AAEC/E417

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS

CRYSTALLOGRAPHIC COMPUTER PROGRAMS FOR THE ELECTRON MICROSCOPIST

by

J.G. NAPIER

February 1978

ISBN 0 642 59645 X

AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT
LUCAS HEIGHTS

CRYSTALLOGRAPHIC COMPUTER PROGRAMS FOR THE ELECTRON MICROSCOPIST

by

JOHN NAPIER

ABSTRACT

A series of six main programs and eighteen associated subroutines is presented which covers most of the crystallographic computational requirements likely to arise in electron microscopy. The programs are designed to handle all crystal systems.

The programs are written in FORTRAN-4, specifically to be run on an IBM360 computer using the FORTRAN H compiler. Where this is likely to cause difficulties in portability, the problems are mentioned in the text.

National Library of Australia card number and ISBN 0 642 59645 X

The following descriptors have been selected from the INIS Thesaurus to describe the subject content of this report for information retrieval purposes. For further details please refer to IAEA-INIS-12 (INIS: Manual for Indexing) and IAEA-INIS-13 (INIS: Thesaurus) published in Vienna by the International Atomic Energy Agency.

COMPUTER CALCULATIONS; COMPUTER CODES; CRYSTALLOGRAPHY; ELECTRON MICROSCOPY; CRYSTAL STRUCTURE; FORTRAN

CONTENTS

CONTENTS (Continued)

1. INTRODUCTION

The programs and subroutines contained in this report were developed over a number of years as a self-contained set of procedures to handle almost any problem in crystallography likely to arise in electron microscopy.

The eighteen subroutines have been divided into four groups dealing respectively with crystal geometry, crystal structure, crystal indices and general problems. Six main programs have been included to cover the major computations required; any further main programs can then be written with minimum difficulty by using the subroutines, as illustrated by the program DT4BMS.

Following this introduction, which discusses portability problems, arithmetic precision, and program elaboration, are two sections outlining the subroutines and main programs; these are listed in Appendices A and B respectively. Appendix C is in two parts which give examples and a listing of typical input to the programs. Because of the voluminous output produced, only selected portions are given in Appendix D, in order to illustrate the output format.

The procedures have been written in FORTRAN-4 for the IBM-H compiler. Use has been made of the facility available in this compiler for assigning values other than .TRUE. or .FALSE. to LOGICAL*1 variables (variables occupying 1 storage byte). This has been used in subroutine LABAL to form two character strings of one word length each (4 storage bytes). If the user's compiler does not include this facility, this subroutine will have to be rewritten in ASSEMBLER or the four numbers plotted separately.

Floating point arithmetic is carried out in double precision, giving a nominal accuracy of 14 digits. The sole reason is to minimise summation errors in structure factor calculations. Single precision accuracy (6 to 7 digits) may give sufficient accuracy. All other subroutines should run satisfactorily if the real variables and functions are redefined to standard length.

Some elaboration is included within the programs and subroutines; for example, it is possible to specify the order in which indices will be incremented. In all cases, default options are built in if the option is not exercised.

A word of caution should be given about the elaboration available in the subroutines and programs, specifically about the use of systematic absences. If, for instance, a table of angles or d-spacings is being

computed for a close packed hexagonal crystal and the systematic absence parameter KABS is specified as H on the index parameter card, then the index set $(11\bar{2}1)$ is rejected but $(22\bar{4}2)$ is passed. The plane $(11\bar{2}1)$ is a valid plane and should be included in a table of d-spacings and is easier to find in a table of angles than the $(22\bar{4}2)$ entry. Systematic absences are automatically rejected in program STRFSB and thus need not be specified.

When calculating tables, the value of KABS should always be specified as primitive (the default option) although the programs have been written to allow use of this parameter if it is so desired. Use of the systematic absence option is certainly justified to prevent clutter on a stereographic projection, and to reject non-allowed curves on Kossel and Kikuchi patterns.

2. SUBROUTINES

2.1 Subroutines Relating to Crystal Geometry

There are four subroutines in this class. The first of these, XTLSYS, is used to define the crystal system under consideration. IDCALC is a subroutine with multiple entry points which performs commonly required calculations, such as interplanar spacings or angles between directions. Subroutine TCSET is used to transform crystallographic planes and directions from a general (triclinic)crystal system to cartesian (cubic) coordinates. Input of PERP2 is a set of general indices for a plane and the output is indices of two directions lying in this plane. If the coordinates are cartesian, then the input and output indices are mutually perpendicular.

2.1.1 Subroutine XTLSYS

The first requirement in any computation is to define the crystal system under consideration. This is carried out by subroutine XTLSYS which reads one card containing the crystal parameters A, B, C, α, β, γ and the name or identification (maximum of 20 characters) according to the format (6F 10.0, 5A4). Only those parameters actually required to specify the crystal system need to be punched; for example, only A, C and 120.0 in the γ field need to be specified for hexagonal crystals. The subroutine will set unspecified parameters as needed and return a number from 1 to 8 in the integer variable NSYS to signify the crystal system. The occurrence of 8 rather than 7 crystal systems is brought about by the distinction of 2 alternative settings for the monoclinic system as outlined by the International Union of Crystallography [1965]. The crystal systems are summarised in Table 1.

TABLE 1

NSYS	Crystal System	Specify
1	Cubic	a
2	Rhombohedral	a α
3	Tetragonal	a c
4	Hexagonal	a c γ = 120.
5	Orthorhombic	a b c
6	Monoclinic 1	a b c γ
7	Monoclinic 2	a b c β
8	Triclinic	a b c α β γ

This should be the first subroutine called. It can also be used to terminate execution of a program by using a blank card at the end of the input data, since a STOP statement is executed if parameter A is zero.

Required values and useful constants are set in common XTLCOM before returning from this subroutine. These include π, the volume of the unit cell, and the cosine and sine of the angles α, β and γ.

2.1.2 Subroutine IDCALC

This is a multiple entry point subroutine used for common crystallographic calculations. Entry points ADCALC and IDCALC calculate the d-spacing of the plane (H,K,L). The former name should be used if the indices are real variables, whereas the latter is used when the indices are integer variables. This use of two entry points also holds for the remaining calculations performed and is summarised in Table 2.

TABLE 2

Entry Point for		Calculation
Real Indices	Integer Indices	
ADCALC	IDCALC	d-spacing
ADENT	IDENT	identity spacing along a direction
ANGDIR	INGDIR	angle between directions
ANGPLN	INGPLN	angle between planes
ANGDP	INGDP	angle between a direction and a plane

Constants required in the calculations are found in common XTLCOM and all arguments are passed by argument list in the call statement. The following examples illustrate the lists for calculations using integer indices:

```
CALL IDCALC (H, K, L, D)
CALL IDENT (IU, IV, IW, AD)
CALL INGDIR (IU, IV, IW, AD, IU2, IV2, IW2, AD2, ANG, COSANG)
CALL INGPLN (H, K, L, D, H2, K2, L2, D2, ANG, COSANG)
CALL INGDP (IU, IV, IW, AD, H, K, L, D, ANG, SINANG)
```

In these examples, H,K,L are the indices of a plane of interplanar spacing D, whilst IU, IV, IW are the indices of a direction with identity distance AD. The variables D and AD are calculated real values in the first two call statements but are required values in the remainder. The last three call statements return the required angle in degrees in the variable ANG, together with the sine or cosine of the angle.

2.1.3 Subroutine PERP2

This is another multiple entry point subroutine. Entry point PERP2 is used when dealing with real values of indices, whereas IPERP2 is used for integer indices.

If, in the general crystal system, the input is the indices of a plane, the output would be the indices of two directions lying in this plane; alternatively if the indices of a direction are given as input then the output would be the indices of two planes containing the direction.

In the cubic or cartesian case, the input and output values would be orthogonal (but not normalised). In this case the subroutine can be used to generate a set of orthogonal axes for plotting of stereographic projection, or Kikuchi patterns.

2.1.4 Subroutine TCSET

Conversion of direction indices from any crystal system to cartesian coordinates is carried out in this subroutine. An initial call to TCSET is required to initialise the conversion parameters. Subsequent calls to either AT2CD or IT2CD (depending upon whether the initial indices were real or integer) will then convert the direction indices to cartesian coordinates such that the cartesian X axis lies parallel to the crystallographic 'a' direction, the Y axis lies in the 'ab' plane, and the Z axis is perpendicular to the X and Y axes. The values returned are always real. Entry points AC2TD or IC2TD are used to transform directions back from cartesian to the system under consideration, and again the returned values

are always real variables.

Conversion of planes from the general system to cartesian coordinates is carried out by calls to AT2CP or IT2CP, whereas the inverse transformation is performed by AC2TP or IC2TP.

In all cases the program has been arranged so that the same variable names can be used for both input and output values, where the input values are real. For example in using the statement

CALL AC2TD (X, Y, Z, X, Y, Z)

the variables (X, Y, Z) would be a cartesian direction before the CALL statement, but would be the coordinates (indices) of the same direction referred to the crystal system after the CALL statement.

2.2 Subroutine Relating to Crystal Structure

Structure factor and its related parameter, extinction distance, are dependent upon the position of each atom within the lattice and its scattering factor. Since scattering factors are widely disseminated either in tabulated form, or by analytical expressions, two subroutines are provided. Subroutine ATMPDT uses the tabulated values of Doyle & Turner [1968]; use of other tabulated values will require changes within the subroutine to allow for variations of the independent variable at which the scattering factors are listed (real array XS). Subroutine ATMPSB was written to use the expression of Smith & Burge [1962], the sum of three Gaussian terms, but has been amplified to allow up to four Gaussian terms with or without a constant, thus permitting the use of expressions derived by other authors.

The final subroutine WAVESP enables the electron beam to be specified either by wavelength or accelerating voltage.

2.2.1 Subroutine ATMPSB

Initialisation of input data for calculation of structure factors and extinction distances is made by a call to ATMPSB.

The first card read defines the wavelength, number of species of atoms, format (F10.5, I2). The wavelength may be expressed as electron wavelength or accelerating voltage of the electron microscope and is referred to subroutine WAVESP for clarification.

For each atomic element present, the following data is then read in:

(a) The number of positions it occupies, its atomic symbol and atomic number and the Debye temperature correction factor.

The atomic symbol and atomic number are used only for identification on the printout. One card, format (I2, IA4, I2, 2X, F10.5).

(b) Either 1 or 2 cards detailing the atomic positions and their filling factor, format (16F5.3), four positions per card.

(c) The parameters of the scattering power expression, format (9F8.0).

The subroutine allows a maximum of 10 atomic species so that the above series of cards may be repeated up to 10 times. Each atomic species may occupy up to 8 lattice sites, thus 4 numbers are required to specify each site uniquely. These are respectively, the fractional lattice translations along the a, b, and c axes together with the fraction of each of these sites which is occupied. If unspecified (and thus read in as zero occupancy), the default value of 1.0 is assumed. The following points should be kept in mind. If more than 8 sites are to be specified, then this species may be repeated with a new set of positions and scattering powers, so long as the total number of species does not exceed 10. The use of the filling factor allows the program to compute structure factors for alloys, thus the filling factors for iron, chromium and nickel in 18/8 stainless steel would be specified as 0.733, 0.192, 0.075 (atomic fraction). High concentrations of vacancies can be included by decreasing the filling factor of the sites, and ordering can be handled by varying the filling factor proportions between different sites.

The analytical expression for the scattering equation is of the form:

$$F = C + \sum_{n=1}^{4} A_n \exp(-B_n X^2)$$

where A_n and B_n are the Gaussian terms (maximum of 4) as defined in Smith & Burge [1962]. Some expressions also include a constant C and this has been included for completeness. If all of these values are not used and the appropriate column is left unpunched, then zero values are read from the card and the expression is still correctly evaluated. The values are read in the order A_1, B_1, A_2, B_2, A_3, B_3, A_4, B_4, C according to the format (9F 8.0). All values read in are listed on the line printer before returning to the main program.

Calls are made to entry point FCALSB for computation of structure factors. Input information required is the 3 indices of the plane and its d-spacing. Calculation of the structure factor is carried out using both cosine and sine terms since the structure is not assumed to be centrosymmetric. The correction for relativistic mass is performed by multiplication of the filling factor by the required constant (assumed to be 1.0 for X-rays) in the initial call to ATMPSB, so it does not appear explicitly in this calculation.

When the structure factor has been calculated and corrected for Debye temperature effects, the extinction distance is calculated. If the structure factor is less than 1.0D-06 then the particular reflection is assumed to be a systematic absence, and values of minus one are returned for the structure factor and extinction distance.

Entry point FOOOSB returns the zero order extinction distance. Since the d-spacing cannot be assigned to indices (0,0,0), Debye temperature correction is not performed.

2.2.2 Subroutine ATMPDT

The second method of utilising atomic scattering factors is 'table lookup'. Tables of atomic scattering factors for specific values of $\sin(\theta)/\lambda$ are given in numerous references, the values used in this subroutine being those of Doyle & Turner [1968]. The 27 values of $\sin(\theta)/\lambda$ used are stored in the dimensioned variable XS; these values would have to be changed if other tables are used. The layout of input cards for this subroutine is the same as that used in ATMPSB with the exception that 4 cards are required to specify the table of explicit values instead of one card of analytical coefficients. These values are read in using the format (7F10.7), the remaining 10 columns being available for card identification.

Calculation of structure factor and extinction distance is by way of the entry point FCALDT. Table lookup is carried out using quadratic interpolation, ensuring that the centre value is that one closest to the interpolation value. Calculation procedures otherwise are the same as those employed in FCALSB.

Entry point FOOODT calculates the zero order extinction distance.

2.2.3 Subroutine WAVESP

Calculation of electron scattering factors in subroutines ATMPSB and ATMPDT requires the value of the beam wavelength and relativistic mass correction. Electron beams are normally specified by their accelerating

potential in kV, and only rarely by their wavelength. The value read in by the two subroutines as the variable WAVE could be either of these possibilities, so subroutine WAVESP was written to resolve this problem.

If WAVE is zero, then a 200 kV electron beam is assumed; any other value lower than 0.2 is then assumed to be an electron wavelength. If the WAVE is greater than or equal to 5.0, then an electron accelerating voltage is being specified. Between these limits, an X-ray wavelength is assumed. Integer IND returns a value of 3, 2 or 1 respectively for electron wavelength, X-ray wavelength or accelerating voltage specification. In the case of X-ray wavelength the relativistic mass is assigned the value of 1.0, but in the electron cases, the remaining quantities are calculated before returning to the calling subroutine.

2.3 Subroutines Relating to Crystal Indices

This is the largest group of subroutines having eight numbers in all. They are designed to be simple in use yet give greatest flexibility in the way in which the indices can be manipulated.

Provision has been made to vary the order in which the indices are indexed: to test the indices for most of the common systematic absences, and make provision for a user specified test procedure; to test for and reject (if so required) multiples of an allowed set of indices; and to permute the indices without duplication.

2.3.1 Subroutine INDEX

Calls to this subroutine return the next set of incremented indices. Indices are incremented in the order determined in INSDET. If the selected index is already at its maximum value, it is reset to minimum and the next selected index is incremented. The number of indices which are at their maximum value is returned in NTST. If all indices are at maximum (NTST equal to 3 or 4), then NTST is set to 4 and an immediate return is effected. Tests may be carried out before returning to ensure that the indices are not systematic absences, are not multiples of an allowed reflection or are ordered (generally $H \geqslant K \geqslant L \geqslant 0$) before return, as outlined below.

Entry point INDSET should be called prior to calls to INDEX. Three parameters located in common INDCOM are specified in the call to INDSET as outlined in Table 3.

TABLE 3

SUMMARY OF INPUT PARAMETERS TO INDSET

NSABS	= 0	Reset depending upon systematic absences specified on the input data card
	= 1	Reject systematic absences
	= 2	No check carried out for systematic absences
NMULT	= 1	No multiples of allowed reflections returned
	= 2	No check carried out for multiple values
NORDER	= 1	General ordering required ($H \geqslant K \geqslant L \geqslant 0$)
	= 2	Ordering required with some relaxations depending upon zero values
	= 3	No checks for ordering are carried out

The number of allowable values returned increases as any of these parameters rise in value.

NSABS should be specified as 0 if systematic absences are affected by permutation and the indices are to be permuted in the main program. The returned value of 1 or 2 is then used in the main program. If systematic absences are independent of permutation of the indices, then systematic absences are rejected in this subroutine (NSABS = 1). Where permutation of the indices influences systematic absences, for example a rhombohedral crystal specified on a hexagonal lattice, then systematic absences must be checked in the main program (NSABS = 2).

Multiple values of allowed reflections, which are not systematic absences themselves, are passed for such calculations as d-spacing or structure factor, (NMULT = 2). They are not required when plotting stereographic projections or calculating angles between planes, (NMULT = 1).

Ordering is important when permuting indices and NORDER should be specified as 1 (together with NSABS initially set to 0) so that a given set is not generated a second time. In this case, a strict requirement is that $H \geqslant K \geqslant L \geqslant 0$ except for hexagonal indices where $H \geqslant K \geqslant 0$, $L \geqslant 0$. In crystal systems of low symmetry further relaxations are allowed when calculating d-spacings so that by specifying NORDER = 2, all possible values may be calculated. In this case, indices K and L may have any value within the ranges specified and H must be either zero or positive. If H is zero then K may not be negative but if both H and K are zero then L cannot be negative. No ordering requirements at all are required, for example when

plotting stereographic projections and NORDER is specified as 3.

Upon entry to INDSET, the following values are read from one card: KABS, NPARM, (NORD(J), J = 1,3), HMIN, HMAX, KMIN, KMAX, LMIN, LMAX, INDMAX, DMIN, according to the format (1A1, 1I1, 3X, 3I1, 2X, 6(2X,I3), 3X, I2, F5.3).

KABS specifies the systematic absence criteria to be employed and may be specified numerically or by character as outlined in Table 4.

TABLE 4

SYSTEMATIC ABSENCES

KABS	JABS	Test
A	1	centred on A face
B	2	centred on B face
C	3	centred on C face
F	4	centred on all 3 faces
I	5	internal (body) centred
R	6	obverse rhombohedral on hexagonal lattice
(R)	7	inverse rhombohedral on hexagonal lattice
H	8	close packed hexagonal
D	9	diamond lattice
P	(0) (blank)	primitive, no test
*		specified by user supplied subroutine XTSYAB

Upon return, KABS contains the character representation and JABS the numeric value shown in the table. R specifies inverse rhombohedral restriction but is also returned in KABS if 7 is specified on the card. A blank or zero on the card returns P in KABS and 10 in JABS, whereas * returns 11 in JABS. Default option is therefore primitive.

NORD specifies the order in which the indices will be incremented. Default is 321, that is L will be incremented most frequently between its limits but H will be incremented least frequently.

Minimum and maximum values are self explanatory. If unspecified, the default values are zero for HMIN and 4 for HMAX, KMIN is set equal to -HMAX and KMAX is set equal to HMAX, and LMIN is set equal to KMIN and LMAX is set equal to KMAX.

INDMAX has a default value of 100 and sets an upper limit, if required, to the sum of the absolute values of the three indices which will be generated, (variable KOUNT).

DMIN and NPARM allow a real and an integer value to be passed to a main program. They are used, for example, to specify a minimum value for d-spacing calculations, and to specify whether angles are to be calculated between planes or directions. After the input values have been read in and checked the indices are set equal to their minimum value and then tested to ensure that they conform to any restriction imposed by NSABS, NORDER, NMULT and systematic absences before being returned as IH, IK, IL and ITST in common INDCOM.

Initially, the indices are set equal to IH, IK and IL in the main calling program after returning from INDSET. Subsequent call to entry point INDEX will increment the indices to the next allowed set of values. When all have reached their maximum value, NTST is set to 3, signifying that this is the last allowed set. After calculations have been completed, termination of this program phase can be carried out by testing that NTST is greater than or equal to 3, before recalling INDEX. Alternatively, a call to INDEX may be made followed by a test of NTST to determine whether it has a value of 4. The latter method is preferred when testing for systematic absences since the set (HMAX, KMAX, LMAX) may be a systematic absence and hence not be returned.

An index set of (0,0,0) is not returned by calls to INDEX. Submission of this index set is used to reset the initial value of IH, IK and IL after parameters such as NSABS, NMULT and NORDER have been altered in a main program. Such a call returns the starting value in the variables used in the calling statement.

2.3.2 Subroutine SYSABS

Input to this subroutine consists of the three indices (H,K,L) and integer JABS which indicates which test for systematic absences is to be applied in this subroutine. Specific tests are outlined in the section dealing with JABS, KABS in subroutine INDEX. When the indices are a systematic absence, then MTST = 1 upon return, otherwise it is zero. A value of 11 for JABS passes the indices to the user supplied subroutine XTSYAB for determination. If this returns a value of MTST less than zero, then JABS is reset to 10 (primitive, no test), MTST is set to zero and RETURN is executed.

2.3.3 Subroutine XTSYAB

This subroutine is included in the library mainly to satisfy the requirements of the linkage editor phase of the operating system, thus preventing an 'unresolved external reference' error message. A user supplied subroutine with this name should be supplied to test indices for systematic absences not included in SYSABS. It should return a value of + 1 when the indices are a systematic absence, and zero when they form an allowable set. Execution of this particular subroutine, if not over-ridden by another subroutine of the same name, will print an error message on the line printer and return a value of - 1 to SYSABS. This latter subroutine will reset MTST to zero and JABS to 10, thus assuming that the lattice is primitive and no further calls to XTSYAB will be made.

2.3.4 Subroutine MULTPL

In some applications, it is required that a set of otherwise acceptable indices be rejected if they are a multiple of another allowed set. Such a situation occurs in calculation of angles between planes. Subroutine MULTPL determines whether this condition applies, and returns a value of one in MTST for indices which are such a multiple, otherwise MTST equals zero.

Subroutine COMFAC is first called to find the highest common factor of the indices. A returned value of 1 causes immediate return; otherwise each multiple of the basic index set, up to, but not including the set supplied to the subroutine, is tested to determine if it is a systematic absence. If all lower order sets are systematic absences, then the current set is passed as satisfactory, the assumption being that this set of indices was tested elsewhere against systematic absence requirements.

