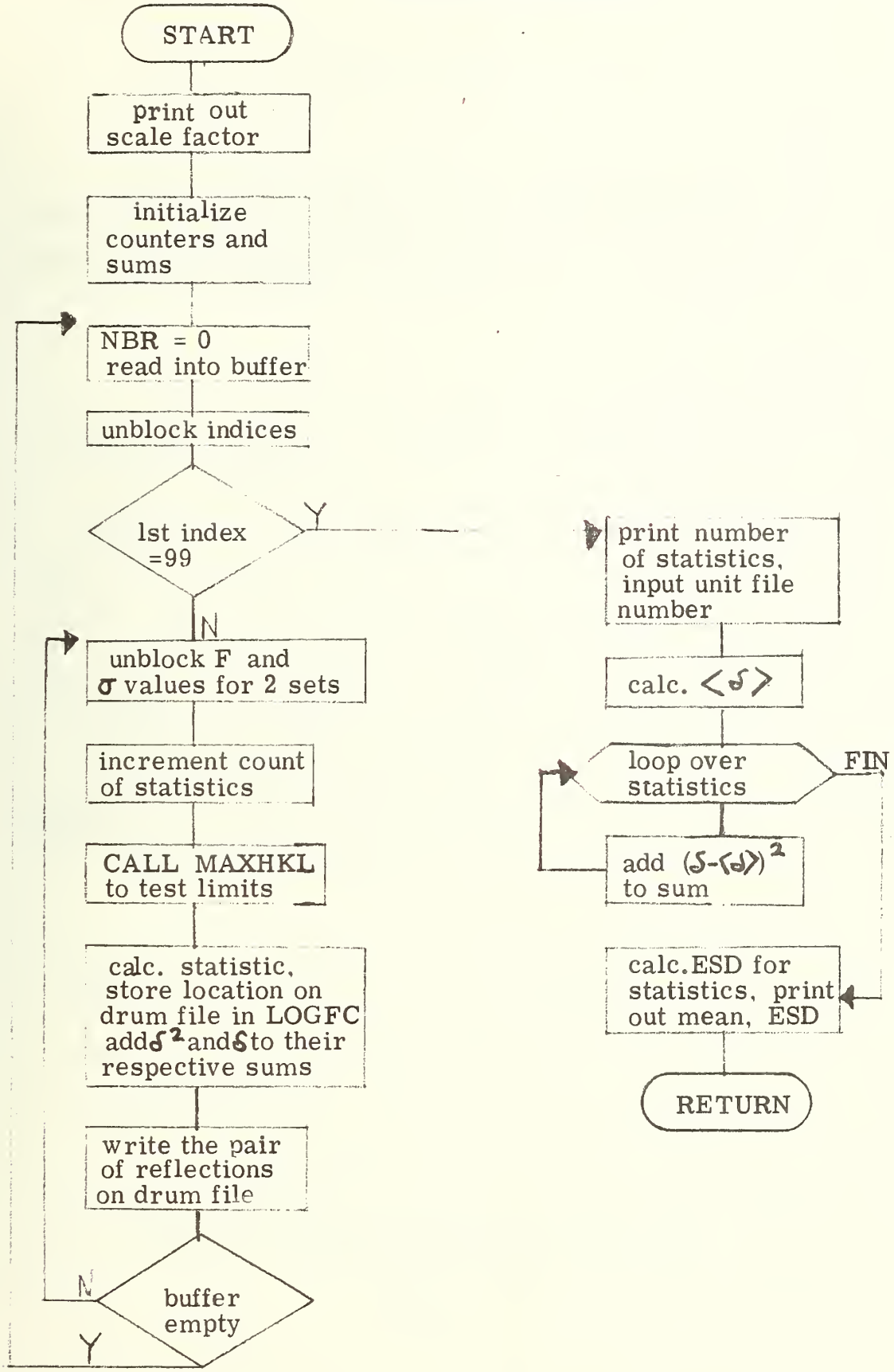
JJ, KK, LL - order in which to unpack the indices for matched pair.

SKALE -scale factor relating the two observed data sets.

SUMM- mean of the distribution of statistics.

SSUM- standard deviation of distribution of statistics.

Subroutine REREAD



III. J. Subroutine SCALE

Purpose:

Match reflections from two sets of observed data, store drum addresses of the pair on LOGFC word and calculate the scale factor relating the two observed sets of data. Optional tasks include applying spherical absorption corrections to the data sets and writing out matched pairs on unit NR.

Procedure:

Reads the packed index word, Fo and σ(Fo) values for each data set from the file written on drum by subroutine FREAD. This information is read in by batches, a batch being all those reflections with the same value of the slowest varying Miller index (batching reflections saves time in the sort procedure). Corresponding reflections are matched by comparing the index words JKLA and JKLB and matched indices, Fo and σ(Fo) values are optionally copied into a buffer which is then output on unit NR. The buffer holds information corresponding to 70 pairs of reflections. The corresponding drum addresses of the pair are stored in LOGFC(I) for the I-th statistic. If requested, a spherical absorption correction is applied before pairs are written out on unit NR. Subroutine ABSORB is called to calculate the transmission factor and its error for each reflection. The sum of Fo for each data set is accumulated. Finally, the scale factor is calculated and printed out for reference together with the total number of statistics, the number of reflections in each data set and the input unit number.

Important Variables:

IDA, IDB - arrays holding values of slowest varying index for the data sets.

JA, JB - number of reflections in data set one and two.

LOGFC - array holding drum address for reflections making up the statistic.

SKALE - scale factor relating the two data sets.

IJMINA, IJMAXA - minimum and maximum values of slowest varying index for set one.

JKLA - packed indices for data set one.

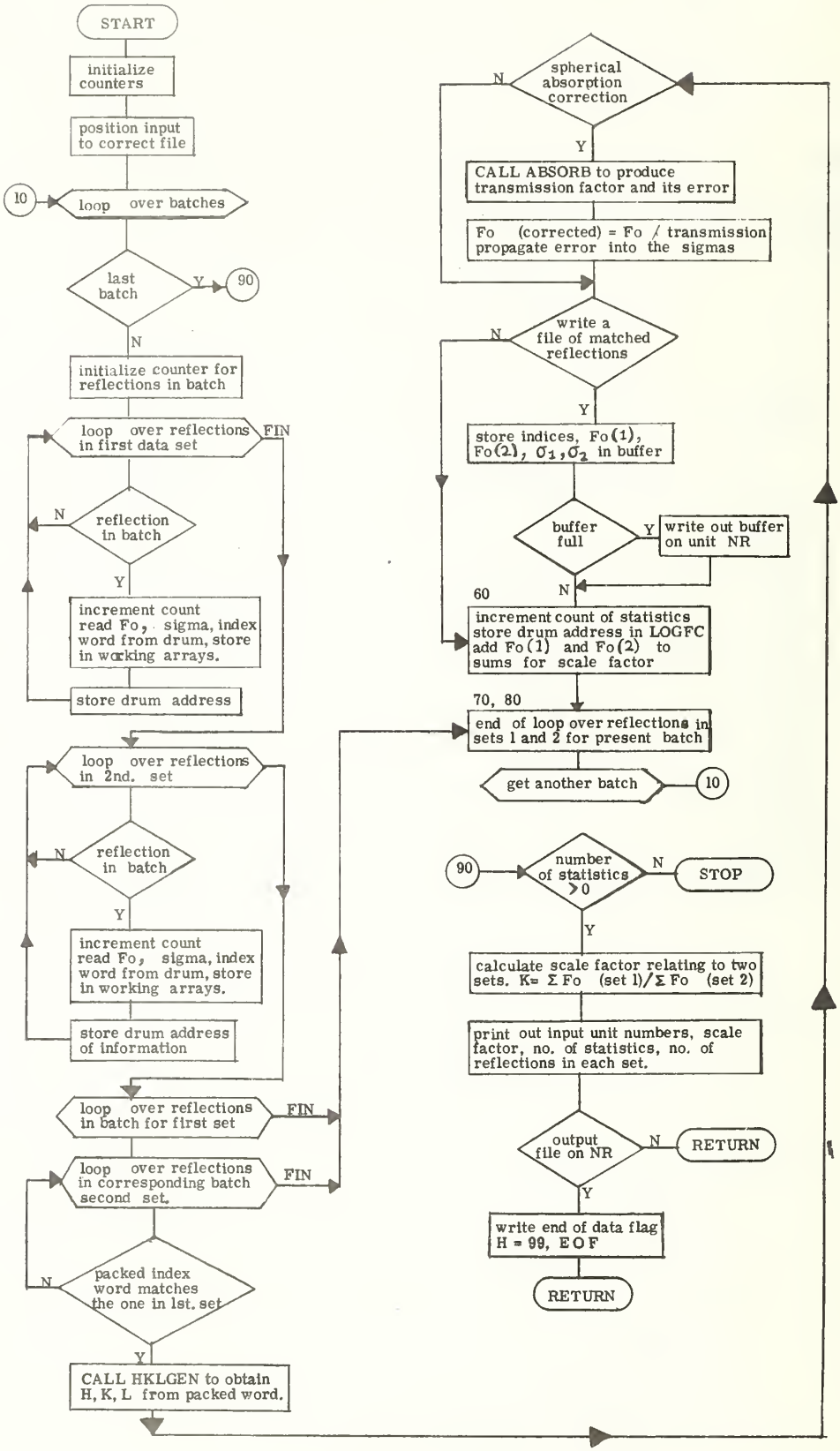
JKLB - packed indices for data set two.

KA, KB - arrays holding drum addresses of the matched pair.

FA, FB - arrays holding Fo values for the two data sets.

SGA, SGB - arrays holding sigma values for the two data sets.

NBUFF - size of buffer BUFF.


START
initialize counters
position input to correct file
10
loop over batches
last batch
Y
90
N
initialize counter for reflections in batch
loop over reflections in first data set
FIN
N
reflection in batch
Y
increment count read Fo, sigma, index word from drum, store in working arrays.
store drum address
loop over reflections in 2nd. set
N
reflection in batch
Y
increment count read Fo, sigma, index word from drum, store in working arrays.
store drum address of information
loop over reflections in batch for first set
FIN
loop over reflections in corresponding batch second set.
FIN
N
packed index word matches the one in 1st. set
Y
CALL HKLGEN to obtain H, K, L from packed word.
N
spherical absorption correction
Y
CALL ABSORB to produce transmission factor and its error
Fo (corrected) = Fo / transmission propagate error into the sigmas
N
write a file of matched reflections
Y
store indices, Fo(1), Fo(2), σ1, σ2 in buffer
buffer full
Y
write out buffer on unit NR
N
60
increment count of statistics store drum address in LOGFC add Fo(1) and Fo(2) to sums for scale factor
70, 80
end of loop over reflections in sets 1 and 2 for present batch
get another batch
10
90
number of statistics > 0
N
STOP
Y
calculate scale factor relating to two sets. K= Σ Fo (set 1)/Σ Fo (set 2)
print out input unit numbers, scale factor, no. of statistics, no. of reflections in each set.
output file on NR
N
RETURN
Y
write end of data flag H = 99, EOF
RETURN

IV. FORTRAN Code

IV. A. Introduction

STATUS was written using UNIVAC 1108 FORTRAN V code and is generally compatible with FORTRAN IV. The section on Common Blocks and Arrays describes common block storage and its purpose. Arrays requiring similar dimensions and the parameter variables which set the dimensions are also given. A number of equivalences used to conserve storage are indicated along with the names of the arrays that are later written over. Storage allotment can be changed by changing the value of the parameter variables. Users whose compilers will not allow the parameter statement can remove it and dimension the arrays explicitly in the main routine, STATUS and, if required, in the subroutines also. The section on Features Specific to UNIVAC 1108 FORTRAN V indicates where these statements and all other features of the code that are not American Standard FORTRAN are located in STATUS. The main control parameters are described in the section on Input Parameters.

IV. B. Features Specific to UNIVAC 1108 FORTRAN V

NTRAN - a UNIVAC routine which handles transfers of information between central memory and mass storage or tape devices. It performs such functions as reading blocks of data into arrays, skipping files, closing files, etc.

NTRAN is called by the following subroutines:

DRUMRD	Line number	17
DRUMRT	" "	23

FLD - a UNIVAC FORTRAN V function for manipulation of specified bits in a computer word.

Called by the following subroutines:

CATLOG	Line number	38 - 47, 61 -63
FITCHK	" "	44
INFING	" "	33 - 34
LSFIT	" "	52

PARAMETER statement - Assigns numerical values to variables specifying array dimensions and enables one to change program storage by redimensioning arrays conveniently.

Used in the following routines:

STATUS	Line number	29
FINGFO	" "	19
FITCHK	" "	10
FREAD	" "	11
INFING	" "	18
LSFIT	" "	12
REREAD	" "	13
SCALE	" "	17
USER	" "	4

I/O statements with END = statement number optional in the following routines:

STATUS	Line number	98
FREAD	" "	17

Dimension statements including PARAMETER variables:

STATUS	Line number	55

Statement involving array ATITLE (variable = 6H text)

STATUS	Line number	84 - 88
	" "	254 - 256
	" "	276 - 278
	" "	305 - 307
	" "	378 - 386
	" "	453 - 454
	" "	476 - 477
PLOTEM	Check entire subroutine	

```
1       C  STATUS MAIN PROGRAM
2       C  THE PROCEDURE IN THIS PROGRAM IS THAT DESCRIBED IN THE PAPER ' NORMAL
3       C  PROBABILITY PLOT ANALYSIS OF ERROR IN MEASURED AND DERIVED QUANTITIES
4       C  STANDARD DEVIATIONS' BY S.C.ABRAHAMS AND E.T.KEVE ACTA CRYST.(1971)
5       C  A27, P.157-165.
6       C  THIS PROCEDURE IS TO COMPARE (DELTA F)/(SIGMA F) WITH A NORMAL
7       C  DISTRIBUTION. THE PLOT SHOULD IDEALLY BE A STRAIGHT LINE WITH A
8       C  SLOPE =1.0.
9       C
10      C  RESIDUAL ANALYSIS DEVELOPED BY L.W.SCHROEDER BASED ON N.R. DRAPER
11      C  AND H.SMITH, 'APPLIED REGRESSION ANALYSIS' (1966)WILEY. CHAPTER 3.
12      C  NOTE THAT DELTA F/SIGMA F IS THE RESIDUAL IN 'UNIT DEVIATE' FORM.
13      C  PLOTS OF RESIDUALS VS INDEPENDENT VARIABLES (D-STAR,ETC.) SHOULD NOT
14      C  SHOW ANY TREND,ALSO, THE SCATTER SHOULD BE UNIFORM WHEN CORRECT
15      C  WEIGHTS HAVE BEEN APPLIED.
16      C
17      C  PROGRAM WRITTEN BY B. DICKENS AND L.W. SCHROEDER (311.05) AT THE
18      C  NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. 20234.
19      C  PHONE (301) 926-2455.
20      C
21      C  USE AT YOUR OWN RISK--PROGRAM ONLY TESTED ON UNIVAC 1108-EXEC-8.
22      C
23      C
24      C     ***************************************************************
25      C
26      C  PARAMETER STATEMENT ALLOWS EASY ADJUSTMENT OF ARRAY SIZES.
27      C QUANTITIES TO DO WITH AVAILABLE CORE AND ARRAY DIMENSIONS
28      C  CAUTION--- DON,T MAKE MM MORE THAN 100 WITHOUT ALTERING EQUIVALENCES
29              PARAMETER N=50, NNA=1800, NN=7200, NNB=1801, NNC=3601, NND=5401,
30             2 MM=60, MMM=400, NBUFF=700, MMMM=32
31      C  INITIALLIZE MAXIMUM AND MINIMUM VALUES OF MILLER INDICES TO
32      C  RIDICULOUS VALUES
33              DATA MAXH,MAXK,MAXL,MINH,MINK,MINL /-1000,-1000,-1000,1000,1000,
34             2 1000/
35              COMMON /D/ MAXH,MINH,MAXK,MINK,MAXL,MINL
36              COMMON /TITLE/ ATITLE(14)
37      C  INITIALLIZE HIGHEST AND LOWEST VALUES OF MILLER INDEX KEPT IN CORE IN
38      C  SAME WAY
39              DATA IJMAXA,IJMAXB,IJMINA,IJMINB /-1000,-1000,1000,1000/
40              COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
41      C  DUMMY IS USED IN EQUIVALENCE STATEMENTS SOME SUBROUTINES TO SAVE
42      C  SPACE
43              COMMON /H/ DUMMY(NN)
44      C  JJA,KKA AND LLA ARE THE ORDER IN WHICH THE MILLER INDICES ARE PACKED
45      C  INTO THE JKL WORD.  THEY ARE IN COMMON WITH NR AND NABS BECAUSE
46      C  THESE QUANTITIES ARE ALL USED AT THE END OF THE SCALE SUBROUTINE.
47              COMMON /ORDER/ JJA,KKA,LLA,NR,NFILER,NABS
48              DIMENSION TITLE(12),X(NN),DMSMAL(N,5),XSMALL(N,5),DM(NN),
49             2 LOGHKL(NN),LOGFC(NN)
50              DIMENSION DOC(2) /6HJAN    ,6H 1975 /
51      C  THESE ARE THE RECIPROCAL CELL VALUES
52              COMMON /G/ ASTAR,BSTAR,CSTAR,COSAST,COSBST,COSGST
53              COMMON /EXPT/ WAVE,SKALE,UR(2),DUR(2),NF
54      C  THESE ARE USED IN SUBROUTINE SCALE
55              DIMENSION IDA(NN),IDB(NN),FA(NNA),FB(NNA),SGA(NNA),SGB(NNA),
56             2 JKLA(NNA),JKLB(NNA),KA(NNA),KB(NNA)
57      C  THESE ARE USED FOR THE ATOMIC PARAMETER TESTING.
```

```
      DIMENSION NAMEA(MM),OCA(MM),XYZA(3,MM),SIGXA(3,MM),BA(6,MM),
     2 SIGBA(6,MM),NAMEB(MM),OCB(MM),XYZB(3,MM),SIGXB(3,MM),BB(6,MM),
     3 SIGBB(6,MM),SIGOA(MM),SIGOB(MM),LOG(NN),MATCH(MM)
C  LOTS OF EQUIVALENCES TO SAVE SPACE, SO THAT LOTS OF REFLECTIONS CAN
C  BE TREATED.
      EQUIVALENCE (DUMMY,LOGHKL)
      EQUIVALENCE (DM,IDA), (X,IDB), (LOGHKL(1),JKLA(1)), (LOGHKL(NNB),
     2 JKLB(1)), (LOGHKL(NNC),KA(1)), (LOGHKL(NND),KB(1))
      EQUIVALENCE (LOGHKL(200),OCA(1))
      EQUIVALENCE (LOGHKL(300),OCB(1))
      EQUIVALENCE (LOGHKL(400),SIGOA(1))
      EQUIVALENCE (LOGHKL(500),SIGOB(1))
      EQUIVALENCE (LOGHKL(600),XYZA(1,1))
      EQUIVALENCE (LOGHKL(900),XYZB(1,1))
      EQUIVALENCE (LOGHKL(1200),SIGXA(1,1))
      EQUIVALENCE (LOGHKL(1500),SIGXB(1,1))
      EQUIVALENCE (LOGHKL(1800),BA(1,1))
      EQUIVALENCE (LOGHKL(2400),BB(1,1))
      EQUIVALENCE (LOGHKL(3000),SIGBA(1,1))
      EQUIVALENCE (LOGHKL(3600),SIGBB(1,1))
      EQUIVALENCE (LOGHKL(4200),NAMEA(1))
      EQUIVALENCE (LOGHKL(4300),NAMEB(1))
      EQUIVALENCE (LOGHKL(4400),MATCH(1))
      EQUIVALENCE (LOGFC(1),LOG(1))
C INITIALLISE SOME OF TITLE OF PLOTS
10    DO 20 I=1,14
20    ATITLE(I)=6H
      ATITLE(8)=6H PLOT
      ATITLE(9)=6HOF DM(
      ATITLE(10)=6HVERT)
      ATITLE(11)=6H AGAIN
C  ASSIGN UNIT NUMBERS
      CALL PRIME
      WRITE (NOUT,370) DOC,NDRUM
C
C
C     **************************************************************
C
C  READ AND WRITE TITLE
30    CONTINUE
      READ (IN,460,END=340) TITLE
      WRITE (NOUT,470) TITLE
C  READ IN UNIT CELL
      CALL CELL
C  READ IN REFLECTION TAPE NUMBER AND ORDER IN WHICH INDICES VARY FOR
C  TAPE A AND TAPE B.ALSO READ FLAGS FOR READ REFLECTIONS AND READ
C  PARAMETERS, ETC.
      READ (IN,480) CARD,NTAPEA,JJA,KKA,LLA,NTAPEB,NFTEST,NPTEST,NFILEA,
     2NFILEB,NPRINT,NO,NOC,NLAYO,NOCTO,ND,NDC,NLAYD,NOCTD,NQ,NQC,NLAYQ,N
     3OCTQ,NR,NABS,MINDM,NF
      WRITE (NOUT,380) NTAPEA,NFILEA,NTAPEB,NFILEB,JJA,KKA,LLA,NFTEST,NP
     2TEST,NPRINT
      IF (CARD.NE.3HCTL) GO TO 330
C     READ 4TH CARD ONLY IF NECESSARY, I.E. FOR SCALING, Q-PLOTS, OR
C     IF ABSORPTION CORRECTION WANTED.
      IF(NQ.LE.0.AND.NABS.LE.0.AND.NFTEST.NE.4) GO TO 35
      READ(IN,390) WAVE,SKALE,UR,DUR
35 CONTINUE
```

```
116       C
117       C  END OF CARD READ-IN, PRINT HEADINGS
118             IF (NR.GT.0) WRITE (NOUT,400) NR
119             IF (NABS.GT.0) WRITE (NOUT,410)
120             IF (MINDM.GT.0) WRITE (NOUT,420)
121       C
122       C  CHECK INPUT PARAMETERS FOR ERRORS - STOP IF NECESSARY.
123             IF(JJA.LT.0.OR.KKA.LT.0.OR.LLA.LT.0) GO TO 330
124             IF (JJA.GT.3.OR.KKA.GT.3.OR.LLA.GT.3) GO TO 330
125             IF (NFTEST.LT.0.OR.NFTEST.GT.5) GO TO 330
126             IF (NPTEST.LT.0.OR.NPTEST.GT.2) GO TO 330
127             IF (NOC.GT.7.OR.NDC.GT.7.OR.NQC.GT.7) GO TO 330
128             IF (NABS.LT.0.OR.NR.LT.0) GO TO 330
129             IF (WAVE.LT.-.000001.OR.WAVE.GT.3.0) GO TO 330
130             IF (SKALE.LT.-.01) GO TO 330
131             IF (NLAYO.LT.0.OR.NLAYO.GT.10) GO TO 330
132             IF (NOCTO.LT.0.OR.NOCTO.GT.7) GO TO 330
133             IF (NOCTO.EQ.5.OR.NOCTO.EQ.6) NOCTO=7
134             IF (NLAYD.LT.0.OR.NLAYD.GT.10) GO TO 330
135             IF (NOCTD.LT.0.OR.NOCTD.GT.7) GO TO 330
136             IF (NOCTD.EQ.5.OR.NOCTD.EQ.6) NOCTD=7
137             IF (NLAYQ.LT.0.OR.NLAYQ.GT.10) GO TO 330
138             IF (NOCTQ.LT.0.OR.NOCTQ.GT.7) GO TO 330
139             IF (NOCTQ.EQ.5.OR.NOCTQ.EQ.6) NOCTQ=7
140             IF(NFLAG.LT.-01) GO TO 330
141       C
142       C  PRINT OUT WHAT THE PROGRAM WILL ATTEMPT TO DO
143             IF (NO.EQ.1) WRITE (NOUT,430) NOC,NLAYO,NOCTO
144             IF (ND.EQ.1) WRITE (NOUT,440) NDC,NLAYD,NOCTD
145             IF (NQ.EQ.1) WRITE (NOUT,450) NQC,NLAYQ,NOCTQ
146       C  COMPUTE THE NUMBER OF PLOTS PROGRAM WILL ATTEMPT TO GENERATE
147             NOPLTS=0
148             ELEVO=11
149             ELEVD=11
150             ELEVQ=11
151             IF (NOC.EQ.2.OR.NOC.EQ.3.OR.NOC.EQ.4) ELEVO=0
152             IF (NOC.LE.1) NLAYO=0
153             IF (NDC.EQ.2.OR.NDC.EQ.3.OR.NDC.EQ.4) ELEVD=0
154             IF (NDC.LE.1) NLAYD=0
155             IF (NQC.EQ.2.OR.NQC.EQ.3.OR.NQC.EQ.4) ELEVQ=0
156             IF(NOC.LE.0)ELEVO=0
157             IF(NDC.LE.0)ELEVD=0
158             IF(NQC.LE.0)ELEVQ=0
159             IF (NQC.LE.1) NLAYQ=0
160             IF (NO.EQ.1) NOPLTS=NOPLTS+1+(ELEVO+NLAYO)*(NOCTO+1)
161             IF (ND.EQ.1) NOPLTS=NOPLTS+1+(ELEVD+NLAYD)*(NOCTD+1)
162             IF (NQ.EQ.1) NOPLTS=NOPLTS+1+(ELEVQ+NLAYQ)*(NOCTQ+1)
163             IF(NPTEST.GT.0) NOPLTS=NOPLTS + 3
164             NPAGES=2*NOPLTS + 5
165             WRITE (NOUT,350) NOPLTS,NPAGES
166             IF (NPAGES.GT.250) WRITE (NOUT,360)
167             IF (NPAGES.GT.250) STOP
168       C  ABSORPTION CORRECTION VALID FOR MU R=0.0 TO 1.0
169             DO 40 JC=1,2
170             IF (UR(JC).LT.-.0001.OR.UR(JC).GT.1.0) GO TO 330
171       40    IF (DUR(JC).GT.UR(JC).OR.DUR(JC).LT.-.0001) GO TO 330
172       C  GET READY FOR ASSIGNED GO TO STATEMENTS LATER
173             NFTEST=NFTEST+1
```

```
174           NPTEST=NPTEST+1
175     C  POSITION INPUT TAPES NTAPEA AND NTAPEB TO NFILEA AND NFILEB
176           IF (NTAPEA.GT.6) CALL POSITN (NTAPEA,NFILEA)
177           JC=0
178           JZ=-1
179           JA=0
180           JB=0
181     C    IF FO AND FC BEING READ, COPY OVER SORT ORDER.
182           JJB=JJA
183           KKB=KKA
184           LLB=LLA
185     C SET FLAG FOR CATALOGING-LATER TURNED OFF IF NO  CLASSES NEEDED.
186           NCL=1
187     C IN PROGRAM,  NFTEST= 1 NO F,S
188     C                NFTEST = 2 READ FOBS AND FC FROM FINGER FOURIER TAPE
189     C      NFTEST = 3 READ ONE SET OF FOBS FROM FIRST FINGER FOURIER TAPE AND
190     C SET FROM SECOND  FINGER FOURIER TAPE
191     C                NFTEST = 4 F,S FROM XRAY TYPE INPUT, 2 DIFFERENT TAPES,2
192     C                SETS OF F,S.
193     C      NFTEST=5   READ FO'S FROM TAPE PREVIOUSLY WRITTEN BY SUBROUTINESCA
194     C
195     C      NFTEST=6 READ FO AND FC FROM X-RAY 70 BINARY DATA FILE OR USER
196     C      FILE IF INFOFC SUITABLY MODIFIED.
197           GO TO (140,50,50,60,80,  90), NFTEST
198     C READ F,S FROM FINGER FOURIER TAPE
199     50     CALL FINGFO (JA,LOGFC,DM,JJA,KKA,LLA,NFTEST,IDA,IJMINA,IJMAXA,NTAP
200          2EA,JZ)
201     C  ADD THESE REFLECTIONS TO TOTAL
202           JC=JC+JA
203     C  IF WORKING WITH FO AND FC THEY NEED NOT BE SORTED AND SCALED.  JUMP
204     C  TO SORTING OF DM VALUES, WHICH WERE CALCULATED IN FINGFO FOR THIS
205     C  CASE.
206           IF (NFTEST.EQ.2) GO TO 100
207     C  IF ANOTHER FO SET IS NEEDED, READ IT NOW
208           IF (NTAPEB.GT.6) CALL POSITN (NTAPEB,NFILEB)
209           CALL FINGFO (JB,LOGFC,DM,JJB,KKB,LLB,NFTEST,IDB,IJMINB,IJMAXB,NTAP
210          2EB,JZ)
211     C  ADD THESE REFLECTIONS TO TOTAL
212           JC=JC+JB
213           GO TO 70
214     C READ FIRST REFLECTION SET FROM XRAY67 TYPE INPUT
215     60     CALL FREAD (NTAPEA,IDA,JA,JJA,KKA,LLA,IJMINA,IJMAXA)
216     C READ IN SECOND REFLECTION SET
217           IF (NTAPEB.GT.6) CALL POSITN (NTAPEB,NFILEB)
218           CALL FREAD (NTAPEB,IDB,JB,JJB,KKB,LLB,IJMINB,IJMAXB)
219     C  FIND COMMON REFLECTIONS AND SCALE FACTOR
220     70     CONTINUE
221           CALL SCALE (IDA,JA,IDB,JB,JC,LOGFC,SKALE,IJMINA,IJMAXA,FA,FB,SGA,S
222          2GB,JKLA,JKLB,KA,KB)
223     C  CALL SPEC FOR MINIMUM OF SUM(DM**2) IF REQUESTED
224           IF (MINDM.GT.0) CALL SPEC (JC,SKALE,DM,LOGFC,NN)
225     C  CALCULATE DM QUANTITIES FROM F,S AND SIGMAS
226           IF (MINDM.LE.0) CALL DMCALC (JC,SKALE,DM,LOGFC,NN)
227           GO TO 100
228     C
229     80     CALL REREAD (JC,LOGFC,DM,JJA,KKA,LLA,NTAPEA,JZ,SKALE)
230     C  CALL SPEC FOR MINIMUM OF SUM(DM**2) IF REQUESTED
231           IF (MINDM.GT.0) CALL SPEC (JC,SKALE,DM,LOGFC,NN)
```

```
232             GO TO 100
233       C
234       90    CALL INFOFC (NTAPEA,JC,JZ,NFTEST,IJMINA,IJMAXA,LOGFC,DM,NN)
235       C
236       C
237       C     ****************************************************************
238       C
239       C  SORT DM VALUES IN ORDER OF INCREASING MAGNITUDE
240       100   CALL SORT (DM,X,LOGFC,LOGHKL,JC,NN)
241       C CALCULATE NORMAL PROBABLITY QUANTILES IF NEEDED.
242             IF (NPRINT.GT.0.OR.NO.GE.1) CALL PROB (DM,X,JC,NN)
243       C  PRINT DM VALUES AS REQUESTED VIA NPRINT
244             IF (NPRINT.GT.0) CALL PRYNT (DM,LOGFC,LOGHKL,JC,JJA,KKA,LLA,JJB,KK
245            2B,LLB,NPRINT,X,NFTEST,NN)
246       C
247       C
248       C     ****************************************************************
249       C
250       C  THIS SECTION FOR PLOTS OF RESIDUALS VS NORMAL QUANTITLES
251             IF (NO.NE.1) GO TO 110
252             M=0
253       C SET UP REMAINDER OF TITLE.
254             ATITLE(12)=6HST NOR
255             ATITLE(13)=6HMAL QU
256             ATITLE(14)=6HANTILE
257             WRITE (NOUT,550)
258       C CATALOG REFLECTIONS IF REQUIRED.
259             IF (NOC.GT.0) CALL DMDQ (DM,LOGFC,JC,JJA,KKA,LLA,X,LOGHKL,0,0,NOC,
260            2NN)
261       C PLOT DM VS THE NORMAL QUANTILES.
262             CALL PLOTEM (JC,X,DM,DMSMAL,XSMALL,M,II,LOGHKL,NN,N)
263             IF (NOC.LE.0) GO TO 110
264       C  PLOT REFLECTION CLASSES
265             M=NOC*1000+NLAYO*10+NOCTO
266             CALL ANISO (JC,LOGHKL,DMSMAL,XSMALL,DM,X,M,NN,N)
267       110   CONTINUE
268       C
269       C
270       C     ****************************************************************
271       C
272       C  SKIP OVER PLOTTING OF DM AGAINST D-STAR IF THESE PLOTS NOT REQUESTED.
273             IF (ND.NE.1) GO TO 120
274             M=0
275       C  FIX UP TITLE
276             ATITLE(12)=6HST
277             ATITLE(13)=6HDSTAR
278             ATITLE(14)=6H
279             WRITE (NOUT,550)
280       C   OBTAIN DSTAR  VALUES AND CATALOG IF REQUIRED.
281             CALL DMDQ (DM,LOGFC,JC,JJA,KKA,LLA,X,LOGHKL,1,0,NDC,NN)
282       C  DO OVERALL PLOT
283       C SORT X ARRAY HOLDING DSTAR**2 VALUES AND REARRANGE CORRESPONDING ARRAY
284             CALL SORT (X,DM,LOGFC,LOGHKL,JC,NN)
285             CALL PLOTEM (JC,X,DM,DMSMAL,XSMALL,M,II,LOGHKL,NN,N)
286       C  SKIP PLOTTING OF REFLECTION CLASSES IF SO REQUESTED
287             IF (NDC.LE.0) GO TO 120
288             M=NDC*1000+NLAYD*10+NOCTD
289       C PLOT DM AS FUNCTION OF H,K,L OF REFLECTION. THE NECESSARY INFORMATION
```

```
C HAS ALREADY BEEN STORED IN THE LOGHKL ARRAY IN THE DMD SUBROUTINE.
      CALL ANISO (JC,LOGHKL,DMSMAL,XSMALL,DM,X,M,NN,N)
120   CONTINUE
C
C
C
C     ******************************************************************
C
      IF (NQ.NE.1) GO TO 130
C THIS SECTION FOR PLOTS OF RESIDUALS VS. CRYSTAL Q (F**2/SIN(2THETA))
C  GET Q VALUES NOW
      M=0
C CALCULATE Q VALUES AND STORE IN X ARRAY
      CALL DMDQ (DM,LOGFC,JC,JJA,KKA,LLA,X,LOGHKL,0,1,NQC,NN)
C  FIX UP GENERAL TITLE
      ATITLE(12)=6HST LOG
      ATITLE(13)=6HF**2/S
      ATITLE(14)=6HIN(2T)
      WRITE (NOUT,550)
C  DO OVERALL PLOT FIRST.
C SORT X ARRAY HOLDING Q VALUES.
      CALL SORT (X,DM,LOGFC,LOGHKL,JC,NN)
      CALL PLOTEM (JC,X,DM,DMSMAL,XSMALL,M,II,LOGHKL,NN,N)
C  PLOTS FOR CLASSES IF DESIRED.
      IF (NQC.LE.0) GO TO 130
C  PLOT RESIDUAL VS Q FOR VARIOUS CLASSES OF REFLECTIONS.
      M=NQC*1000+NLAYQ*10+NOCTQ
      CALL ANISO (JC,LOGHKL,DMSMAL,XSMALL,DM,X,M,NN,N)
130   CONTINUE
C
C
C     ******************************************************************
C
C  TREAT ATOMIC PARAMETERS IN HALF NORMAL PLOTS
C
C
C              NPTEST = 1 NO PARAMETERS
C              NPTEST = 2 PARAMETERS FROM FINGER FOURIER TAPE
C              NPTEST = 3 PARAMETERS READ IN.
C
140   GO TO (30,160,150), NPTEST
C SPECIAL USER INPUT TO BE SPECIFIED BY USER.
150   CALL USER (NAMEA,OCA,SIGOA,XYZA,SIGXA,BA,SIGBA,LA,NTAPEA)
      CALL USER (NAMEB,OCB,SIGOB,XYZB,SIGXB,BB,SIGBB,LB,NTAPEB)
      GO TO 180
C  READ PARAMETERS FROM END OF FINGER FOURIER TAPE
C  SKIP OVER REFLECTION SET 1 IF NECESSARY
160   IF (NFTEST.EQ.3) GO TO 170
      CALL FINGFO (JA,LOGFC,DM,JJA,KKA,LLA,NFTEST,IDA,IJMINA,IJMAXA,NTAP
     2EA,JZ)
C  SKIP OVER REFLECTION SET 2 IF NECESSARY (HAVE TO DO SAME AS FOR
C  SET 1)
      CALL FINGFO (JB,LOGFC,DM,JJB,KKB,LLB,NFTEST,IDB,IJMINB,IJMAXB,NTAP
     2EB,JZ)
170   CONTINUE
C  READ ATOM SET 1.
      CALL INFING (NAMEA,OCA,SIGOA,XYZA,SIGXA,BA,SIGBA,LA,NTAPEA)
C  READ ATOM SET 2
```

```
348             CALL INFING (NAMEB,OCB,SIGOB,XYZB,SIGXB,BB,SIGBB,LB,NTAPEB)
349         C
350         C
351         C     ******************************************************************
352         C
353         C  PRELIMINARY CHECK ON ATOM SETS-EQUAL NUMBERS OF ATOMS EXPECTED AT
354         C  FIRST SIGHT.
355         180   IF (LA.NE.LB) WRITE (NOUT,490) LA,LB
356         C CHECK THERE ARE AT LEAST TWO SIMILAR ATOMS IN THE 2 SETS
357         C  FIND CORRESPONDING ATOMS IN THE 2 ATOM SETS
358             CALL MTCHEK (NAMEA,LA,NAMEB,LB,MATCH,IP,MM)
359         C  TEST FOR MORE DATA IF LESS THAN 3 ATOMS IN COMMON BETWEEN ATOMIC SETS
360             IF (IP.LE.2) GO TO 10
361         C THERE MUST BE AT LEAST TWO EQUVALENT ATOMS IF WE GET THIS FAR
362         C CALCULATE DP VALUES AND ASSOCIATED QUANTITIES FOR OCCUPANCY,POSITIONAL
363         C PARAMETERS AND THERMAL PARAMETERS SEPARATELY.
364         C CHECK WHETHER STANDARD DEVIATIONS ARE GREATER THAN ZERO TO SEE WHICH
365         C OF THESE QUANTITIES HAVE BEEN VARIED.
366         C
367         C
368         C     ******************************************************************
369         C
370         C  HALF-NORMAL PROBABILITY PLOTS DONE IN FOLLOWING SECTION.
371         C DO OCCUPANCIES HERE.
372             NWHAT=0
373         C  TELLS SUBROUTINE PLOTEM HOW TO MAKE REST OF TITLE FOR THIS PART
374         C  NO SPECIAL REFLECTION CLASSES