2.3.5 Subroutines COMFAC and FACTOR

Subroutine COMFAC determines the highest common factor of the three indices supplied. Zero values are ignored unless all three indices are zero, in which case a negative value is returned. Ignoring zeros, two input values are passed to FACTOR which reduces them to their highest common factor. This value and the third index are then resubmitted to FACTOR to obtain the required value.

2.3.6 Subroutine PERM

Whereas all other subroutines operate on 3 indices only (H,K,L) four indices (H,K,I,L) must be specified for this procedure, the index I being required for the hexagonal system but ignored in all other cases.

Initial call is to entry point PRMSET at which time the indices should be strictly ordered (by specifying NORD = 1 in calling INDSET). The indices are examined to determine if any duplication will occur (e.g. hkk or hko), so that, by setting an indicator value in variable NZ, duplicate values are not generated during permutation. Permutation follows a strict routine and all possible values are generated including negative indices. Variable N shows the current position throughout the sequence generated, which can be listed out if the subroutine is tested with an initial input of H = 3, K = 2 and L = 1. There is a maximum of 48 combinations possible in the general case, which is restricted to 24 for hexagonal values, since L is not permuted with H, K and I, but can assume only positive and negative values. By carrying the values N and NZ with the indices, it is possible to interleave the permutation of more than one set of indices simultaneously.

2.4 General Subroutines

This final section consists of the four remaining subroutines. They contain procedures for initialising data, forming a table of values ordered from the highest to the lowest value, a procedure for solving the general conic and a method of forming two character strings from a set of indices.

2.4.1 Subroutine BLOCK DATA

In the IBM system, data initialisation of variables in common can only be carried out by using a subroutine called BLOCK DATA (4). No executable statements may be included in this control block and the variables must be in a Named Common. This subroutine is automatically linked into the fully compiled and edited program, without being explicitly called, when a FORTRAN deck is passed through the three steps (COMPILE, LINKEDIT and GO) of the procedure FORTHCLG.

When a library of subroutines and main programs is being set up this data initialisation step is not included unless an instruction to the linkage editor step is included. In this case the following cards should be included each time a main program is compiled and placed in the library:

```
//LKED.SYSINØDDØ*
ØINCLUDEØSYSLIB(SYMCOM)
```

where Ø signifies a blank column in the card and SYMCOM is the name of the first named common in the BLOCK DATA subroutine.

2.4.2 Subroutine INSRT8

This is the first of a series of entry points designed to tabulate d-spacings of planes in decreasing size, together with the associated indices, such that a call to one of the entry points will insert the new values in their correct positions. If the value set in NDUP is 1, then duplicate values of the d-spacings already present will not be inserted, whereas if NDUP=2, then duplicate values will be inserted. Parameter ERROR specifies the relative error within which two values will be considered to be identical for non-insertion purposes. These values are passed from the main program to the subroutine via common INSRT.

The variables N and NDIM are, respectively, the number of values in the table and the maximum number of values that can be accommodated. N should be set to zero before being used. D is the d value that is to be inserted into the array DAR; it is the value to be tested against existing values in DAR (if duplicate values are to be excluded). Integer values H, K, I and L are the normal crystallographic indices; RH, RK and RL are integer rhombohedral indices associated with the hexagonal indices when JABS is either 6 or 7 in the main program. These values are inserted into the arrays HAR, KAR, IAR, LAR, RHAR, RKAR and RLAR.

Entry points are INSRT1, INSRT4, INSRT5 and INSRT8. The appropriate entry point to be selected depends upon the number of values to be inserted. If a table of angles between planes is being constructed, then only a single value is to be entered and the correct entry point is INSRT1. When calculating a table of d-spacings for the hexagonal lattice then the indices H, K, I and L are to be inserted together with the d-spacing and the entry point INSRT5 is used.

NTST is normally set to 0 before return but will be +1 if a duplicate value has been detected and no insertion occurs, or -1 if the tables are already filled.

2.4.3 Subroutine FUNZ

Kikuchi and Kossel patterns are lines formed by the intersection of a general central conic and the projection plane. Plotting these lines requires that the equation

$$Ax^2 + By^2 + Cz^2 + 2Eyz + 2Fxz + 2Gxy = 0$$

should be solved for a fixed value of Z.

Entry point FUNZ sets the value of Z and the coefficients, and reduces the equation to a biquadratic in X and Y. A test is carried out at

this stage to determine whether the curve is an ellipse, a parabola or a hyperbola, and a value of 1, 2 or 3 respectively, is returned in the integer variable NTYPE.

Entry points FUNOFY and FUNOFX are called to solve the biquadratic equation for given values of Y and X respectively. Entry point FUNOFY specifies a value of Y as input; this then reduces the equation to a quadratic in X, which in general can be solved for two values of X. Whether the higher or lower value is returned depends upon whether the integer variable NSY has been set to plus or minus one. The integer variable JTST is set to plus one unless the solution of the quadratic will be an imaginary number, in which case JTST is negative and X is not determined.

Entry point FUNOFX carries out a similar operation using Y as the dependent variable and X as the independent variable.

2.4.4 Subroutine LABAL

Labelling of stereographic projections and Kikuchi and Kossel lines is facilitated by subroutine LABAL. Input parameters are the four indices H, K, I and L together with N, the number of indices to be used. This should be set to four for hexagonal indices, and any other number for all others, in which case I is ignored. It is acceptable to use NSYS for N.

Two character strings A1 and A2 are produced by examining each index in turn. A blank is inserted into the second string if the index is positive or zero, a negative sign if it is less than zero. The alphanumeric representation of the absolute value of the index is placed into the first string.

Two character strings are thus produced which can be written as text and, by plotting A2 above A1, the indices are represented in normal (bar) form.

2.5 Outline of the Plotting Subroutines

The plotting subroutines used at the AAEC were produced locally by Dr. G. Cox and Mr. R. Backstrom of the Applied Mathematics and Computing Division. Plots may be viewed on a Tektronix T-4002 Graphic Computer Terminal and plotted on a Calcomp 565 Drum Plotter. Calcomp output is specified by pages which are 10.9 inches in the Y direction and multiples of 16.54 inches in the X direction.

2.5.1 GPSEND(N,M)

This subroutine should be the first and last call of the procedure. To initialise the plotter, specify N = 1; M then specifies the

maximum number of plotting pages to be used. The subroutine identifies the plot by writing the job name, programmer's name, the time and the date along the Y axis and leaves the pen positioned at the minimum X and Y position clear of the text.

To finalise the plot, specify N = 2; M is not used.

2.5.2 GPLOT(X,Y,N)

For N = 1, the current position of the pen is defined as being located at the position (X,Y).

For N = 2, X and Y are the number of units of variables X and Y required to move the pen by 1 inch. To scale the plotter in centimetres, for example, CALL GPLOT (2.54, 2.54, 2).

For N = 3, move the pen to the position (X,Y) with the pen up.

For N = 4, move the pen to the position (X,Y) with the pen down.

For N = 5, move the pen to the position (X,Y) drawing a dashed line.

2.5.3 GPTEXT(STRING,NC,NHT,THETA)

Commencing with the pen at its current position,write NC characters from STRING at an angle THETA degrees (anticlockwise) to the X axis. The characters will be NHT multiples of the basic size, which is 0.05 x 0.07 inches with a separation of 0.01 inches between them.

If NCH is zero, then the first character in STRING is written, centred on the current pen position. This is useful for plotting points.

2.5.4 GPNUMB(NUM,FMT,NHT,THETA)

This subroutine writes the single number NUM according to the format FMT. NHT and THETA are the same as defined in GPTEXT. The format FMT may be a variable name containing the format as a character string within brackets, or the format may be defined within the call statement by enclosing it within brackets enclosed within quotes. Only one number may be written by a single call, which is the reason for writing subroutine LABAL.

3. MAIN PROGRAMS

Six programs are presented in this section and include procedures for calculating tables of interplanar d-spacings (program DSPACE) and angles between planes or directions (program ANGLES). Two programs deal with structure factors and extinction distance in the simple dynamical two-beam case (program STRFSB) and four-beam corrections (DT4BMS). The final two programs are plotting routines. The first (program STEREO) plots projections of directions or poles of planes whilst the second produces Kikuchi or Kossel maps (program KIKU).

The two plotting procedures require that the projection axis be specified. This must be in the 4-index (Miller-Bravis) notation for hexagonal crystals. In any other system the L index may be punched in either the I or L field with impunity, but the remaining field must be left blank.

3.1 DSPACE - Calculation of Interplanar d-spacings

This program calculates the d-spacings of planes for any crystal system and produces two columns of output. The first column lists the calculated values in index order; the second column gives the same values in decreasing order of d-spacing.

Only two data cards are required; the first card defines the crystal system, as outlined under subroutine XTLSYS, and the second card defines the index parameters required by subroutine INDSET. These two cards may be repeated indefinitely specifying new computations to be performed. The input data set should be terminated with a blank card in order to avoid abnormal termination of the program.

The statement

```
CALL INDSET (1, 2, 3)
```

specifies that

(a) systematic absences are to be tested before indices are returned,

(b) multiples are to be allowed, and

(c) no ordering of indices is required.

Statement 20 rejects negative values of index H (if HMIN is less than 0), and the d-spacing is then calculated by a call to IDCALC. If the crystal is hexagonal or rhombohedral on a hexagonal lattice, such additional indices as are required are then set.

The d-spacing and indices are stored twice, firstly using the appropriate INSRT call to obtain a table of decreasing d-spacings, and then sequentially to obtain a table in index order. When the sequence of indices has been completed, or the tables filled, the values are printed out on the line printer. Dimension statements at the beginning of the program limit the tables to a maximum of 400 values.

3.2 ANGLES - Angles Between Plane Directions

Tables of angles between directions and angles between planes are frequently required in solving crystallographic problems. When setting up a program to produce such tables, consideration must be given to the quantity of data generated.

The simplest programming procedure, having selected two sets of indices, is to generate all possible angles that can be obtained by permuting and changing the sign of the indices. Considering the index sets (3, 2, 1) and (5, 4, 3) and the cubic system, a total of 34 distinct angles exist (between 0° and 180°), whereas 1152 distinct angles exist in the triclinic system, the latter case requiring approximately two pages of computer listing. Such a table becomes almost useless. If in this latter case the permutations of the (3, 2, 1) set are collected into groups having the same d-spacing (and thus assumed to be symmetry related) then twenty four sets of forty eight values are produced (a total again of 1152 values), some of which may be useful. In this program, the latter approach has been adopted.

The mandatory calls are made to XTLSYS and INDSET to set the required parameters. The value of the integer NPARM read from column 2 of the index card is passed via common block INDCOM and is used in the main program as integer NDP. If NDP=1, then angles between directions are calculated, whereas NDP=2 signifies that tables of angles between planes are required. Any other value of NDP (e.g. 0, 3) will produce both sets of tables.

Two sets of indices are used within the program and two copies of each are used. The first set (HX, KX, LX) is incremented from minimum to maximum once throughout the computation. The second copy of this set (HA, KA, LA) is permuted to generate a table of indices with their associated d-spacing or identity distance, excluding duplicate values (IDUP=1). If systematic absences are independent of permutation, such indices are rejected in calls to INDEX. Where permutation has an effect, each set is tested immediately after permutation.

The second set of indices (HY, KY, LY) is incremented from the current value of (HX, KX, LX) to the maximum values and copied into (HB, KB, LB). For each set of values, a table is constructed of allowed indices with associated distance without excluding duplicate values (IDUP=2) by permuting (HB, KB, LB). Restriction of the first table to non-duplicate values minimises the amount of printout; on the other hand, allowing duplicate values in the second table ensures that all values are determined.

Each set of indices in the first table is selected in turn and the angles between the selected set and all sets of indices in the second table are calculated and printed out.

Where the complete table has been calculated and printed, and variable NDP has a value of 1 or 2, execution is recommended at the call to XTLSYS to determine whether a new set of data follows, or to terminate the program. Where NDP has the default value of 3 and the angles between planes have been calculated then NDP, NSABS and NMULT are all reset to 1, JABS is set to 10 (primitive) and a call made to INDEX specifying zero values for the indices. This resets the index parameters so that a table of angles between directions may be produced free of systematic absences. Execution recommences at the statement labelled 100.

Because of the large amount of printout which can be produced, the maximum values for indices should be restricted for crystals of low symmetry. Where crystallographic work must be carried out on low symmetry crystals, consideration should be given to modifying the program to calculate the angles between two specific directions or planes instead of the procedure adopted here of one specific and one general form. An alternative worth considering is a set of stereographic projections about the three main axes produced by program STEREO.

3.3 STRFSB - Structure Factors and Extinction Distances

Calculation of structure factors and extinction distances using analytical expressions for the scattering factors is carried out in this program.

Data input commences with the two standard cards for crystal system and indices and a call is then made to ATMPSB. This subroutine reads in the remaining data as outlined in Section 2.2.1. The indices are incremented throughout their ranges and FCALSB is called to calculate the required values. These values are not stored but printed as calculated. If the structure factor is less than 1.0D-05, then values of -1 returned for FHKL and TO and the reflection is assumed to be a systematic absence.

When all values have been calculated, program execution is returned to subroutine XTLSYS in order to determine if a new set of data follows.

If it is desired to calculate structure factors and extinction distances using the table lookup procedure of Doyle & Turner [1968], then only four cards need to be changed. The program name in format statement 201 should be changed to STRFDT, the statement CALL ATMPSB (...) should be changed to CALL ATMPDT (...) with the same argument list, the statement CALL FCALSB (...) should be changed to FCALDT (...), again without altering the argument list, and finally the call to FOOOSB should be changed to FOOODT for the zero order extinction distance.

3.4 DT4BMS - Four-beam Corrections to Structure Factors

This program was a late development, written when a need arose to determine the effect of multiple systematic beam excitation on the extinction distance of selected metals, varying over a range of atomic numbers from aluminium (AN=13) to gold (AN=79). The effect was to be determined for specific indices only using both the tabulated values of Doyle & Turner [1968] and the analytic approach of Smith & Burge [1962]. The latter program SB4BM is derived from this program by changing the two cards executing calls to ATMPDT and FCALDT to execute calls to ATMPSB and FCALSB.

The first card is the customary data card read by XTLSYS to define the crystal system. The next set of cards defines the lattice site and scattering parameters required by ATMPDT. The following cards then define the H, K, (I), and L indices required, punched one set per card using the format 4I3. The end of this sequence of cards is signified by a blank card defining all indices to be zero. This passes control back to XTLSYS which reads the next card to determine whether a new set of data follows, or the end of the data set has been reached (a second blank card).

After each of the indices is read in, the structure factors and associated crystal potential (variable U) are calculated for the first three systematic reflections of this set. The corrected first order crystal potential, and hence the extinction distance, are then calculated by formulae given by Howie [1966] and the values listed on the line printer.

The correction procedure is repeated in Howie [1970] together with a correction procedure for the second order structure factor, this being a 3-beam correction. This latter correction has not been incorporated into the program. In each case, the corrected value is for the condition $S = 0$ for the specified diffracted beam.

3.5 Program STEREO - Plotting of Directions or Poles of Planes

This program plots the projections of crystallographic directions and poles of planes using gnomonic, stereographic, or orthographic projection.

After setting the crystal system and index parameters, the program reads one card which defines the crystallographic indices of the Z axis (projection direction or pole of a plane) together with the variables NDP, NGSO, THETA and RADIUS, format (4I5, 4X, I1, 4X, I1, 3F10.5).

NDP defines whether directions (NDP=1) or planes (NDP=2) are to be plotted (default is planes), and NGSO is the projection to be employed, 1 for gnomonic, 2 for stereographic (and default), 3 for orthographic. Plot scaling is controlled by THETA or, if this is undefined (thus zero), then by RADIUS. THETA is the angle in degrees from the centre of the plot to the centre of the YMAX boundary (plotting being centred within a 31 x 26 cm rectangle). RADIUS specifies the distance from the origin at which a vector perpendicular to the Z axis will be plotted in stereographic or orthographic projection, and a vector at 45° to the Z axis will be plotted for gnomonic projection. Default options are THETA = 45° for gnomonic projection and RADIUS = 10 cm for stereographic and orthographic projection.

Orthogonal axes are set up such that the Z axis corresponds with the specified crystallographic direction or pole of the crystal plane. The height of the projection plane (ZPLN) is then calculated to give the required scaling for the plot.

The indices are incremented and permuted throughout their range, systematic absences and multiple orders being rejected. The angle from the pole of the plane or the direction to the three axes is then determined. The cosines of the three angles then constitute a unit vector which is plotted. In the gnomonic case, the vector is rejected if the third cosine is equal to or less than zero (on or below the equatorial plane). Orthographic projection is permissible if the vector is not below the equatorial plane. All vectors in stereographic projection which are at a greater angle than approximately 25.8° to the -Z axis (cosine greater than -0.9) are acceptable at this part of the program.

The plotting points (X, Y) are determined using the appropriate conversion and all points falling within the plot boundaries are plotted. If variable KOUNT is less than 5, the points are also labelled with their indices. These are written at an angle of 30° to the X axis of the plot in order to minimise overlap. The indices of all plotted points, together with some pertinent data are printed on the line printer. Points which have been labelled are signified by a plus sign on the listing.

On completion, the plot is labelled with its pole and the name taken from the crystal system card. A new card is then read to determine if further poles of this crystal system are to be plotted. If the indices are all zero, then execution of the program recommences at the call to XTLSYS to specify a new crystal system or to stop program execution. Thus, one

blank card should follow a sequence of data cards in order to recommence computation with a new crystal system, and two blank cards should be appended at the end of all data to bring about normal termination of the program.

3.6 Program KIKU - Plotting of Kikuchi and Kossel Patterns

Kikuchi and Kossel patterns are used for accurate determination of crystal orientation. This program produces maps of such patterns to allow easy indexation of experimentally produced determinations.

Once the crystal system and index parameters have been determined, the main program reads in one card specifying the pole (NPOLE), the pole location (CX, CY), the scale of the plot (SCALE) and the radiation wavelength (WAVE). The plot is restricted to a region 12" by 8" centred on the plotting page. Parameters CX and CY specify the location of the pole from the centre of the plotting region and they do not have to be within the plot; by keeping all other parameters unchanged and varying CX and CY, a series of plots can be produced which may be overlapped to produce an extended map of large scale. The parameter SCALE specifies the angular displacement corresponding to a displacement of one inch from the pole and has a minimum and default value of 3° for Kikuchi, and a minimum of 3° and default value of 20° for Kossel patterns.

After the plotter has been initialised and scaled, the angular displacement from the point (CX, CY) to the furthest corner of the plot is calculated (variable THETAM). This is subsequently used in conjunction with the Bragg angle to determine whether the plotted line is likely to fall within the plot boundaries. The plotting axis is then set up and the pole position is plotted and labelled with the letter 'P'.

The indices are incremented throughout their range and permuted to generate all possible sets. An index set is rejected if the d-spacing is below DMIN, if the sine of the Bragg angle is greater than 1.0 or if the Bragg angle, the angle to the Z axis and THETAM are all such that the plotted line would not lie within the plot boundaries.

The parameters of the general conic are then set and plotting commenced. If the cosine of the angle between the X axis and the projection of the pole onto the XY plane is greater than 0.555 (within 33.4°), plotting commences at the boundary Y=YMAX. If it is less than -0.555, plotting commences at the boundary Y=MIN; otherwise it will commence at the boundaries X=XMAX or X=XMIN depending upon whether the cosine of the angle to the Y axis is positive or negative. Since the plotting strategy

is similar for plots across the page (X being the incremented variable) or up and down the page (Y being incremented), only the former will be described.

The Calcomp 565 Drum Plotter is an incremental plotter having a step length of 0.01 inch and the variable XSTPX has been set to this value. The starting value XST has been set equal to either XMAX or XMIN and the step length STP to either minus or plus XSTPX. Variable STP2 is equal to twice XSTPX and is used later in the program to decide if step lengths of 0.01 or 0.10 inch are being taken. STP10 is ten times STP and is used to obtain large step lengths when the curve is outside the plot boundaries.

Plotting commences at the selected boundary and proceeds across the page until the opposite boundary is encountered. The procedure is then repeated a second time in the opposite direction with the value of NSX reversed in sign in order to plot the second branch of the curve (if it exists). This is not carried out for Kikuchi lines where the second branch corresponds to the first branch of a curve with indices which are the negative of the current set. Once the second pass is completed plotting ceases. If, during plotting, the curve passes beyond the YMIN or YMAX boundaries, then plotting is suspended until the plot is again within limits. During this period, the increment in X is increased to STP10 until a correct value of Y is found, X is then stepped back by STP10 and reapproaches the correct value in increments of 0.01 inch, and plotting is restarted. If Y is imaginary at the beginning of the plot, a similar procedure is adopted until Y is real and within limits, and plotting commences. If Y becomes imaginary during plotting on the first pass across the page, the direction is reversed to plot the second branch. If this occurs on the second pass, then plotting is assumed to be finished. This will occur if Y is imaginary at the beginning of the first pass, for which condition the ends of the curve are closed. All points at which the curve meets any boundary or becomes imaginary are listed on the lineprinter.

The integer variable IP used in calls to GPLOT is normally set to 4. When plotting Kikuchi patterns for indices which are a multiple of an allowed set, IP is set to 5. A full curve is plotted for IP = 4; such curves are labelled with their indices at the finishing end. If IP = 5, the curves are plotted with a dashed line and are not labelled.

When all indices have been plotted, the pattern is labelled with the pole, the name and other pertinent data, and the program proceeds to plot the next pole if specified, seek a new crystal system if one blank card is encountered, or stop execution if two blank cards are found.

4. ACKNOWLEDGEMENTS

It is a pleasure to acknowledge the encouragement given by Dr. A. Jostsons whilst these programs were being developed. Thanks are due also to Dr. G. Cox and Mr. R. Backstrom of the Applied Mathematics and Computing Division for allowing the use of the plotting subroutines, and to many other colleagues for their tuition in computing and their assistance in correcting my errors.

5. REFERENCES

Doyle, P.A. & Turner, P.S. [1968] - Acta Crystallogr. A24:390.

IBM(Undated) - IBM System/360, FORTRAN-4 Language, Form C28-6515-4, p.104.

The International Union of Crystallography [1965] - International Tables for X-ray Crystallography (ed. N.F.M. Henry & K. Lonsdale) The Knoch Press, Birmingham, England.

Howie, A. [1966] - Phil. Mag. 14:223.

Howie, A. [1970] - 'Theory of High Energy Electron Diffraction,' in 'Modern Diffraction and Imaging Techniques in Material Science' (ed. S. Amelincks, R. Gevers, G. Remaut & J. Van Landugt). North Holland Publishing Company.

Smith, G.M. & Burge, R.E. [1962] - Acta Crystallogr. 15:182.

APPENDIX A

LISTING OF SUBROUTINES

APPENDIX A

LISTING OF SUBROUTINES

***** SUBROUTINE XTLSYS *****

```
      SUBROUTINE XTLSYS
C
      COMMON/XTLCOM/ A,B,C,ALFA,BETA,GAMA,NSYS,NAME(5),VOLUME,
     &       ZERO,ONE,TWO,PY02,PY,RAD,DEG,CA,CB,CG,SA,SB,SG,
     &       S11,S22,S33,S12,S23,S31,  A2,B2,C2,VOLSQ
      COMMON/UNITS/ NCARD,LNPTR
C
      DATA    DM6/1.0E-06/,  A90/9.0E+01/,  A120/1.2E+02/
C
C  SET CRYSTAL PARAMETERS
C   NSYS=1    CUBIC                      SPECIFY: A
C   NSYS=2    RHOMBOHEDRAL               SPECIFY: A,ALFA
C   NSYS=3    TETRAGONAL                 SPECIFY: A,C
C   NSYS=4    HEXAGONAL                  SPECIFY: A,C,GAMMA=120.0
C   NSYS=5    ORTHORHOMBIC               SPECIFY: A,B,C
C   NSYS=6    MONOCLINIC,1ST.SETTING     SPECIFY: A,B,C,GAMMA
C   NSYS=7    MONOCLINIC,2ND.SETTING     SPECIFY: A,B,C,BETA
C   NSYS=8    TRICLINIC                  SPECIFY: A,B,C,ALPHA,BETA,GAMMA
C
C
      READ(NCARD,1) A,B,C,ALFA,BETA,GAMA,NAME
    1 FORMAT(6F10.6,5A4)
      IF(A.LT.DM6) STOP
      IF(B.LT.DM6)  B=A
      IF(C.LT.DM6)  C=A
      IF(ALFA.LT.DM6)  ALFA=A90
      IF(BETA.LT.DM6)  BETA=ALFA
      IF(GAMA.LT.DM6)  GAMA=ALFA
C
   10 IF(ALFA.EQ.A90) GO TO 20
      NSYS=2
      IF(A.NE.B)  NSYS=8
      IF(A.NE.C)  NSYS=8
      IF(BETA.NE.ALFA)  NSYS=8
      IF(GAMA.NE.ALFA)  NSYS=8
      GO TO 60
C
   20 IF(BETA.EQ.A90) GO TO 30
      NSYS=7
      IF(GAMA.NE.A90) NSYS=8
      GO TO 60
C
   30 IF(GAMA.NE.A90) GO TO 40
      NSYS=1
      IF(C.NE.A)  NSYS=3
      IF(B.NE.A)  NSYS=5
```

(continued)

```
      GO TO 60
C
   40 IF(GAMA.EQ.A120) GO TO 50
      NSYS=6
      GO TO 60
C
   50 NSYS=4
      IF(B.NE.A)  NSYS=6
C
   60 CONTINUE
      MSYS=NSYS
C
C    SET CONSTANTS FOR CRYSTALLOGRAPHIC CALCULATIONS, AND RETURN CELL VOLUME
C
      ZERO=0.0D+00
      ONE=1.0D+00
      TWO=ONE + ONE
      PYO2=TWO* DATAN(ONE)
      PY=TWO*PYO2
      DEG=(180.0D+00)/PY
      RAD=ONE/DEG
      ABC= A*B*C
      A2=A*A
      B2=B*B
      C2=C* C
      CA= DCOS(ALFA*RAD)
      CB= DCOS(BETA*RAD)
      CG= DCOS(GAMA*RAD)
      CABG=CA*CB*CG
      SA= DSIN(ALFA*RAD)
      SB= DSIN(BETA*RAD)
      SG= DSIN(GAMA*RAD)
      S11=B2*C2*SA*SA
      S22=C2*A2*SB*SB
      S33=A2*B2*SG*SG
      S12= ABC*C*(CA*CB-CG)
      S23= ABC*A*(CB*CG-CA)
      S31= ABC*B*(CG*CA-CB)
      SQRANG= DSQRT(ONE+TWO*CABG-CA*CA-CB*CB-CG*CG)
      VOLUME = ABC*SQRANG
      VOLSQ=VOLUME*VOLUME
      WRITE(LNPTR,100) NAME,NSYS,
     &                 A,B,C,ALFA,BETA,GAMA,CA,CB,CG,SA,SB,SG,S11,S22,
     &  S33,S12,S23,S31,SQRANG,VOLUME
  100 FORMAT('1SUBROUTINE XTLSYS',T50,5A4,T99,'NSYS= ',I2/
     &                 T50,'CRYSTAL DATA'/
     &  T5,'A= ',F8.5,T25,'B= ',F8.5,T45,'C= ',F8.5,T65,'ALFA= ',F6.2,
     &  T85,'BETA= ',F6.2,T105,'GAMA= ',F6.2//T50,'CALCULATION DATA'/
     &  T5,'CA= ',F7.5,T25,'CB= ',F7.5,T45,'CG= ',F7.5,T65,'SA= ',F7.5,
     &  T85,'SB= ',F7.5,T105,'SG= ',F7.5/T5,'S11= ',F10.5,T25,'S22= ',
     &  F10.5,T45,'S33= ',F10.5,T65,'S12= ',F10.5,T85,'S23= ',F10.5,
     &  T105,'S31= ',F10.5/T5,'SQRANG= ',F7.5,T55,'VOLUME OF UNIT CELL=
     &',F10.5)
C*****************************************************************************
      RETURN
      END
```

(continued)

***** SUBROUTINE IDCALC *****

```
      SUBROUTINE IDCALC(IH,IK,IL,D)
C
C    CALCULATE 'D' SPACING OF THE PLANE (H,K,L)
C
      IMPLICIT  REAL*8(A-H,O-Z),  INTEGER*4(I-N)
      REAL*8    H,K,L,H1,K1,L1,H2,K2,L2
      COMMON/UNITS/NCARD,LNPTR
      COMMON/XTLCOM/ A,B,C,ALFA,BETA,GAMA,NSYS,NAME(5),VOLUME,
     &       ZERO,ONE,TWO,PYO2,PY,RAD,DEG,CA,CB,CG,SA,SB,SG,
     &       S11,S22,S33,S12,S23,S31,A2,B2,C2,VOLSQ
      DATA A999/0.99999999D+00/,A000/1.0D-08/
C
      H=IH
      K=IK
      L=IL
      GO TO 20
C
      ENTRY ADCALC  (H,K,L,D)
   20 D=H*H*S11+K*K*S22+L*L*S33+TWO*S12*H*K
     &  +TWO*S23*K*L +TWO*S31*L*H
      D=VOLUME/DSQRT(D)
      RETURN
C*********************************************************************
C
C  CALCULATE IDENTITY DISTANCE ALONG THE DIRECTION (U,V,W)
C
      ENTRY IDENT(IU,IV,IW,AID)
      U=IU
      V=IV
      W=IW
      GO TO 30
C
      ENTRY ADENT(U1,V1,W1,AID)
      U=U1
      V=V1
      W=W1
   30 IF  (NSYS.NE.4) GO TO 35
      T=-U-V
      U=U-T
      V=V-T
   35 AID=DSQRT(A2*U*U +B2*V*V +C2*W*W +TWO*B*C*V*W*CA
     &       +TWO*C*A*W*U*CB +TWO*A*B*U*V*CG )
      RETURN
C*********************************************************************
C
C  CALCULATE THE  ANGLE BETWEEN DIRECTIONS (U1,V1,W1) & (U2,V2,W2)
C
      ENTRY INGDIR(IU1,IV1,IW1,AID,IU2,IV2,IW2,BID,ANG,COSANG)
      U1=IU1
      V1=IV1
      W1=IW1
      U2=IU2
      V2=IV2
      W2=IW2
      GO TO 40
C
```

(continued)

```
      ENTRY ANGDIR(UA,VA,WA,AID,UB,VB,WB,BID,ANG,COSANG)
      U1=UA
      V1=VA
      W1=WA
      U2=UB
      V2=VB
      W2=WB
   40 IF(NSYS.NE.4) GO TO 45
      T=-U1-V1
      U1=U1-T
      V1=V1-T
      T=-U2-V2
      U2=U2-T
      V2=V2-T
   45 ANG = A2*U1*U2 +B2*V1*V2 +C2*W1*W2 +B*C*(V2*W1+V1*W2)*CA
     &       +A*C*(U1*W2+U2*W1)*CB +A*B*(U1*V2+U2*V1)*CG
      COSANG=ANG/(AID*BID)
      GO TO 100
C
C*************************************************************************
C   CALCULATE THE ANGLE BETWEEN PLANES (H1,K1,L1) & (H2,K2,L2)
C
      ENTRY INGPLN(IH1,IK1,IL1,AID,IH2,IK2,IL2,BID,ANG,COSANG)
      H1=IH1
      K1=IK1
      L1=IL1
      H2=IH2
      K2=IK2
      L2=IL2
      GO TO 50
C
      ENTRY ANGPLN(H1,K1,L1,AID,H2,K2,L2,BID,ANG,COSANG)
   50 ANG=S11*H1*H2 +S22*K1*K2 +S33*L1*L2 +S12*(H1*K2+H2*K1)
     &  +S23*(K1*L2+K2*L1) +S31*(L1*H2+L2*H1)
      COSANG=(ANG*AID*BID)/(VOLSQ)
      GO TO 100
C*************************************************************************
C
C  CALCULATE THE ANGLE BETWEEN THE DIRECTION (U,V,W) & THE PLANE (H,K,L)
C
      ENTRY INGDP(IU,IV,IW,AID,IH,IK,IL,D,ANG,SINANG)
      U=IU
      V=IV
      W=IW
      H=IH
      K=IK
      L=IL
      GO TO 60
C
      ENTRY ANGDP(UA,VA,WA,AID,H,K,L,D,ANG,SINANG)
      U=UA
      V=VA
      W=WA
   60 IF(NSYS.NE.4) GO TO 65
      T=-U-V
      U=U-T
      V=V-T
   65 SINANG=(H*U + K*V + L*W)*(D/AID)
      GO TO 200
C*************************************************************************
```

(continued)

```
  100 IF(COSANG.GT.A999) GO TO 101
      IF(COSANG.LT.-A999) GO TO 102
      IF(DABS(COSANG).LT.A000) COSANG=ZERO
      ANG=DEG*DARCOS(COSANG)
      RETURN
  101 COSANG=ONE
      ANG=ZERO
      RETURN
  102 COSANG=-ONE
      ANG=180.0D+00
      RETURN
  200 IF(SINANG.GT.A999) GO TO 201
      IF(SINANG.LT.-A999) GO TO 202
      IF(DABS(SINANG).LT.A000) SINANG=ZERO
      ANG=DEG*DARSIN(SINANG)
      RETURN
  201 SINANG=ONE
      ANG=90.0D+00
      RETURN
  202 SINANG=-ONE
      ANG=-90.0D+00
      RETURN
      END
```

***** SUBROUTINE PERP2 *****

```
      SUBROUTINE PERP2 (X,Y,Z,  XA,YA,ZA,  XB,YB,ZB)
      IMPLICIT  REAL*8(A-G,O-Z),  INTEGER*4(H-N)
C
C  GIVEN A PLANE(X,Y,Z), FIND TWO DIRECTIONS (XA,YA,ZA) & (XB,YB,ZB)
C  WHICK LIE IN THE PLANE.  THE CROSS PRODUCT IN THE ORDER,
C  (SET A)X(SET B) WILL GIVE THE PLANE (X,Y,Z).
C
      ZERO=0.0D+00
      ONE=1.0D+00
      DM10=1.0D-10
      XA=ZERO
      YA=ZERO
      ZA=ZERO
C
  200 IF(DABS(X).GT.DM10) GO TO 210
      X=ZERO
      XA=ONE
      GO TO 240
  210 IF(DABS(Y).GT.DM10) GO TO 220
      Y=ZERO
      YA=ONE
      GO TO 240
  220 IF(DABS(Z).GT.DM10) GO TO 230
      Z=ZERO
      ZA=ONE
      GO TO 240
  230 XA=Y
      YA=-X
  240 XB=Y*ZA - Z*YA
      YB=Z*XA - X*ZA
      ZB=X*YA - Y*XA
  250 RETURN
```

(continued)

```
C
      ENTRY IPERP2 (IH,IK,IL,  HA,KA,LA,  HB,KB,LB)
C
      HA=Ø
      KA=Ø
      LA=Ø
  1ØØ IF(IH.NE.Ø) GO TO 11Ø
      HA=1
      GO TO 14Ø
  11Ø IF(IK.NE.Ø) GO TO 12Ø
      KA=1
      GO TO 14Ø
  12Ø IF(IL.NE.Ø) GO TO 13Ø
      LA=1
      GO TO 14Ø
  13Ø HA=IK
      KA=-IH
  14Ø HB=IK*LA - IL*KA
      KB=IL*HA - IH*LA
      LB=IH*KA - IK*HA
  15Ø RETURN
      END
```

***** SUBROUTINE TCSET *****

```
      SUBROUTINE TCSET
C
C
C  SET PARAMETERS FOR THE TRANSFORMATION OF INDICES (U,V,W) IN TRIGONAL
C  SYSTEM (A,B,C,ALFA,BETA,GAMA) TO INDICES (X,Y,Ż) IN CUBIC SYSTEM (A)
C
C  'X' LIES ALONG 'U'.
C  'Y' LIES IN THE 'UV' PLANE.
C  'Ż' IS PERPENDICULAR TO THE 'XY' PLANE.
C
      COMMON /XTLCOM/ A,B,C,ALFA,BETA,GAMA,NSYS,NAME(5) , VOLUME,
     &       ŻERO,ONE,TWO,PYO2,PY,RAD,DEG,CA,CB,CG,SA,SB,SG
      COMMON/UNITS/NCARD,LNPTR
C
      G=(CA-CB*CG)/SG
      H=DSQRT(ONE - CB*CB - G*G)
      VX=B*CG/A
      WX=C*CB/A
      VY=B*SG/A
      WY=C*G/A
      WŻ=C*H/A
      ŻW=ONE/WŻ
      YV=ONE/VY
      WV=-(C*G)/(B*SG)
      PHTK=CG/SG
      PKTL=(A*G)/(B*H*SG)
      PHTL=(SG*CB-CG*G)/(H*SG)
      RETURN
C
C  TRANSFORMATION OF DIRECTION FROM TRICLINIC(U,V,W) TO CARTESIAN(X,Y,Ż)
C
      ENTRY IT2CD(IU,IV,IW,  X,Y,Ż)
C
```

(continued)

```
      U1=IU
      V1=IV
      W=IW
      IF(NSYS.NE.4) GO TO 20
      GO TO 15
C
      ENTRY AT2CD(U,V,W,  X,Y,Z)
C
   10 U1=U
      V1=V
      IF(NSYS.NE.4) GO TO 20
   15 T=-U1-V1
      U1=U1-T
      V1=V1-T
   20 X=W*WX+V1*VX+U1
      Y=W*WY+V1*VY
      Z=W*WZ
      RETURN
C
C  TRANSFORM DIRECTION FROM CARTESIAN(X,Y,Z) TO TRICLINIC(U,V,W)
C
      ENTRY IC2TD(IX,IY,IZ,  U,V,W)
C
      X=IX
      Y=IY
      Z=IZ
      GO TO 30
C
      ENTRY AC2TD(X,Y,Z,  U,V,W)
C
   30 W=Z*ZW
      V=W*WV+Y*YV
      U=-W*WX-V*VX+X
      IF(NSYS.NE.4) RETURN
      U1=U
      V1=V
      U=(2.0*U1-V1)/3.0
      V=(2.0*V1-U1)/3.0
      RETURN
C
C  TRANSFORM PLANE FROM TRICLINIC(H,K,L) TO CARTESIAN(X,Y,Z)
C
      ENTRY IT2CP(IH,IK,IL,  X,Y,Z)
C
   40 AH=IH
      AK=IK
      AL=IL
      GO TO 45
C
      ENTRY AT2CP(AH,AK,AL,  X,Y,Z)
C
   45 Z=AL*ZW-AK*PKTL-AH*PHTL
      Y=AK*YV-AH*PHTK
      X=AH
      RETURN
C
C  TRANSFORM PLANE FROM CARTESIAN(X,Y,Z) TO TRICLINIC(AH,AK,AL)
C
      ENTRY IC2TP(IX,IY,IZ,  AH,AK,AL)
C
```

(continued)

```
   50 X=IX
      Y=IY
      Z=IZ
      GO TO 55
C
      ENTRY AC2TP(X,Y,Z,  AH,AK,AL)
C
   55 AH=X
      AK=Y*VY+AH*VX
      AL=WZ*(Z+AK*PKTL+AH*PHTL)
      RETURN
      END
```

***** SUBROUTINE ATMPSB *****

```
      SUBROUTINE ATMPSB (VOLUME,WAVE,RELMAS,KV,IND)
C
      IMPLICIT  REAL*8(A-G,O-Z),  INTEGER*4(H-N)
      COMMON /UNITS/ NCARD,LNPTR
      DIMENSION    POSN(3,8,10),NPOSN(10),AA(4,10),BB(4,10),CC(10),
     &             NSYMB(10),FILL(8,10),DBYE(10),DBYEX(10)
      REAL*8  KV
C
      ZERO=0.0D+00
      ONE=1.0D+00
      TWO=2.0D+00
      FOUR=4.0D+00
      PY=FOUR*DATAN(ONE)
      DM6=1.0D-06
C
    1 READ(NCARD,101) WAVE,NATOMS
  101 FORMAT(F10.5,I2)
      CALL WAVESP(WAVE,KV,RELMAS,IND)
    2 WRITE(LNPTR,3)
    3 FORMAT( //' SUBROUTINE ATMPSB,   VERSION 1 : 12TH JULY,1973'/
     & '  READS IN  DATA RELEVANT TO ATOMIC POSITIONS IN THE LATTICE, FI
     &LLING FACTORS, AND SCATTERING FACTORS.'/
     &' ENTRY FCALSB  CALCULATES THE STRUCTURE FACTOR & EXTINCTION DISTA
     &NCE FOR THE GIVEN INDICES '//T6,'NO.',T14,'SYMBOL',T22,'A.N.',
     &T36,'POSITION / FILL.'/)
      DO 8  L=1,NATOMS
    4 READ(NCARD,104)     NPOSN(L),NSYMB(L),NAN,DBYE(L)
  104 FORMAT(I2,A4,I2,2X,F10.5)
      NP=NPOSN(L)
    5 READ(NCARD,105)    ((POSN(J,K,L),J=1,3),FILL(K,L),K=1,NP)
  105 FORMAT(16F5.3)
      DO 15 J=1,NP
      IF(FILL(J,L).LT.DM6)  FILL(J,L)=ONE
   15 CONTINUE
      WRITE(LNPTR,204)
     &            L,NSYMB(L),NAN,((POSN(J,K,L),J=1,3),FILL(K,L),K=1,NP)
  204 FORMAT(1H0,5X,I1,7X,1A4,4X,I2,T28,3(3(F5.3,1X),'/',F5.3,4X)/
     1,1H ,T28, 3(3(F5.3,1X),'/',F5.3,4X)    /
     2,1H ,T28, 3(3(F5.3,1X),'/',F5.3,4X)    /
     3,1H ,T28, 3(3(F5.3,1X),'/',F5.3,4X)    )
      DO 16 J=1,NP
   16 FILL(J,L)=FILL(J,L)*RELMAS
      READ(NCARD,106)  ((AA(J,L),BB(J,L)),J=1,4),CC(L)
```

(continued)

```
  106 FORMAT(9F8.0)
    8 CONTINUE
      WRITE(LNPTR,205)
  205 FORMAT(//1H ,'          SCATTERING EQUATION CONSTANTS'//1H ,'ATOM N
     10.  SYMBOL',T20,'DEBYE',T34,'A1',T44,'B1',T54,'A2',T64,'B2',T74,
     2  'A3',T84,'B3',T94,'A4',T104,'B4',T114,'C')
      DO 9 L = 1,NATOMS
      WRITE(LNPTR,206)L,NSYMB(L),DBYE(L),((AA(I,L),BB(I,L)),I=1,4),CC(L)
  206 FORMAT(/1H ,4X,I1,T13,1A4,T20,F5.3,T28,9(2X,F7.4,1X))
    9 CONTINUE
      RETURN
C
      ENTRY FCALSB  (NH,NK,NL,D,FHKL,TO )
C
      X=ONE/(TWO*D)
      SINTHE=WAVE*X
      IF(SINTHE.GE.ONE) GO TO 70
            THETA=DARSIN(SINTHE)
   40 COSFX=ZERO
      SINFX=ZERO
      DO 30 J=1,NATOMS
   30 DBYEX(J)=DEXP(-DBYE(J)*X*X)
C
C     CALCULATE OVER NUMBER OF ATOMS IN UNIT CELL
C
      DO 60  J1= 1,NATOMS
C
C     CALCULATES ELECTRON SCATTERING FACTOR FOR SELECTED ATOM
C     USING THE ANALYTICAL REPRESENTATION OF SMITH & BURGE
C     (ACTA.CRYST.) ,1962,VOL15,PAGE182, WHICH SEE FOR CONSTANTS
      F=CC(J1)
      DO 50 J=1,4
   50 F=F+AA(J,J1)*DEXP(-BB(J,J1)*X*X)
      G=DBYEX(J1)
      J2= NPOSN (J1)
C
C     CALCULATE OVER EACH POSITION OF EACH ATOMIC SPECIES
C
      DO 60  J3= 1,J2
             PHI=TWO*PY*(NH*POSN(1,J3,J1)+NK*POSN(2,J3,J1)+
     1                   NL*POSN(3,J3,J1))
      DPOS=F*G*FILL(J3,J1)
      COSFX=COSFX+DCOS(PHI)*DPOS
      SINFX=SINFX+DSIN(PHI)*DPOS
   60 CONTINUE
C
C
      FHKL= DSQRT(COSFX*COSFX + SINFX*SINFX)
      IF(FHKL.LT.DM6)  GO TO 70
      TO=PY*VOLUME*DCOS(THETA)/(WAVE *FHKL)
      RETURN
   70 TO=-ONE
      RETURN
      ENTRY FOOOSB(FHKL,TO)
      DO 75 J=1,NATOMS
   75 DBYE(J)=ZERO
      NH=0
      NK=0
      NL=0
      X=0.0
```