375             M=0
376             II=0
377         C  FIX UP TITLE
378             ATITLE(1)=6H DM QU
379             ATITLE(2)=6HANTITI
380             ATITLE(3)=6HES BAS
381             ATITLE(4)=6HED ON
382             ATITLE(5)=6HOCCUPA
383             ATITLE(6)=6HNCIES
384             ATITLE(12)=6HST 1/2
385             ATITLE(13)=6HNORMAL
386             ATITLE(14)=6HQUANT.
387         C  TEST FOR ZERO IN FLOATING POINT.
388             TEST=.000001
389             WRITE (NOUT,500)
390         C  CHECK STANDARD DEVIATIONS OF OCCUPANCY TO SEE IF THEY ARE NON ZERO
391             JC=0
392             DO 190 I=1,MMM
393         C  LOG WILL TELL PRYNTT SUBROUTINE WHICH PARAMETERS ON WHICH ATOMS HAVE
394         C  WHICH DM VALUES
395         190   LOG(I)=0
396         C  TELL PRYNTT SUBROUTINE THESE ARE OCCUPANCIES
397             KIND=1
398             DO 200 I=1,IP
399         C GET THE TWO ATOMIC CATALOGUE NUMBERS.
400             J=MATCH(I)/100
401             K=MATCH(I)-J*100
402         C  TEST FOR NON-ZERO SIGMAS
403             IF(SIGOA(J).LT.TEST.OR.SIGOB(K).LT.TEST) GO TO 200
404         C  CALCULATE DM VALUE IF APPROPRIATE
405             CALL DPCALC (D,OCA(J),OCB(K),SIGOA(J),SIGOB(K))
```

```
406             JC=JC+1
407         C  STORE IT AWAY
408             DM(JC)=D
409         C  STORE WHERE IT CAME FROM
410             LOG(JC)=MATCH(I)*100+1
411         200   CONTINUE
412         C  CONTINUE PROCESS IF THERE ARE ENOUGH PARAMETERS FOUND
413         210   IF (JC.GE.2) GO TO 220
414             WRITE (NOUT,510) JC
415         C  GO TO NEXT STAGE
416             GO TO 240
417         C  ARRAYS X AND IDB ARE DUMMYS AT PRESENT.
418         220   CALL SORT (DM,X,LOG,IDB,JC,NN)
419         C CALCULATE HALF NORMAL PROBABILITY DISTRIBUTION FOR THESE PARAMETERS.
420         C  GET APPROXIMATION TO MEDIAN OF I-TH ORDER STATISTIC
421             CALL UNIMED (JC,X)
422         C
423         C  GET EXPECTED VALUES VIA PERCENTAGE POINTS AND MEDIAN.
424         C REFERENCE DANIEL ,TECHNUMETRICS,1959,PAGES 311-341
425             DO 230 LS=1,JC
426             Q=X(LS)
427             Q=(1.0-Q)/2.0
428             X(LS)=PINV(Q)
429         230   CONTINUE
430         C  PRINT OUT ALL DM VALUES FOR ATOMIC PARAMETERS
431             NPRINT=1
432             IF (NPRINT.GT.0) CALL PRYNTT (JC,DM,X,LOG,NPRINT,KIND,OCA,OCB,SIGO
433            2A,SIGOB,XYZA,XYZB,SIGXA,SIGXB,BA,BB,SIGBA,SIGBB,NAMEA,NAMEB,NN,MM)
434         C  PLOT DM VALUES AGAINST X VALUES, ETC.
435             WRITE (NOUT,550)
436             CALL PLOTEM(JC,X,DM,DMSMAL,XSMALL,M,II,LOGHKL,NN,N)
437         C  GO TO NEXT STAGE
438         240   NWHAT=NWHAT+1
439         C
440         C  RE-INITIALLIZE
441             JC=0
442             DO 250 I=1,400
443         250   LOG(I)=0
444         C  OFF WE GO
445         C
446             GO TO (260,290,320), NWHAT
447         C DO POSITIONAL PARAMETERS HERE
448         C
449         260   WRITE (NOUT,520)
450         C  TELLS PRYNTT SUBROUTINE WE  ARE NOW DOING POSITIONAL PARAMETERS
451             KIND=2
452         C  FIX UP TITLE
453             ATITLE(5)=6HX,Y,Z*
454             ATITLE(6)=6HS
455         C  GET DM'S IF APPROPRIATE (NON-ZERO SIGMAS OF PARAMETERS) AND PLOT THEM
456             DO 280 I=1,IP
457             J=MATCH(I)/100
458             K=MATCH(I)-J*100
459             DO 270 L=1,3
460             IF (SIGXA(L,J).LT.TEST.OR.SIGXB(L,K).LT.TEST) GO TO 270
461             CALL DPCALC (D,XYZA(L,J),XYZB(L,K),SIGXA(L,J),SIGXB(L,K))
462             JC=JC+1
463             DM(JC)=D
```

```
464             LOG(JC)=MATCH(I)*100+L+1
465       270   CONTINUE
466       280   CONTINUE
467             GO TO 210
468       C
469       C DO THERMAL PARAMETERS HERE(ASSUMED ANISOTROPIC)
470       C
471       290   WRITE (NOUT,530)
472       C  THERMAL PARAMETERS NOW, O SUBROUTINE PRYNTT(PRYNTTE, 3RD DECLENSION
473       C  (CAN'T WASTE 6 YEARS OF LATIN, GOT TO WORK IT IN SOMEWHERE))
474             KIND=3
475       C  FIX UP TITLE
476             ATITLE(5)=6HBETAS
477             ATITLE(6)=6H
478       C  AGAIN, GET DM VALUES, PLOT THEM.
479             DO 310 I=1,IP
480             J=MATCH(I)/100
481             K=MATCH(I)-J*100
482             DO 300 L=1,6
483             IF (SIGBA(L,J).LT.TEST.OR.SIGBB(L,K).LT.TEST) GO TO 300
484             CALL DPCALC (D,BA(L,J),BB(L,K),SIGBA(L,J),SIGBB(L,K))
485             JC=JC+1
486             DM(JC)=D
487             LOG(JC)=MATCH(I)*100+L+4
488       300   CONTINUE
489       310   CONTINUE
490             GO TO 210
491       C  FINISHED THIS PART
492       320   WRITE (NOUT,540)
493       C  CHECK FOR MORE DATA, BEGINNING WITH TITLE.
494             GO TO 30
495       C
496       C
497       C     ***************************************************************
498       C
499       330   CONTINUE
500       C  COMES HERE WHEN INPUT PARAMETERS FAULTY.
501             WRITE (NOUT,560)
502       340   STOP
503       C
504       C
505       C
506       C
507       C
508       350   FORMAT (1H0,I5,15H PLOTS POSSIBLE,I5,15H PAGES OF PLOTS)
509       360   FORMAT (1H0,49H TOO MUCH OUTPUT POSSIBLE-EXECUTION STOPPED-THINK)
510       370   FORMAT (1H1,98HLIST SUBROUTINE CHANGE TO KEEP UP TO DATE ON THE EV
511            2OLUTION OF THIS PROGRAM.THIS IS THE VERSION OF ,2A6,//9H*****UNIT,
512            3I4,52HMUST BE ASSIGNED FOR INTERMEDIATE SCRATCH FILE.*****//)
513       380   FORMAT (1H0,28HFIRST INPUT DATA SET ON UNIT,I3,6H FILE ,I5,30H SEC
514            2OND INPUT DATA SET ON UNIT,I3,6H FILE ,I5/1H0,10HSORT ORDER,3I2,8H
515            3 NFTEST=,I2,8H NPTEST=,I2,13H PRINT OPTION,I2)
516        390  FORMAT(2F10.7,1X,4F5.2)
517       400   FORMAT (1H0,34HMATCHED REFLECTIONS OUTPUT ON UNIT,I4)
518       410   FORMAT (1H0,51HABSORPTION CORRECTIONS WILL BE APPLIED TO DATA SETS
519            2)
520       420   FORMAT (1H0,66HSUBROUTINE SPEC WILL BE CALLED TO VARY SCALE TO MIN
521            2IMIZE SUM DM**2)
```

```
  430  FORMAT (1H0,47H STATISTICS VS NORMAL QUANTILES WILL BE PLOTTED/5H
 2NOC=,I2,7H NLAYO=,I3,15H OCTANT OPTION=,I3)
  440  FORMAT (1H0,62H STATISTICS VS D-STAR(2SIN(THETA)/WAVELENGTH) WILL
 2BE PLOTTED /5H NDC=,I2,7H NLAYD=,I3,15H OCTANT OPTION=,I3)
  450  FORMAT (1H0,57H STATISTICS VS CRYST.Q(F**2/SIN(2*THETA)) WILL BE P
 2LOTTED/5H NQC=,I2,7H NLAYQ=,I3,15H OCTANT OPTION=,I3)
  460  FORMAT (12A6)
  470  FORMAT (1H1,5X,12A6)
  480  FORMAT (A3,I2,8I5,I3,16I2)
  490  FORMAT (14H *** -WARNING-,I6,15H ATOMS IN SET 1,I6,9H IN SET 2)
  500  FORMAT (60H1TEST OCCUPANCIES IF THEY HAVE NON-ZERO STANDARD DEVIAT
 2IONNS)
  510  FORMAT (64H LESS THAN 2 COMMON ATOMIC PARAMETERS VARIED,GO ON TO N
 2EXT STAGE)
  520  FORMAT (27H1TEST POSITIONAL PARAMETERS)
  530  FORMAT (24H1TEST THERMAL PARAMETERS)
  540  FORMAT (23H *** JOB COMPLETED *** )
  550  FORMAT (1H1)
  560  FORMAT (1H0,57HCONTROL PARAMETER INCORRECT-CHECK INPUT-EXECUTION S
 2TOPPED)
C
       END
```

```
      SUBROUTINE ABSORB (J,K,L,AA,AB,DAA,DAB)
C
C     ****************************************************************
C
C  SUBROUTINE ABSORB CALCULATES THE ABSORPTION CORRECTIONS TO BE APPLIED
C  TO THE F VALUES IF REQURED.  IT IS SET UP FOR SPHERICAL CRYSTALS AT
C  PRESENT.
C  RECIPROCAL CELL CONSTANTS
      COMMON /G/ ASTAR,BSTAR,CSTAR,COSAST,COSBST,COSGST
C  IO UNITS
      COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
      COMMON /EXPT/ WAVE,SKALE,UR(2),DUR(2),NF
      DIMENSION A(2),DADUR(2),ANORM(2)
C  NF= FLAG FOR F OR F**2, 0 FOR F,  1 FOR F**2
C  SPHERICAL ABSORPTION CORRECTIONS FOR 1) CRYSTAL A  2) CRYSTAL B
C  DEGREE TO RADIAN CONVERSION PARAMETER
      RAD=3.1416/180.
      IF (NTIME.EQ.1) GO TO 40
      DO 10 L=1,2
      IF (DUR(L).LT.(.02*UR(L))) DUR(L)=.02*UR(L)
C  NORMALIZE THE CORRECTION TO FIRST DATA SET SO SCALE DOES NOT CHANGE
C  VERY MUCH. COMPUTE FACTOR FOR 2-THETA =0 CASE.
C  ABSORPTION CORRECTION EQUATION FROM  ROUSE,COOPER,ETAL.  ACTA CRYST.
C  (1970) A26, P.682-691.
      A(L)=EXP(-1.5108*UR(L)+.0951*UR(L)**2)
10    CONTINUE
      ANORM(1)=1.0
      ANORM(2)=A(1)/A(2)
      WRITE (NOUT,30) NF,WAVE,UR,DUR,ANORM
C SET FLAG TO GO TO STATEMENT 40 ON THE NEXT CALL.
      NTIME=1
   40 CONTINUE
      FJ=J
      FK=K
      FL=L
C  CALCULATE D SPACING AND THEN 2THETA FOR J,K,L REFLECTION
      TEMPTH=(FJ*ASTAR)**2+(FK*BSTAR)**2+(FL*CSTAR)**2+2.*FK*FL*BSTAR*CS
     2TAR*COSAST+2.*FL*FJ*ASTAR*CSTAR*COSBST+2.*FJ*FK*ASTAR*BSTAR*COSGST
      D=1./SQRT(TEMPTH)
      TEMPTH=(ASIN(WAVE/(2.*D)))/RAD*2.
      TH2=TEMPTH
C  CALCULATE ABSORPTION CORRECTION VIA ROUSE EQUATION.
      FACT1=1.5108-.0315*SIN(TH2)**2
      FACT2=-.0951-.2898*SIN(TH2)**2
C  CALCULATE TRANSMISSION FACTOR FOR TWO DATA SETS.
      DO 20 L=1,2
      A(L)=EXP(-1.0*FACT1*UR(L)-1.0*FACT2*UR(L)**2)*ANORM(L)
      DADUR(L)=A(L)*(-1.0*FACT1-2.0*FACT2*UR(L))
20    CONTINUE
      AA=A(1)
      AB=A(2)
      IF (NF.NE.1) AA=SQRT(AA)
      IF (NF.NE.1) AB=SQRT(AA)
      DAA=DADUR(1)*DUR(1)
      DAB=DADUR(2)*DUR(2)
C
      RETURN
```

```
58        C
59        C
60        C
61        30     FORMAT (2X,3HFXX,I1,5HWAVE=,F10.7,7HMUR(1)=,F5.2,7HMUR(2)=,F5.2,10
62               2HDELMUR(1)=,F5.2,10HDELMUR(2)=,F5.2,22HNORMALIZATION FACTORS=,2F5.
63               33)
64        C
65              END
```

```
      SUBROUTINE ANISO (JC,LOGHKL,DMSMAL,XSMALL,DM,X,M,NN,N)
C
C     ****************************************************************
C
C  SUBROUTINE ANISO MAKES PLOTEM PLOT EACH CLASS OF REFLECTIONS IN TURN.
C  DM IS WHOLE SAMPLE POPULATION FOR ORDINATE IN PLOTS, X IS SAME FOR
C  ABSCISSA, DMSMAL IS THE SUBSET OF DM WHICH IS ACTUALLY PLOTTED,
C  XSMALL IS SAME FOR X, JC IS TOTAL NUMBER IN DM POPULATION, LOGHKL
C  IS ONE WORD FOR EACH DM WHICH TELLS WHICH CLASSES THE REFLECTION
C  FALLS IN .
      DIMENSION DM(NN),X(NN),DMSMAL(N,5),XSMALL(N,5),LOGHKL(NN)
C  M IS FLAG INDICATING  CLASS TYPE
C  SET UP THE BIT NUMBER ,II, WHICH GOVERNS CATALOGUE LOOK-UP FOR
C   REFLECTION CLASS.
C  MAKE SURE M.GT.0.
      IF (M.LT.0) RETURN
C  NON-ZERO VALUE OF M TELLS REFLECTION PLOTTING AND CURVE FITTING
C  ROUTINES TO DIVIDE REFLECTIONS UP INTO CLASSES.
      IBEG=0
      ISTOP=10
      II=0
C  SET OPTION SWITCH =0 FOR PLOTTING ALL OCTANTS TOGETHER IN ONE PLOT
C  PER CLASS.
      NOCT=0
C  UNPACK M TO ESTABLISH WHAT WILL BE PLOTTED.
C  GET CLASS OPTION
      MT=M/1000
C  GET NUMBER OF LAYERS TO BE PLOTTED.
      MLAY=(M-MT*1000)/10
C  GET OCTANT OPTION WHICH IS 0,1,2,3,4, OR 7
      MOCT=(M-MT*1000-MLAY*10)
C  DO STANDARD CLASSES FIRST, IF WANTED.
      IF (MT.EQ.1.OR.MT.GE.5) GO TO 20
10    CONTINUE
C  SET UP MT VARIABLE FOR LAYER TYPE,1=STD CLASSES,2=H,3=K,4=L LAYERS.
      IF (MT.GE.5) MT=MT-3
C  SET UP RANGE OF BITS TO BE CHECKED.
      IBEG=11
      ISTOP=IBEG+MLAY
C  LOOP THROUGH CLASSES AND  LAYERS SETTING BIT AND OCTANT NUMBERS.
C
20    CONTINUE
C   INITIALIZE II
      II=IBEG
C    INITIALIZE OCTANT NUMBER IF SEPARATE OCTANTS ARE TO BE CONSIDERED.
      IF (MOCT.GT.0) NOCT=1
30    CONTINUE
C  II IS THE REFLECTION CLASS BEING CURVE-FITTED AND PLOTTED
C  CALL PLOTTING AND CURVE FITTING ROUTINES
C  VARIABLE MCLT TELLS LAYER TYPE AND OCTANT WANTED.
      MCLT=NOCT*100+MT
      CALL PLOTEM (JC,X,DM,DMSMAL,XSMALL,MCLT,II,LOGHKL,NN,N)
      IF (MOCT.LE.0) GO TO 50
C  0 TH OPTION SPECIFIES ALL OCTANTS.
      IF (NOCT.EQ.8) GO TO 50
C  INCREMENT OCTANT NUMBER
40    NOCT=NOCT+1
```

```
58      C  CHECK IF OCTANT OPTION SPECIFIES THIS OCTANT.
59            IF (MOCT.EQ.1.AND.NOCT.LT.8) GO TO 40
60      C  FIRST OPTION SPECIFIES +H,+K,+L (OCTANT 1) AND -H,-K,-L (OCTANT 8)
61            IF (MOCT.EQ.2.AND.(NOCT.GT.4.AND.NOCT.LT.8)) GO TO 40
62            IF (MOCT.EQ.2.AND.NOCT.EQ.2) GO TO 40
63      C  SECOND OPTION SPECIFIES OCTANTS 1,3 AND 8.
64            IF (MOCT.EQ.3.AND.(NOCT.GT.3.AND.NOCT.LE.7)) GO TO 40
65      C  THIRD OPTION SPECIFIES OCTANTS 1,2,3 AND 8.
66            IF (MOCT.EQ.4.AND.(NOCT.EQ.4.OR.NOCT.EQ.6.OR.NOCT.EQ.7)) GO TO 40
67      C  FOURTH OPTION SPECIFIES ALL OCTANTS EXCEPT 4,6, AND 7
68      C  LAST OPTION SPECIFIES ALL OCTANTS.
69      C    NOW THAT WE HAVE THE CORRECT OCTANT NUMBER-PRODUCE PLOT.
70            GO TO 30
71      C
72      C     INCREMENT BIT NUMBER - BEGIN A NEW CLASS OR LAYER.
73      50    II=II+1
74      C  RESET OCTANT NUMBER TO FIRST OCTANT.
75            IF (NOCT.EQ.8) NOCT=1
76            IF (II.LT.ISTOP) GO TO 30
77            IF (MT.GE.5) GO TO 10
78            RETURN
79      C
80            END
```

```
      SUBROUTINE CATLOG (J,K,L,LOGHKL,I,LAY,NN)
C
C     ****************************************************************
C
C  SUBROUTINE CATLOG PACKS UP THE LOGHKL WORD ACCORDING TO WHICH OF THE
C  CLASSES BELOW THE REFLECTION J,K,L FALLS IN.
C  MAXIMUM AND MINIMUM VALUES OF MILLER INDICES IN THIS DATA SET
      COMMON /D/ MAXH,MINH,MAXK,MINK,MAXL,MINL
      DIMENSION LOGHKL(NN)
C SET BITS IN LOGHKL WORD DEPENDING ON WHICH REFLECTION CLASSES THE
C REFLECTION WITH INDICES J,K,L  FALLS IN.
C     BIT 0     H,0,0
C         1     0,K,0
C         2     0,0,L
C         3     H,K,0
C         4     H,0,L
C         5     0,K,L
C         6     H,K,L (H,K OR L NOT EQUAL TO ZERO)
C         7     H,H,H
C         8     H,H,L
C         9     H,K,K
C         10    H,K,H
C     BITS FOR LAYER DATA
C         11   1KL,OR H1L OR HK1 LAYER
C         12   2KL,OR H2L OR HK2 LAYER
C         13   3KL,OR H3L OR HK3 LAYER
C         ETC.
C         20   10KL,OR H10L OR HK10 LAYER
C
C     BITS  21 THROUGH 33 FOR FUTURE USE
C
C     BIT 33  0 IF H IS PLUS, 1 IF H IS MINUS
C     BIT 34  0 IF K IS PLUS, 1 IF K IS MINUS
C     BIT 35  0 IF L IS PLUS, 1 IF L IS MINUS
C     THUS  000 OR OCTAL 0 WILL BE THE +++ OCTANT.
C     LAY IS CLASS OR LAYER TYPE
C     DO STANDARD CLASSES
      IF (K.EQ.0.AND.L.EQ.0) FLD(0,1,LOGHKL(I))=1
      IF (J.EQ.0.AND.L.EQ.0) FLD(1,1,LOGHKL(I))=1
      IF (J.EQ.0.AND.K.EQ.0) FLD(2,1,LOGHKL(I))=1
      IF (L.EQ.0) FLD(3,1,LOGHKL(I))=1
      IF (K.EQ.0) FLD(4,1,LOGHKL(I))=1
      IF (J.EQ.0) FLD(5,1,LOGHKL(I))=1
      IF (J.NE.0.AND.K.NE.0.AND.L.NE.0) FLD(6,1,LOGHKL(I))=1
      IF (J.EQ.K.AND.K.EQ.L) FLD(7,1,LOGHKL(I))=1
      IF (J.EQ.K) FLD(8,1,LOGHKL(I))=1
      IF (K.EQ.L) FLD(9,1,LOGHKL(I))=1
C
C     CLASSIFY THE APPROPRIATE LAYER
      IF (LAY.LE.1) GO TO 20
      IF (LAY.EQ.2) NH=J
      IF (LAY.EQ.3) NH=K
      IF (LAY.EQ.4) NH=L
      DO 10 N=1,10
      NB=N+10
      IF (NH.EQ.N.OR.NH.EQ.-N) FLD(NB,1,LOGHKL(I))=1
10    CONTINUE
```

```
58       C
59       20     CONTINUE
60       C      ASSIGN REFLECTION TO ITS PROPER OCTANT.
61              IF (J.LT.0) FLD(33,1,LOGHKL(I))=1
62              IF (K.LT.0) FLD(34,1,LOGHKL(I))=1
63              IF (L.LT.0) FLD(35,1,LOGHKL(I))=1
64              RETURN
65       C
66              END
```

```
      SUBROUTINE CELL
C
C     ******************************************************************
C
C  SUBROUTINE TO READ THE CELL CARD, TRANSFORM THE ANGLES IF NECESSARY,
C  WRITE OUT THE REAL CELL, AND CALCULATE THE RECIPROCAL CELL
C  I/O UNITS
      COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
C  RECIPROCAL CELL PARAMETERS
      COMMON /G/ ASTAR,BSTAR,CSTAR,COSAST,COSBST,COSGST
C  PARAMETER REQUIRED TO CHANGE DEGREES TO RADIANS
      RAD=2.*3.1416/360.
C  READ CELL, TRANSFORM ANGLES IF NECESSARY,WRITE OUT CELL
      READ (IN,10) A,B,C,AL,BE,GA
C  ANGLES MAY BE DEGREES OR COSINES
      IF (AL.GT.1.) COSA=COS(AL*RAD)
      IF (AL.LE.1.) COSA=AL
      IF (AL.LE.1.) AL=ACOS(AL)/RAD
      IF (BE.GT.1.) COSB=COS(BE*RAD)
      IF (BE.LE.1.) COSB=BE
      IF (BE.LE.1.) BE=ACOS(BE)/RAD
      IF (GA.GT.1.) COSG=COS(GA*RAD)
      IF (GA.LE.1.) COSG=GA
      IF (GA.LE.1.) GA=ACOS(GA)/RAD
      WRITE (6,20) A,B,C,AL,BE,GA
      AL=AL*RAD
      BE=BE*RAD
      GA=GA*RAD
      S=(AL+BE+GA)/2.
      VOL=2.*A*B*C*SQRT(SIN(S)*SIN(S-AL)*SIN(S-BE)*SIN(S-GA))
C  CALCULATE RECIPROCAL CELL
      ASTAR=B*C*SIN(AL)/VOL
      BSTAR=C*A*SIN(BE)/VOL
      CSTAR=A*B*SIN(GA)/VOL
      COSAST=(COSB*COSG-COSA)/(SIN(BE)*SIN(GA))
      COSBST=(COSG*COSA-COSB)/(SIN(GA)*SIN(AL))
      COSGST=(COSA*COSB-COSG)/(SIN(AL)*SIN(BE))
      RETURN
C
C
C
10    FORMAT (13X,3F8.3,3F9.5)
20    FORMAT (32H THE CELL DIMENSIONS READ IN ARE,3F10.4,3F10.2)
C
      END
```

```
      SUBROUTINE DMCALC (JC,SKALE,DM,LOGFC,NN)
C
C     ****************************************************************
C
C  ROUTINE CALCULATES DM,THE 'UNIT DEVIATE'.
C   ALSO THE MEAN WHICH SHOULD BE CLOSE TO ZERO.
C   ALSO THE E.S.D. WHICH SHOULD BE CLOSE TO ONE.
C
C WE NOW HAVE JC COMMON REFLECTIONS,THEIR CATALOGUE NUMBERS ARE PACKED
C INTO THE LOGFC ARRAY.
      COMMON /IO/ IN,NOUT
C  NOW CALCULATE DM(I),S
      DIMENSION DM(NN),LOGFC(NN)
      RNUM=0.0
      RDEN=0.0
      SUMM=0.
      SUM=0.
      SS=SKALE**2
      DO 10 I=1,JC
C  FIND THE SEQUENCE NUMBERS OF THE TWO CORRESPONDING REFLECTIONS, ONE
C  IN EACH DATA SET
      J=LOGFC(I)/100000
      K=LOGFC(I)-100000*J
C  READ THEM FROM THE DRUM
      CALL DRUMRD (JHKL,FA,SGA,NT,J)
      CALL DRUMRD (JHKL,FB,SGB,NT,K)
C  CALCULATE THE DM QUANTITY
      DM(I)=(FA-SKALE*FB)/SQRT(SGA**2+SS*SGB**2)
      SUM=SUM+DM(I)**2
      SUMM=SUMM+DM(I)
      RNUM=RNUM+ABS(DM(I))
      RDEN=RDEN+FA/SGA
C CALCULATE MEAN AND ESD OF THE DM DISTRIBUTION
10    CONTINUE
      SUMMA=SUMM/JC
      ESD=0.0