(continued)

```
      THETA=0.0
      GO TO 40
      END

                    *****     SUBROUTINE ATMPDT     *****

      SUBROUTINE ATMPDT   (VOLUME,WAVE,RELMAS,KV,IND)
C
      IMPLICIT  REAL*8(A-G,O-Z),  INTEGER*4(H-N)
      DIMENSION      POSN(3,8,10),NPOSN(10),TABLE(27,10),XS(27) ,
     &              NSYMB(10),FILL(8,10),DBYE(10),DBYEX(10)
      DATA  XS/0.0,0.05,0.10,0.15,0.20,0.25,0.30,0.35,0.40,0.45,0.50,
     &  0.6,0.7,0.8,0.9,1.0,1.2,1.4,1.6,1.8,2.0,2.5,3.0,3.5,4.0,5.0,6.0/
      COMMON /UNITS/ NCARD,LNPTR
      REAL*8  KV
C
      ZERO=0.0D+00
      ONE=1.0D+00
      TWO=2.0D+00
      FOUR=4.0D+00
      PY=FOUR*DATAN(ONE)
      DM6=1.0D-06
C
    1 READ(NCARD,101) WAVE,NATOMS
  101 FORMAT(F10.5,I2)
      CALL WAVESP(WAVE,KV,RELMAS,IND)
    2 WRITE(LNPTR,3)
    3 FORMAT( //'  SUBROUTINE ATMPDT,   VERSION 1 : 21ST. NOV.1974'//
     & '  READS IN  DATA RELEVANT TO ATOMIC POSITIONS IN THE LATTICE, FI
     &LLING FACTORS, AND SCATTERING FACTORS.'/
     &' ENTRY FCALDT  CALCULATES THE STRUCTURE FACTOR & EXTINCTION DISTA
     &NCE FOR THE GIVEN INDICES '//T6,'NO.',T14,'SYMBOL',T22,'A.N.',
     &T36,'POSITION / FILL.'/)
      DO 8  L=1,NATOMS
    4 READ(NCARD,104)      NPOSN(L),NSYMB(L),NAN,DBYE(L)
  104 FORMAT(I2,1A4,I2,2X,F10.5)
      NP=NPOSN(L)
    5 READ(NCARD,105)     ((POSN(J,K,L),J=1,3),FILL(K,L),K=1,NP)
  105 FORMAT(16F5.3)
      DO 15 J=1,NP
      IF(FILL(J,L).LT.DM6)  FILL(J,L)=ONE
   15 CONTINUE
      WRITE(LNPTR,204)
     &             L,NSYMB(L),NAN,((POSN(J,K,L),J=1,3),FILL(K,L),K=1,NP)
  204 FORMAT(1H0,5X,I1,7X,1A4,4X,I2,T28,3(3(F5.3,1X),'/',F5.3,4X)/
     1,1H ,T28, 3(3(F5.3,1X),'/',F5.3,4X)     /
     2,1H ,T28, 3(3(F5.3,1X),'/',F5.3,4X)     /
     3,1H ,T28, 3(3(F5.3,1X),'/',F5.3,4X)     )
      DO 16 J=1,NP
   16 FILL(J,L)=FILL(J,L)*RELMAS
      READ(NCARD,106)(TABLE(J,L),J=1,27)
  106 FORMAT(7F10.7)
    8 CONTINUE
      WRITE(LNPTR,205)
  205 FORMAT(//1H ,'          SCATTERING EQUATION CONSTANTS'//1H ,'ATOM N
     10.  SYMBOL',T20,'DEBYE',T54,'(THETA/WAVE) : SCATTERING FACTOR')
      DO 9 L = 1,NATOMS
      WRITE(LNPTR,206)  L,NSYMB(L),DBYE(L),(XS(I),TABLE(I,L),I=1,27)
```

(continued)

```
  206 FORMAT(/1H ,4X,I1,T13,1A4,T20,F5.3/
     &    T11,6(F5.2,' : ',F7.4,4X)  /
     &    T11,6(F5.2,' : ',F7.4,4X)  /
     &    T11,6(F5.2,' : ',F7.4,4X)  /
     &    T11,6(F5.2,' : ',F7.4,4X)  /
     &    T11,6(F5.2,' : ',F7.4,4X)  )
      9 CONTINUE
        RETURN
C
        ENTRY FCALDT (NH,NK,NL,D,FHKL,TO )
C
        X=ONE/(TWO*D)
        SINTHE=WAVE*X
        IF(SINTHE.GE.ONE) GO TO 70
              THETA=DARSIN(SINTHE)
     40 COSFX=ZERO
        SINFX=ZERO
        DO 45 J=1,NATOMS
     45 DBYEX(J)=DEXP(-DBYE(J)*X*X)
C
C    CALCULATE OVER NUMBER OF ATOMS IN UNIT CELL
C
        DO 60  J1= 1,NATOMS
C   CALCULATE ELECTRON SCATTERING FACTOR FOR SELECTED ATOM USING THE
C   TABULATED VALUES OF DOYLE & TURNER, ACTA.CRYST.1968,A4,390
C   WHICH SEE FOR VALUES.
C   FIND TABLE POSITION
        DO 20 J=1,27
        K=J
        IF(X.LT.XS(J)) GO TO 21
     20 CONTINUE
        K=27
     21 CONTINUE
        DEL=XS(K)-XS(K-1)
        DELX=X-XS(K-1)
        IF(DELX*TWO.LT.DEL) K=K-1
        IF(K.LT.2) K=2
        IF(K.GT.26) K=26
C   CALCULATE  F USING QUADRATIC INTERPOLATION OF THE TABLE
        XSA=XS(K-1)
        XSB=XS(K)
        XSC=XS(K+1)
        TABA=TABLE(K-1,J1)
        TABB=TABLE(K,J1)
        TABC=TABLE(K+1,J1)
        F=TABA*(X-XSB)*(X-XSC)/((XSA-XSB)*(XSA-XSC))
       & +TABB*(X-XSA)*(X-XSC)/((XSB-XSA)*(XSB-XSC))
       & +TABC*(X-XSA)*(X-XSB)/((XSC-XSA)*(XSC-XSB))
C
        J2= NPOSN (J1)
        G=DBYEX(J1)
C
C    CALCULATE OVER EACH POSITION OF EACH ATOMIC SPECIES
C
        DO 60  J3= 1,J2
                PHI=TWO*PY*(NH*POSN(1,J3,J1)+NK*POSN(2,J3,J1)+
      1                     NL*POSN(3,J3,J1))
        DPOS=F*G*FILL(J3,J1)
        COSFX=COSFX+DCOS(PHI)*DPOS
        SINFX=SINFX+DSIN(PHI)*DPOS
```

(continued)

```
   60 CONTINUE
C
      FHKL= DSQRT(COSFX*COSFX + SINFX*SINFX)
      IF(FHKL.LT.DM6) GO TO 70
      TO=PY*VOLUME*DCOS(THETA)/(WAVE *FHKL)
      RETURN
   70 TO=-ONE
      RETURN
      ENTRY FOOOOT(FHKL,TO)
      DO 75 J=1,NATOMS
   75 DBYE(J)=ZERO
      NH=0
      NK=0
      NL=0
      X=0.0
      THETA=0.0
      GO TO 40
      END
```

```
                *****     SUBROUTINE WAVESP     *****


      SUBROUTINE WAVESP(WAVE,KV,RELMAS,IND)
      IMPLICIT  REAL*8(A-G,O-Z),  INTEGER*4(H-N)
      COMMON/UNITS/NCARD,LNPTR
      REAL*8 KV,AX/0.9789D-06/,BX/1.226D+01/,ZERO/0.0D+00/,ONE/1.0D+00/,
     &       TWO/2.0D+00/,THOU/1.0D+03/
      IF(WAVE.LE.ZERO) WAVE=200.D+00
      IF(WAVE.GE.5.0D+00) GO TO 12
      IF(WAVE.LT.0.2D+00) GO TO 11
C*  X-RAY WAVELENGTH SPECIFIED          IND=2
   10 RELMAS=ONE
      KV=ZERO
      IND=2
      WRITE(LNPTR,110) WAVE,IND
  110 FORMAT(/' SUBROUTINE WAVESP'/
     &T5,'X-RAY WAVELENGTH OF ',F8.5,' ANGSTROMS WAS SPECIFIED,  IND= ',
     &I2/)
      RETURN
C*  ELECTRON WAVELENGTH SPECIFIED       IND=3
   11 TO=TWO*BX/WAVE
      TO=TO*TO
      EMF=(-ONE+DSQRT(ONE+AX*TO))/(TWO*AX)
      KV=EMF/THOU
      RELMAS=ONE + TWO*AX*EMF
      IND=3
      WRITE(LNPTR,111) WAVE,KV,RELMAS,IND
  111 FORMAT(/' SUBROUTINE WAVESP'/
     &T5,'ELECTRON WAVELENGTH OF ',F7.5,' ANGSTROMS WAS SPECIFIED,  APRO
     &XIMATE ACCELERATING VOLTAGE = ',F6.1,' KILOVOLTS'/
     &T5,'RELATIVISTIC MASS CORRECTION FACTOR = ',F8.5,10X,'IND= ',I2/)
      RETURN
C*  KV. OF ELECTRON BEAM SPECIFIED      IND=1
   12 KV=WAVE
      EMF=THOU*KV
      WAVE=BX/DSQRT(EMF*(ONE + AX*EMF))
      RELMAS=ONE + TWO*AX*EMF
      IND=+1
      WRITE(LNPTR,112) KV,WAVE,RELMAS,IND
```

(continued)

```
  112 FORMAT(/' SUBROUTINE WAVESP'/
     &T5,'ELECTRON ACCELERATING VOLTAGE OF ',F6.1,' KILOVOLTS WAS SPECIF
     &IED, WAVELENGTH = ',F7.5,' ANGSTROMS'/
     &T5,'RELATIVISTIC MASS CORRECTION FACTOR = ',F8.5,10X,'IND= ',I2/)
      RETURN
      END
```

***** SUBROUTINE INDEX *****

```
      SUBROUTINE INDEX(H,K,L,NTST)
C
      IMPLICIT  REAL*8(A-G,O-Z),  INTEGER*4(H-N)
      LOGICAL*1 CHARS(12),NUM(13),KABS
C
      COMMON/XTLCOM/ A,B,C,ALFA,BETA,GAMA,NSYS
      COMMON/INDCOM/ IH,IK,IL,ITST,KOUNT,INDMAX,OMIN,NPARM,
     &               NSABS,NMULT,NORDER,JABS,KABS
      COMMON/SYMCOM/CHARS,NUM
      COMMON/INDVAL/HMAX,HMIN,KMAX,KMIN,LMAX,LMIN,NORD(4)
      COMMON/UNITS/ NCARD,LNPTR
C
      IENT=3
      J=IABS(H) + IABS(K) + IABS(L)
      IF(J.EQ.0) GO TO 660
      IENT=1
    3 IF(NTST.LT.3) GO TO 2
      NTST=4
      RETURN
    2 CONTINUE
      N=1
      NTST=0
    1 J=NORD(N)
      N=N+1
      GO TO (10,20,30,50),J
   10 H=H+1
      IF(H.LE.HMAX) GO TO 50
      H=HMIN
      GO TO 1
   20 K=K+1
      IF(K.LE.KMAX) GO TO 50
      K=KMIN
      GO TO 1
   30 L=L+1
      IF(L.LE.LMAX) GO TO 50
      L=LMIN
      GO TO 1
   50 IF(H.GE.HMAX) NTST=NTST+1
      IF(K.GE.KMAX) NTST=NTST+1
      IF(L.GE.LMAX) NTST=NTST+1
      KOUNT=IABS(H)+IABS(K)+IABS(L)
      IF(KOUNT.EQ.0) GO TO 3
      IF(KOUNT.GT.INDMAX) GO TO 3
C*  CHECK FOR SYSTEMATIC ABSENCE
  200 MTST=0
      GO TO (210,220),NSABS
  210 CALL SYSABS(H,K,L,JABS,MTST)
      IF(MTST.NE.0) GO TO 3
  220 GO TO (230,240),NMULT
```

(continued)

```
  230 CALL MULTPL(H,K,L,JABS,MTST)
      IF(MTST.NE.0) GO TO 3
  240 GO TO (250,270,300),NORDER
  250 IF(NSYS.EQ.4) GO TO 260
      IF(L.GT.K) GO TO 3
  260 IF(L.LT.0) GO TO 3
      IF(K.GT.H) GO TO 3
      IF(K.LT.0) GO TO 3
      IF(H.LT.0) GO TO 3
      GO TO 300
  270 IF(H.LT.0) GO TO 3
      IF(H.GT.0) GO TO 300
      IF(K.LT.0) GO TO 3
      IF(K.GT.0) GO TO 300
      IF(L.LT.0) GO TO 3
  300 IF(IENT.EQ.1) RETURN
      IH=H
      IK=K
      IL=L
      ITST=NTST
      IF(IENT.EQ.2)
     &WRITE(LNPTR,301) HMIN,HMAX,KMIN,KMAX,LMIN,LMAX,IH,IK,IL,INDMAX,
     &  NPARM,DMIN,KABS,JABS,NSABS,NMULT,NORDER,(NORD(I),I=1,3)
  301 FORMAT(//' SUBROUTINE INDEX'/T5,'HMIN  = ',I3,T25,'HMAX  = ',I3,
     &  T45,'KMIN  = ',I3,T65,'KMAX  = ',I3,T85,'LMIN  = ',I3,T105,
     &  'LMAX  = ',I3/T5,'IH    = ',I3,T25,'IK    = ',I3,T45,'IL    = ',
     &  I3,T65,'INDMAX = ',I3,T85,'NPARM  = ',I3,T105,'DMIN   = ',F9.6/
     &  T5,'KABS   = ',1A2,T25,'JABS   = ',I3,T45,'NSABS  = ',I3,T65,
     &  'NMULT  = ',I3,T85,'NORDER = ',I3,T105,'NORD   = ',3I2)
      RETURN
C
      ENTRY INDSET
C
      READ(NCARD,610) KABS,NPARM,(NORD(I),I=1,3),HMIN,HMAX,KMIN,KMAX,
     &                LMIN,LMAX,INDMAX,DMIN
  610 FORMAT(1A1,1I1,3X,  3I1,2X,  6(2X,I3),  3X,I2,  F5.3)
COLUMNS:      1 , 2 ,     6---10   11---40    41-45   46-50
C*
C* NSABS=1      CHECK SYSABS BEFORE RETURN
C* NSABS=0      SET ACCORDING TO JABS
C* NSABS=2      CHECK SYSABS IN MAIN PROGRAM, DEPENDS UPON PERMUTATION.
C*
C* NMULT=1      NO MULTIPLES RETURNED FROM INDEX.
C* NMULT=2      MULTIPLE VALUES ALLOWED.
C*
C* NORDER=1     GENERAL ORDERING OF INDICES.   H.GE.K.GE.L.GE.0
C* NORDER=2     ORDERING WITH SOME RELAXATIONS, DEPENDING ON ZERO VALUES.
C* NORDER=3     NO ORDERING, ALL INDICES PASSED.
C*
      IENT=2
      IF(INDMAX.LE.0) INDMAX=100
C* CHECK RANGE OF INDICES
  612 IF(HMAX.GT.HMIN) GO TO 613
      HMAX=4
      HMIN=0
  613 IF(KMAX.GT.KMIN) GO TO 614
      KMAX=HMAX
      KMIN=-HMAX
  614 IF(LMAX.GT.LMIN) GO TO 615
      LMAX=KMAX
```

(continued)

```
      LMIN=KMIN
  615 CONTINUE
C* CHECK SYSTEMATIC ABSENCES
      K=0
      DO 620 J=1,9
      IF(KABS.EQ.CHARS(J)) K=J
      IF(KABS.EQ.NUM(J))   K=J
      IF(K.NE.0) GO TO 621
  620 CONTINUE
      K=10
      IF(KABS.EQ.CHARS(11)) K=11
  621 JABS=K
C* CHECK NORD, THE SEQUENCE FOR ALTERING THE INDICES.
      L=1
      J=0
      DO 630 I=1,3
      K=NORD(I)
      L=L*K
  630 J=J+K
      IF(J.NE.6) GO TO 640
      IF(L.EQ.6) GO TO 650
  640 NORD(1)=3
      NORD(2)=2
      NORD(3)=1
  650 NORD(4)=4
C*  NSABS MAY DEPEND UPON PERM. OF INDICES
  660 IF(NSABS.GT.0) GO TO 670
      K=(JABS-1)*(JABS-2)*(JABS-3)*(JABS-6)*(JABS-7)*(JABS-11)
      NSABS=1
      IF(K.EQ.0) NSABS=2
      IF(K.EQ.0) NMULT=2
C* SET STARTING VALUES OF INDICES.
  670 H=HMIN
      K=KMIN
      L=LMIN
      NTST=0
      KABS=CHARS(JABS)
      GO TO 50
      END
```

***** SUBROUTINE SYSABS *****

```
      SUBROUTINE SYSABS(H,K,L,JABS,MTST)
      IMPLICIT INTEGER*4(A-Z)
  200 CONTINUE
      MTST=0
      GO TO (210,220,230,240,250,260,270,280,290,300,310),JABS
  210 IF((L+K).NE.((L+K)/2)*2) MTST=1
      GO TO 300
  220 IF((H+L).NE.((H+L)/2)*2) MTST=1
      GO TO 300
  230 IF((H+K).NE.((H+K)/2)*2) MTST=1
      GO TO 300
  240 IF(H.EQ.(H/2)*2) GO TO 245
      IF(K.EQ.(K/2)*2) MTST=1
      IF(L.EQ.(L/2)*2) MTST=1
      GO TO 300
  245 IF(K.NE.(K/2)*2) MTST=1
```

(continued)

```
      IF(L.NE.(L/2)*2) MTST=1
      GO TO 300
  250 IF((H+K+L).NE.((H+K+L)/2)*2) MTST=1
      GO TO 300
  260 IF((K+L-H).NE.((K+L-H)/3)*3) MTST=1
      GO TO 300
  270 IF((H+L-K).NE.((H+L-K)/3)*3) MTST=1
      GO TO 300
  280 IF(L.EQ.2*(L/2)) GO TO 300
      J=H+2*K
      IF(J.EQ.3*(J/3)) MTST=1
      GO TO 300
  290 J=H*H + K*K + L*L
      MTST=1
      IF(J.EQ.((J/8)*8)) MTST=0
      IF(J.EQ.((J/8)*8+3)) MTST=0
  300 CONTINUE
      RETURN
  310 CALL XTSYAB(H,K,L,MTST)
      IF(MTST.GE.0) GO TO 315
      MTST=0
      JABS=10
  315 RETURN
      END
```

******* SUBROUTINE XTSYAB ********

```
      SUBROUTINE XTSYAB (I1,I2,I3,MTST)
      IMPLICIT INTEGER*4(A-Z)
      COMMON/UNITS/ NCARD,LNPTR
      WRITE(LNPTR,1)
    1 FORMAT('  ENTRY TO DUMMY SUBROUTINE XTSYAB;'/'  THIS SUBROUTINE SH
     &OULD BE PROVIDED BY THE USER FOR SPECIAL SYSTEMATIC ABSENCES'/
     & '  NOT COVERED BY THE RANGE PROVIDED IN SUBROUTINE SYSABS'//
     &'  SYSTEMATIC ABSENCES ARE NOT CONSIDERED TO BE PRESENT')
      MTST=-1
      RETURN
      END
```

******* SUBROUTINE MULTPL *******

```
      SUBROUTINE MULTPL(H,K,L,JABS,NTST)
      IMPLICIT  INTEGER*4 (A-Z)
      NTST=0
   10 CALL COMFAC(H,K,L,J)
   15 IF(J.EQ.1) RETURN
      HT=H/J
      KT=K/J
      LT=L/J
      J=J-1
   20 DO 30 JT=1,J
      HX=HT*JT
      KX=KT*JT
      LX=LT*JT
      CALL SYSABS(HX,KX,LX,JABS,MTST)
      IF(MTST.EQ.0) GO TO 40
```

(continued)

```
   30 CONTINUE
      RETURN
   40 NTST=1
C  A SUBMULTIPLE HAS BEEN FOUND WHICH IS NOT A SYSTEMATIC ABSENCE.
C  THIS SUBMULTIPLE HAS BEEN (OR WILL BE) PROCESSED IN ITS DUE TURN.
      RETURN
      END
```

***** SUBROUTINE COMFAC *****

```
      SUBROUTINE COMFAC (I1,I2,I3,J)
C     DETERMINES H.C.F. OF FIRST THREE TERMS AND RETURNS VALUE IN LAST TERM
C    TERMS CONTAINING ZEROS ARE IGNORED UNLESS ALL ARE ZERO,
C    IN WHICH CASE A VALUE OF -100 IS RETURNED
      IMPLICIT INTEGER*4(A-Z)
      I=IABS(I1)
      J=IABS(I2)
      K=IABS(I3)
      IF(I*J.EQ.0) GO TO 20
      CALL FACTOR(I,J)
      IF (J.EQ.1) GO TO 10
      IF (K.EQ.0) GO TO 10
      CALL FACTOR(J,K)
   10 RETURN
   20 J=J+I
      IF (J.EQ.0) GO TO 30
      IF (K.EQ.0) GO TO 40
      CALL FACTOR(J,K)
      RETURN
   30 J=K
      IF (K.EQ.0)  J=-100
   40 RETURN
      END
```

***** SUBROUTINE FACTOR *****

```
      SUBROUTINE FACTOR(I,J)
C     REDUCES BOTH TERMS TO THEIR HIGHEST COMMON FACTOR
      IMPLICIT INTEGER*4(A-Z)
   10 IF (J.EQ.0) GO TO 20
      I=I-(I/J)*J
      ITEMP=I
          I=J
          J=ITEMP
       GO TO 10
   20 J=I
      RETURN
      END
```

(continued)

***** SUBROUTINE PERM *****

```
      SUBROUTINE PERM(AH,AK,AI,AL,N,NZ )
      IMPLICIT INTEGER*4(A-Z)
  100 IF(NZ.LT.0) RETURN
    1 N=N+1
      NY=(N-1)/6
      NX=N-6*NY
      NY=NY+1
      IF(NSYS.EQ.4)GO TO 18
      GO TO (5,2,3,2,4,2),NX
    2 T=AK
      AK=AL
      AL=T
      GO TO 6
    3 T=AH
      AH=AL
      AL=AK
      AK=T
      GO TO 6
    4 T=AH
      AH=AK
      AK=AL
      AL=T
      GO TO 6
    5 T=AH
      AH=AL
      AL=T
    6 IF(NX.NE.1)GO TO 10
      GO TO (10,9,8,9,7,9,8,9,7),NY
    7 AH=-AH
    8 AK=-AK
    9 AL=-AL
   10 GO TO(14,11,13,12,14,11,13),NZ
   11 IF(NX.EQ.6)GO TO 14
      IF(NX.EQ.4)GO TO 14
   12 IF(NX.EQ.1)GO TO 14
      GO TO 1
   13 IF(NX.EQ.6)GO TO 14
      IF(NX.EQ.3)GO TO 14
      IF(NX.EQ.1)GO TO 14
      GO TO 1
   14 IF(NZ.LE.4)GO TO 17
      IF(NZ.LE.6)GO TO 16
   15 IF(NY.EQ.1)GO TO 17
      IF(NY.EQ.5)GO TO 17
      IF(NY.EQ.9)GO TO 17
      GO TO 1
   16 IF(NY.EQ.(NY/2)*2)GO TO 1
   17 IF(N.GT.NV) NZ=-1
      RETURN
   18 CONTINUE
      GO TO (118,120,120,119,122,121,121), NZ
  119 N=N+11
      AL=-AL
      GO TO 24
  120 IF(NX.LT.4) GO TO 118
      N=N+3
```

(continued)

```
      T=-AK
      AK=-AI
      AI=-AH
      AH=T
      IF(N.EQ.13) AL=-AL
      IF(N.EQ.25) AL=-AL
      GO TO 24
  121 IF(N.LT.10) GO TO 120
      T=AI
      AI=-AH
      AH=-AK
      AK=-T
      N=25
      NZ=-1
      RETURN
  122 IF(N.LT.13) GO TO 118
      AH=-AI
      AK=-AK
      N=25
      NZ=-1
      RETURN
  118 GO TO (19,20,20,19,20,20),NX
   19 T=AI
      AI=AH
      AH=T
      GO TO 21
   20 T=AH
      AH=AK
      AK=AI
      AI=T
   21 IF(NX.NE.1) GO TO 24
      GO TO (24,23,22,23,22),NY
   22 AL=-AL
   23 AH=-AH
      AK=-AK
      AI=-AI
   24 IF(N.IE.NV) RETURN
      NZ=-1
   25 RETURN
      ENTRY PRMSET(H,K,I,L,N,NZ,NSYS)
      N=1
   37 NZ=1
      IF(L.EQ.0) NZ=5
      IF(H.EQ.K) NZ=NZ+1
      IF(NSYS.EQ.4) GO TO 38
      IF(K.EQ.L) NZ=NZ+2
      NV=48
      RETURN
   38 IF(K.EQ.0) NZ=NZ+2
      NV=24
      RETURN
      END
```

(continued)

```
                *****     SUBROUTINE BLOCK DATA     *****

      BLOCK DATA
      IMPLICIT REAL*8(A-H,O-Z), INTEGER*4(I-N)
      LOGICAL*1 CHARS(12),NUM(13)
      COMMON/SYMCOM/CHARS,NUM
      DATA  NUM /'1','2','3','4','5','6','7','8','9','0','*',' ','-' /
      DATA CHARS/'A','B','C','F','I','R','R','H','D','P','*'         /
      COMMON/UNITS/NCARD,LNPTR
      DATA NCARD,LNPTR/1,3/
      END

                *****     SUBROUTINE INSRT8     *****

      SUBROUTINE INSRT8(N,NDIM,NTST,D,DAR,H,K,I,L,RH,RK,RL,
     &HAR,KAR,IAR,LAR,RHAR,RKAR,RLAR)
      IMPLICIT  REAL*8(A-H,O-Z),  INTEGER*4(I-N)
      COMMON/INSRT/ERROR,IDUP
C
C*  TO REJECT DUPLICATE 'D' VALUES NDUP=1
C*  TO INSERT DUPLICATE 'D' VALUES NDUP=2
C*  NTST=-1 ,(N.GE.NDIM) ARRAYS FILLED PRIOR TO THIS CALL
C*  NTST=0    VALUE INSERTED
C*  NTST=+1   VALUE NOT INSERTED AS EQUAL VALUE ALREADY ENTERED.
C
      INTEGER*4 RHAR(NDIM),RKAR(NDIM),RLAR(NDIM),RH,RK,RL
      INST=8
      JRH=RH
      JRK=RK
      JRL=RL
      GO TO 10
C
      ENTRY INSRT5(N,NDIM,NTST,D,DAR,H,K,I,L,HAR,KAR,IAR,LAR)
      INST=5
   10 CONTINUE
      INTEGER*4  IAR(NDIM),I
      JI=I
      GO TO 20
C
      ENTRY INSRT4(N,NDIM,NTST,D,DAR,H,K,L,HAR,KAR,LAR)
      INST=4
   20 CONTINUE
      INTEGER*4  HAR(NDIM),KAR(NDIM),LAR(NDIM),H,K,L
      JH=H
      JK=K
      JL=L
      GO TO 30
C
      ENTRY INSRT1(N,NDIM,NTST,D,DAR)
      INST=1
   30 CONTINUE
      DIMENSION DAR(NDIM)
      DM7=1.0D-07
      AD=D
      NTST=0
      IF(N.LT.NDIM) GO TO 100
```

(continued)

```
      NTST=-1
      RETURN
  100 CONTINUE
      IF(N.EQ.0) GO TO 150
      DO 130 J=1,N
      JA=J
      IF(DABS(AD).LT.DM7) GO TO 123
      IF(DABS((AD-DAR(J))/AD).LT.ERROR) GO TO 125
      GO TO 124
  123 IF(DABS(AD-DAR(J)).LT.DM7) GO TO 125
  124 IF(DAR(J).LT.AD) GO TO 135
      GO TO 130
  125 GO TO (170,124),IDUP
  130 CONTINUE
      GO TO 150
  135 N=N+1
      DO 140 J=JA,N
      T= DAR(J)
      DAR(J)=AD
      AD=T
      IF(INST.LE.1) GO TO 140
      NT=HAR(J)
      HAR(J)=JH
      JH=NT
      NT=KAR(J)
      KAR(J)=JK
      JK=NT
      NT=LAR(J)
      LAR(J)=JL
      JL=NT
      IF(INST.LE.4) GO TO 140
      NT=IAR(J)
      IAR(J)=JI
      JI=NT
      IF(INST.LE.5) GO TO 140
      NT=RHAR(J)
      RHAR(J)=JRH
      JRH=NT
      NT=RKAR(J)
      JRK=NT
      RKAR(J)=JRK
      NT=RLAR(J)
      RLAR(J)=JRL
      JRL=NT
  140 CONTINUE
      RETURN
  150 N=N+1
      DAR(N)=AD
      IF(INST.LE.1) RETURN
      HAR(N)=JH
      KAR(N)=JK
      LAR(N)=JL
      IF(INST.LE.4) RETURN
      IAR(N)=JI
      IF(INST.LE.5) RETURN
      RHAR(N)=JRH
      RKAR(N)=JRK
      RLAR(N)=JRL
      RETURN
  170 NTST=+1
```

(continued)

```
      RETURN
      END

                    *****     SUBROUTINE FUNZ     *****

       SUBROUTINE FUNZ (A1,B1,C1,E1,F1,G1,Z,NTYPE)
      IMPLICIT  REAL*8(A-G,O-Z),  INTEGER*4(H-N)
C
C        SOLVES THE EQN.: A*X*X+B*Y*Y+C*Z*Z+2*E*Y*Z+2*F*X*Z+2*G*X*Y=0
C
C        FOR A FIXED VALUE OF Z:
C        X IS DETERMINED AS A FUNCTION OF Y (ENTRY FUNOFY)
C        Y IS DETERMINED AS A FUNCTION OF X (ENTRY FUNOFX)
C
      DATA ZERO/0.0D+00/, ONE/1.0D+00/, TWO/2.0D+00/, DM8/1.0D-08/
      E2=TWO*E1
      F2=TWO*F1
      G2=TWO*G1
   10 EZ=E2*Z
      FZ=F2*Z
      CZ2=C1*Z*Z
      BSQ4AC=G2*G2-TWO*TWO*A1*B1
      IF(DABS(BSQ4AC).LT.DM8) BSQ4AC=ZERO
      NTYPE=1
      IF(BSQ4AC.EQ.ZERO) NTYPE=2
      IF(BSQ4AC.GT.ZERO) NTYPE=3
C* NTYPE=1, CURVE IS AN ELLIPSE.
C* NTYPE=2, CURVE IS A PARABOLA.
C* NTYPE=3, CURVE IS A HYPERBOLA.
      RETURN
C
      ENTRY FUNOFY(X,Y,NSY,JTST)
   20 JTST=+1
      A=A1
      B=FZ+G2*Y
      C=B1*Y*Y+CZ2+EZ*Y
   22 DEL=B*B-TWO*TWO*A*C
      IF(DEL.LT.DM8) GO TO 100
   24 DEL=DSQRT(DEL)
      X=(-B+ NSY*DEL)/(TWO*A)
      RETURN
C
      ENTRY FUNOFX(X,Y,NSX,JTST)
   30 JTST=+1
      A=B1
      B=EZ+G2*X
      C=A1*X*X+CZ2+FZ*X
   32 DEL=B*B-TWO*TWO*A*C
      IF(DEL.LT.DM8) GO TO 100
   34 DEL=DSQRT(DEL)
      Y=(-B+ NSX*DEL)/(TWO*A)
      RETURN
  100 JTST=-1
      RETURN
      END
```

(continued)

***** SUBROUTINE LABAL *****

```
      SUBROUTINE LABAL (H,K,I,L,A1,A2,NSYS)
C     THE ABSOLUTE VALUES OF THE INTEGERS (H,K,I,L) ARE PLACED IN A1,
C     AND THEIR SIGN IS PLACED IN A2. IF N=0 INSTEAD OF 4 THEN INTEGER I IS
C     IGNORED AND INTEGER L PLACED IN THE THIRD POSITION.
C     IF ANY INTEGER IS GREATER THAN 9, IT IS REPLACED WITH *.
      LOGICAL*1 CHARS(12),NUM(13),A1(4),A2(4)
      INTEGER*4 H,K,I,L,M,N,IB(4),NSYS
      COMMON/SYMCOM/CHARS,NUM
      IB(1)=H
      IB(2)=K
      IB(3)=I
      IB(4)=L
      N=NSYS
      IF(N.NE.4) N=3
      IF(N.EQ.3) IB(3)=L
      A1(4)= NUM(12)
      A2(4)= NUM(12)
      DO 10 J=1,N
      A2(J)= NUM(12)
      IF(IB(J).LT.0) A2(J)= NUM(13)
      M=IABS(IB(J))
      IF(M.GT.9) M=11
      IF(M.EQ.0) M=10
      A1(J)= NUM(M)
   10 CONTINUE
      RETURN
      END
```

APPENDIX B

LISTING OF MAIN PROGRAMS

APPENDIX B

LISTING OF MAIN PROGRAMS

```
C*  PROGRAM DSPACE
      IMPLICIT  REAL*8(A-G,O-Z),  INTEGER*4(H-N)
      LOGICAL*1 KABS
      INTEGER*4 RH,RK,RL,RHAR,RKAR,RLAR,RHARB,RKARB,RLARB
      COMMON/XTLCOM/ A,B,C,ALFA,BETA,GAMA,NSYS,NAME(5)
      COMMON/INDCOM/ IH,IK,IL,ITST,KOUNT,INDMAX,DMIN,NDP,
     &               NSABS,NMULT,NORDER,JABS,KABS
      COMMON/UNITS/NCARD,LNPTR
      COMMON/INSRT/ERROR,IDUP
      DIMENSION HAR(400),KAR(400),IAR(400),LAR(400),HARB(400),KARB(400),
     & IARB(400),RHAR(400),RKAR(400),RLAR(400),LARB(400),RHARB(400),
     & RKARB(400),RLARB(400),DAR(400),DARB(400)
      NDIM=400
    1 CALL XTLSYS
      IF(A.LT.0.001) GO TO 1000
      NSABS=1
      NMULT=2
      NORDER=3
      CALL INDSET
      J=0
      NUM=0
      IDUP=2
      ERROR=1.0D-05
   10 H=IH
      K=IK
      L=IL
      JTST=ITST
   20 IF(H.LT.0) GO TO 80
      CALL IDCALC(H,K,L,AID)
      IF(AID.LT.DMIN) GO TO 80
      IF(NSYS.NE.4) GO TO 70
      I=-(H+K)
      IF(JABS.NE.6) GO TO 40
   30 RH=(2*H+K+L)/3
      RK=(-H+K+L)/3
      RL=(-H-2*K+L)/3
      GO TO 50
   40 IF(JABS.NE.7) GO TO 60
      RH=(H-K+L)/3
      RK=(H+2*K+L)/3
      RL=(-2*H-K+L)/3
   50 IF(H.LT.0) GO TO 80
      IF(K.LT.0) GO TO 80
      CALL INSRT8(NUM,NDIM,NTST,AID,DARB,H,K,I,L,RH,RK,RL,HARB,KARB,
     &            IARB,LARB,RHARB,RKARB,RLARB)
      IF(NTST.LT.0) GO TO 200
   55 RHAR(NUM)=RH
      RKAR(NUM)=RK
      RLAR(NUM)=RL
      GO TO 65
   60 IF(K.LT.0) GO TO 80
      IF(L.LT.0) GO TO 80
      IF(K.GT.H) GO TO 80
      CALL INSRT5(NUM,NDIM,NTST,AID,DARB,H,K,I,L,HARB,KARB,IARB,LARB)
```

(continued)

```
      IF(NTST.LT.Ø) GO TO 2ØØ
   65 IAR(NUM)=I
      GO TO 78
   7Ø IF(NSYS.GT.3) GO TO 74
      GO TO (71,73,72),NSYS
   71 IF(L.GT.K) GO TO 8Ø
   72 IF(L.LT.Ø) GO TO 8Ø
   73 IF(K.GT.H) GO TO 8Ø
      IF(K.LT.Ø) GO TO 8Ø
      GO TO 77
   74 IF(NSYS.EQ.8) GO TO 77
      IF(NSYS.EQ.7) GO TO 76
      IF(NSYS.EQ.6) GO TO 75
      IF(K.LT.Ø) GO TO 8Ø
   75 IF(L.LT.Ø) GO TO 8Ø
      IF(H.NE.Ø) GO TO 77
      IF(K.LT.Ø) GO TO 8Ø
      GO TO 77
   76 IF(K.LT.Ø) GO TO 8Ø
      IF(H.NE.Ø) GO TO 77
      IF(L.LT.Ø) GO TO 8Ø
   77 CALL INSRT4(NUM,NDIM,NTST,AID,DARB,H,K,L,HARB,KARB,LARB)
      IF(NTST.LT.Ø) GO TO 2ØØ
   78 HAR(NUM)=H
      KAR(NUM)=K
      LAR(NUM)=L
      DAR(NUM)=AID
   8Ø CALL INDEX(H,K,L,JTST)
      IF(JTST.LT.4) GO TO 2Ø
  2ØØ CONTINUE
  2Ø1 J=Ø
C*************************************************************************
  2Ø2 WRITE(LNPTR,2Ø3) NAME,KABS
  2Ø3 FORMAT(1H1,T1Ø,'PROGRAM DSPACE',T5Ø,5A4/
     &  T5Ø,'REFLECTION RESTRICTION:  ',1A1)
      GO TO (21Ø,22Ø,23Ø,24Ø,25Ø,26Ø,27Ø,28Ø),NSYS
C*************************************************************************
  21Ø WRITE(LNPTR,211) A
  211 FORMAT(T1Ø,'CUBIC SYSTEM'/T1Ø,'A=',F7.4/)
      GO TO 3ØØ
C*************************************************************************
  22Ø WRITE(LNPTR,221) A,ALFA
  221 FORMAT(T1Ø,'RHOMBØHEDRAL SYSTEM'/
     & T1Ø,'A=',F7.4,5X,'ALPHA=',F6.2)
      GO TO 3ØØ
C*************************************************************************
  23Ø WRITE(LNPTR,231) A,C
  231 FORMAT(T1Ø,'TETRAGONAL SYSTEM'/
     & T1Ø,'A=',F7.4,5X,'C=',F7.4)
      GO TO 3ØØ
C*************************************************************************
  24Ø WRITE(LNPTR,241) A,C
  241 FORMAT(T1Ø,'HEXAGONAL SYSTEM'/
     & T1Ø,'A=',F7.4,5X,'C=',F7.4)
      GO TO 4ØØ
C*************************************************************************
  25Ø WRITE(LNPTR,251) A,B,C
  251 FORMAT(T1Ø,'ORTHORHOMBIC SYSTEM'/
     &  T1Ø,'A=',F7.4,5X,'B=',F7.4,5X,'C=',F7.4)
      GO TO 3ØØ
```

(continued)

```
C*************************************************************************
  260 WRITE(LNPTR,261) A,B,C,GAMA
  261 FORMAT(T10,'MONOCLINIC SYSTEM : 1 ST. SETTING'/
     &  T10,'A=',F7.4,5X,'B=',F7.4,5X,'C=',F7.4,5X,'GAMMA=',F6.2)
      GO TO 300
C*************************************************************************
  270 WRITE(LNPTR,271) A,B,C,BETA
  271 FORMAT(T10,'MONOCLINIC SYSTEM : 2 ND. SETTING'/
     &  T10,'A=',F7.4,5X,'B=',F7.4,5X,'C=',F7.4,5X,'BETA=' ,F6.2)
      GO TO 300
C*************************************************************************
  280 WRITE(LNPTR,281) A,B,C,ALFA,BETA,GAMA
  281 FORMAT(T10,'TRICLINIC SYSTEM'/T10,'A=',F7.4,5X,'B=',F7.4,5X,
     & 'C=',F7.4,5X,'ALPHA=',F6.2,5X,'BETA=',F6.2,5X,'GAMMA=',F6.2)
C*************************************************************************
  300 WRITE(3,301)
  301 FORMAT(/T16,'H   K   L',T30,'D',T46,'H   K   L',T60,'D')
  302 J=J+1
C     *     *     *     *     *     *     *     *     *     *     *     *     *     *
      WRITE(LNPTR,310) HAR(J),KAR(J),LAR(J),DAR(J),HARB(J),KARB(J),
     &                 LARB(J),DARB(J)
  310 FORMAT(T13,3I4,T28,F6.3,T43,3I4,T58,F6.3)
      IF(J.GE.NUM) GO TO 900
      IF(J.EQ.(J/50)*50) GO TO 202
  320 GO TO 302
C*************************************************************************
  400 IF(JABS.EQ.6) GO TO 410
      IF(JABS.EQ.7) GO TO 420
      WRITE(3,401)
  401 FORMAT(/T16,'H   K   I   L',T33,'D',T46,'H   K   I   L',T63,'D')
  402 J=J+1
      WRITE(LNPTR,405) HAR(J),KAR(J),IAR(J),LAR(J),DAR(J),
     &                 HARB(J),KARB(J),IARB(J),LARB(J),DARB(J)
  405 FORMAT(T13,4I4,T31,F6.3,T43,4I4,T61,F6.3 )
      IF(J.GE.NUM) GO TO 900
      IF(J.EQ.(J/50)*50) GO TO 202
  409 GO TO 402
C     *     *     *     *     *     *     *     *     *     *     *     *     *     *
  410 WRITE(3,411)
  411 FORMAT(T10,'OBVERSE RHOMBOHEDRAL ORIENTATION RESTRICTIONS')
      GO TO 430
  420 WRITE(3,421)
  421 FORMAT(T10,'INVERSE RHOMBOHEDRAL ORIENTATION RESTRICTIONS')
  430 WRITE(3,431)
  431 FORMAT(/T6,'HEXAGONAL',T20,'RHOMBOHEDRAL',T56,'HEXAGONAL',T70,
     &  'RHOMBOHEDRAL'/T6,'H  K  I  L',T22, 'H  K  L',T36,'D',
     &  T56,'H  K  I  L',T72,'H  K  L',T86,'D')
C     *     *     *     *     *     *     *     *     *     *     *     *     *     *
  440 J=J+1
      WRITE(LNPTR,441) HAR(J),KAR(J),IAR(J),LAR(J),RHAR(J),RKAR(J),
     & RLAR(J),DAR(J),HARB(J),KARB(J),IARB(J),LARB(J),RHARB(J),RKARB(J)
     &,RLARB(J),DARB(J)
  441 FORMAT(T4,4I3,T20,3I3,T34,F6.3,T54,4I3,T70,3I3,T84,F6.3)
      IF(J.GE.NUM) GO TO 900
      IF(J.EQ.(J/50)*50) GO TO 202
  450 GO TO 440
C*************************************************************************
  900 WRITE(LNPTR,901) NUM
  901 FORMAT(/' ',T20,'THERE ARE ',I4,' ENTRIES IN THE TABLES')
      GO TO 1
```