      DO 20 I=1,JC
      ESD=ESD+(DM(I)-SUMMA)**2
20    CONTINUE
      ESD=SQRT(ESD/(JC-1))
      R=RNUM/RDEN
      WRITE (NOUT,30) JC,SUM,SUMM,SUMMA,ESD
      WRITE (NOUT,40) R
      RETURN
C
C
30    FORMAT (1H0,16H SUMMARY OF THE ,I5,14H DM STATISTICS//17H SUM OF D
     2M**2 IS ,E10.4,14H SUM OF DM IS ,E10.4//28H MEAN OF DM DISTRIBUTIO
     3N =  ,F8.3,27H ESD. OF DM DISTRIBUTION = ,F6.3)
40    FORMAT (1H0,6H WR = ,F8.4,51H FOR COMPARISON WITH WR OBTAINED IN
     2   REFINEMENTS)
C
      END
```

```
      SUBROUTINE DMDQ (DM,LOGFC,JC,JJ,KK,LL,X,LOGHKL,ND,NQ,NCL,NN)
C
C     ****************************************************************
C
C  SUBROUTINE TO READ EACH REFLECTION IN TURN FROM THE DRUM
C  GENERATE THE MILLER INDICES   CALCULATE THE  D VALUE AND PREPARE FOR
C  D-STAR PLOT OR Q PLOTS.
C  DM IS ORDINATE VALUES, X IS ABSCISSA, SEQUENCE NUMBERS OF
C  CORRESPONDING REFLECTIONS ARE PACKED INTO LOGFC, INFORMATION ON
C  REFLECTIONS CLASSES EACH REFLECTION BELONGS TO IS PACKED INTO LOGHKL
C  JJ,KK AND LL ARE PACKING ORDER OF MILLER INDICES IN JKL WORD, ND IS
C  A PARAMETER WHICH CONTROLS WHAT IS TO BE PLOTTED AGAINST WHAT.
C  RECIPROCAL CELL PARAMETERS
      COMMON /G/ ASTAR,BSTAR,CSTAR,COSAST,COSBST,COSGST
      COMMON /EXPT/ WAVE,SKALE,UR(2),DUR(2),NF
      DIMENSION DM(NN),LOGFC(NN),X(NN),LOGHKL(NN)
C  FIRST INITIALLIZE LOGHKL AWAY SO THAT WE HAVE ALL BITS ZERO.
      DO 10 I=1,JC
      LOGHKL(I)=0
10    CONTINUE
      DO 30 I=1,JC
C  OBTAIN INDICES OF REFLECTIONS ASSOCIATED WITH DM VALUE.
C  READ REFLECTION INDICES AND FCALC FROM THE DRUM
      LB=LOGFC(I)/100000
      LA=LOGFC(I)-LB*100000
      CALL DRUMRD (JKL,F,SG,NT,LA)
C  GENERATE THE MILLER INDICES
      CALL HKLGEN (JKL,J,K,L,JJ,KK,LL)
C  FLOAT MILLER INDICES.
      HJ=J
      HK=K
      HL=L
C  CALCULATE D-STAR VALUE IF WANTED.
      IF (ND.NE.1.AND.NQ.NE.1) GO TO 20
      TEMPTH=(HJ*ASTAR)**2+(HK*BSTAR)**2+(HL*CSTAR)**2+2.*HK*HL*BSTAR*CS
     2TAR*COSAST+2.*HL*HJ*ASTAR*CSTAR*COSBST+2.*HJ*HK*ASTAR*BSTAR*COSGST
C  STORE D VALUE IN X ARRAY WHICH IS LATER PLOTTED AGAINST DM ARRAY
      X(I)=SQRT(TEMPTH)
C  CHECK IF Q (F**2/SIN(2THETA)) VALUES ARE WANTED.
      IF (NQ.NE.1) GO TO 20
      SIN2T=.5*WAVE*X(I)
C  STORE LOG Q IN ARRAY X
C  CHECK IF DATA F OR F**2
      IF (NF.LE.0) F=F**2
      X(I)=ALOG10(F/SIN2T)
20    CONTINUE
C  CHECK IF REFLECTION CLASSES ARE TO BE PLOTTED SEPARATELY.  IF NOT,
C  RETURN.
C  OBTAIN CLASS OR LAYER TYPE
      NLAY=NCL
      IF(NLAY.LE.0) GO TO 30
      IF(NLAY.GE.5) NLAY=NLAY-3
C  CATALOG REFLECTIONS ACCORDING TO CLASS OR LAYER AND OCTANT.
      CALL CATLOG(J,K,L,LOGHKL,I,NLAY,NN)
30    CONTINUE
      RETURN
C
      END
```

```
1             SUBROUTINE DPCALC (DP,PA,PB,SA,SB)
2       C
3       C     ************************************************************************
4       C
5       C  STANDARD CALCULATION OF DP QUANTITY FOR HALF NORMAL PLOT
6             DP=ABS(ABS(PA)-ABS(PB))/SQRT(SA**2+SB**2)
7             RETURN
8       C
9             END
```

```
      SUBROUTINE DRUMRD (JHKL,F,SG,NT,I)
C
C     ************************************************************
C
C  SUBROUTINE TO READ RECORD (REFLECTION) FROM RANDOM ACCESS MASS
C  STORAGE (DISK,DRUM).
C  READS ONE RECORD PER CALL
C  MAY BE EASILY MODIFIED FOR DIFFERENT I/O.
      COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
C  NEXT TWO STATEMENTS ARE A MECHANISM TO READ BOTH FLOATING AND FIXED
C  POINT QUANTITIES FROM THE DRUM IN ONE OPERATION.
      DIMENSION LDRUM(4),FDRUM(4)
      EQUIVALENCE (LDRUM,FDRUM)
C  J POINT INDICATES LOCATION OF REFLECTION ON DRUM
      JPOINT=(I-1)*4
C  READ FROM DRUM, BUT FIRST COUNT FROM BEGINNING OF DRUM.
      CALL NTRAN (NDRUM,6,-100000,6,JPOINT,2,4,LDRUM,LOPT,22)
      JHKL=LDRUM(1)
      NT=LDRUM(2)
      F=FDRUM(3)
      SG=FDRUM(4)
      RETURN
C
      END
```

```
      SUBROUTINE DRUMRT (N,IJ,F,S)
C
C     ******************************************************************
C
C  SUBROUTINE TO WRITE REFLECTION AND RANDOM ACCESS MASS STORAGE (DISK,
C  DRUM).
C  MAY BE EASILY MODIFIED FOR OTHER TYPR I/O.
C  N IS FLAG FOR UNOBSERVED REFLECTIONS.
C  IJ ARRAY IS MILLER INDICES, F IS STRUCTURE FACTOR, S IS SIG(FO)
      COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
      DIMENSION IJ(3),LDRUM(4),FDRUM(4)
C  LDRUM AND FDRUM ARE EQUIVALENT SO THAT FIXED AND FLOATING POINT
C  QUANTITIES CAN BE WRITTEN ON THE DRUM IN ONE OPERATION, WHICH SAVES
C  TIME.
      EQUIVALENCE (LDRUM,FDRUM)
C  PACK MILLER INDICES INTO ONE WORD TO SAVE COMPUTER SPACE AND WRITING
C  TIME.
      LDRUM(1)=(IJ(2)+100)*1000+IJ(3)+100+(IJ(1)+100)*1000000
      LDRUM(2)=N
      FDRUM(3)=F
      FDRUM(4)=S
C  WRITE THE DRUM.
      CALL NTRAN (NDRUM,1,4,LDRUM,LOP,22)
      RETURN
C
      END
```

```
      SUBROUTINE FINGFO (JY,LOGFC,DM,JJ,KK,LL,NFTEST,ID,IJMIN,IJMAX,
     2 NTAPE,JZ)
C
C     ****************************************************************
C
C  SUBROUTINE TO READ REFLECTIONS FROM FINGLS FOURIER TAPE
C  FINGLS FOURIER TAPE IS WRITTEN BY PROGRAM RFINE 2.
C  RFINE 2 WRITTEN BY L.W.FINGER,GEOPHYSICAL LABORATORY,CARNEGIE
C  INSTITUTE OF WASHINGTON,D.C.
C  NTAPEA AND NTAPEB NOT IN COMMON HERE BECAUSE WE WANT TO USE SAME
C  SUBROUTINE FOR BOTH TAPES. THEY ARE TRANSFERRED IN VIA NTAPE ARGUMENT
C  IN CALL STATEMENT.
      INTEGER A
      COMMON /IO/ IN,NOUT,NDRUM
C  ID ARRAY IS ONE MILLER INDEX WHICH IS USED TO PULL REFLECTIONS OFF
C  DRUM IN BATCHES OF (SAY) CONSTANT K.  THE SORTING ON HKL GOES MUCH
C  FASTER WHEN ONLY REFLECTIONS WITH ONE INDEX THE SAME ARE SORTED AT
C  THE SAME TIME.
      PARAMETER N=50, NNA=1800, NN=7200, NNB=1801, NNC=3601, NND=5401,
     2 MM=60, MMM=400, NBUFF=700, MMMM=32
      DIMENSION ID(NN)
C  DUMMY IS USED TO OVERLAP IN CORE SOME QUANTITIES WHICH ARE NOT
C  NEEDED AT SAME TIME.
C  FBUF AND IFBUF ARE BUFFERS USED IN READING FINGLS FOURIER TAPE F,
      COMMON /H/ DUMMY(NN)
C  IJ(3) IS MILLER INDICES, DM IS ORDINATE IN PLOTS, LOGFC IS WORD
C  CONTAINING PACKED SEQUENCE NUMBERS ON DRUM OF CORRESPONDING
C  REFLECTIONS, RMAT AND TRANS ARE SYMMETRY ELEMENT COMPONENTS WHICH ARE
C  NOT NEEDED HERE.
      DIMENSION FBUF(504),IFBUF(504),IJ(3),DM(NN),LOGFC(NN),RMAT(3,3,24)
     2,TRANS(3,24)
C  EQUIVALENCES TO CONSERVE STORAGE
      EQUIVALENCE (FBUF(1),IFBUF(1),DUMMY(1),VOL,RMAT(1,1,1),
     2 TRANS(1,1))
C  READ FIRST RECORD, CONTAINING SYMMETRY ELEMENT INFORMATION, ON FINGLS
C  FOURIER TAPE.
      READ (NTAPE) NSYM,VOL,(((RMAT(I,J,K),I=1,3),J=1,3),K=1,NSYM),((TRA
     2NS(J,I),J=1,3),I=1,NSYM)
      JX=0
C  BEGIN BY READING REFLECTIONS INTO BUFFER
      GO TO 20
C  CHECK WHETHER ALL BUFFER PROCESSED.
10    IF (NFBR-500) 30,20,20
C  READ INTO BUFFER.
20    READ (NTAPE) FBUF
C  ZERO COUNTER
      NFBR=0
C  GET MILLER INDICES
30    IJ(JJ)=IFBUF(NFBR+1)
C  CHECK FOR END OF LIST, INDICATED BY J=99.
      IF (IJ(JJ)-99) 40,60,40
40    IJ(KK)=IFBUF(NFBR+2)
      IJ(LL)=IFBUF(NFBR+3)
C  FIND MAXIMUM H K AND L
      CALL MAXHKL (IJ,JJ,KK,LL)
      FOBS=FBUF(NFBR+4)
      FC=FBUF(NFBR+5)
```

```
      C  LESS THAN INDICATOR.
           A=IFBUF(NFBR+8)
      C  SIG(FO)
           SGA=FBUF(NFBR+9)
      C  EXTINCTION CORRECTION FOR THIS REFLECTION
           EXY=FBUF(NFBR+10)
           NFBR=(NFBR+10)
           DIMENSION NSAMP(20),FEXY(20)
           NEXY=(EXY+.02)*20
           IF(NEXY.LE.0) NEXY=1
           NSAMP(NEXY)=NSAMP(NEXY)+1
           NTOTAL=NTOTAL+1
      C REJECT REFLECTIONS REJECTED FOR SEVERE EXTINCTION IN RFINE.
           IF(EXY.LE..7) GO TO 10
      C  REJECT LESS-THANS
           IF (A.EQ.1) GO TO 10
      C  INCREMENT VARIOUS COUNTERS. JX IS NUMBER OF REFLECTIONS READ IN OFF
      C  THIS TAPE.  JY IS TOTAL NUMBER OF REFLECTIONS READ IN, JZ IS TO SPACE
      C  REFLECTIONS IN PAIRS ON DRUM BECAUSE FO AND FC ARE TREATED AS
      C  SEPARATE REFLECTIONS, BUT THEY ARE NOT SCALED BECAUSE IT IS ASSUMED
      C  THAT FINGLS SCALING IS MORE APPROPRIATE.
           JX=JX+1
           JY=JY+1
           JZ=JZ+2
      C  DO NOT OVERWRITE REFLECTION ARRAYS.
           IF (JX.GE.NN) GO TO 60
      C  IF TWO SEPARATE SETS OF FO'S ARE BEING READ IN, WE HAVE TO SAVE THE
      C  VALUES USED LATER IN THE SORT ROUTINE--I.E., THE MAXIMUM AND MINIMUM
      C  VALUES OF THE INDEX KEPT IN CORE.
           IF (NFTEST.NE.3) GO TO 50
      C GET MAXIMUM AND MINIMUM INDICES FOR LATER PLOTTING OF DM VALUES
      C AGAINST RANGES OF INDICES.
           CALL MAXHKL (IJ,JJ,KK,LL)
      C THE ID ARRAY LATER GIVES THE POSITION OF THE REFLECTION ON THE DRUM,
      C THE CONTENTS OF THE ID ARRAY IS THE MOST SLOWLY VARYING INDEX OF THE
      C REFLECTION.
           ID(JY)=IJ(1)
      C IJMIN AMD IJMAX GIVE THE RANGE IN THE VALUES OF THE MOST SLOWLY VARY-
      C ING INDEX.
           IF (IJ(1).LT.IJMIN) IJMIN=IJ(1)
           IF (IJ(1).GT.IJMAX) IJMAX=IJ(1)
      C  WRITE REFLECTION ON DRUM.
           CALL DRUMRT (A,IJ,FOBS,SGA)
           GO TO 10
      C  IF FO AND FC ARE BEING COMPARED, GET DM VALUE HERE.
      50   DM(JY)=(ABS(FOBS)-ABS(FC))/SGA
      C  STORE LOCATIONS OF FO AND FC ON DRUM.
           LOGFC(JY)=JZ*100000+JZ+1
      C  WRITE REFLECTION ON DRUM WITH FOBS FOR FIRST WRITE.
           CALL DRUMRT (A,IJ,FOBS,SGA)
      C WRITE SECOND REFLECTION ON DRUM OR IN THIS CASE SECOND HALF OF FIRST
      C REFLECTION
      C EXY IS EXTINCTION FACTOR. STICK IT IN PLACE OTHERWISE USED FOR SECOND
      C SIGMA(FO).
           CALL DRUMRT (A,IJ,FC,EXY)
           GO TO 10
      C  WRITE HOW MANY REFLECTIONS READ IN AND WHICH TAPE THEY ARE READ FROM
      60   WRITE (NOUT,90) NTOTAL,NTAPE,JX
```

```
116             DO 70 I=1,20
117       70    FEXY(I)=I/20.
118             WRITE (6,100)
119             WRITE (6,110) (FEXY(I),NSAMP(I),I=1,20)
120             DO 80 I=1,20
121             FEXY(I)=0.
122       80    NSAMP(I)=0
123             RETURN
124       C
125       C
126       C
127       90    FORMAT (I10,28H REFLECTIONS READ FROM UNIT ,I6,9H OF THESE,I6,63H
128            2HAD EXTINCTION COEFFICIENTS GREATER THAN 0.7 AND WERE OBSERVED/37H
129            3 ONLY THESE WERE SAVED FOR LATER USE.)
130       100   FORMAT (49H DISTRIBUTION OF EXTINCTION FACTORS IN INPUT LIST/40H
131            2  TOP OF RANGE   NUMBER OF REFLECTIONS)
132       110   FORMAT (F12.2,8X,I10)
133       C
134             END
```

```
      SUBROUTINE FITCHK (DM,X,JC,DMSMAL,XSMALL,L,K,ALINE,BLINE,ACURVE,
     2 BCURVE,CCURVE,SUMLN,SUMCV,M,II,LOGHKL,J,ACUBIC,BCUBIC,CCUBIC,
     3 DCUBIC,SUMCUB)
C
C     ******************************************************************
C
C  SUBROUTINE TO CALCULATE POINTS FOR LEAST SQUARES LINE AND CURVE AN4
C  TO ALSO CALCULATE GOODNESS OF FITS FOR L.S. LINE, QUADRATIC AND
C  CUBIC CURVES.
      PARAMETER N=50, NNA=1800, NN=7200, NNB=1801, NNC=3601, NND=5401,
     2 MM=60, MMM=400, NBUFF=700, MMMM=32
      DIMENSION DM(NN),X(NN),DMSMAL(N,5),XSMALL(N,5),LOGHKL(NN)
C
C J=1 INDICATES THAT X ARRAY CONTAINS NORMAL QUANTILES, SET FLAG NQ
      NQ=0
      IF (J.EQ.1) NQ=1
C     RESET J SO IT CAN KEEP COUNT OF THE NUMBER OF POINTS IN PLOT
      J=0
      JK=0
C  LHALF IS CONCERNED WITH PICKING THE PLOTTED POINT FROM THE MIDDLE OF
C  THE BATCH WHEN EVERY LTH POINT IS PLOTTED.
C  PLOTTING ROUTINES ARE SET UP FOR 50 POINTS MAXIMUM.
      LHALF=L/2
      SUMLN=0
      SUMCV=0
      SUMCUB=0
      SNEG=0.0
      SPOS=0.0
      NPOS=0
      NEG=0
      NIN=0
C  LOOK OVER ALL QUANTITIES--REFLECTIONS, PARAMETERS, ETC.
      DO 80 I=1,JC
C   LEAVE OUT TAILS OF DISTRIBUTION IF X ARRAY CONTAINS NORMAL QUANTILES
      IF (NQ.NE.1) GO TO 10
      IF (ABS(X(I)).GT.2.0) GO TO 80
10    CONTINUE
C  PULL OUT OCTANT NUMBER, 0 THROUGH 8, 0=ALL OCTANTS,1=+++,ETC.
      NOCT=M/100
      IF((M-NOCT*100).LE.0) GO TO 20
C  IF ONLY PARTIAL DATA IS TO BE PLOTTED, SELECT DATA HERE
C  THIS PULLS OUT THE APPROPRIATE REFLECTIONS BASED ON THE BITS PLACED
C  IN THE LOGHKL WORD BY SUBROUTINE CATLOG
      NTEST=FLD(II,1,LOGHKL(I))
      NTEST=ABS(NTEST)
      IF (NTEST.EQ.0) GO TO 80
20    CONTINUE
C  CHECK IF REFLECTION BELONGS TO OCTANT WANTED
      IF(NOCT.LE.0) GO TO 25
      NTEST=FLD(33,3,LOGHKL(I))
      NTEST=ABS(NTEST)
      IF((NOCT-1).NE.NTEST) GO TO 80
   25 CONTINUE
C  CALCULATE POINTS ON LEAST SQUARES LINES
      D=ALINE+BLINE*X(I)
C  CALCULATE POINTS ON LEAST SQUARES CURVES
      E=ACURVE+BCURVE*X(I)+CCURVE*X(I)**2
```

```
58            IF (K.LT.4) GO TO 30
59      C  CALCULATE POINTS ON L.S. CUBIC CURVE IF ENOUGH POINTS IN PLOT.
60            F=ACUBIC+BCUBIC*X(I)+CCUBIC*X(I)**2+DCUBIC*X(I)**3
61      C  SUM UP FOR GOODNESS OF FIT PARAMETER.
62            SUMCUB=SUMCUB+(F-DM(I))**2
63      30    CONTINUE
64            SUMLN=SUMLN+(D-DM(I))**2
65            SUMCV=SUMCV+(E-DM(I))**2
66      C  COPY ONLY  REQUIRED POINTS  INTO PLOTTING ARRAYS
67      C CALC QUANTITIES FOR INDICATION OF DM SCATTER.
68      C FIRST CHECK IF NECESARY CONDITIONS ARE MET.
69      C OMIT IF NORMAL QUANTILES OR LESS THAN 6 POINTS IN INTERVAL
70            IF (L.LT.6.OR.NQ.EQ.1) GO TO 70
71      C CALC DIFFERENCE FOR POINT ON LS LINE
72            DEL=DM(I)-D
73            IF (DEL) 40,50,50
74      C SUMM UP NEGATIVE DEVIATIONS
75      40    SNEG=SNEG+DEL
76            NEG=NEG+1
77            GO TO 60
78      C SUMM UP POSITIVE DEVATIONS
79      50    SPOS=SPOS+DEL
80            NPOS=NPOS+1
81      60    NIN=NIN+1
82            IF (NIN.LT.L) GO TO 70
83            IF(NEG.LE.0)NEG=1
84            IF(NPOS.LE.0)NPOS=1
85      C HAVE ALL POINTS IN INTERVAL NOW
86      C STORE QUANTITIES IN DSMALL ARRAY
87            DMSMAL(J,4)=SNEG/NEG
88            DMSMAL(J,5)=SPOS/NPOS
89      C RESET COUNTERS AND SUMS
90            SPOS=0.0
91            SNEG=0.0
92            NPOS=0
93            NEG=0
94            NIN=0
95      70    CONTINUE
96      C  JK COUNTS HOW MANY POINTS WE HAVE SO FAR
97            JK=JK+1
98      C  NTEST IS THE POINT IN THE MIDDLE OF THIS RANGE.
99            NTEST=(JK/L)*L+LHALF
100           IF (JK.NE.NTEST) GO TO 80
101     C  IF THIS IS THE POINT WE WANT, COME HERE.
102           J=J+1
103     C  STORE ORIGINAL DM VALUE IN PLOTTING ARRAYS. THESE ARE DMSMAL AND
104     C  XSMALL.
105           DMSMAL(J,1)=DM(I)
106     C  STORE VALUE FOR LINE
107           DMSMAL(J,2)=D
108     C  STORE VALUE FOR CUBIC CURVE
109           DMSMAL(J,3)=F
110     C  STORE DISTRIBUTION QUANTILE FOR PLOTTING ARRAYS. THEY HAVE TO BE TOLD
111     C  3 TIMES.
112           XSMALL(J,1)=X(I)
113           XSMALL(J,2)=X(I)
114           XSMALL(J,3)=X(I)
115           XSMALL(J,4)=X(I)
```

```
116             XSMALL(J,5)=X(I)
117       80    CONTINUE
118       C  CALCULATE GOODNESS OF FIT OF LEAST SQUARES LINE AND CURVES TO DATA.
119             SUMLN=SQRT(SUMLN/(K-2))
120             SUMCV=SQRT(SUMCV/(K-3))
121             IF (K.GT.4) SUMCUB=SQRT(SUMCUB/(K-4))
122             RETURN
123       C
124             END
```

```
      SUBROUTINE FREAD (NTAPE,ID,J,JJ,KK,LL,IJMIN,IJMAX)
C
C     ***************************************************************
C
C  SUBROUTINE TO READ THE INPUT REFLECTIONS FROM N TAPE TO WRITE THOSE R
C  NS ON THE DRUM AND TO KEEP IN STORE IN THE ID ARRAY THE SLOWLY VARYIN
C  AS A CATALOGUE NUMBER
C  MAY EASILY BE MODIFIED FOR UNFORMATTED I/O
      INTEGER A
      COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
      PARAMETER N=50, NNA=1800, NN=7200, NNB=1801, NNC=3601, NND=5401,
     2 MM=60, MMM=400, NBUFF=700, MMMM=32
      DIMENSION IJ(3),ID(NN)
      J=0
      DO 10 I=1,NN
C  READ IN REFLECTION, STOP READING ON END OF FILE.
      READ (NTAPE,30,END=20,ERR=10) A,IJ(JJ),IJ(KK),IJ(LL),F,S
C  DISREGARD ANY ZERO VALUES OF F
      IF (F.LT..001) GO TO 10
C  THROW OUT UNOBSERVED REFLECTIONS
      IF (A.EQ.2) GO TO 10
C  MAKE SURE WE DON'T GET 0,0,0, FROM BLANK CARD OR SOME OTHER GOOF.
      IF (IJ(1).EQ.0.AND.IJ(2).EQ.0.AND.IJ(3).EQ.0) GO TO 10
C  WE HAVE GOT A GOOD REFLECTION.
      J=J+1
C GET MAXIMUM AND MINIMUM INDICES FOR LATER PLOTTING OF DM VALUES
C AGAINST RANGES OF INDICES.
      CALL MAXHKL (IJ,JJ,KK,LL)
C THE ID ARRAY LATER GIVES THE POSITION OF THE REFLECTION ON THE DRUM,
C THE CONTENTS OF THE ID ARRAY IS THE MOST SLOWLY VARYING INDEX OF THE
C REFLECTION.
      ID(J)=IJ(1)
C IJMIN AMD IJMAX GIVE THE RANGE IN THE VALUES OF THE MOST SLOWLY VARY-
C ING INDEX.
      IF (IJ(1).LT.IJMIN) IJMIN=IJ(1)
      IF (IJ(1).GT.IJMAX) IJMAX=IJ(1)
C  WRITE REFLECTION ON DRUM
      CALL DRUMRT (A,IJ,F,S)
10    CONTINUE
C  READING OF REFLECTIONS COMPLETED
20    WRITE (NOUT,40) J,NTAPE
      RETURN
C
C      X-RAY BCD FORMAT
C
C
30    FORMAT (13X,I2,3I4,F10.2,F10.4)
40    FORMAT (I10,27H REFLECTIONS READ FROM TAPE,I5)
C
      END
```

```
      SUBROUTINE HKLGEN (JKL,J,K,L,JJ,KK,LL)
C
C     ****************************************************************
C
C  SUBROUTINE TO UNPACK THE MILLER INDICES
      DIMENSION IJ(3)
      IJ(1)=JKL/1000000-100
      IJ(2)=(JKL-(IJ(1)+100)*1000000)/1000-100
      IJ(3)=JKL-(IJ(1)+100)*1000000-(IJ(2)+100)*1000-100
      J=IJ(JJ)
      K=IJ(KK)
      L=IJ(LL)
      RETURN
C
      END
```

```
      SUBROUTINE INFING (NAME,OCCA,SIGO,XYZ,SIGXYZ,BBETA,SIGB,NATOM,
     2 NTAPE)
C
C     ****************************************************************
C
C  SUBROUTINE TO READ PARAMETERS INFORMATION FROM END OF FINGER  FOURIER