(continued)

```
 1000 STOP
      END
```

******* PROGRAM ANGLES *******

```
C* PROGRAM ANGLES                J.G.NAPIER.    E4
      IMPLICIT REAL*8(A-G,O-Z),  INTEGER*4(H-N)
      LOGICAL*1 KABS
      COMMON/XTLCOM/A,B,C,ALFA,BETA,GAMA,NSYS,NAME(5)
      COMMON/INDCOM/IH,IK,IL,ITST,KOUNT,INDMAX,DMIN,NDP,
     &              NSABS,NMULT,NORDER,JABS,KABS
      COMMON/UNITS/NCARD,LNPTR
      COMMON/INSRT/ERROR,IDUP
      ERROR=1.0D-07
      ZERO=0.0D+00
C
      DIMENSION ANGLES(2500),TABA(50),TABB(50),
     & HAR(50),KAR(50),IAR(50),LAR(50),IPA(50),
     & HBR(50),KBR(50),IBR(50),LBR(50),IPB(50)
C
    1 CALL XTLSYS
      IF(A.LT.ERROR) STOP
      NSABS=0
      NMULT=1
      NORDER=1
      CALL INDSET
      IF(NDP.LE.0) NDP=3
      IF(NDP.GT.3) NDP=3
    2 HX=IH
      KX=IK
      LX=IL
      NTSTA=ITST
  100 JX=1000
      IDUP=1
    3 HA=HX
      KA=KX
      LA=LX
    4 IA=-(HA+KA)
      MTST=0
      NUMA=0
      CALL PRMSET(HA,KA,IA,LA,NA,NZA,NSYS)
C*  N*=PERM. NO.       NZ* IS POSATIVE UNTILL END, THEN GOES NEGATIVE.
      IDUP=1
      IF(NSABS.NE.1) GO TO 10
    5 GO TO (6,7,7),NDP
    6 CALL IDENT(HA,KA,LA,AIDA)
      GO TO 8
    7 CALL IDCALC(HA,KA,LA,AIDA)
      IF(AIDA.LT.DMIN) GO TO 9
    8 CALL INSRT5(NUMA,50,MTST,AIDA,TABA,HA,KA,IA,LA,HAR,KAR,IAR,LAR)
    9 CALL PERM(HA,KA,IA,LA,NA,NZA)
      IF(NZA.LT.0) GO TO 12
      IF(HA.LT.0) GO TO 9
      GO TO (11,10),NSABS
   10 CALL SYSABS(HA,KA,LA,JABS,MTST)
      IF(MTST.NE.0) GO TO 9
   11 IF(NMULT.NE.2) GO TO 5
      CALL MULTPL(HA,KA,LA,JABS,MTST)
```

(continued)

```
      IF(MTST.NE.0) GO TO 9
      GO TO 5
   12 IF(NUMA.EQ.0) GO TO 50
C* NOW HAVE TABLES OF PERMED INDICES AND AID( NO DUPLICATES )
C* GENERATE SIMILAR TABLES FOR THE SECOND SET OF INDICES ( WITH DUPLICATES )
   13 HY=HX
      KY=KX
      LY=LX
      NTSTB=NTSTA
   14 HB=HY
      KB=KY
      LB=LY
      HT=HB
      KT=KB
      IT=-HT-KT
      LT=LB
   15 IB=-(HB+KB)
      MTST=0
      NUMB=0
      IDUP=2
      CALL PRMSET(HB,KB,IB,LB,NB,NZB,NSYS)
      IF(NSABS.NE.1) GO TO 21
   16 GO TO (17,18,18),NDP
   17 CALL IDENT(HB,KB,LB,AIDB)
      GO TO 19
   18 CALL IDCALC(HB,KB,LB,AIDB)
      IF(AIDB.LT.DMIN) GO TO 20
   19 CALL INSRT5(NUMB,50,MTST,AIDB,TABB,HB,KB,IB,LB,HBR,KBR,IBR,LBR)
   20 CALL PERM(HB,KB,IB,LB,NB,NZB)
      IF(NZB.LT.0) GO TO 23
      IF(HB.LT.0) GO TO 20
      GO TO (22,21),NSABS
   21 CALL SYSABS(HB,KB,LB,JABS,MTST)
      IF(MTST.NE.0) GO TO 20
   22 IF(NMULT.NE.2) GO TO 16
      CALL MULTPL(HB,KB,LB,JABS,MTST)
      IF(MTST.NE.0) GO TO 20
      GO TO 16
   23 IF(NUMB.EQ.0) GO TO 40
C* TWO SETS OF TABLES NOW READY, NOW USE THEM.
      IDUP=1
C* SELECT FIRST AVAILABLE SET OF INDICES
   30 DO 1000 JA=1,NUMA
      HA=HAR(JA)
      KA=KAR(JA)
      IA=-HA-KA
      LA=LAR(JA)
      AIDA=TABA(JA)
C* SELECT FIRST AVAILABLE SET OF THE SECOND INDICES
      NANG=0
   32 DO 2000 JB=1,NUMB
      HB=HBR(JB)
      KB=KBR(JB)
      LB=LBR(JB)
      AIDB=TABB(JB)
      GO TO (33,34,34),NDP
   33 CALL INGDIR(HA,KA,LA,AIDA,HB,KB,LB,AIDB,ANG,COSANG)
      GO TO 35
   34 CALL INGPLN(HA,KA,LA,AIDA,HB,KB,LB,AIDB,ANG,COSANG)
   35 ANG=DABS(ANG)
```

(continued)

```
      ANG=DMIN1(ANG,(180.D+00-ANG))
      IF(ANG.LT.ERROR) ANG=ZERO
      CALL INSRT1(NANG,2500,MTST,ANG,ANGLES)
      IF(NDP.NE.1) GO TO 2000
      ANG=180.0D+00-ANG
      CALL INSRT1(NANG,2500,MTST,ANG,ANGLES)
 2000 CONTINUE
  199 IF(NSYS.EQ.4) GO TO 403
      GO TO 303
C*  PAGE TITLES AND PRINTOUT
  200 CONTINUE
  201 JX=0
  202 WRITE(LNPTR,203) NAME,KABS
  203 FORMAT(1H1,T10,'PROGRAM ANGLES',T50,5A4/
     &  T50,'REFLECTION RESTRICTION:  ',1A1)
      GO TO (210,220,230,240,250,260,270,280),NSYS
  210 WRITE(LNPTR,211) A
  211 FORMAT(T10,'CUBIC SYSTEM'/T10,'A=',F7.4/)
      GO TO 300
  220 WRITE(LNPTR,221) A,ALFA
  221 FORMAT(T10,'RHOMB0HEDRAL SYSTEM'/
     & T10,'A=',F7.4,5X,'ALPHA=',F6.2)
      GO TO 300
  230 WRITE(LNPTR,231) A,C
  231 FORMAT(T10,'TETRAGONAL SYSTEM'/
     & T10,'A=',F7.4,5X,'C=',F7.4)
      GO TO 300
  240 WRITE(LNPTR,241) A,C
  241 FORMAT(T10,'HEXAGONAL SYSTEM'/
     & T10,'A=',F7.4,5X,'C=',F7.4)
      GO TO 400
  250 WRITE(LNPTR,251) A,B,C
  251 FORMAT(T10,'ORTHORHOMBIC SYSTEM'/
     &  T10,'A=',F7.4,5X,'B=',F7.4,5X,'C=',F7.4)
      GO TO 300
  260 WRITE(LNPTR,261) A,B,C,GAMA
  261 FORMAT(T10,'MONOCLINIC SYSTEM : 1 ST. SETTING'/
     &  T10,'A=',F7.4,5X,'B=',F7.4,5X,'C=',F7.4,5X,'GAMMA=',F6.2)
      GO TO 300
  270 WRITE(LNPTR,271) A,B,C,BETA
  271 FORMAT(T10,'MONOCLINIC SYSTEM : 2 ND. SETTING'/
     &  T10,'A=',F7.4,5X,'B=',F7.4,5X,'C=',F7.4,5X,'BETA=' ,F6.2)
      GO TO 300
  280 WRITE(LNPTR,281) A,B,C,ALFA,BETA,GAMA
  281 FORMAT(T10,'TRICLINIC SYSTEM'/T10,'A=',F7.4,5X,'B=',F7.4,5X,
     & 'C=',F7.4,5X,'ALPHA=',F6.2,5X,'BETA=',F6.2,5X,'GAMMA=',F6.2)
  300 IF(NDP.EQ.1) WRITE(LNPTR,301)
  301 FORMAT(/T8,'SPECIFIC',T22,'GENERAL',
     &        /T8,'U   V   W',T22,'U   V   W',
     &     T40,'ANGLES BETWEEN DIRECTIONS')
      IF(NDP.NE.1) WRITE(LNPTR,302)
  302 FORMAT(/T8,'SPECIFIC',T22,'GENERAL',
     &        /T8,'H   K   L',T22,'H   K   L',
     &     T40,'ANGLES BETWEEN PLANES')
      GO TO 304
  303 IF(JX.GE.50) GO TO 200
  304 WRITE(LNPTR,310) HA,KA,LA,HT,KT,LT,(ANGLES(I),I=1,NANG)
  310 FORMAT(/T5,3I4,T19,3I4,(T34,12F8.2))
      GO TO 1000
  400 IF(NDP.EQ.1) WRITE(LNPTR,401)
```

(continued)

```
  401 FORMAT(/T5,'SPECIFIC',T21,'GENERAL'
     &        /T4,'U  V  T  W',T20,'U  V  T  W',
     &      T40,'ANGLES BETWEEN DIRECTIONS')
      IF(NDP.NE.1) WRITE(LNPTR,402)
  402 FORMAT(/T5,'SPECIFIC',T21,'GENERAL'
     &        /T4,'H  K  I  L',T20,'H  K  I  L',
     &      T40,'ANGLES BETWEEN PLANES')
      GO TO 404
  403 IF(JX.GE.50) GO TO 200
  404 WRITE(LNPTR,410) HA,KA,IA,LA,HT,KT,IT,LT,(ANGLES(I),I=1,NANG)
  410 FORMAT(/T2,4I3,T18,4I3,(T34,12F8.2))
 1000 JX=(NANG-1)/12+2+JX
   40 CALL INDEX(HY,KY,LY,NTSTB)
      IF(NTSTB.LT.4) GO TO 14
      WRITE(LNPTR,41)
   41 FORMAT(' ')
      JX=1+JX
   50 CALL INDEX(HX,KX,LX,NTSTA)
      IF(NTSTA.LT.4) GO TO 3
      IF(NDP.LT.3) GO TO 1
      NDP=1
      NSABS=1
      JABS=10
      NMULT=1
      HX=0
      KX=0
      LX=0
      NTSTA=0
      CALL INDEX(HX,KX,LX,NTSTA)
      GO TO 100
      END
```

***** PROGRAM STRFSB *****

```
C*  PROGRAM STRFSB
      IMPLICIT  REAL*8(A-G,O-Z),  INTEGER*4(H-N)
     ·LOGICAL*1 KABS
      CALL ERRSET(208,0,-1,0)
      INTEGER*4  RH,RK,RL
      REAL*8  KV
      COMMON/XTLCOM/ A,B,C,ALFA,BETA,GAMA,NSYS,NAME(5),VOLUME
      COMMON/INDCOM/ IH,IK,IL,ITST,KOUNT,INDMAX,DMIN,$N,
     &               NSABS,NMULT,NORDER,JABS,KABS
      COMMON/UNITS/NCARD,LNPTR
      COMMON/INSRT/ERROR,IDUP
      DIMENSION FHKLX(50)
      ERROR=5.0D-03
      IDUP=1
      DM2=1.0D-02
    1 CALL XTLSYS
      IF(A.LT.ERROR) GO TO 1000
      NSABS=0
      NMULT=2
      NORDER=1
      CALL INDSET
      J=0
      CALL ATMPDT ( VOLUME,WAVE,RELMAS,KV,INDIC )
   10 HX=IH
```

(continued)

```
      KX=IK
      LX=IL
      JTST=ITST
   11 H=HX
      K=KX
      L=LX
   12 I=-H-K
      NUMT=0
      CALL PRMSET(H,K,I,L,N,NZ,NSYS)
   13 GO TO (15,14),NSABS
   14 CALL SYSABS(H,K,L,JABS,MTST)
      IF(MTST.NE.0) GO TO 80
   15 CALL IDCALC(H,K,L,AID)
      IF(AID.LT.DMIN) GO TO 80
      CALL FCALDT(H,K,L,AID,FHKL,TO)
      IF(TO.GT.DM2) GO TO 16
      TO=(NUMT+2)*(1.0E+10)
   16 CALL INSRT1(NUMT,50,MTST,FHKL,FHKLX)
      IF(MTST.NE.0) GO TO 80
      IF(NSYS.NE.4) GO TO 320
      IF(JABS.NE.6) GO TO 40
C* RHOMBOHEDRAL ON OBVERSE HEXAGONAL INDICES CHECK
   30 RH=(2*H+K+L)/3
      RK=(-H+K+L)/3
      RL=(-H-2*K+L)/3
      GO TO 470
C* RHOMBOHEDRAL ON INVERSE HEXAGONAL INDICES CHECK
   40 IF(JABS.NE.7) GO TO 440
      RH=(H-K+L)/3
      RK=(H+2*K+L)/3
      RL=(-2*H-K+L)/3
      GO TO 470
C*************************************************************************
  200 WRITE(LNPTR,201) NAME,KABS
  201 FORMAT('1   PROGRAM STRFDT',T50,5A4,T90,'REFLECTION RESTRICTION= '
     &,1A1)
      GO TO ( 203,205,207 ),INDIC
  203 WRITE(LNPTR,204) KV,WAVE,RELMAS
  204 FORMAT(/' SPECIFIED BEAM KV. = ',F6.1, 8X,'WAVELENGTH = ',F7.4,
     &    8X,'RELATIVISTIC MASS = ',F8.5)
      GO TO 209
  205 WRITE(LNPTR,206) WAVE
  206 FORMAT(/' SPECIFIED X-RAY WAVELENGTH = ',F8.5,' ANGSTROMS')
      GO TO 209
  207 WRITE(LNPTR,208) WAVE,KV,RELMAS
  208 FORMAT(/' SPECIFIED ELECTRON WAVELENGTH = ',F6.4,'ANGSTROMS',
     &   8X,'CALCULATED KV. = ',F6.1, 8X,'RELATIVISTIC MASS = ',F8.5)
  209 GO TO (210,220,230,240,250,260,270,280),NSYS
C*************************************************************************
  210 WRITE(LNPTR,211) A
  211 FORMAT(T10,'CUBIC SYSTEM'/T10,'A=',F7.4/)
      GO TO 300
C*************************************************************************
  220 WRITE(LNPTR,221) A,ALFA
  221 FORMAT(T10,'RHOMBOHEDRAL SYSTEM'/
     & T10,'A=',F7.4,5X,'ALPHA=',F6.2)
      GO TO 300
C*************************************************************************
  230 WRITE(LNPTR,231) A,C
  231 FORMAT(T10,'TETRAGONAL SYSTEM'/
```

(continued)

```
     & T10,'A=',F7.4,5X,'C=',F7.4)
      GO TO 300
C*****************************************************************************
  240 WRITE(LNPTR,241) A,C
  241 FORMAT(T10,'HEXAGONAL SYSTEM'/
     & T10,'A=',F7.4,5X,'C=',F7.4)
      GO TO 400
C*****************************************************************************
  250 WRITE(LNPTR,251) A,B,C
  251 FORMAT(T10,'ORTHORHOMBIC SYSTEM'/
     &  T10,'A=',F7.4,5X,'B=',F7.4,5X,'C=',F7.4)
      GO TO 300
C*****************************************************************************
  260 WRITE(LNPTR,261) A,B,C,GAMA
  261 FORMAT(T10,'MONOCLINIC SYSTEM : 1 ST. SETTING'/
     &  T10,'A=',F7.4,5X,'B=',F7.4,5X,'C=',F7.4,5X,'GAMMA=',F6.2)
      GO TO 300
C*****************************************************************************
  270 WRITE(LNPTR,271) A,B,C,BETA
  271 FORMAT(T10,'MONOCLINIC SYSTEM : 2 ND. SETTING'/
     &  T10,'A=',F7.4,5X,'B=',F7.4,5X,'C=',F7.4,5X,'BETA=' ,F6.2)
      GO TO 300
C*****************************************************************************
  280 WRITE(LNPTR,281) A,B,C,ALFA,BETA,GAMA
  281 FORMAT(T10,'TRICLINIC SYSTEM'/T10,'A=',F7.4,5X,'B=',F7.4,5X,
     & 'C=',F7.4,5X,'ALPHA=',F6.2,5X,'BETA=',F6.2,5X,'GAMMA=',F6.2)
C*****************************************************************************
  300 WRITE(3,301)
  301 FORMAT(/T16,'H   K   L',T34,'D',T47,'STR.FAC',T60,'EXT.DIST.')
      GO TO 325
C     *     *     *     *     *     *     *     *     *     *     *     *     *
  320 IF(J.EQ.(J/50)*50)  GO TO 200
  325 J=J+1
  330 WRITE(LNPTR,335) H,K,L,AID,FHKL,TO
  335 FORMAT(T13,3I4,T30,F7.4,T45,F8.3,T60,F8.1)
      GO TO 80
C*****************************************************************************
  400 IF(JABS.EQ.6) GO TO 405
      IF(JABS.EQ.7) GO TO 410
      WRITE(LNPTR,401)
  401 FORMAT(/T12,'H   K   I   L',T37,'D',T47,'STR.FAC.',
     &        T63,'EXT.DIST.')
      GO TO 445
  405 WRITE(LNPTR,406)
  406 FORMAT(T10,'OBVERSE RHOMBOHEDRAL ORIENTATION RESTRICTIONS')
      GO TO 415
  410 WRITE(LNPTR,411)
  411 FORMAT(T10,'INVERSE RHOMBOHEDRAL ORIENTATION RESTRICTIONS')
  415 WRITE(LNPTR,416)
  416 FORMAT(/T6,'H  K  I  L',T21,'RH RK RL',T39,'D',
     &        T51,'STR.FAC.',T65,'EXT.DIST')
      GO TO 475
C     *     *     *     *     *     *     *     *     *     *     *     *     *
  440 IF(J.EQ.(J/50)*50) GO TO 200
  445 J=J+1
  450 WRITE(LNPTR,455) H,K,I,L,AID,FHKL,TO
  455 FORMAT( T9,4I4,T33,F7.4,T46,F8.3,T62,F8.1)
      GO TO 80
C     *     *     *     *     *     *     *     *     *     *     *     *     *
  470 IF(J.EQ.(J/50)*50) GO TO 200
```

(continued)

```
  475 J=J+1
  480 WRITE(LNPTR,485) H,K,I,L,RH,RK,RL,AID,FHKL,TO
  485 FORMAT(T4,4I3,T20,3I3,T36,F7.4,T50,F8.3,T65,F8.1)
C*****************************************************************************
   80 CALL PERM(H,K,I,L,N,NZ)
      IF(NZ.LE.0) GO TO 81
      IF(H.LT.0) GO TO 80
      GO TO 13
   81 CALL INDEX(HX,KX,LX,JTST)
      IF(JTST.LT.4) GO TO 11
      WRITE(LNPTR,82) J
   82 FORMAT(//T10,'THERE ARE ',I4,' ENTRIES IN THE TABLE')
      CALL FOOODT (FHKL,TO)
      WRITE(LNPTR,83) FHKL,TO
   83 FORMAT(//T4,'THE (0,0,0) REFLECTION HAS A STRUCTURE FACTOR OF ',F8
     &.3,' AND AN EXTINCTION DISTANCE OF ',F8.1,' ANGSTROMS'/
     & T4,'FOR A DBYE TEMPERATURE CORRECTION FACTOR OF 0.0')
      GO TO 1
 1000 STOP
      END
```

***** PROGRAM DT4BMS *****

```
C*  PROGRAM DT4BMS   (DOYLE & TURNER )
      IMPLICIT  REAL*8(A-G,O-Z)  ,INTEGER*4(H-N)
      COMMON/XTLCOM/A,B,C,ALFA,BETA,GAMA,NSYS,NAME(5),VOL
      COMMON/UNITS/NCARD,LNPTR
      EQUIVALENCE (TABU(1),U1),  (TABU(2),U2),  (TABU(3),U3)
      REAL*8  KV,  TABU(3),  ONE/1.0D+00/,  TWO/2.0D+00/
   10 CALL XTLSYS
      JLINE=0
   20 CALL ATMPDT (VOL,WAVE,RELMAS,KV,IND)
   30 READ(NCARD,31) HA,KA,IA,LA
   31 FORMAT(4I3)
      IF(NSYS.NE.4) LA=IA + LA
      IF(HA*HA+KA*KA+LA*LA.EQ.0) GO TO 10
      KLINE=0
      IF(JLINE.EQ.3*(JLINE/3)) KLINE=1
      WRITE(LNPTR,32) KLINE,NAME,WAVE,RELMAS
      JLINE=1+JLINE
   32 FORMAT(I1,////' PROGRAM DT4BMS',T30,5A4,T60,'WAVELENGTH= ',F6.4,
     &  T90,'RELMAS= ',F5.3/)
      IF(NSYS.EQ.4) WRITE(LNPTR,33) HA,KA,IA,LA
   33 FORMAT(' INPUT DATA: HA,KA,IA,LA= ',4I4/)
      IF(NSYS.NE.4) WRITE(LNPTR,34) HA,KA,LA
   34 FORMAT(' INPUT DATA: HA,KA,LA= ',3I3/)
      IF(NSYS.EQ.4) WRITE(LNPTR,35)
   35 FORMAT(T12,'H  K  I  L',T30,'D SPACE',T47,'STR.FAC.',T62,'SI(G)',
     &  T80,'U'//)
      IF(NSYS.NE.4) WRITE(LNPTR,36)
   36 FORMAT(T12,'H  K  L',T30,'D SPACE',T47,'STR.FAC.',T62,'SI(G)',
     &  T80,'U'//)
      DO 50 J=1,3
      HB=HA*J
      KB=KA*J
      LB=LA*J
      CALL IDCALC(HB,KB,LB,D)
      IF(J.EQ.1) G=ONE/D
```