C  NAME IS THE ATOM NAME, TO BE WRITTEN IN A6
C  OCCA IS THE ATOM OCCUPANCY
C  SIGO IS THE STANDARD DEVIATION OF THE OCCUPANCY
C  XYZ ARE THE POSITIONAL PARAMETERS
C  SIGXYZ ARE THE STANDARD DEVIATIONS ON THE POSITIONAL PARAMETERS.
C  BBETA ARE THE ANISOTROPIC THERMAL PARAMETERS.
C  SIGB ARE THE STANDARD DEVIATIONS ON BBETA.
C  TAG IS THE FINGLS REPRESENTATION OF THE ATOM NAME.
C  ISOT, ISCAT, OCCUP AND SITE ARE CONCERNED WITH TOTAL SITE
C  OCCUPANCY, NUMBERS OF SCATTERING FACTORS, ETC. AND ARE NOT USED HERE.
      COMMON /IO/ IN,NOUT
      PARAMETER N=50, NNA=1800, NN=7200, NNB=1801, NNC=3601, NND=5401,
     2 MM=60, MMM=400, NBUFF=700, MMMM=32
      COMMON /H/ DUMMY(NN)
      DIMENSION NAME(MM),OCCA(MM),SIGO(MM),XYZ(3,MM),SIGXYZ(3,MM),
     2 BBETA(6,MM),SIGB(6,MM),TAG(2,MM),ISOT(MM),ISCAT(2,MM),OCCUP(MM),
     3 SITE(MM)
C  WRITE UNWANTED ATOM PARAMETERS ON TOP OF ONE ANOTHER
      EQUIVALENCE (ISOT(1),ISCAT(1,1),OCCUP(1),SITE(1),DUMMY(1))
C  READ END OF TAPE F AS WRITTEN BY FINGLS.  REFLECTIONS HAVE ALREADY
C  BEEN READ.
      READ (NTAPE) NATOM,((TAG(J,I),J=1,2),ISOT(I),(ISCAT(J,I),J=1,2),OC
     2CA(I),OCCUP(I),SITE(I),(XYZ(J,I),J=1,3),(BBETA(J,I),J=1,6),I=1,NAT
     3OM),(SIGO(I),(SIGXYZ(J,I),J=1,3),(SIGB(J,I),J=1,6),I=1,NATOM)
C  PACK ATOM NAME INTO ONE LOCATION
      DO 10 I=1,NATOM
      FLD(0,18,NAME(I))=FLD(0,18,TAG(1,I))
      FLD(18,18,NAME(I))=FLD(0,18,TAG(2,I))
10    CONTINUE
      WRITE (NOUT,20) NTAPE
      WRITE (NOUT,30) (NAME(I),OCCA(I),SIGO(I),(XYZ(J,I),SIGXYZ(J,I),J=1
     2,3),(BBETA(J,I),SIGB(J,I),J=1,6),I=1,NATOM)
      RETURN
C
C
C
20    FORMAT (36H1ATOMIC PARAMETERS AS READ FROM UNIT,I4/99H QUANTITIES
     2ARE E OCCUPANCIES, SIGMA OF OCCUPANCIES, X, SIG(X), Y, SIG(Y), Z,
     3SIG(Z) ON FIRST LINE,/55H  AND BETAS, EACH FOLLOWED BY ITS SIGMA,
     4ON SECOND LINE/55H  ORDER OF BETAS IS (1,1) (2,2) (3,3) (1,2) (1,3
     5) (2,3)/)
30    FORMAT (2XA6,2X2F9.6,6F10.6/8X12F10.7)
C
      END
```

```
      SUBROUTINE INFOFC (NTAPE,JY,JZ,NFTEST,IJMIN,IJMAX,LOGFC,DM,NN)
C
C     ****************************************************************
C
C  ROUTINE TO READ IN FOBS AND FCALC VALUES,CALCULATE THE CORRESPONDING
C  DM,WRITE OUT FOBS AND FCALC IN PAIRS ON DRUM.
C  PRESENTLY SET UP FOR READING THE X-RAY 70 SYSTEM BINARY DATA FILE.
C  X-RAY 70 SYSTEM OF COMPUTER PROGRAMS,EDITED BY J.M.STEWART,UNIVERSITY
C  OF MARYLAND, COLLEGE PARK, MARYLAND . 20742.
C
C  NTAPE IS LOGICAL UNIT FOR INPUT FILE.
C  JY IS COUNT OF REFLECTIONS READ.
C  JZ IS POINTER TO SPACE FOBS AND FCALC IN PAIRS.
C  LOGFC ARRAY HOLDS LOCATIONS OF FOBS AND FCALC ON DRUM.
C
C  REPLACE STATEMENTS BETWEEN '10 CONTINUE TO 20 CONTINUE' AND '30
C  CONTINUE TO 50 CONTINUE' FOR  YOUR OWN READ IN
C
      COMMON /IO/ IN,NOUT,NDRUM
      COMMON /H/ DUMMY(501)
      DIMENSION DM(NN),LOGFC(NN),IJ(3)
      DIMENSION IOBUF(500),FOBUF(500)
      EQUIVALENCE (IOBUF(1),FOBUF(1),DUMMY(1))
C  INITIALIZE COUNTERS
      JY=0
      JZ= -1
C
C  NEXT SECTION CONTAINS READ IN STATEMENTS FOR X-RAY 70 BINARY FILE.
C  THIS FILE CONTAINS 25 LOGICAL RECORDS, THE FIRST FOUR WORDS OF EACH
C  RECORD CHARACTERIZE THE RECORD TYPE.
C
C  WORD 1  IS COUNT OF WORDS IN THE 'PHYSICAL' RECORD (PRODUCED BY THE
C           FORTRAN WRITE).
C
C  WORD 2  NUMBER OF THE PHYSICAL RECORD BELONGING TO LOGICAL RECORD,
C           =0 IF LOGICAL RECORD = PHYSICAL RECORD.
C
C  WORD 3  IDENTIFICATION OF LOGICAL RECORD TYPE, I.E. HISTORY,CELL
C           CONSTANTS, REFLECTIONS,ETC.
C
C  WORD 4  NUMBER OF WORDS PER BLOCK IN A BLOCKED RECORD,E.G. NUMBER
C           OF WORDS PER REFLECTION IN LOGICAL RECORD 15.
C
C  BEGIN READING THE BINARY FILE.
10    CONTINUE
C
      READ (NTAPE) IOBUF1,(IOBUF(J),J=2,IOBUF1)
C  SKIP OVER FIRST 14 LOGICAL RECORDS AS REFLECTION INFORMATION IS IN
C  THE 15TH.
      IF (IOBUF(3).LT.15) GO TO 10
      IF (IOBUF(3).GT.15) GO TO 30
C  INIALIZE UNBLOCKING POINTER,AND GET NUMBER OF WORDS IN BLOCK.
      IPT=4
      NWORD=IOBUF(4)
C  UNBLOCK THE REFLECTION RECORDS FROM THE BUFFER.
20    CONTINUE
      IJ(1)=IOBUF(IPT+1)
```

```
58              IJ(2)=IOBUF(IPT+2)
59              IJ(3)=IOBUF(IPT+3)
60              JCODE=IOBUF(IPT+9)
61              FOBS=FOBUF(IPT+12)
62              SIG=FOBUF(IPT+13)
63              FCALC=FOBUF(IPT+14)
64              TBAR=FOBUF(IPT+21)
65        C  TBAR IS CARRIED ALONG FOR CHECKING POSSIBLE EXTINCTION EFFECTS.
66        C
67        C  STATEMENTS FROM '20 CONTINUE TO 30 CONTINUE' PROVIDE COMPATABILITY
68        C  WITH OTHER ROUTINES IN STATUS.
69        C
70        C  INCREMENT REFLECTION COUNT
71              JY=JY+1
72              JZ=JZ+2
73        C  DO NOT OVER WRITE ARRAYS
74              IF (JY.GT.NN) GO TO 30
75        C  STORE LOCATIONS OF FO AND FC ON THE DRUM
76              LOGFC(JY)=JZ*100000+JZ+1
77        C  CALCULATE DM VALUE
78              DM(JY)=(FOBS-FCALC)/SIG
79        C  WRITE REFLECTION ON DRUM
80              CALL DRUMRT (JCODE,IJ,FOBS,SIG)
81              CALL DRUMRT (JCODE,IJ,FCALC,TBAR)
82              IPT=IPT+NWORD
83        C  CHECK IF MORE REFLECTIONS IN RECORD
84              IF (IPT.LT.IOBUF1) GO TO 20
85        C  CHECK IF MORE PHYSICAL RECORDS IN LOGICAL RECORD 15.
86              IF (IOBUF(2).NE.0) GO TO 10
87        30    CONTINUE
88        C  OUTPUT SUMMARY OF REFLECTION READ-IN
89              WRITE (NOUT,40) JY,NTAPE
90              RETURN
91        C
92        C
93        40    FORMAT (I10,27H REFLECTIONS READ FROM UNIT,I5)
94        C
95              END
```

```
      SUBROUTINE LSFIT (Y,X,J,ALINE,BLINE,M,II,LOGHKL,K,ACURVE,BCURVE,
     2 CCURVE,ACUBIC,BCUBIC,CCUBIC,DCUBIC)
C
C     ****************************************************************
C
C  SUBROUTINE TO CALCULATE EQUATIONS OF LEAST SQUARES LINE, AND
C  QUADRATIC AND CUBIC CURVES WHICH RELATE VARIABLES X AND Y.  LOGHKL
C  TELLS WHICH REFLECTIONS ARE TO BE INCLUDED IF REFLECTIONS ARE BEING
C  DONE BY CLASS.
C  VARIABLE M WILL BE ZERO IF NO CLASSES.
      COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
      PARAMETER N=50, NNA=1800, NN=7200, NNB=1801, NNC=3601, NND=5401,
     2 MM=60, MMM=400, NBUFF=700, MMMM=32
      DIMENSION Y(NN),X(NN),LOGHKL(NN)
C K=1 WHEN X ARRAY CONTAINS NORMAL QUANTILES, SET FLAG NQ
      NQ=0
      IF (K.EQ.1) NQ=1
C  RESET K TO ZERO SO IT CAN COUNT THE NUMBER OF DM IN THE FIT.
      K=0
C  INITIALLIZE
      SIGY=0
      SIGYX=0
      SIGYXX=0
      SIGYX3=0
      SIGX=0
      SIGXX=0
      SIGX3=0
      SIGX4=0
      SIGX5=0
      SIGX6=0
      ACUBIC=0
      BCUBIC=0
      CCUBIC=0
      DCUBIC=0
      DO 30 I=1,J
C  LEAVE TAILS OUT OF DISTRIBUTION IF X ARRAY CONTAINS NORMAL QUANTILES
      IF (NQ.NE.1) GO TO 10
      IF (ABS(X(I)).GT.2.0) GO TO 30
10    CONTINUE
C  SKIP TO STATEMENT 20 IF ALL REFLECTIONS ARE TO BE INCLUDED
C  PULL OUT OCTANT NUMBER, 0 THROUGH 8, 0=ALL OCTANTS, 1=+++,ETC.
      NOCT=M/100
C  CHECK IF CLASSES OR LAYERS
      IF((M-NOCT*100).LE.0) GO TO 20
C
C  IF ONLY SELECTED REFLECTIONS ARE TO BE INCLUDED, FILTER THEM OUT HERE
      NTEST=FLD(II,1,LOGHKL(I))
      IF (NTEST.EQ.0) GO TO 30
20    CONTINUE
C  CHECK IF REFLECTION BELONGS TO OCTANT IF SPECIFIED
      IF(NOCT.LE.0) GO TO 25
      NTEST =FLD(33,3,LOGHKL(I))
      NTEST = ABS(NTEST)
      IF((NOCT-1).NE.NTEST) GO TO 30
25    CONTINUE
C  SUM UP QUANTITIES FOR LINE, QUADRATIC AND CUBIC.
      XX=X(I)**2
```

```
      SIGY=SIGY+Y(I)
      SIGYX=SIGYX+Y(I)*X(I)
      SIGYXX=SIGYXX+Y(I)*XX
      SIGYX3=SIGYX3+Y(I)*XX*X(I)
      SIGX=SIGX+X(I)
      SIGXX=SIGXX+XX
      SIGX3=SIGX3+XX*X(I)
      SIGX4=SIGX4+XX*XX
      SIGX5=SIGX5+XX*XX*X(I)
      SIGX6=SIGX6+(XX*X(I))**2
C  COUNT HOW MANY IN SUMMATION
      K=K+1
30    CONTINUE
C  MAKE SURE WE HAVE ENOUGH
      IF (K.LT.3) RETURN
      DIMENSION A(6,6),C(6),B(6)
C  LS LIN5
      C(1)=SIGY
      C(2)=SIGYX
      A(1,1)=K
      A(1,2)=SIGX
      A(2,1)=SIGX
      A(2,2)=SIGXX
      CALL SIMLEQ (2,A,C,B)
      ALINE=B(1)
      BLINE=B(2)
C  SOLVE FOR LS QUADRATIC, REMEMBER WE HAVE SOME QUANTITIES ALREADY
C  STORED FOR LINE CALCULATION.
      C(3)=SIGYXX
      A(1,3)=SIGXX
      A(2,3)=SIGX3
      A(3,1)=SIGXX
      A(3,2)=SIGX3
      A(3,3)=SIGX4
      CALL SIMLEQ (3,A,C,B)
      ACURVE=B(1)
      BCURVE=B(2)
      CCURVE=B(3)
      IF (K.GT.4) GO TO 40
      RETURN
C  SOLVE FOR LS CUBIC
40    C(4)=SIGYX3
      A(1,4)=SIGX3
      A(2,4)=SIGX4
      A(3,4)=SIGX5
      A(4,1)=SIGX3
      A(4,2)=SIGX4
      A(4,3)=SIGX5
      A(4,4)=SIGX6
      CALL SIMLEQ (4,A,C,B)
      ACUBIC=B(1)
      BCUBIC=B(2)
      CCUBIC=B(3)
      DCUBIC=B(4)
      RETURN
C
      END
```

```
      SUBROUTINE MAXHKL (IJ,JJ,KK,LL)
C
C     ****************************************************************
C
C  SUBROUTINE TO OBTAIN THE MAXIMUM AND MINIMUM VALUES AT THE MILLER I
      DIMENSION IJ(3)
      COMMON /D/ MAXH,MINH,MAXK,MINK,MAXL,MINL
      IF (IJ(JJ).LT.MINH) MINH=IJ(JJ)
      IF (IJ(JJ).GT.MAXH) MAXH=IJ(JJ)
      IF (IJ(KK).LT.MINK) MINK=IJ(KK)
      IF (IJ(KK).GT.MAXK) MAXK=IJ(KK)
      IF (IJ(LL).LT.MINL) MINL=IJ(LL)
      IF (IJ(LL).GT.MAXL) MAXL=IJ(LL)
      RETURN
C
      END
```

```
      SUBROUTINE MTCHEK (NAMEA,LA,NAMEB,LB,MATCH,KOUNT,MM)
C
C     ************************************************************
C
C  SUBROUTINE TO CHECK NUMBER OF COMMON ATOMS BETWEEN TWO ATOMIC SETS
C  ONE ATOMIC SET MAY HAVE MORE,E.G. HYDROGEN ATOMS,ETC.
      COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
C  NAMEA IS FIRST SET OF ATOM NAMES, NAMEB IS SECOND SET, MATCH IS A
C  WORD INTO WHICH THE SEQUENCE NUMBERS OF CORRESPONDING ATOMS ARE
C  PACKED.
      DIMENSION NAMEA(MM),NAMEB(MM),MATCH(MM)
      KOUNT=0
C  DO OVER BOTH ATOM SETS, LA ATOMS IN SET 1, LB IN SET 2.
      DO 20 I=1,LA
      DO 10 J=1,LB
      IF (NAMEA(I).NE.NAMEB(J)) GO TO 10
      KOUNT=KOUNT+1
C  STORE CATALOGUE NUMBER OF MATCHED ATOMS
C  I ISPOSITION OF FIRST ATOM  IN FIRST SET, J IS POSITION OF SECND  ATO
C  SECOND SET
      MATCH(KOUNT)=I*100+J
10    CONTINUE
20    CONTINUE
      WRITE (NOUT,30) KOUNT
      RETURN
C
C
C
30    FORMAT (10H THERE ARE,I4,46H ATOMS IN COMMON BETWEEN ATOMIC SETS 1
     2   AND 2/61H IF THIS IS LESS THAN 3 ATOMS,ABANDON PARAMETER CHECKI
     3NG PART)
C
      END
```

```
      FUNCTION NFACT (N)
C
C     ***************************************************************
C
C  INTEGER FUNCTION TO CALCULATE N FACTORIAL.
      NT=N-1
      NFACT=N
      DO 10 J=1,NT
      NFACT=NFACT*(N-J)
10    CONTINUE
      RETURN
C
      END
```

```
      FUNCTION PINV (Q)
C
C     ****************************************************************
C
C        INVERSE PROBABILITY FUNCTION        BY R.J.ARMS       2/27/67
C        Q BETWEEN 0. AND 1.
C        ERROR LESS THAN 4.5E-4 . SEE NBS HANDBOOK  26.2.23 .
      DATA C0,C1,C2,D1,D2,D3,E1,E2,K /2.515517,.802853,.010328,
     2 1.432788,.189269,.001308,1.E-06,1.E-35,4/
      IF (Q.LT.1.+E1.AND.Q.GT.-E1) GO TO 10
      PINV=0.
      IF (K.EQ.0) RETURN
      K=K-1
      WRITE (6,30) Q
      RETURN
C
   10 F=1.
      R=AMAX1(0.,AMIN1(1.,Q))
      IF (Q.LT..5) GO TO 20
      F=-1.
      R=1.-R
   20 A=SQRT(-2.*ALOG(AMAX1(E2,R)))
      PINV=F*(A-((C2*A+C1)*A+C0)/(((D3*A+D2)*A+D1)*A+1.))
      RETURN
C
C
C
   30 FORMAT (1H0,48HINVERSE PROBILITY FUNCTION HAS INVALID ARGUMENT=E16
     2.8,/1X,18HVALUE SET TO ZERO.)
C
      END
```

```
      SUBROUTINE PLOTEM (JC,X,DM,DMSMAL,XSMALL,M,II,LOGHKL,NN,N)
C
C     ****************************************************************
C
C  SUBROUTINE TO CALCULTE LEAST SQUARES LINE AND LEAST SQUARES QUADRATI
C  FOR DM AGAINST X AND THEN PLOT DM AGAINST X. LEAST-SQUARES
C  LINE AND CUBIC CURVE INCLUDED IN 3RD. PLOT.
      COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
C  ROUTINE SETS UP PLOT TITLES,CALLS ROUTINES TO CALCULATE LEAST-SQUARES
C  LINE, QUADRATIC AND CUBIC.
      COMMON /TITLE/ ATITLE(14)
C  DM AND X ARE ALL QUANTITIES, DMSMAL AND XSMALL ARE SUBSETS OF DM AND
C  X SELECTED FOR PLOTS, NUMBER IS NUMBER OF POINTS IN PLOT.
      DIMENSION DM(NN),X(NN),DMSMAL(N,5),XSMALL(N,5),NUMBER(5)
      DIMENSION ALAY(10)/6H   1  ,6H   2  ,6H   3  ,6H   4  ,6H   5  ,
     16H   6  ,6H   7  ,6H   8  ,6H   9  ,6H  10  /
      DIMENSION OCT(9)/6H  +++ ,6H  ++- ,6H  +-+ ,6H  +-- ,6H  -++ ,6H
     1-+- ,6H  --+ ,6H  -+- ,6H ALL  /
C  LOGHKL TELLS REFLECTION CLASSES.  IT IS NEEDED FOR TRANSFER TO FITCH
C  AND LSFIT
      DIMENSION LOGHKL(NN)
      K=0
C  GET EQUATIONS FOR LEAST-SQUARES LINE, QUADRATIC, CUBIC.
C  THESE ARE DM = ALINE + BLINE*X
C  DM = ACURVE + BCURVE*X + CCURVE*X**2
C  AND DM = ACUBIC + BCUBIC*X + CCUBIC*X**2 + DCUBIC*X**3
C
C  IN THE ARGUMENT LIST, M IS NON-ZERO IF SOME SUBSET OF POINTS IS TO
C  BE SOUGHT AND II IS THE NUMBER OF THE BIT IN THE LOGHKL WORD TO
C  EXAMINE (0=IGNORE, 1=INCLUDE) IF REFLECTION CLASSES ARE BEING PICKED
C SIGNAL =LSFIT= AND =FITCHK= IF X ARRAY HOLDS NORMAL QUANTILES.
      IF(ATITLE(12).EQ.6HST NOR.OR.(ATITLE(12).EQ.6HST 1/2))K=1
      J=K
C
C     K WILL CONTAIN NUMBER OF POINTS IN THE SET AFTER WEEDING OUT IS
C     DONE IN =LSFIT=.
C JC IS TOTAL NUMBER OF POINTS CONSIDERED.
      CALL LSFIT (DM,X,JC,ALINE,BLINE,M,II,LOGHKL,K,ACURVE,BCURVE,CCURVE
     2,ACUBIC,BCUBIC,CCUBIC,DCUBIC)
C  MAKE SURE WE HAVE ENOUGH POINTS TO PROCEED.
      IF (K.GT.4) GO TO 10
      WRITE (NOUT,240) K
      RETURN
C
10    L=K/N+1
C  CALCULATE GOODNESS OF FITS AND FILL OUT PLOTTING ARRAYS DMSMAL AND
C  XSMALL. N IS DIMENSION OF PLOTTING ARRAYS, EVERY L*TH POINT WILL BE
C  PLOTTED, J IS NUMBER OF POINTS IN FINAL PLOT, SUMLN, SUMCV AND SUMCUB
C  ARE GOODNESS OF FITS FOR LS LINE, QUADRATIC AND CUBIC,
C  RESPECTIVELY.
      CALL FITCHK (DM,X,JC,DMSMAL,XSMALL,L,K,ALINE,BLINE,ACURVE,BCURVE,C
     2CURVE,SUMLN,SUMCV,M,II,LOGHKL,J,ACUBIC,BCUBIC,CCUBIC,DCUBIC,SUMCUB
     3)
C  TELL PLOTS HOW MANY POINTS TO PLOT FOR LINE, CURVE AND ACTUAL DATA.
      DO 20 I=1,3
      NUMBER(I)=J
20    CONTINUE
```

```
58            NPT=3
59            IF(ATITLE(12).EQ.6HST 1/2) GO TO 200
60            IF (ATITLE(12).EQ.6HST NOR.OR.L.LT.6) GO TO 30
61      C SET UP FOR PLOTS INVOLVING DM, SCATTER
62            NPT=5
63            NUMBER(4)=J
64            NUMBER(5)=J
65      30    CONTINUE
66      C  THE NEXT PART GETS THE APPROPRIATE TITLE.
67            IF (M.LE.0) GO TO 180
68      C  GET OCTANT NUMBER AND CLASS OR LAYER TYPE,(1=STD CLASSES,2=H,3=K,4=L)
69            NOCT=M/100
70            LAYT=M-NOCT*100
71            IF(NOCT.EQ.0)NOCT=9
72      C  SET UP TITLE
73            ATITLE(2)=6H
74            ATITLE(3)=6H  REFL
75            ATITLE(4)=6HECTION
76            ATITLE(5)=6HS
77            ATITLE(6)=6H
78            ATITLE(7)=6H
79            IF (II.GT.10) GO TO 150
80            NTEST=II+1
81            GO TO (40,50,60,70,80,  90,100,110,120,130,  140), NTEST
82      C  FOLLOWING SECTION FOR REFLECTION CLASSES
83      40    ATITLE(1)=6HH,0,0
84            GO TO 190
85      C
86      50    ATITLE(1)=6H0,K,0
87            GO TO 190
88      C
89      60    ATITLE(1)=6H0,0,L
90            GO TO 190
91      C
92      70    ATITLE(1)=6HH,K,0
93            GO TO 190
94      C
95      80    ATITLE(1)=6HH,0,L
96            GO TO 190
97      C
98      90    ATITLE(1)=6H0,K,L
99            GO TO 190
100     C
101     100   ATITLE(1)=6HH,K,L
102           ATITLE(5)=6HH,K,AN
103           ATITLE(6)=6HD L NO
104           ATITLE(7)=6HT = 0.
105           GO TO 190
106     C
107     110   ATITLE(1)=6HH,H,H
108           GO TO 190
109     C
110     120   ATITLE(1)=6HH,H,L
111           GO TO 190
112     C
113     130   ATITLE(1)=6HH,K,K
114           GO TO 190
115     C
```

```
116       140    ATITLE(1)=6HH,K,H
117              GO TO 190
118       C  TITLES FOR LAYERS FOLLOW.
119       150    CONTINUE
120              NTEST=II-10
121              IF(LAYT.NE.4) GO TO 160
122              ATITLE(1)=6H  H  K
123              ATITLE(2)=ALAY(NTEST)
124              GO TO 190
125       C
126       160    CONTINUE
127       C  TEST IF K LAYERS WANTED
128              IF(LAYT.NE.3) GO TO 170
129              ATITLE(1)=6H    H
130              ATITLE(2)=ALAY(NTEST)
131              ATITLE(3)=6HL REFL
132              GO TO 190
133       C
134       170    CONTINUE
135       C  DO H LAYERS HERE
136              IF(LAYT.NE.2) GO TO 190
137              ATITLE(1)=ALAY(NTEST)
138              ATITLE(2)=6HK  L
139              GO TO 190
140       C
141       180    ATITLE(1)=6H ALL R
142              ATITLE(2)=6HEFLECT
143              ATITLE(3)=6HIONS I
144              ATITLE(4)=6HNCLUDE
145              ATITLE(5)=6HD
146              ATITLE(6)=6H
147              ATITLE(7)=6H
148              GO TO 200
149       C  WRITE OUT HEADING
150        190   WRITE(NOUT,260)OCT(NOCT)
151       C  WRITE OUT HEADING
152       200    WRITE (NOUT,270) ATITLE
153       C  WRITE OUT EQUATIONS OF LINE ETC, AND GOODNESS OF FITS
154              WRITE (NOUT,280) L,K,J,ALINE,BLINE,ACURVE,BCURVE,CCURVE,ACUBIC,BCU
155             2BIC,CCUBIC,DCUBIC,SUMLN,SUMCV,SUMCUB
156       C  PLOT X SMALL ARRAYS AGAINST DM SMALL ARRAYS
157              CALL PLOTS (NPT,XSMALL,DMSMAL,NUMBER,N)
158       C  TELL USER WHAT SYMBOLS IN PLOTS MEAN.
159              WRITE (NOUT,290)
160              IF (ATITLE(12).EQ.6HST NOR.OR.L.LT.6) GO TO 210
161              WRITE (NOUT,220) L
162       210    CONTINUE
163              WRITE (NOUT,230)
164       C  WRITE OUT NUMBERS USED IN PLOT.
165              WRITE (NOUT,300) ((I,DMSMAL(I,1),DMSMAL(I,2),DMSMAL(I,3),XSMALL(I,
166             21)),I=1,J)
167              RETURN
168       C
169       C
170       C
171       220    FORMAT (1H ,64H SYMBOLS L, U ARE FOR AVERAGE - AND + DEVIATIONS (A
172             2-B VALUES FOR,I5,25H POINTS IN THE INTERVAL.))
173       230    FORMAT (1H ,52H   POINT          A          B          C          D
```

```
174             2 )
175       240    FORMAT (I4,59H     DATA IN DATA SET                    === NOT ENOUGH
176             2TO PLOT)
177        260  FORMAT(1H1,A6,9HOCTANT(S),)
178       270    FORMAT (1H+,26X,14A6)
179       280    FORMAT (6H EVERY,I3,18H TH POINT IN PLOTS,I4,35H POINTS IN THIS SE
180             2T                  ,I4,20H POINTS IN THIS PLOT/32H  LEAST SQUARES LI
181             3NE       IS DM=,F9.3,3H + ,F9.3,2H*X,34H -- LEAST SQUARES QUADRATIC
182             4 IS DM=,F9.3,3H + ,F9.3,4H *X+,F9.3,6H *X**2/27H LEAST SQUARES CUB
183             5IC IS DM=,F9.3,3H + ,F9.3,6H *X + ,F9.3,9H* X**2 + ,F9.3,5H*X**3/1
184             69H GOODNESS OF FIT IS,F9.3,14H FOR LS LINE, ,F9.3,18H FOR  LS QUAD
185             7RATIC,5H AND ,F9.3,41H FOR CUBIC FIT IF MORE THAN 4 PTS IN PLOT)
186       290    FORMAT (1H ,107HA = DM(I) (ORDINATE), SYMBOL IS .  B = DM(I)- L. S
187             2. LINE, SYMBOL IS *  C = DM(I) - L. S. CUBIC, SYMBOL IS +/74H D =
188             3ABSCISSA VALUES (DSTAR, LOG F**2/SIN(2*THETA), OR QUANTILE
189             4    )
190       300    FORMAT (I10,3F10.3,E15.3)
191       C
192             END
```

```
      SUBROUTINE PLOTS (NARGS,X,Y,NRMX,NROW)
C
C     ****************************************************************
C
C  PLOTS UP TO 5 PAIRS OF COLUMNS.   THE NOTATION IS AS FOLLOWS.
C
C     X - A MATRIX HAVING UP TO 5 COLUMNS  AND  NROW ROWS
C
C     Y - A MATRIX HAVING UP TO 5 COLUMNS  AND  NROW ROWS
C
C          THE FIRST COLUMN OF Y IS PLOTTED VERSUS THE FIRST COLUMN OF X
C             SECOND          Y                          SECOND          X
C                                    ETC
C
C    NROW - THE NUMBER OF ROWS IN  X AND Y IN THEIR DIMENSION STATEMENTS
C           IN THE MAIN PROGRAM.  E.G. IF  X AND Y ARE DIMENSIONED
C           AS   X(130,4) , Y(130,4)  IN THE MAIN PROGRAM THEN SET NROW=
C
C    NARGS - THE NUMBER OF PAIRS OF COLUMNS TO BE PLOTTED.
C
C    NRMX  - A VECTOR CONTAINING THE LENGTHS OF THE DATA  TO BE
C            PLOTTED IN THE SEVERAL COLUMN PAIRS.  THAT IS,
C            NRMX(1)  IS NUMBER OF POINTS TO BE PLOTTED IN FIRST COLUMN
C            NRMX(2)                                      SECOND
C            NRMX(3)                                      THIRD
C                                ETC.
C
C  THE SYMBOLS USED ARE  FIRST  SECOND  THIRD  FOURTH  FIFTH   COLUMN PA
C                          .      *       +       L      U
C
C  IF MORE THAN ONE POINT FALLS ON THE SAME POSITION A TALLEY IS KEPT AN
C     THE NUMBER IS PRINTED.
C
C  LIMITS FOR THE PLOT ARE FIGURED OUT AUTOMATICALLY BASED ON THE SMALLE
C     AND LARGEST DATA POINTS.
C****
C****   DOES NOT CALL A NEW PAGE.
C
C**** WRITTEN BY  S. PEAVY    5/1/67
C  COMMENTS BY B.L.JOINER   6/4/69
C
C
      DIMENSION NRMX(5),X(NROW,5),Y(NROW,5)
      DIMENSION PRINT(101),XP(6),BOOL(5),IDGT(9)
      INTEGER PRINT,BLANK
      EQUIVALENCE (X0,XMIN), (X1,XMAX), (Y0,YMIN), (Y1,YMAX)
      INTEGER BOOL
      DATA BOOL(1),BOOL(2),BOOL(3),BOOL(4),BOOL(5) /1H.,1H*,1H+,1HL,
     2 1HU/ ,COLX /6HCOLUMN/ ,BLANK /1H /
      DATA IDGT(1),IDGT(2),IDGT(3),IDGT(4),IDGT(5),IDGT(6),IDGT(7),
     2 IDGT(8),IDGT(9) /1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HX/
      IPRINT=6
      X1=X(1,1)
      X0=X1
      DO 10 IX=1,NARGS
      K2=NRMX(IX)
      DO 10 I=1,K2
```

```
      IF (X1.LT.X(I,IX)) X1=X(I,IX)
      IF (X0.GT.X(I,IX)) X0=X(I,IX)
10    CONTINUE
      Y1=Y(1,1)
      Y0=Y1
      DO 20 J=1,NARGS
      K4=NRMX(J)
      DO 20 I=1,K4
      IF (Y1.LT.Y(I,J)) Y1=Y(I,J)
20    IF (Y0.GT.Y(I,J)) Y0=Y(I,J)
      WRITE (IPRINT,250)
C**** DETERMINE X AND Y INCREMENTS FOR PLOT
      YDELTA=(YMAX-YMIN)/50.
      XDELTA=(XMAX-XMIN)/100.
      YL=YMAX-YDELTA/2.
      YT=YMAX
C**** THE I LOOP CONTROLS THE 5 DIVISIONS OF THE Y ORDINATE
      DO 190 I=1,6
      L=1
C**** THE J LOOP IS FOR EACH LINE OF PRINT WITHIN THE DIVISIONS
      DO 190 J=1,10
C**** BLANK OUT PRINT BUFFER LINE.
      DO 30 K=1,101
30    PRINT(K)=BLANK
C**** THE KK INDEX IS FOR EACH CURVE.  KK LESS THAN 6.
      DO 150 KK=1,NARGS
      K4=NRMX(KK)
      K5=1
C**** THIS DETERMINES IF Y(K) VALUE IS ON THE PRESENT PRINT LINE
      DO 140 K=1,K4
      IF (Y(K,KK)-YT) 40,40,140
40    IF (Y(K,KK)-YL) 140,140,50
C**** YES. Y(K) BELONGS ON THIS PRINT LINE
C**** THEREFORE DETERMIND  WHERE ALL THE X(K5) FALL ON THE X-AXIS
50    XL=XMIN
      XT=XMIN+XDELTA/2.
      DO 130 KA=1,101
      IF (X(K5,KK)-XL) 120,60,60
60    IF (X(K5,KK)-XT) 70,120,120
70    IF (PRINT(KA)-BLANK) 90,80,90
80    PRINT(KA)=BOOL(KK)
      GO TO 140
C**** IF MORE THEN ONE POINT FALLS ON THE PRINT POSITION, TALLY THE NO.
C**** OF POINTS.
90    DO 100 KKK=1,9
      IF (PRINT(KA)-IDGT(KKK)) 100,110,100
100   CONTINUE
      PRINT(KA)=IDGT(1)
      GO TO 140
110   IF (PRINT(KA).NE.IDGT(9)) PRINT(KA)=IDGT(KKK+1)
      GO TO 140
120   XL=XT
130   XT=XT+XDELTA
140   K5=K5+1
150   CONTINUE
      YT=YL
      YL=YL-YDELTA
      GO TO (160,180), L
```

```
      160   IF (I-5) 170,170,200
      170   L=2
      C**** THIS PATH IS EXECUTED ONCE IN EVERY DIVISION OF THE Y-AXIS. EVERY
      C**** TENTH LINE, STARTING WITH ZERO LINE
            YP=YT+YDELTA/2.
            WRITE (IPRINT,220) YP,PRINT
            GO TO 190
      180   WRITE (IPRINT,230) PRINT
      C**** PRINTS LINE
      190   CONTINUE
      200   WRITE (IPRINT,220) YMIN,PRINT
      C**** LAST LINE OF PRINT OUT PLUS X VALUES ALONG X-AXIS.
            WRITE (IPRINT,250)
            XP(1)=XMIN
            XP(6)=XMAX
            XR=20.*XDELTA
            DO 210 I=2,5
      210   XP(I)=XP(I-1)+XR
            WRITE (IPRINT,240) XP
            RETURN
      C
      C
      C
      220   FORMAT (1X,E12.5,1H+,101A1,1H+)
      230   FORMAT (13X,1H-,101A1,1H-)
      240   FORMAT (6(7X,E13.5))
      250   FORMAT (14X,1H+,10(10H---------+))
      C
            END
```

```
      SUBROUTINE POSITN (NTAPE,NFILE)
C
C     ***************************************************************
C
C  SUBROUTINE TO POSITION ANY TAPE =NTAPE= WITH FILE =NFILE= READY
C  SEPARATE ROUTINE ALLOWS EASY MODIFICATION FOR LOCAL COMPUTER CONVEN-
C  TIONS.
      COMMON /IO/ IN,NOUT
C  MAYBE THE USER DOESN'T WANT TO POSITION HIS TAPE. IF SO, RETURN.
      IF (NFILE.EQ.0) RETURN
C  PUT IN THE NECESSARY STATEMENTS HERE.
      WRITE (NOUT,10) NTAPE,NFILE
      RETURN
C
C
10    FORMAT (5H TAPE,I4,19H POSITIONED TO FILE,I4)
C
      END
```

```
      SUBROUTINE PRIME
C
C     ******************************************************************
C
CC  ALLOWS USER TO SET UNITS NUMBERS FOR ADAPTION TO LOCAL COMPUTER
C  CONVENTIONS.
      COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
C  CARD READER
      IN=5
C  PRINTED OUTPUT
      NOUT=6
C NDRM IS DRUM UNIT NUMBER
      NDRUM=27
      RETURN
C
      END
```

```
      SUBROUTINE PROB (DM,X,JC,NN)
C
C     ***************************************************************
C
      DIMENSION DM(NN),X(NN)
C WE NOW HAVE JC VALUES OF DM ARRANGED IN INCREASING ORDER.COMPARE
C DISTRIBUTION OF DM,S WITH NORMAL PROBABILITY FUNCTION.
C PROBABILITY OF INDIVIDUAL DM
C  DM VALUES GO ON BOTH SIDES OF ZERO
C  FIND HALFWAY
      NTEST=JC/2
      FJC=JC
      DO 10 I=1,JC
C MAKE PROBABILITY SUITABLE FOR HALF NORMAL PLOT PROGRAM PINV(Q),SUPPLI-
C ED BY R.J.ARMS OF NBS.
      FI=I
C  CALCULATE PROBABILITY  IN FLOATING POINT ARITHEMATIC
      PROB=ABS((FJC-2.*FI+1.)/(FJC))
C  MAKE PROB SUITABLE FOR USE IN ARM'S ROUTINE
      IF (PROB.GT.1.0) PROB=1.
      Q=(1.-PROB)/2.
C  OBTAIN CORRESPONDING NORMAL QUANTILE VIA PINV(Q).
      X(I)=PINV(Q)
C CONVERT TO FULL NORMAL PLOT VALUE.
C  END TEST GOVERNS SIGN  OF X VALUE
      IF (I.LE.NTEST) X(I)=-X(I)
10    CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE PRYNT (DM,LOGFC,LOGHKL,JC,JJA,KKA,LLA,JJB,KKB,LLB,
     2 NPRINT,X,NFTEST,NN)
C
C     ***************************************************************
C
C  SUBROUTINE TO PRINT ALL DM VALUES WITH ATTACHED H,K,L AND X, OR
C  PRINT THOSE VALUES WITH/DM/ GREATER THAN 2.
      COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
      DIMENSION DM(NN),LOGFC(NN),X(NN),LOGHKL(NN)
      NOUT=20
      WRITE (NOUT,30)
      IF (NFTEST.EQ.2.OR.NFTEST.EQ.6) WRITE (NOUT,20)
      DO 10 I=1,JC
C  SELECT DM VALUES AS REQUESTED IF NPRINT=2
      IF (NPRINT.EQ.2.AND.ABS(DM(I)).LT.2.) GO TO 10
C  GET REFLECTION LOCATIONS ON DRUM
      J=LOGFC(I)/100000
      K=LOGFC(I)-100000*J
C  READ REFLECTIONS OFF DRUM.
      CALL DRUMRD (JHKLA,FA,SGA,NA,J)
      CALL DRUMRD (JHKLB,FB,SGB,NB,K)
C  FIND HKL FOR EACH REFLECTION.
      CALL HKLGEN (JHKLA,JA,KA,LA,JJA,KKA,LLA)
      CALL HKLGEN (JHKLB,JB,KB,LB,JJB,KKB,LLB)
C  WRITE OUT QUANTITIES.
      WRITE (NOUT,40) JA,KA,LA,JB,KB,LB,NA,NB,FA,FB,SGA,SGB,DM(I),X(I),J
     2,K
10    CONTINUE
      NOUT=6
      END FILE 20
      RETURN
C
C
C
20    FORMAT (1H0,63HF(2) IS FCALC AND SQ(2) GIVES EXTINCTION FACTOR, OR
     2 , TBAR,ETC.)
30    FORMAT (90H1 H1  K1  L1  H2  K2  L2 LT      F(1)     F(2)    SG(1)
     2SG(2)    DM(I)       X(I) LIST PLACES//)
40    FORMAT (6I4,2I2,2F8.3,2F8.4,2F8.3,2I6)
C
      END
```

```
      SUBROUTINE PRYNTT (JC,DM,X,LOG,NPRINT,KIND,OCA,OCB,SIGOA,SIGOB,
     2 XYZA,XYZB,SIGXA,SIGXB,BA,BB,SIGBA,SIGBB,NAMEA,NAMEB,NN,MM)
C
C     ****************************************************************
C
C  SUBROUTINE PRINTS DM VALUES TOGETHER WITH
C  CORRESPONDING PARAMETERS AND THEIR NAMES.
      COMMON /IO/ IN,NOUT
      DIMENSION DM(NN),X(NN),LOG(NN),OCA(MM),OCB(MM),SIGOA(MM),SIGOB(MM)
     2,XYZA(3,MM),XYZB(3,MM),SIGXA(3,MM),SIGXB(3,MM),BA(6,MM),BB(6,MM),
     3 SIGBA(6,MM),SIGBB(6,MM),NAMEA(MM),NAMEB(MM)
      DIMENSION PARAM(10)
      DATA PARAM(1),PARAM(2),PARAM(3),PARAM(4),PARAM(5),PARAM(6),
     2 PARAM(7),PARAM(8),PARAM(9),PARAM(10) /6HOCCUP ,6H  X   ,6H  Y   ,
     3 6H  Z   ,6HB(1,1),6HB(2,2),6HB(3,3),6HB(1,2),6HB(1,3),6HB(2,3)/
      WRITE (NOUT,60)
      DO 50 I=1,JC
      IF (NPRINT.EQ.2.AND.ABS(DM(I)).LT.2.) GO TO 50
C  FIND NUMBERS OF ATOMS ASSOCIATED WITH EACH DM VALUE
      MATCH=LOG(I)/100
C  FIND WHICH PARAMETER OF THESE ATOMS
      NPARA=LOG(I)-MATCH*100
      NA=MATCH/100
      NB=MATCH-NA*100
C  WRITE OUT PART OF LINE
      WRITE (NOUT,70) DM(I),X(I),NAMEA(NA),NAMEB(NB)
C  WRITE OUT ACTUAL PARAMETERS, BASED ON VALUE OF KIND =1 FOR OCCUPANCY
C  2 FOR POSITIONAL, 3 FOR THERMAL
      GO TO (10,20,30), KIND
10    WRITE (NOUT,80) OCA(NA),OCB(NB),SIGOA(NA),SIGOB(NB)
      GO TO 40
20    NP=NPARA-1
      WRITE (NOUT,80) XYZA(NP,NA),XYZB(NP,NB),SIGXA(NP,NA),SIGXB(NP,NB)
      GO TO 40
30    NP=NPARA-4
      WRITE (NOUT,80) BA(NP,NA),BB(NP,NB),SIGBA(NP,NA),SIGBB(NP,NB)
C  WRITE OUT HOLLERITH LITERAL TO TELL EXPLICITLY TO THE OUTSIDE WORLD
C  WHICH PARAMETER THAT DM IS FOR.
40    WRITE (NOUT,90) PARAM(NPARA)
50    CONTINUE
      RETURN
C
C
C
60    FORMAT (7X,79H    DM       X    ATOM 1  ATOM 2     PARAMETERS      ST
     2ANDARD DEVIATIONS  VARIABLE)
80    FORMAT (1H+,36X,2F8.5,5X,2F8.5)
70    FORMAT (7X,2F7.3,2(2XA6))
90    FORMAT (1H+,80X,A6)
C
      END
```

```
      SUBROUTINE REREAD (JY,LOGFC,DM,JJ,KK,LL,NTAPE,JZ,SKALE)
C
C     ****************************************************************
C  IN ORDER ON UNIT NR BY SCALE ROUTINE IN STATUS PROGRAM.
C
C   SUBROUTINE TO REREAD REFLECTION INFORMATION PREVIOUSLY WRITTEN
C  WRITING OUT AT THAT TIME AND READING IN FROM UNIT NTAPE HERE SAVES
C  THE TIME REQUIRED FOR THE SEARCH FOR MATCHES IN 2 LISTS OF REFLECTION
C  NTAPEA AND NTAPEB NOT IN COMMON HERE BECAUSE WE WANT TO USE SAME
C  SUBROUTINE FOR BOTH TAPES. THEY ARE TRANSFERRED IN VIA NTAPE ARGUMENT
C  IN CALL STATEMENT.
      COMMON /IO/ IN,NOUT,NDRUM
      PARAMETER N=50, NNA=1800, NN=7200, NNB=1801, NNC=3601, NND=5401,
     2 MM=60, MMM=400, NBUFF=490
C  DUMMY IS USED TO OVERLAP IN CORE SOME QUANTITIES WHICH ARE NOT
C  NEEDED AT SAME TIME.
      COMMON /H/ DUMMY(NN)
C  IJ(3) IS MILLER INDICES, DM IS ORDINATE IN PLOTS, LOGFC IS WORD
C  CONTAINING PACKED SEQUENCE NUMBERS ON DRUM OF CORRESPONDING
C  REFLECTIONS.  BUFF AND IBUFF ARE INPUT BUFFERS FOR THE TAPE READING
      DIMENSION BUFF(NBUFF),IBUFF(NBUFF),IJ(3),DM(NN),LOGFC(NN)
C  EQUIVALENCES TO CONSERVE STORAGE
      EQUIVALENCE (BUFF(1),IBUFF(1),DUMMY(1))
      WRITE (NOUT,60) SKALE
      SS=SKALE**2
      JX=0
      NT=0
      SUM=0.
      SUMM=0.
C  BEGIN BY READING REFLECTIONS INTO BUFFER
10    NBR=0
      READ (NTAPE) BUFF
20    CONTINUE
      IJ(JJ)=IBUFF(NBR+1)
      IF (IJ(JJ).EQ.99) GO TO 30
      IF (JX.GT.NN) GO TO 30
      IJ(KK)=IBUFF(NBR+2)
      IJ(LL)=IBUFF(NBR+3)
      FA=BUFF(NBR+4)
      FB=BUFF(NBR+5)
      SGA=BUFF(NBR+6)
      SGB=BUFF(NBR+7)
      NBR=NBR+7
C  INCREMENT VARIOUS COUNTERS. JX IS NUMBER OF REFLECTIONS READ IN OFF
C  THIS TAPE. JY IS TOTAL NUMBER OF REFLECTIONS READ IN,JZ IS TO SPACE
C  REFLECTIONS IN PAIRS ON DRUM BECAUSE FO AND FC ARE TREATED AS
C  SEPARATE REFLECTION.
      JX=JX+1
      JY=JY+1
      JZ=JZ+2
C GET MAXIMUM AND MINIMUM INDICES FOR LATER PLOTTING OF DM VALUES
C AGAINST RANGES OF INDICES.
      CALL MAXHKL (IJ,JJ,KK,LL)
C CALCULATE DM VALUE AND RECORD REFLECTION LOCATIONS ON DRUM IN LOGFC
C  WORD.
      DM(JY)=(FA-SKALE*FB)/(SQRT(SGA**2+SS*SGB**2))
      LOGFC(JY)=JZ*100000+JZ+1
```

```
58            SUM=SUM+DM(JY)**2
59            SUMM=SUMM+DM(JY)
60      C  WRITE REFLECTION ON DRUM.
61            CALL DRUMRT (NT,IJ,FA,SGA)
62            CALL DRUMRT (NT,IJ,FB,SGB)
63            IF (NBR.EQ.NBUFF) GO TO 10
64            GO TO 20
65      C  WRITE HOW MANY REFLECTIONS READ IN AND WHICH TAPE THEY WERE READ FROM
66      30    WRITE (NOUT,70) JY,NTAPE
67      C CALCULATE MEAN OF DM DISTRIBUTION
68            SUMM=SUMM/JY
69      C CALCULATE E.S.D. OF DM DISTRIBUTION.
70            SSUM=0.0
71            DO 40 JX=1,JY
72            SSUM=SSUM+(DM(JX)-SUMM)**2
73      40    CONTINUE
74            SSUM=SQRT(SSUM/(JY-1))
75            WRITE (NOUT,50) SUM,SUMM,SSUM
76            RETURN
77      C
78      C
79      C
80      50    FORMAT (1H0,13H SUM DM**2 = ,E10.4,11H MEAN DM = ,F6.3,31H E.S.D.
81           20F   DM DISTRIBUTION = ,F6.3)
82      C
83      60    FORMAT (81H  SCALE FACTOR APPLIED TO SECOND SET OF STRUCTURE FACTO
84           2RS IN SUBROUTINE REREAD IS,F10.3)
85      70    FORMAT (I10,27H REFLECTIONS READ FROM TAPE,I6)
86      C
87            END
```

```
      SUBROUTINE SCALE (IDA,JA,IDB,JB,JC,LOGFC,SKALE,IJMINA,IJMAXA,FA,FB
     2,SGA,SGB,JKLA,JKLB,KA,KB)
C
C     ***************************************************************
C
C  SUBROUTINE TO  CALCULATE THE LEAST SQUARES SCALE BETWEEN THE TWO SETS
C  REFLECTIONS
C  ALSO APPLIES CORRECTION FOR ABSORPTION IF NEEDED.
      COMMON /IO/ IN,NOUT,NDRUM,NTAPEA,NTAPEB
C  JJ,KK, AND LL TELL PACKING ORDER OF H K AND L IN JKL WORD.
C  NR, IF NON-ZERO, IS NUMBER OF UNIT SORTED VALUES ARE TO BE WRITTEN ON
C  (SAVES SORTING NEXT TIME). NABS IS NON-ZERO IF ABSORPTION
C  CORRECTIONS ARE TO BE MADE.
      COMMON /ORDER/ JJA,KKA,LLA,NR,NFILER,NABS
C  FA ETC ARE THE ARRAYS WHICH ARE ACTUALLY SORTED.  THE SORTING IS DONE
C  IN BATCHES WITH ONE CONSTANT INDEX TO SAVE TIME.
      PARAMETER N=50, NNA=1800, NN=7200, NNB=1801, NNC=3601, NND=5401,
     2 MM=60, MMM=400, NBUFF=490
      DIMENSION FA(NNA),FB(NNA),SGA(NNA),SGB(NNA),JKLA(NNA),JKLB(NNA),
     2 KA(NNA),KB(NNA)
C  IDA AND IDB ARE THE VALUES OF ONE OF THE MILLER INDICES.  THEY ARE
C  USED TO SELECT REFLECTIONS FROM THE DRUM. THE LOGFC ARRAY TELLS
C  WHERE THE SELECTED REFLECTIONS ARE ON THE DRUM.
      DIMENSION LOGFC(NN),IDA(NN),IDB(NN)
C  THIS IS A BUFFER FOR WRITING OUT ON UNIT NR IF REQUESTED.
      DIMENSION BUFF(NBUFF),IBUFF(NBUFF)
      EQUIVALENCE (BUFF,IBUFF)
      SUMA=0
      SUMB=0
      NBR=0
      JC=0
      AA=1.
      AB=1.
      IF (NR.GT.6) CALL POSITN (NR,NFILER)
C  BEGIN LOOKING FOR WHICH REFLECTIONS ARE READ AND SORTED
      NJKL=IJMINA-1
C  LOOP OVER REFLECTION BATCHES BEGINS HERE.
10    NJKL=NJKL+1
C  STOP, LOOK IF LAST HKL RANGE HAS BEEN PROCESSED ON THE COMPUTER
      IF (NJKL.GT.IJMAXA) GO TO 90
      KOUNTA=0
      KOUNTB=0
C READ REFLECTIONS OFF DRUM IN BATCHES WITH SAME SLOWLY VARYING INDEX.
C  DO FIRST REFLECTION SET HERE
      DO 20 I=1,JA
      IF (IDA(I).NE.NJKL) GO TO 20
      KOUNTA=KOUNTA+1
C  PREVENT POSSIBLE ARRAY OVERFLOW
      IF(KOUNTA.GT.NNA) GO TO 25
      CALL DRUMRD (J,FT,SGT,NT,I)
      FA(KOUNTA)=FT
      SGA(KOUNTA)=SGT
      JKLA(KOUNTA)=J
      KA(KOUNTA)=I
20    CONTINUE
C  DO SECOND REFLECTION SET HERE
 25   DO 30 I=1,JB
```

```
      IF (IDB(I).NE.NJKL) GO TO 30
      KOUNTB=KOUNTB+1
      IF(KOUNTB.GT.NNA) GO TO 35
      IJA=I+JA
      CALL DRUMRD (J,FT,SGT,NT,IJA)
      FB(KOUNTB)=FT
      SGB(KOUNTB)=SGT
      JKLB(KOUNTB)=J
      KB(KOUNTB)=IJA
30    CONTINUE
C  CHECK THAT BOTH SETS INCLUDE SOME REFLECTIONS