(continued)

```
      CALL FCALDT  (HB,KB,LB,D,FHKL,ETA,DBYE)
      TABU(J)=ONE/(WAVE*ETA)
      IF(FHKL.LT.Ø.ØD+ØØ) TABU(J)=Ø.ØD+ØØ
      IF(NSYS.NE.4) WRITE(LNPTR,4Ø) HB,KB,LB,D,FHKL,ETA,TABU(J)
   4Ø FORMAT(T1Ø,3I3,T3Ø,F7.4,T45,F8.3,T6Ø,F9.1,T75,F7.5)
      IB=-HB-KB
      IF(NSYS.EQ.4) WRITE(LNPTR,41) HB,KB,IB,LB,D,FHKL,ETA,TABU(J)
   41 FORMAT(T1Ø,4I3,T3Ø,F7.4,T45,F8.3,T6Ø,F9.1,T75,F7.5)
   5Ø CONTINUE
      UNEW =(U1+U3+DSQRT((G*G+(U1-U3)/TWO)**TWO+(U1+U2)*(U1+U2))
     &             -DSQRT((G*G-(U1-U3)/TWO)**TWO+(U1-U2)*(U1-U2)))/TWO
      SINEW=ONE/(WAVE*UNEW)
      IF(NSYS.NE.4) WRITE(LNPTR,7Ø) HA,KA,LA,UNEW,SINEW
   7Ø FORMAT(//' NEW VALUE FOR H,K,L=',3I3,5X,'U=',F7.5,5X,'SI(G)=',
     &           F8.1)
      IF(NSYS.EQ.4) WRITE(LNPTR,71) HA,KA,IA,LA,UNEW,SINEW
   71 FORMAT(//' NEW VALUE FOR H,K,I,L=',4I3,5X,'U=',F7.5,5X,'SI(G)=',
     &           F8.1)
      GO TO 3Ø
      END
```

***** PROGRAM STEREO *****

```
C* PROGRAM STEREO
      IMPLICIT REAL*8(A-G,O-Z), INTEGER*4(H-N)
C*************************************************************************
      LOGICAL*1 PLUS,P,KABS,BL,MARK
      REAL*4  A1,A2
      DATA PLUS/'+'/,P/'P'/,BL/' '/
      DIMENSION PROJG(2),PROJS(2),PROJO(2),NPROJ(3),
     &          FMT1(4),FMT2(3),NTY(2),FPOLE(3),NPOLE(4)
      DATA PROJG/'GNOMONIC '/,  PROJS/'STEREOGRAPHIC '/,
     &     PROJO/'ORTHOGRAPHIC '/,  NPROJ/9,14,13/,  NTY/28,24/,
     &     FMT1/'PROJECTION OF DIRECTIONS OF '/,
     &     FMT2/'PROJECTION OF PLANES OF '/,
     &     FPOLE/'(''POLE = '',1A4)'/,   TEN/1.ØD+Ø1/,
     &     XMAX/1.55D+Ø1/,XMIN/-1.55D+Ø1/,YMAX/1.3D+Ø1/,YMIN/-1.3D+Ø1/,
     &     FIFTEN/1.5D+Ø1/,FOURT5/4.5D+Ø1/,CM/2.54D+ØØ/,NHT/2/
      COMMON/XTLCOM/ A,B,C,ALFA,BETA,GAMA,NSYS,NAME(5),VOLUME,
     &       ZERO,ONE,TWO,PYO2,PY,RAD,DEG
      COMMON/INDCOM/ IH,IK,IL,ITST,KOUNT,INDMAX,DMIN,$NP,
     &               NSABS,NMULT,NORDER,JABS,KABS
      COMMON/UNITS/NCARD,LNPTR
      EQUIVALENCE (NPOLE(1),IPH),(NPOLE(2),IPK),(NPOLE(3),IPI),
     &            (NPOLE(4),IPL)
C*************************************************************************
      STP=NHT*(Ø.13D+ØØ)
      STPX=Ø.5ØØ*STP
      STPY=Ø.866*STP
    1 CALL XTLSYS
      IF(A.LT.(Ø.1D-Ø6)) GO TO 99999
      N=4
      NSABS=Ø
      NMULT=1
      NORDER=1
      CALL INDSET
      IKOUNT=KOUNT
      CALL TCSET
```

(continued)

```
    3 READ(NCARD,4) NPOLE, NDP, NGSO, THETA, RADIUS
    4 FORMAT(4I5,    4X,I1,    4X,I1,    2F10.5)
COLUMNS:     1-20,      25,       30,    31-50
      J=IABS(IPH) + IABS(IPK) + IABS(IPL) + IABS(IPI)
      IF(J.EQ.0) GO TO 1
      IF(NSYS.EQ.4) GO TO 6
      IPL=IPL+IPI
      IPI=IPL
      N=3
    6 IF(NDP.LE.0) NDP=2
      IF(NDP.GT.2) NDP=2
      IF(NGSO.LE.0) NGSO=2
      IF(NGSO.GT.3) NGSO=2
      WRITE(LNPTR,5) NDP, NGSO, NAME, (NPOLE(J),J=1,N)
    5 FORMAT('1 PROGRAM STEREO'/
     & T40,'NDP=',I2,T50,'NGSO=',I2,T60,5A4,T10,'POLE=',4I4)
C* SET UP AXIS
      X3=IPH
      Y3=IPK
      Z3=IPL
      IF(NDP.GT.1) GO TO 7
C* GIVEN DIRECTION (X3,Y3,Z3), FIND TWO ORTHOGONAL DIRECTIONS
      CALL AT2CD(X3,Y3,Z3,  X,Y,Z)
      CALL PERP2(X,Y,Z,  X1,Y1,Z1,  X2,Y2,Z2)
      CALL AC2TD(X1,Y1,Z1,  X1,Y1,Z1)
      CALL AC2TD(X2,Y2,Z2,  X2,Y2,Z2)
      CALL ADENT(X1,Y1,Z1,AID1)
      CALL ADENT(X2,Y2,Z2,AID2)
      CALL ADENT(X3,Y3,Z3,AID3)
      GO TO 8
C*  GIVEN PLANE (X3,Y3,Z3) , FIND TWO ORTHOGONAL PLANES.
    7 CALL AT2CP(X3,Y3,Z3,  X,Y,Z)
      CALL PERP2(X,Y,Z,  X1,Y1,Z1,  X2,Y2,Z2)
      CALL AC2TP(X1,Y1,Z1,  X1,Y1,Z1)
      CALL AC2TP(X2,Y2,Z2,  X2,Y2,Z2)
      CALL ADCALC(X1,Y1,Z1,AID1)
      CALL ADCALC(X2,Y2,Z2,AID2)
      CALL ADCALC(X3,Y3,Z3,AID3)
C*  INITIALISE THE PLOTTER.
    8 CALL GPSEND(1,1)
      CALL GPLOT(CM,CM,2)
      CALL GPLOT(-17.0, -13.8, 1)
      CALL GPLOT(ZERO,ZERO,3)
      CALL GPTEXT(PLUS,0,NHT,ZERO)
      CALL GPLOT(-0.42, -0.12, 3)
      CALL GPTEXT(P,1, 2 ,ZERO)
C* CALCULATION OF ZPLN & THMAX
      GO TO (10,20,30),NGSO
C* GNOMONIC PROJECTION
   10 IF(THETA.EQ.ZERO) GO TO 15
   11 IF(THETA.GT.FOURT5) THETA=FOURT5
   12 ZPLN=XMAX/DTAN(THETA*RAD)
      GO TO 40
   15 IF(RADIUS.GT.ZERO) GO TO 17
      THETA=FOURT5
      GO TO 12
   17 ZPLN=RADIUS
      GO TO 40
C* STEREOGRAPHIC PROJECTION
   20 IF(THETA.EQ.ZERO) GO TO 25
```

(continued)

```
   21 ZPLN=XMAX/(TWO*DATAN((THETA*RAD)/TWO))
      GO TO 40
   25 IF(RADIUS.EQ.ZERO) RADIUS= TEN
      ZPLN=RADIUS
      GO TO 40
C* ORTHOGRAPHIC PROJECTION
   30 IF(THETA.EQ.ZERO) GO TO 35
   31 ZPLN=XMAX/DSIN(THETA*RAD)
      GO TO 40
   35 IF(RADIUS.EQ.ZERO) RADIUS= TEN
   37 ZPLN=RADIUS
   40 CONTINUE
      WRITE(LNPTR,45) THETA,RADIUS,ZPLN
   45 FORMAT(/T40,'THETA= ',F10.5,T60,'RADIUS= ',F10.5,T80,
     &    'ZPLN= ',F10.5//)
      IF(NSYS.NE.4) WRITE(LNPTR,46)
   46 FORMAT(T7,'H    K    L',T26,'COS(1)',T36,'COS(2)',T46,'COS(3)',
     &    T62,'ANG(1)',T72,'ANG(2)',T82,'ANG(3)',T97,'X',T105,'Y',
     &    T112,'CONST',T120,'LABEL'/)
      IF(NSYS.EQ.4) WRITE(LNPTR,47)
   47 FORMAT(T6,'H   K   I   L',T26,'COS(1)',T36,'COS(2)',T46,'COS(3)',
     &    T62,'ANG(1)',T72,'ANG(2)',T82,'ANG(3)',T97,'X',T105,'Y',
     &    T112,'CONST',T120,'LABEL'/)
  100 H=IH
      K=IK
      L=IL
      KOUNT=IKOUNT
      NTST=ITST
  105 I=-(H+K)
  106 CALL PRMSET(H,K,I,L,NUM,NZ,NSYS)
  107 GO TO (109,108),NSABS
  108 CALL SYSABS(H,K,L,JABS,MTST)
      IF(MTST.NE.0) GO TO 190
  109 RH=DFLOAT(H)
      RK=DFLOAT(K)
      RL=DFLOAT(L)
      IF(NDP.EQ.2) GO TO 110
      CALL ADENT(RH,RK,RL,AID)
      CALL ANGDIR(RH,RK,RL,AID,X1,Y1,Z1,AID1,ANG1,COSAN1)
      CALL ANGDIR(RH,RK,RL,AID,X2,Y2,Z2,AID2,ANG2,COSAN2)
      CALL ANGDIR(RH,RK,RL,AID,X3,Y3,Z3,AID3,ANG3,COSAN3)
      GO TO 120
  110 CALL ADCALC (RH,RK,RL,D)
      CALL ANGPLN(X3,Y3,Z3,AID3,RH,RK,RL,D,ANG3,COSAN3)
      CALL ANGPLN(X2,Y2,Z2,AID2,RH,RK,RL,D,ANG2,COSAN2)
      CALL ANGPLN(X1,Y1,Z1,AID1,RH,RK,RL,D,ANG1,COSAN1)
  120 CONTINUE
      ZC=COSAN3
      GO TO (130,140,150),NGSO
 CONSTANT FOR GNOMONIC
  130 IF(ZC.LE.ZERO) GO TO 190
      CONST=ZPLN/ZC
      GO TO 160
 CONSTANT FOR STEREOGRAPHIC
  140 IF(ZC.LT.-9.0D-01) GO TO 190
      CONST= ZPLN/(ONE+ZC)
      GO TO 160
 CONSTANT FOR ORTHOGRAPHIC
  150 IF(ZC.LT.ZERO) GO TO 190
      CONST=ZPLN
```

(continued)

```
  160 X=CONST*COSAN1
      IF(X.GT.XMAX) GO TO 190
      IF(X.LT.XMIN) GO TO 190
      Y=CONST*COSAN2
      IF(Y.GT.YMAX) GO TO 190
      IF(Y.LT.YMIN) GO TO 190
      MARK=PLUS
      IF(KOUNT.GE.5) MARK=BL
      IF(NSYS.NE.4) WRITE(LNPTR,161)
     &  H,K,L,COSAN1,COSAN2,COSAN3,ANG1,ANG2,ANG3,X,Y,CONST,MARK
  161 FORMAT(T3,3I5,5X,3F10.5,5X,3F10.3,5X,3F8.3,6X,1A1)
      IF(NSYS.EQ.4) WRITE(LNPTR,162)
     &  H,K,I,L,COSAN1,COSAN2,COSAN3,ANG1,ANG2,ANG3,X,Y,CONST,MARK
  162 FORMAT(T3,4I4,4X,3F10.5,5X,3F10.3,5X,3F8.3,6X,1A1)
      CALL GPLOT(X,Y,3)
      CALL GPTEXT(PLUS,0,NHT,ZERO)
      IF(KOUNT.GE.5) GO TO 190
      CALL LABAL(H,K,I,L,A1,A2,N)
      CALL GPLOT(X+STP,Y,3)
      CALL GPTEXT(A1,N,NHT,30.)
      CALL GPLOT(X+STPX,Y+STPY,3)
      CALL GPTEXT(A2,N,NHT,30.)
  190 CALL PERM(H,K,I,L,NUM,NZ)
      IF(NZ.GE.0) GO TO 107
  195 CALL INDEX(H,K,L,NTST)
  197 IF(NTST.LT.4) GO TO 105
C* DRAW EQUITORIAL CIRCLE FOR STEREOGRAPHIC & ORTHOSCOPIC PLOTS.
      IF(NGSO.EQ.1) GO TO 300
      NP=3
      DO 200 J=1,361,1
      ANG1=RAD*DFLOAT(J)
      X=DSIN(ANG1)*RADIUS
      IF(X.GT.XMAX) GO TO 199
      IF(X.LT.XMIN) GO TO 199
      Y=DCOS(ANG1)*RADIUS
      IF(Y.GT.YMAX) GO TO 199
      IF(Y.LT.YMIN) GO TO 199
      CALL GPLOT(X,Y,NP)
      NP=4
      GO TO 200
  199 NP=3
  200 CONTINUE
C* LABEL THE PLOT
  300 X=17.0D+00
      CALL GPLOT(X,YMIN,3)
      IF(NGSO.EQ.1) CALL GPTEXT(PROJG,9,3,90.0)
      IF(NGSO.EQ.2) CALL GPTEXT(PROJS,14,3,90.0)
      IF(NGSO.EQ.3) CALL GPTEXT(PROJO,12,3,90.0)
      Y=YMIN+0.38*NPROJ(NGSO)
      CALL GPLOT(X,Y,3)
      IF(NDP.EQ.1) CALL GPTEXT(FMT1,28,3,90.0)
      IF(NDP.EQ.2) CALL GPTEXT(FMT2,24,3,90.0)
      Y=Y+0.38*NTY(NDP)
      CALL GPLOT(X,Y,3)
      CALL GPTEXT(NAME,20,3,90.0)
      CALL GPLOT(18.2,YMIN,3)
      CALL LABAL(IPH,IPK,IPI,IPL,A1,A2,N)
      CALL GPNUMB(A1,FPOLE,3,90.0)
      CALL GPLOT(18.2-NHT*(0.35D+00),YMIN,3)
      CALL GPNUMB(A2,'(7X,1A4)',3,90.0)
```

(continued)

```
       CALL GPLOT(18.2,YMIN+8.5,3)
       IF(THETA.EQ.ZERO)
      & CALL GPNUMB(RADIUS,'(''SCALE: RADIUS= '',F6.2,'' CMS.'')',3,90.)
       IF(THETA.NE.ZERO)
      & CALL GPNUMB(THETA,'(''SCALE: THETA= '',F6.2,'' DEGREES'')',3,90.)
       CALL GPLOT(19.4, YMIN, 3)
       CALL GPNUMB(A,'(''A= '',F7.4)',3,90.)
       CALL GPLOT(19.4, YMIN+8.5, 3)
       CALL GPNUMB(B,'(''B= '',F7.4)',3,90.)
       CALL GPLOT(19.4,YMIN+17.0,3)
       CALL GPNUMB(C,'(''C= '',F7.4)',3,90.)
       CALL GPLOT(20.6,YMIN,3)
       CALL GPNUMB(ALFA,'(''ALFA= '',F6.2)',3,90.)
       CALL GPLOT(20.6,YMIN+8.5,3)
       CALL GPNUMB(BETA,'(''BETA= '',F6.2)',3,90.)
       CALL GPLOT(20.6,YMIN+17.,3)
       CALL GPNUMB(GAMA,'(''GAMA= '',F6.2)',3,90.)
       CALL GPSEND(2,1)
       GO TO 3
99999  STOP
       END
```

***** PROGRAM KIKU *****

```
C**  PROGRAM   KIKU:
C**       CALCULATE AND PLOT KOSSEL OR KIKUCHI PATTERNS
C**       AROUND A SELECTED POLE IN ANY CRYSTAL SYSTEM.
C**  INPUT PARAMETERS
C**   NPOLE: POLE AROUND WHICH THE KIKUCHI PATTERN IS TO BE PLOTTED
C**   CX,CY: POSITION AT WHICH THE POLE IS REQUIRED WITHIN THE BOUNDARIES
C**   NLABEL: =0,NO LINE LABELING; =1,PLOTTED LINES ARE LABELED WITH INDICES
C**   SCALE: NUMBER OF DEGREES FOR FIRST INCH OUT FROM POLE
C**   WAVE : WAVELENGTH OF INCIDENT BEAM, OR KV. OF ELECTRON BEAM
C**
       IMPLICIT REAL*8(A-G,O-Z), INTEGER*4(H-N)
       LOGICAL*1 KABS,TITLE(40),P/'P'/,PLUS/'+'/
       REAL*8 KV,TEN/1.0D+01/,A555/0.555D+00/
       DIMENSION XXX(10),YYY(10),NPOLE(4)
       COMMON/XTLCOM/ A,B,C,ALFA,BETA,GAMA,NSYS,NAME(5),VOLUME    ,
      &        ZERO,ONE,TWO,PYO2,PY,RAD,DEG
       COMMON/INDCOM/IH,IK,IL,ITST,KOUNT,INDMAX,DMIN,NPARM,
      &              NSABS,NMULT,NORDER,NABS,KABS
       COMMON/UNITS/NCARD,LNPTR
     1 CALL XTLSYS
       IF(A.LT.1.0D-06) STOP
       NSABS=0
       NMULT=2
       NORDER=1
       CALL INDSET
       NLINES=23
       CALL TCSET
COLUMN STARTING:     1       11     21            41
     2 FORMAT        (4I2,2X,2F5.0,I5,2F5.0,5X,40A1)
     3 READ(NCARD,2) NPOLE,CX,CY,NLABEL,SCALE,WAVE,TITLE
       H=0
       DO 4 J=1,4
     4 H=H+NPOLE(J)*NPOLE(J)
       IF(H.EQ.0) GO TO 1
```

(continued)

```
      IF(NSYS.EQ.4) GO TO 5
      NPOLE(3)=NPOLE(3)+NPOLE(4)
      NPOLE(4)=NPOLE(3)
    5 CONTINUE
      CALL WAVESP(WAVE,KV,RELMAS,IND)
C     IF(IND.NE.2.AND.SCALE.LT.3.0) SCALE=3.0D+00
      IF(IND.NE.2.AND.SCALE.LT.0.5) SCALE=0.5D+00
      IF(IND.EQ.2.AND.SCALE.LT.3.0) SCALE=20.D+00
      NHT=2
      XSTPX=+0.1D-01
      IF(DABS(CX).LT.0.01)  CX = +6.00D+00
      IF(DABS(CY).LT.0.01)  CY=4.0D+00
C**   INITIALISE THE PLOTTER
      CALL GPSEND(1,2)
      CALL GPLOT(ZERO,ZERO,1)
      CALL GPLOT(ONE,ONE,2)
      ZX= CX +1.5D+00
      ZY= CY +1.5D+00
      CALL GPLOT(ZX,ZY,3)
      CALL GPLOT(ZERO,ZERO,1)
C**   SET LIMITS FOR X & Y DISPLACEMENTS; CALCULATE Z & THETAM
      XMIN = -CX
      YMIN = -CY
      XMAX=-CX+12.0D+00
      YMAX=-CY +8.0D+00
      XXC=(XMIN+XMAX)/TWO
      YYC=(YMIN+YMAX)/TWO
      Z=+ONE/DTAN(SCALE*RAD)
      IF(IND.NE.2)  Z=-Z
      ZX=DMAX1(DABS(XMAX),DABS(XMIN))
      ZY=DMAX1(DABS(YMAX),DABS(YMIN))
      ZZ=DSQRT(ZX*ZX+ZY*ZY)
      THETAM=DEG*DABS(DATAN(ZZ/Z))
      WRITE(3,65) TITLE,CX,CY,NLABEL,SCALE,Z,THETAM
   65 FORMAT( /' PROGRAM KIKU',T50,40A1/T5,'CX,CY= ',2F8.3,T40,'NLABEL='
     &,I2,T60,'SCALE= ',F8.2,' DEGREES FOR THE FIRST INCH FROM THE POLE'
     &/T5,'Z ( THE PROJECTION DISTANCE) = ',F8.2,' INCHES',T60,'THETAM='
     &',F8.2,' DEGREES FROM THE POLE TO THE FURTHUREST POINT'/ )
      NLINES=NLINES + 20
C**   DRAW BOUNDARIES AROUND THE PLOT
      CALL GPLOT(XMIN,YMIN,3)
      CALL GPLOT(XMIN,YMAX,4)
      CALL GPLOT(XMAX,YMAX,4)
      CALL GPLOT(XMAX,YMIN,4)
      CALL GPLOT(XMIN,YMIN,4)
C**   SET UP AN AXES SYSTEM FOR PLOTTING
      ZA=NPOLE(1)
      ZB=NPOLE(2)
      ZC=NPOLE(4)
      CALL AT2CD(ZA,ZB,ZC,  ZX,ZY,ZZ)
      CALL PERP2(ZX,ZY,ZZ,  XA,XB,XC,  YA,YB,YC)
      CALL AC2TD(XA,XB,XC,  XA,XB,XC)
      CALL AC2TD(YA,YB,YC,  YA,YB,YC)
      CALL ADENT(XA,XB,XC,XID)
      CALL ADENT(YA,YB,YC,YID)
      CALL ADENT(ZA,ZB,ZC,ZID)
      ZI=-(ZA+ZB)
      XI=-(XA+XB)
      YI=-(YA+YB)
C**   MARK POLE POSITION, IF WITHIN THE PLOT
```