35    IF(KOUNTA.EQ.0.OR.KOUNTB.EQ.0) GO TO 10
C WE NOW HAVE KOUNTA REFLECTIONS IN THE FA,SGA,ETC. ARRAYS,AND KOUNTB
C REFLECTIONS IN THE CORRESPONDING FB,SGB,ARRAYS.
C TAKE EACH JKLA IN TURN AND SEARCH JKLB LIST FOR MATCH.
C ASSUME EACH REFLECTION IS UNIQUE IN EACH SET
      DO 80 I=1,KOUNTA
      DO 70 M=1,KOUNTB
      IF (JKLA(I)-JKLB(M)) 70,40,70
C  CALCULATE QUANTITIES NEEDED FOR SCALE FACTOR  ESTIMATION IF
C  REFLECTIONS MATCH.
40    INDEX=JKLA(I)
      CALL HKLGEN(INDEX,J,K,L,JJA,KKA,LLA)
      IF (NABS.LE.0) GO TO 50
C  CALCULATE ABSORPTION CORRECTIONS IF NABS IS NON-ZERO.
C  APPLY ABSORPTION CORRECTIONS
      CALL ABSORB (J,K,L,AA,AB,DAA,DAB)
C PROPAGATE ERROR IN ABSORPTION CORRECTION INTO SIGMAS
      VARA=(SGA(I)/FA(I))**2+(DAA/AA)**2
      VARB=(SGB(M)/FB(M))**2+(DAB/AB)**2
      FA(I)=FA(I)/AA
      FB(M)=FB(M)/AB
      SGA(I)=FA(I)*SQRT(VARA)
      SGB(M)=FB(M)*SQRT(VARB)
50    IF (NR.EQ.0) GO TO 60
C  FILL UP BUFFER
      IBUFF(NBR+1)=J
      IBUFF(NBR+2)=K
      IBUFF(NBR+3)=L
      BUFF(NBR+4)=FA(I)
      BUFF(NBR+5)=FB(M)
      BUFF(NBR+6)=SGA(I)
      BUFF(NBR+7)=SGB(M)
      NBR=NBR+7
      IF (NBR.LT.NBUFF) GO TO 60
C WRITE OUT BUFFER CONTAINING  70 REFLECTIONS
      WRITE (NR) BUFF
C  REZERO BUFFER COUNTER.
      NBR=0
C STORE RE-ORDERED INFORMATION FOR DM CALCULATIONS LATER.
60    JC=JC+1
      IF(JC.GT.NN) GO TO 90
C  LOGFC  CONTAINS THE POSITION OF THE REFLECTIONS ON DRUM
      LOGFC(JC)=KA(I)*100000+KB(M)
      SUMA=SUMA+FA(I)
      SUMB=SUMB+FB(M)
70    CONTINUE
80    CONTINUE
```

```
      C NOW THAT ALL REFLECTIONS FOR GIVEN SLOWLY VARYING INDEX HAVE BEEN READ
      C OFF THE DRUM AND SORTED,INCREASE THE SLOWLY VARYING INDEX BY 1 AND
      CSEE IF THERE ARE ANY MORE REFLECTIONS TO READ AND SORT. CONTINUE UNTIL
      CREFLECTIONS FOR HIGHEST VALUE OF SLOWLY VARYING INDEX HAVE BEEN READ
      C AND SORTED.
            GO TO 10
      C CHECK  IF WE DO HAVE SOME COMMON REFLECTIONS.
      90    IF (JC.GT.0) GO TO 100
            WRITE (NOUT,120)
            STOP
      C  CALCULATE THE SCALE FACTOR BETWEEN TWO SETS OF REFLECTIONS
      100   CONTINUE
            SKALE=SUMA/SUMB
            WRITE (NOUT,130) NTAPEA,NTAPEB,SKALE,JC,JA,JB
            IF (NR.EQ.0) GO TO 110
      C  SET END OF DATA FLAG (H=99) AND WRITE OUT LAST PART OF DATA ON UNIT
      C  NR.
            IBUFF(NBR+1)=99
            WRITE (NR) BUFF
            END FILE NR
      110   RETURN
      C
      C
      C
      120   FORMAT (49H STOPPED IN SCALE ROUTINE,- NO COMMON REFLECTIONS)
      130   FORMAT (46H THE SCALE FACTOR BETWEEN DATASETS 1 FROM TAPE,I6,17H A
           2ND  2 FROM TAPE,I6,3H IS,F8.3/11H THERE WERE,I6,38H REFLECTIONS CO
           3MMON  TO BOTH SETS,WITH,I6/25H REFLECTIONS IN SET 1 AND,I6,2CH REF
           4LECTIONSIN SET 2)
      C
            END
```

```
1             SUBROUTINE SIMLEQ (N,A,C,X)
2       C
3       C     ******************************************************************
4       C
5       C     GENERAL SOLUTION OF A SET OF LINEAR SIMULTANEOUS EQUATIONS
6       C     SIGMA(A(I,J)*X(J))=C(I)      BY  WILLIAM V. LOEBENSTEIN
7             DIMENSION A(6,6),B(6,6),F(6,6),X(6),XR(6),BR(6,6),D(6),DR(6),C(6)
8             D(1)=C(1)
9             DO 10 J=1,N
10      10    B(1,J)=A(1,J)
11            N1=N-1
12            DO 20 I=1,N1
13      20    F(1,I)=-A(I+1,1)/B(1,1)
14            DO 50 L=2,N
15            D(L)=C(L)
16            L1=L-1
17            DO 30 K=1,L1
18      30    D(L)=D(L)+F(K,L1)*D(K)
19            DO 40 J=L,N
20            B(L,J)=A(L,J)
21            DO 40 K=1,L1
22      40    B(L,J)=B(L,J)+F(K,L1)*B(K,J)
23            IF (L.GT.N1) GO TO 60
24            DO 50 I=L,N1
25            F(L,I)=-A(I+1,L)/B(L,L)
26            DO 50 K=1,L1
27      50    F(L,I)=F(L,I)-F(K,I)*B(K,L)/B(L,L)
28      60    DO 70 I=1,N
29            K=N+1-I
30      70    DR(K)=D(I)
31            DO 80 J=1,N
32            DO 80 I=1,J
33            K=N+1-I
34            L=N+1-J
35      80    BR(K,L)=B(I,J)
36            XR(1)=DR(1)/BR(1,1)
37            DO 90 K=2,N
38            XR(K)=DR(K)/BR(K,K)
39            K1=K-1
40            DO 90 I=1,K1
41      90    XR(K)=XR(K)-BR(K,I)*XR(I)/BR(K,K)
42            DO 100 K=1,N
43            I=N+1-K
44      100   X(I)=XR(K)
45            RETURN
46      C
47            END
```

```
      SUBROUTINE SORT (Y,YD1,YD2,YD3,N,NL)
C
C     ******************************************************************
C
C     THIS ROUTINE SORTS THE ELEMENTS OF THE INPUT VECTOR Y.
C     Y IS A SINGLE PRECISION VECTOR OF (UNSORTED) OBSERVATIONS,
C     THE INTEGER VALUE N = NUMBER OF ELEMNETS. NL IS THE
C     MAXIMUM NUMBER OF ELEMENTS DIMENSIONED FOR IN THE CALLING PROGRAM.
C     THE OUTPUT FROM THIS ROUTINE IS THE SINGLE PRECISION VECTOR Y INTO
C     THE SORTED OBSERVATIONS HAVE BEEN PLACED.
C     RESTRICTIONS ON THE MAXIMUM ALLOWABLE VALUE OF N--THE DIMENSIONS
C     OF VECTORS IU AND IL (DEFINED AND USED INTERNALLY WITHIN THIS ROUT
C     DETERMINE THE MAXIMUM ALLOWABLE VALUE OF N FOR THIS
C     ROUTINE.  IF IU AND IL EACH HAVE DIMENSION K, THEN N MAY NOT EXCEE
C     2**(K+1) - 1.  FOR THIS ROUTINE AS WRITTEN, THE DIMENSIONS OF IU A
C     HAVE BEEN SET TO 36, THUS THE MAXIMUM ALLOWABLE VALUE OF N IS
C     APPROXIMATELY 137 BILLION.  SINCE THIS EXCEEDS THE MAXIMUM ALLOWAB
C     VALUE FOR AN INTEGER VARIABLE IN MANY COMPUTERS, AND SINCE A SORT
C     BILLION ELEMENTS IS PRESENTLY IMPRACTICAL AND UNLIKELY, THEREFORE
C     TEST FOR WHETHER THE INPUT SAMPLE SIZE N EXCEEDS 137 BILLION HAS B
C     INCORPORATED INTO THIS ROUTINE.  IT IS THUS ASSUMED THAT THERE IS
C     (PRACTICAL) RESTRICTION ON THE MAXIMUM VALUE OF N FOR THIS ROUTINE
C     PRINTING--NONE UNLESS AN ERROR CONDITION EXISTS
C     THIS ROUTINE IS SINGLE PRECISION IN INTERNAL OPERATION.
C     SUBROUTINES NEEDED--NONE
C     SORTING METHOD--BINARY SORT
C     REFERENCE--CACM MARCH 1969, PAGE 186 (BINARY SORT ALGORITHM BY RIC
C                C. SINGLETON.
C               --CACM JANUARY 1970, PAGE 54.
C               --CACM OCTOBER 1970, PAGE 624.
C               --JACM JANUARY 1961, PAGE 41.
C     WRITTEN BY JAMES J. FILLIBEN, STATISTICAL ENGINEERING LABORATORY (
C     NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. 20234        JUNE 19
C
C    TIME FOR SORTING IS PROPORTIONAL TO N LN N IN CONTRAST TO N**2
C     FOR THE BUBBLE SORT.
      DIMENSION Y(1),YD1(1),YD2(1),YD3(1)
      DIMENSION IU(36),IL(36)
C
      IPR=6
C
C     CHECK THE INPUT ARGUMENTS FOR ERRORS
C
      IF (N.LT.1) GO TO 20
      IF (N.EQ.1) GO TO 30
      HOLD=Y(1)
      DO 10 I=2,N
      IF (Y(I).NE.HOLD) GO TO 40
10    CONTINUE
      WRITE (IPR,180) HOLD
      RETURN
C
20    WRITE (IPR,190)
      WRITE (IPR,210) N
      RETURN
C
30    WRITE (IPR,200)
```

```
 58            RETURN
 59      C
 60      40    CONTINUE
 61      C
 62      C  SLIGHT MODIFICATION BY L.W.SCHROEDER,(NBS,WASHINGTON,D.C.) SO THAT
 63      C   DEPENDENT ARRAYS ARE REARRANGED AS IS ARRAY Y TO PRESERVE RELATIVE
 64      C   INDEXING.
 65      C
 66      C     CHECK TO SEE IF THE INPUT VECTOR IS ALREADY SORTED
 67      C
 68            NM1=N-1
 69            DO 50 I=1,NM1
 70            IP1=I+1
 71            IF (Y(I).LE.Y(IP1)) GO TO 50
 72            GO TO 60
 73      50    CONTINUE
 74            RETURN
 75      C
 76      60    M=1
 77            I=1
 78            J=N
 79      70    IF (I.GE.J) GO TO 140
 80      80    K=I
 81            MID=(I+J)/2
 82            AMED=Y(MID)
 83      C     SAVE CORRESPONDING VALUES FOR DEPENDENT ARRAYS
 84            CORYD1=YD1(MID)
 85            CORYD2=YD2(MID)
 86            CORYD3=YD3(MID)
 87            IF (Y(I).LE.AMED) GO TO 90
 88            Y(MID)=Y(I)
 89            Y(I)=AMED
 90            AMED=Y(MID)
 91      C     EXTEND EXCHANGE TO DEPENDENT ARRAYS
 92            YD1(MID)=YD1(I)
 93            YD2(MID)=YD2(I)
 94            YD3(MID)=YD3(I)
 95            YD1(I)=CORYD1
 96            YD2(I)=CORYD2
 97            YD3(I)=CORYD3
 98            CORYD1=YD1(MID)
 99            CORYD2=YD2(MID)
100            CORYD3=YD3(MID)
101      90    L=J
102            IF (Y(J).GE.AMED) GO TO 110
103            Y(MID)=Y(J)
104            Y(J)=AMED
105            AMED=Y(MID)
106      C     EXTEND EXCHANGE TO DEPENDENT ARRAYS
107            YD1(MID)=YD1(J)
108            YD2(MID)=YD2(J)
109            YD3(MID)=YD3(J)
110            YD1(J)=CORYD1
111            YD2(J)=CORYD2
112            YD3(J)=CORYD3
113            CORYD1=YD1(MID)
114            CORYD2=YD2(MID)
115            CORYD3=YD3(MID)
```

```
116             IF (Y(I).LE.AMED) GO TO 110
117             Y(MID)=Y(I)
118             Y(I)=AMED
119             AMED=Y(MID)
120       C     EXTEND EXCHANGE TO DEPENDENT ARRAYS
121             YD1(MID)=YD1(I)
122             YD2(MID)=YD2(I)
123             YD3(MID)=YD3(I)
124             YD1(I)=CORYD1
125             YD2(I)=CORYD2
126             YD3(I)=CORYD3
127             CORYD1=YD1(MID)
128             CORYD2=YD2(MID)
129             CORYD3=YD3(MID)
130             GO TO 110
131       C
132       100   Y(L)=Y(K)
133             Y(K)=TT
134       C     SET VALUES FOR CORRESPONDING ELEMENTS IN DEPENDENT ARRAYS.
135             YD1(L)=YD1(K)
136             YD2(L)=YD2(K)
137             YD3(L)=YD3(K)
138             YD1(K)=TYD1
139             YD2(K)=TYD2
140             YD3(K)=TYD3
141       110   L=L-1
142             IF (Y(L).GT.AMED) GO TO 110
143             TT=Y(L)
144       C     SAVE VALUES FOR DEPENDENT ARRAYS
145             TYD1=YD1(L)
146             TYD2=YD2(L)
147             TYD3=YD3(L)
148       120   K=K+1
149             IF (Y(K).LT.AMED) GO TO 120
150             IF (K.LE.L) GO TO 100
151             LMI=L-I
152             JMK=J-K
153             IF (LMI.LE.JMK) GO TO 130
154             IL(M)=I
155             IU(M)=L
156             I=K
157             M=M+1
158             GO TO 150
159       C
160       130   IL(M)=K
161             IU(M)=J
162             J=L
163             M=M+1
164             GO TO 150
165       C
166       140   M=M-1
167             IF (M.EQ.0) RETURN
168             I=IL(M)
169             J=IU(M)
170       150   JMI=J-I
171             IF (JMI.GE.11) GO TO 80
172             IF (I.EQ.1) GO TO 70
173             I=I-1
```

```
174       160    I=I+1
175              IF (I.EQ.J) GO TO 140
176              AMED=Y(I+1)
177       C      SAVE CORRESPONDING VALUES FOR DEPENDENT ARRAYS
178              CORYD1=YD1(I+1)
179              CORYD2=YD2(I+1)
180              CORYD3=YD3(I+1)
181              IF (Y(I).LE.AMED) GO TO 160
182              K=I
183       170    Y(K+1)=Y(K)
184       C      REARRANGE DEPENDENT ARRAYS IN LIKE MANNER
185              YD1(K+1)=YD1(K)
186              YD2(K+1)=YD2(K)
187              YD3(K+1)=YD3(K)
188              K=K-1
189              IF (AMED.LT.Y(K)) GO TO 170
190              Y(K+1)=AMED
191       C      SET VALUES FOR CORRESPONDING ELEMENTS IN DEPENDENT ARRAYS
192              YD1(K+1)=CORYD1
193              YD2(K+1)=CORYD2
194              YD3(K+1)=CORYD3
195              GO TO 160
196       C
197       C
198       180    FORMAT (1H ,108H***** NON-FATAL DIAGNOSTIC--THE FIRST  INPUT ARGUM
199              2ENT (A VECTOR) TO THE SORT   SUBROUTINE HAS ALL ELEMENTS =,E15.8,6
200              3H *****)
201       190    FORMAT (1H ,58H FATAL ERROR - NUMBER OF ELEMENTS TO BE SORTED IS N
202              2EGATIVE)
203       200    FORMAT (1H ,64H DIAGONISTIC - ONLY ONE ELEMENT TO BE SORTED BY SOR
204              2T  SUBROUTINE)
205       210    FORMAT (1H ,35H***** THE VALUE OF THE ARGUMENT IS ,I8,6H *****)
206       C
207              END
```

```
      SUBROUTINE SPEC (JC,SKALE,DM,LOGFC,NN)
C
C     ****************************************************************
C
      DIMENSION DM(NN),LOGFC(NN),SK(11),DY(11,11),SDMS(11),SDM(11)
      DIMENSION NSIGN(11)
C  FIND THE SCALE FACTOR WHICH MAKES THE SUM OVER DM**2 A MINIMUM.
      N=5
      DX=0.006
      NTRY=1
10    IF (NTRY.EQ.2) N=11
      NF=N/2+1
      NCH=0
C
C  CALCULATE SCALE FOR POINTS ON EITHER SIDE OF TRIAL VALUE TO MAKE A
C    DIFFERENCE TABLE.
C  INITIALIZE SUMS
      DO 20 I=J,11
      SDMS(J)=0.0
20    SDM(J)=0.0
      DO 40 I=1,N
      FI=I-NF
      SK(I)=SKALE*(1.+DX*FI)
      PRINT 180, SK(I)
      CALL DMCALC (JC,SK(I),DM,LOGFC,NN)
      DO 30 J=1,JC
      SDM(I)=SDM(I)+DM(J)
30    SDMS(I)=SDMS(I)+DM(J)**2
40    CONTINUE
C  FORM FIRST DIFFERENCES (K=1) FOR DIFFERENCE TABLE.
      NN=N-1
      DO 50 J=1,NN
      DY(1,J)=SDMS(J+1)-SDMS(J)
C  CHECK TO SEE THAT SDMS GOES THRU A MINIMUM.
      NSIGN(J)=SIGN(1,DY(1,J))
      IF (NSIGN(J).NE.NSIGN(1)) NCH=1
50    CONTINUE
C  CHECK IF SIGN CHANGED, IF NOT START OVER
      IF (NCH.EQ.1) GO TO 60
      IF (NTRY.EQ.2) PRINT 190
      IF (NTRY.EQ.2) RETURN
      NTRY=NTRY+1
      GO TO 10
C
60    CONTINUE
C PRINT OUT TABLE HEADING
      PRINT 200
C  FORM K TH ORDER ADJUSTED DIFFERENCES
      DO 70 K=2,NN
      L=N-K
      DO 70 J=1,L
70    DY(K,J)=(DY(K-1,J+1)-DY(K-1,J))/NFACT(K)
C  WRITE OUT DIFFERENCES.
      DO 80 I=1,N
80    PRINT 210, SK(I),SDM(I),SDMS(I),(DY(J,I),J=1,N)
C  FIND MINIMUM VIA 0.001 STEPS IN SCALE,STARTING FROM MINIMUM CALCULATE
      SMIN=10**10
```

```
      DO 100 J=1,N
      IF (SDMS(J)-SMIN) 90,100,100
90    SMIN=SDMS(J)
      JMIN=J
100   CONTINUE
C  INTERPOLATE USING ADJUSTED DIFFERENCES VIA NEWTONS FORMULA(3RD.ORDER)
      SC=SK(JMIN)
      NTRY=0
C  TRY MAKING SCALE LARGER
      NEX=2
      SC=SK(JMIN)
      SSAVE=SK(JMIN)
110   CONTINUE
      SC=SC+.001*(-1.0**NEX)
      NTRY=NTRY+1
      MO=(SC-SK(1)+DX)/DX
      IF (MO) 170,170,120
120   IF (MO.GT.N) GO TO 170
      IF (N-3-MO) 130,140,140
130   MO=N-3
140   FMO=MO-1
      U=(SC-SK(JMIN)-FMO*DX)/DX
      SIN=(((U-2.)*DY(3,MO)+DY(2,MO))*(U-1.)+DY(1,MO))*U+SDMS(MO)
      IF (NTRY.EQ.1.AND.SIN.GT.SMIN) GO TO 150
      IF (SIN.GT.SMIN) GO TO 160
      SSAVE=SC
      SSIN=SIN
      GO TO 110
C
150   SC=SK(JMIN)
      NEX=1
      GO TO 110
C
160   PRINT 220,SKALE,SSAVE,SSIN
      SKALE=SSAVE
      CALL DMCALC (JC,SKALE,DM,LOGFC,NN)
      RETURN
C
170   PRINT 190
      RETURN
C
C
180   FORMAT (1H0,15H TRIAL SCALE = ,F10.4,25H SET BY SUBROUTINE =SPEC=)
190   FORMAT (1H0,43HSCALE MINIMIZATION FAILED,SCALE NOT CHANGED)
200   FORMAT (1H0,52H   SCALE    SUM DM   SUM DM**2            DIFFERENCE
     2S)
210   FORMAT (2X,14E9.4)
220   FORMAT (1H0,15HPREVIOUS SCALE=,F10.4,10HNEW SCALE=,F10.4,9HSUM DMS
     2Q=,E9.4)
C
      END
```

```
      SUBROUTINE UNIMED (N,X)
C
C     ******************************************************************
C
C     STATISTIC (FOR I = 1,2,...,N) FROM A UNIFORM DISTRIBUTION (ON THE
C     INTERVAL (0,1)).
C     THIS IS IDENTICAL TO THE MEDIAN OF THE BETA DISTRIBUTION WITH PARA
C     I AND N-I+1 FOR I=1,2,...,N.
C     THE INPUT TO THIS ROUTINE IS THE DESIRED INTEGER SAMPLE SIZE N
C     AND AN EMPTY SINGLE PRECISION VECTOR X (OF DIMENSION AT LEAST N) I
C     WHICH THE N GENERATED UNIFORM ORDER STATISTIC MEDIANS WILL BE PLAC
C     THE OUTPUT FROM THIS ROUTINE IS THE SINGLE PRECISION VECTOR X
C     INTO WHICH THE N GENERATED UNIFORM ORDER STATISTIC MEDIANS
C     HAVE BEEN PLACED.
C     ALL OF THE PROBABILITY PLOT ROUTINES MAKE USE OF THIS ROUTINE.
C     JUSTIFICATION AND ACCURACY OF THE ALGORITHM USED IS FOUND IN AN
C     UNPUBLISHED JJF MANUSCRIPT.
C     THERE IS NO RESTRICTION ON THE MAXIMUM VALUE OF N FOR THIS ROUTINE
C     PRINTING--NONE UNLESS AN ERROR CONDITION EXISTS
C     THIS ROUTINE IS SINGLE PRECISION IN INTERNAL OPERATION
C     SUBROUTINES NEEDED--NONE
C     REFERENCE--UNPUBLISHED JJF MANUSCRIPT
C     WRITTEN BY JAMES J. FILLIBEN, STATISTICAL ENGINEERING LABORATORY (
C     NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. 20234        JUNE 19
C
      DIMENSION X(1)
C
      AN=N
      IPR=6
C
C     CHECK THE INPUT ARGUMENTS FOR ERRORS
C
      IF (N.LT.1) GO TO 10
      IF (N.EQ.1) GO TO 20
      GO TO 30
C
   10 WRITE (IPR,50)
      WRITE (IPR,70) N
      RETURN
C
   20 WRITE (IPR,60)
   30 CONTINUE
C
      GAM=.3
      DO 40 I=1,N
      AI=I
      X(I)=(AI-GAM)/(AN-2.0*GAM+1.0)
   40 CONTINUE
      RETURN
C
C
   50  FORMAT (1H ,91H***** FATAL ERROR--THE FIRST  INPUT ARGUMENT TO THE
     2 UNIMED SUBROUTINE IS NON-POSITIVE *****)
   60  FORMAT (1H ,100H***** NON-FATAL DIAGNOSTIC--THE FIRST  INPUT ARGUM
     2ENT TO THE UNIMED SUBROUTINE HAS THE VALUE 1 *****)
   70  FORMAT (1H ,35H***** THE VALUE OF THE ARGUMENT IS ,I8,6H *****)
C
      END
```

```
      SUBROUTINE USER (NA,O,SO,X,SX,B,SB,L,NTAPE)
C
C     ****************************************************************
      PARAMETER N=50, NNA=1800, NN=7200, NNB=1801, NNC=3601, NND=5401,
     2 MM=60, MMM=400, NBUFF=700, MMMM=32
      COMMON /H/ DUMMY(NN)
      DIMENSION NA(MM),O(MM),SO(MM),X(3,MM),SX(3,MM),B(6,MM),SB(6,MM)
C PUT YOUR READ IN STATEMENTS HERE FOR SPECIAL USER INPUT OF PARAMETERS
C AND STANDARD DEVIATIONS.
C     NA = NAME OF ATOM
C     O=OCCUPANCY
C     SO = ERROR ON OCCUPANCY
C     X= POSITIONAL PARAMETERS
C     SX = ERROR ON POSITIONAL PARAMETERS
C     B= THERMAL PARAMETERS
C     SB= ERROR ON THERMAL PARAMETERS
C     L= NUMBER OF ATOMS IN LIST
C     NTAPE=UNIT WITH PARAMETERS ON IT.
C  USE EQUIVALENCE STATEMENT WITH DUMMY (1) AS IN SUBROUTINE INFING TO
C  SAVE STORAGE SPACE.
      RETURN
C
      END
```

V. Description of NBS Magnetic Tape 11

This tape contains one file of card images, one 80 character card per record, while comprise the FORTRAN program STATUS described in this report.

The file contains the STATUS main routine and 35 subroutines in the following order: ABSORB, ANISO, CATLOG, CELL, CHANGE, DMCALC, DMDQ, DPCALC, DRUMRD, DRUMRT, FINGFO, FITCHK, FREAD, HKLGEN, INFING, INFOFC, LSFIT, MAXHKL, MITCHEK, NFACT, PINV, PLOTEM, PLOTS, POSITN, PRIME, PROB, PRYNT, PRYNTT, REREAD, SCALE, SIMLEQ, SORT, SPEC, UNIMED, and USER.

The file of card images was recorded on a certified 2400 foot reel in BCD characters, 556 bpi density, even parity and in 7-track mode. The National Technical Information Service can generate tapes in other modes if so requested.

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. 76- 1102	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

Computer Programs for Structural Chemistry: STATUS. A FORTRAN Program for Statistical Analysis of Crystallographic Quantities

5. Publication Date: July, 1976

6. Performing Organization Code

7. AUTHOR(S)
LeRoy W. Schroeder and Brian Dickens

8. Performing Organ. Report No. 76-

9. PERFORMING ORGANIZATION NAME AND ADDRESS

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

10. Project/Task/Work Unit No. 3110585

11. Contract/Grant No. DE 00572

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

American Dental Association
211 East Chicago Avenue
Chicago, Illinois 60611

13. Type of Report & Period Covered

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES
The program is stored on NBS Magnetic Tape 11, available from NTIS.

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 describes a FORTRAN computer program for evaluation of (i) the results of crystallographic least-squares refinements by examination of the residuals, (ii) differences in sets of data collected by different methods from the same crystal, (iii) differences in data sets collected by the same method from different crystals of the same material, and (iv) the differences in parameters in different models representing the crystal structure of the same material.

Part of the evaluation is accomplished by plots of residuals against the expected normal distribution quantiles. Additional plots compare residuals with the independent variable, d*, and with the calculated variable $Fc^2/\sin 2\theta$. The Miller indices can be used to divide a data set into various classes and octants so that the possibility of anisotropic effects can be examined.

The program will treat up to 7200 data points in each of two experimental data sets or 1000 structural parameters, and is oriented specifically towards examining crystallographic data.

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*)

Crystallographic data; errors; probability plots; residuals; statistics; uncertainties

18. AVAILABILITY

[x] Unlimited

[ ] For Official Distribution. Do Not Release to NTIS

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

[X] Order From National Technical Information Service (NTIS) Springfield, Virginia 22151 Manuscript & tape available.

19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED

21. NO. OF PAGES 151

20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED

22. Price $6.75

USCOMM-DC 29042-P74