(continued)

```
      IF(XMIN.GT.ZERO) GO TO 6
      IF(XMAX.LT.ZERO) GO TO 6
      IF(YMIN.GT.ZERO) GO TO 6
      IF(YMAX.LT.ZERO) GO TO 6
      CALL GPLOT(ZERO,ZERO,3)
      CALL GPTEXT(PLUS,0,3,ZERO)
      CALL GPLOT(ZERO,0.1,3)
      CALL GPTEXT(P,0,3,ZERO)
    6 CONTINUE
C**   WRITE OUT THE INDICES OF THE POLES
      IF(NSYS.EQ.4) WRITE(LNPTR,51) XA,XB,XI,XC,YA,YB,YI,YC,ZA,ZB,ZI,ZC
   51 FORMAT(/T5,'X AXIS=',4(F7.3,3X), 5X,'Y AXIS=',4(F7.3,3X)/T5,
     1           'Z AXIS=',4(F7.3,3X))
      IF(NSYS.NE.4) WRITE(LNPTR,52) XA,XB,XC, YA,YB,YC, ZA,ZB,ZC
   52 FORMAT(/T5,'X AXIS=',3(F7.3,3X), 5X,'Y AXIS=',3(F7.3,3X),5X,
     &           'Z AXIS=',3(F7.3,3X))
      IF(NSYS.EQ.4)  WRITE(LNPTR,998)
  998 FORMAT(1H0,T45,'FIRST  LINE',T83,'SECOND LINE'/1H ,T5,'INDICES',
     1  T17,'BRAGG',T28,'D',T40,'START',T56,'FINISH',T78,'START',T94,
     2  'FINISH'/1H ,T4,'H  K  I  L',T17,'ANGLE',T25,'SPACING',T38,'X',
     3  T46,'Y',T55,'X',T63,'Y',T76,'X',T84,'Y',T93,'X',T101,'Y'//)
      IF(NSYS.NE.4) WRITE(LNPTR,997)
  997 FORMAT(1H0,T45,'FIRST  LINE',T83,'SECOND LINE'/1H ,T5,'INDICES',
     1  T17,'BRAGG',T28,'D',T40,'START',T56,'FINISH',T78,'START',T94,
     2  'FINISH'/1H ,T4,'H  K  L   ',T17,'ANGLE',T25,'SPACING',T38,'X',
     3  T46,'Y',T55,'X',T63,'Y',T76,'X',T84,'Y',T93,'X',T101,'Y'//)
      NTST=ITST
      H=IH
      K=IK
      L=IL
   61 I=-(H+K)
      HX=H
      KX=K
      IX=I
      LX=L
      CALL PRMSET(HX,KX,IX,LX,N,NZ,NSYS)
   62 GO TO (64,63),NSABS
   63 CALL SYSABS(HX,KX,LX,MTST)
      IF(MTST.NE.0) GO TO 1020
C**   CALCULATE INTERPLANER SPACING OF SELECTED PLANE
   64 CALL IDCALC(HX,KX,LX,PLN)
      IF(PLN.LT.DMIN) GO TO 1020
      SINBRG=WAVE/(TWO*PLN)
      IF(DABS(SINBRG).GE.ONE) GO TO 1020
      BRG=SINBRG*SINBRG
      BRAGG=DEG*DARSIN(SINBRG)
C**   CALCULATE THE ANGLES BETWEEN THE PLANE AND THE (X,Y,Z) AXES
C**   STORE THE SINE OF THE ANGLE IN ANG:(X=1,Y=2,Z=3)
C**   REJECT IF PLANE AT TOO GREAT AN ANGLE TO POLE (Z AXIS)
      AH=HX
      AK=KX
      AL=LX
      CALL ANGDP(ZA,ZB,ZC,ZID,  AH,AK,AL,PLN,  ANGLE,ANG3)
      XG1=BRAGG-ANGLE
      IF(DABS(XG1).GE.THETAM) GO TO 1020
      IP=4
      IF(IND.EQ.2) GO TO 66
      CALL MULTPL( HX,KX,LX,NABS,NTSTX)
      IF(NTSTX.NE.0) IP=5
   66 CALL ANGDP(YA,YB,YC,YID,  AH,AK,AL,PLN,  ANGLE,ANG2)
```

(continued)

```
      CALL ANGDP(XA,XB,XC,XID,  AH,AK,AL,PLN,  ANGLE,ANG1)
C**   SET VALUES OF EQUATION
      A1=ANG1*ANG1 - BRG
      B1=ANG2*ANG2 - BRG
      C1=ANG3*ANG3 - BRG
      E1=ANG2*ANG3
      F1=ANG3*ANG1
      G1=ANG1*ANG2
      CALL FUNZ (A1,B1,C1,E1,F1,G1,Z,NTYPE)
C* SET STARTING VALUES FOR THE PLOT.
 3000 NT=0
      NSX=+1
      NSY=+1
      NN=0
      IMAG=1
      IMAGST=+1
      S=1.0D-06 + DSQRT(ANG1*ANG1 + ANG2*ANG2)
      ANG1=ANG1/S
      ANG2=ANG2/S
      ISST=0
      IF(ANG1.GT.+A555) GO TO 3040
      IF(ANG1.LT.-A555) GO TO 3020
      IF(ANG2.LT.ZERO)  GO TO 3010
      GO TO 3030
 3010 XST=XMIN
      STP=XSTPX
      NSX=-1
      GO TO 5000
 3020 YST=YMIN
      STP=XSTPX
      NSY=-1
      GO TO 4000
 3030 XST=XMAX
      STP=-XSTPX
      GO TO 5000
 3040 YST=YMAX
      STP=-XSTPX
      GO TO 4000
C*
C*  PLOT UP & DOWN THE PAGE.
C*
 4000 Y=YST
 4010 STP2=XSTPX*TWO
      STP10=TEN*STP
      CALL FUNOFY(X,Y,NSY,JTST)
      IF(JTST.LT.0) IMAGST=2
      GO TO (4021,4051),IMAGST
C* TEST X
 4020 CALL FUNOFY(X,Y,NSY,JTST)
      IF(JTST.LT.0)   GO TO 4051
 4021 IF(X.LT.XMIN)   GO TO 4051
      IF(X.GT.XMAX)   GO TO 4051
C* START PLOTTING
 4030 GO TO (4032,4031),IMAG
 4031 CALL GPLOT(XIM,YIM,3)
      IMAG=1
      GO TO 4033
 4032 CALL GPLOT(X,Y,3)
 4033 NN=NN+1
      XXX(NN)=X
```

(continued)

```
      YYY(NN)=Y
C* CONTINUE PLOTTING
 4040 CALL GPLOT(X,Y,IP)
      XX=X
      YY=Y
      Y=Y+STP
      IF(Y.LT.YMIN) GO TO 4072
      IF(Y.GT.YMAX) GO TO 4073
      CALL FUNOFY(X,Y,NSY,JTST)
      IF(JTST.LT.0) GO TO 4045
      IF(X.LT.XMIN)   GO TO 4046
      IF(X.GT.XMAX)   GO TO 4047
      GO TO 4040
C* X IS IMAG., OR X IS REAL BUT OUTSIDE PLOTTING RANGE
 4045 IMAG=2
      XIM=XX
      YIM=YY
      NT=NT+1
      IF(NT.GT.1) GO TO 5090
      NSY=-NSY
      STP=-STP
      Y=Y+STP10
      STP10=STP
      GO TO 4050
 4046 XX=XMIN
      GO TO 4048
 4047 XX=XMAX
 4048 CALL GPLOT(XX,YY,IP)
 4050 NN=NN+1
      XXX(NN)=XX
      YYY(NN)=YY
 4051 Y=Y+STP10
      IF(Y.GT.YMAX) GO TO 4071
      IF(Y.LT.YMIN) GO TO 4071
      CALL FUNOFY(X,Y,NSY,JTST)
      IF(JTST.LT.0) GO TO 4051
      IF(X.LT.XMIN)   GO TO 4051
      IF(X.GT.XMAX)   GO TO 4051
      IF(DABS(STP10).GT.STP2) GO TO 4060
      STP10=TEN*STP
      GO TO 4030
 4060 Y=Y-STP10
      STP10=STP
      GO TO 4051
C* PLOT BACK, OR FINISHED?
 4070 NN=NN+1
      XXX(NN)=XX
      YYY(NN)=YY
      CALL GPLOT(XX,YY,IP)
 4071 NT=NT+1
      IF(NT.GT.1) GO TO 6000
      IF(IMAGST.EQ.2) GO TO 4075
      IF(NN.EQ.0.AND.IND.EQ.2) GO TO 4075
      IF(NTYPE.EQ.3) GO TO 6000
 4075 STP=-STP
      NSY=-NSY
      STP2=DABS(TWO*STP)
      STP10=TEN*STP
      Y=YMIN + YMAX - YST
      GO TO 4020
```

(continued)

```
 4072 YY=YMIN
      GO TO 4074
 4073 YY=YMAX
 4074 CALL FUNOFY(XX,YY,NSY,JTST)
      CALL GPLOT(XX,YY,IP)
      GO TO 4070
C*
C*  PLOT ACROSS THE PAGE
C*
 5000 X=XST
 5010 STP2=XSTPX*TWO
      STP10=TEN*STP
      CALL FUNOFX(X,Y,NSX,JTST)
      IF(JTST.LT.0) IMAGST=2
      GO TO (5021,5051),IMAGST
C* TEST Y
 5020 CALL FUNOFX(X,Y,NSX,JTST)
      IF(JTST.LT.0) GO TO 5051
 5021 IF(Y.LT.YMIN)   GO TO 5051
      IF(Y.GT.YMAX)   GO TO 5051
C* START PLOTTING.
 5030 GO TO (5032,5031),IMAG
 5031 CALL GPLOT(XIM,YIM,3)
      IMAG=1
      GO TO 5033
 5032 CALL GPLOT(X,Y,3)
 5033 NN=NN+1
      XXX(NN)=X
      YYY(NN)=Y
C* CONTINUE PLOTTING
 5040 CALL GPLOT(X,Y,IP)
      XX=X
      YY=Y
      X=X+STP
      IF(X.LT.XMIN) GO TO 5072
      IF(X.GT.XMAX) GO TO 5073
      CALL FUNOFX(X,Y,NSX,JTST)
      IF(JTST.LT.0) GO TO 5045
      IF(Y.LT.YMIN) GO TO 5046
      IF(Y.GT.YMAX) GO TO 5047
      GO TO 5040
C* Y IS IMAG., OR Y IS REAL BUT OUTSIDE PLOTTING RANGE
 5045 IMAG=2
      YIM=YY
      XIM=XX
      NT=NT+1
      IF(NT.GT.1) GO TO 5090
      NSX=-NSX
      STP=-STP
      X=X+STP10
      STP10=STP
      GO TO 5050
 5046 YY=YMIN
      GO TO 5048
 5047 YY=YMAX
 5048 CALL GPLOT(XX,YY,IP)
 5050 NN=NN+1
      XXX(NN)=XX
      YYY(NN)=YY
 5051 X=X+STP10
```

(continued)

```
      IF(X.GT.XMAX) GO TO 5071
      IF(X.LT.XMIN) GO TO 5071
      CALL FUNOFX(X,Y,NSX,JTST)
      IF(JTST.LT.0) GO TO 5051
      IF(Y.LT.YMIN) GO TO 5051
      IF(Y.GT.YMAX) GO TO 5051
      IF(DABS(STP10).GT.STP2) GO TO 5060
      STP10=TEN*STP
      GO TO 5030
 5060 X=X-STP10
      STP10=STP
      GO TO 5051
C* PLOT BACK, OR FINISHED?
 5070 NN=NN+1
      XXX(NN)=XX
      YYY(NN)=YY
 5071 NT=NT+1
      IF(NT.GT.1) GO TO 6000
      IF(IMAGST.EQ.2) GO TO 5075
      IF(NN.EQ.0.AND.IND.EQ.2) GO TO 5075
      IF(NTYPE.EQ.3) GO TO 6000
 5075   NSX=-NSX
      STP=-STP
      STP2=DABS(TWO*STP)
      STP10=TEN*STP
      X=XMIN + XMAX - XST
      GO TO 5020
 5072 XX=XMIN
      GO TO 5074
 5073 XX=XMAX
 5074 CALL FUNOFX(XX,YY,NSX,JTST)
      CALL GPLOT(XX,YY,IP)
      GO TO 5070
 5090 NN=NN+1
      XXX(NN)=XX
      YYY(NN)=YY
      IF(IMAGST.EQ.2)  CALL GPLOT(XXX(1),YYY(1),IP)
 6000 CONTINUE
C**  COMPLEATION OF PLOT FOR THIS LINE
C**  IF LINE IS OUTSIDE PLOT BOUNDARIES, NN=0
      IF(NN.LT.1) GO TO 1020
      IF(NSYS.EQ.4) WRITE(LNPTR,999) HX,KX,IX,LX,BRAGG,PLN
  999 FORMAT(1H0,4I3,3X,F6.3,4X,F5.3)
      IF(NSYS.NE.4) WRITE(LNPTR,888) HX,KX,LX,BRAGG,PLN
  888 FORMAT(1H0,3I3,6X,F6.3,4X,F5.3)
      WRITE(LNPTR,777)  (XXX(J),YYY(J), J=1,NN)
  777 FORMAT(1H+,(T34,F7.2,F8.2,F9.2,F8.2,F13.2,F8.2,F9.2,F8.2))
      IF(NLABEL.EQ.0) GO TO 1010
      IF(IP.EQ.5) GO TO 1010
C** LABEL THE LINE(S)
      STP=DABS(STP)*TWO
      X=XXX(NN)+STP
      Y=YYY(NN)+STP
      IF(X.LT.XXC) X=X-STP*TWO
      IF(Y.LT.YYC) Y=Y-STP*TWO
      DELX=ZERO
      DELY=ZERO
      ROT=ZERO
      IF(Y.GE.YMAX) GO TO 1022
      IF(Y.LE.YMIN) GO TO 1024
```

(continued)

```
      IF(X.GE.XMAX) GO TO 1026
      IF(X.LE.XMIN) GO TO 1028
      GO TO 1026
 1022 Y=Y+0.4D+00
      DELX=-0.16D+00
      ROT=9.0D+01
      GO TO 1030
 1024 Y=Y-0.9D+00
      DELX=-0.16D+00
      ROT=9.0D+01
      GO TO 1030
 1026 X=X+0.4D+00
      DELY=+0.16D+00
      GO TO 1030
 1028 X=X-0.9D+00
      DELY=0.16D+00
 1030 CALL GPLOT(X,Y,3)
      CALL LABAL(HX,KX,IX,LX,A1,A2,NSYS)
      CALL GPTEXT(A1,4,NHT,ROT)
      CALL GPLOT(X+DELX, Y+DELY, 3)
      CALL GPTEXT(A2,4,NHT,ROT)
 1010 IF(NLINES .LT.56) GO TO 7000
      WRITE(LNPTR,996)
      IF(NSYS.EQ.4) WRITE(LNPTR,998)
      IF(NSYS.NE.4) WRITE(LNPTR,997)
      NLINES=5
 7000 NLINES=NLINES + 2 + (NN-1)/4
C** RETURN FOR THE NEXT LINE
 1020 CALL PERM(HX,KX,IX,LX,N,NZ)
      IF(NZ.GT.0) GO TO 62
      CALL INDEX(H,K,L,NTST)
      IF(NTST.LT.4) GO TO 61
C** AT THIS POINT ALL THE KIKUCHI LINES HAVE BEEN PLOTTED
C**  IDENTIFY THE PLOT WITH THE POLE IDENTIFICATION
       CALL LABAL(NPOLE(1),NPOLE(2),NPOLE(3),NPOLE(4),A1,A2,NSYS)
      X=XMAX+1.5
      Y=YMIN
      CALL GPLOT(X,Y,3)
      CALL GPTEXT(TITLE,40,3,90.0)
      X=XMAX+1.8
      CALL GPLOT(X,Y,3)
      CALL GPTEXT(NAME,20,3,90.0)
      Y=YMIN+4.0
      CALL GPLOT(X,Y,3)
      CALL GPNUMB(WAVE,'(''WAVELENGTH= '',F7.5)',3,90.0)
      X=XMAX+1.9
      Y=YMIN+1.05
      CALL GPLOT(X,Y,3)
      CALL GPTEXT(A2,4,3,90.0)
      X=XMAX+2.15
      Y=YMIN
      CALL GPLOT(X,Y,3)
      CALL GPNUMB(A1,'(''POLE = '',1A4)',3,90.0)
      X=XMAX+2.45
      Y=YMIN
      CALL GPLOT(X,Y,3)
      CALL GPNUMB(SCALE,'(''SCALE= '',F5.2,'' DEGREES FOR THE FIRST INCH
     & FROM POLE'')',3,90.0)
      X=XMAX+2.75
      CALL GPLOT(X,Y,3)
```

(continued)

```
      CALL GPNUMB(A,'(''A= '',F7.4)',3,90.0)
      Y=YMIN+3.0
      CALL GPLOT(X,Y,3)
      CALL GPNUMB(B,'(''B= '',F7.4)',3,90.0)
      Y=YMIN+6.0
      CALL GPLOT(X,Y,3)
      CALL GPNUMB(C,'(''C= '',F7.4)',3,90.0)
      X=XMAX+3.05
      Y=YMIN
      CALL GPLOT(X,Y,3)
      CALL GPNUMB(ALFA,'(''ALFA= '',F6.2)',3,90.0)
      Y=YMIN+3.0
      CALL GPLOT(X,Y,3)
      CALL GPNUMB(BETA,'(''BETA= '',F6.2)',3,90.0)
      Y=YMIN+6.0
      CALL GPLOT(X,Y,3)
      CALL GPNUMB(GAMA,'(''GAMA= '',F6.2)',3,90.0)
      CALL GPSEND(2,1)
C**   RETURN FOR NEW DATA
      WRITE(3,996)
  996 FORMAT(1H1)
       NLINES=0
      GO TO 3
      END
```

APPENDIX C.1

EXAMPLES OF INPUT DATA

APPENDIX C.1
EXAMPLES OF INPUT DATA

A listing of input data for each main program is given in this appendix together with the execution time and core storage requirements. Computation was carried out on an IBM Model 360/65 computer. The programs were executed by catalogued procedure PROGEXEC, a procedure for executing a precompiled program specified by MEM=, and found in the partitioned data set (library) specified by LIB=. Core storage is given in IBM form, that is, in bytes. Since there are four bytes to the word, a program requiring 60 K bytes would occupy 15 K words.

In programs STEREO and KIKU, additional cards set up the buffers for the plot output which is placed in output class F. For these two programs the times quoted are computation times only and do not include plotting, which is carried out off-line and takes considerably longer.

C.1.1 Program DSPACE

The program calculates d-spacing tables for zirconium using the default values for the indices. The second card could thus be replaced by a blank card. A total of 74 entries is generated.

The second computation for copper specifies a limit of eight for all indices and a minimum of 0.5 for the d-spacing. No set of indices that is passed includes the number 8 (the d-spacing for 800 is 0.452), cut-offs occur at 543, 551, 632, 640 and 711, and 63 entries occur in the table.

The first page only of each table is given in Appendix D. Core requirement is 72 K and computation time is 3.14 seconds.

C.1.2 Program ANGLES

In the calculations for zirconium the indices are limited to three with an INDMAX limit of four; consequently cut-offs occur at 0003, $10\bar{1}3$, $11\bar{2}2$, $21\bar{3}1$, $22\bar{4}0$ and $31\bar{4}0$. Since column 2 of the index card is unpunched, tables of angles for both planes and directions are produced, four pages of values being produced in both cases.

The calculation for copper uses the default value of 4 for the indices' maximum. Column 2 has been punched with the value 2 so that a table of angles between planes is produced (no table of angles between directions is calculated). The table covers all indices from 100 to 443 and requires eleven pages of printout.

Appendix D contains the second page of each of the three tables to illustrate the format of the printout. Core requirement is 68 K and execution time is 16.77 seconds.

(continued)

It should be noted that if the default values of 4 are used for the indices in the zirconium calculations (with no INDMAX cut-off), then the execution time increases to 41.0 seconds, 56 pages of values are produced for the angles between planes and 61 pages of values are produced for the angles between directions.

C.1.3 Program STRFSB

Values of the structure factors and extinction distances are calculated for zirconium and copper using a 3-term Gaussian expansion for the scattering power of the atoms. Default options have been used for the index parameters. Although both lattices are specified as primitive, the listing contains only the sets of indices expected for the hexagonal-close-packed and the face-centred-cubic lattices.

In the case of zirconium, only the first two pages of printout are shown in Appendix D, since a total of 60 entries occur in the full table (3 pages of values). The full output is given for the copper computations. This listing also shows, for the first time, the output from subroutines XTLSYS and INDEX, which was not included in previous examples. Core requirement was 56 K and execution time was 4.65 seconds.

C.1.4 Program SB4BMS

The input data to this program are the same as those employed for program STRFSB, but do not include an input card for indices. The input data printout is substantially the same as that shown for STRFSB and is not illustrated.

The first three sets of output for both zirconium and copper are shown in Appendix D.

Core requirement was 44 K and execution time was 1.19 seconds.

C.1.5 Program DT4BMS

This program illustrates the form of the input required to calculate or correct structure factors and extinction distances for an alloy, the first example being an 18/8 stainless steel. The default value of 1.0 cannot be used for the filling factor but must be specified as atomic percentage of the element (not weight percentage as given by the composition). The scattering power now being obtained by table lookup means that more cards are required for data input than are required for analytical calculation. The second example given is titanium and is relatively straightforward. Appendix D includes the listings of the input data and calculated values of the first three sets of indices for each of the structures. Core requirement was 46 K and execution time was 1.39 seconds.

(continued)

C.1.6 Program STEREO

This is the first of the plotting programs. The data cards specify a hexagonal lattice with individual indices permitted up to 3 and a maximum sum of H, K and L (INDMAX) of 4. Three plots are produced about the [0001] and [$2\bar{1}\bar{1}0$] poles, for poles of direction, using gnomonic, stereographic and orthographic projection. The default options were used for scaling (THETA = 45° for gnomonic and RADIUS = 10.0 cm for the stereographic and orthographic projections). The plots for the two stereographic projections, together with the first page of printout for each plot are illustrated in Appendix D. Core requirement was 64 K and execution time was 16.07 seconds.

C.1.7 KIKU

The input data produces three Kossel plots for iron using iron K-α radiation. The indices have individual limits of 2 and an INDMAX limit of 4; systematic absence I (body centred) is applied. The Kikuchi plots use the default value of 200 kV electrons, the same index limits, but no systematic absences are applied. Core requirement was 72 K and execution time was 1 minute 55.9 seconds. The first two pages of list output and the plots for the two 001 orientations are given in Appendix D.

APPENDIX C.2

LISTING OF INPUT DATA

APPENDIX C.2

LISTING OF INPUT DATA

```
COLUMN   1111111111222222222233333333334444444444555555555566666666667777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

//JGNDSP   JOB ('********/########',E4),J.G.NAPIER,
//         CLASS=B,TIME=2
//EXECUTE EXEC PROGEXEC,LIB='MATLIB.JGN.CRYST',MEM=DSPACE
//GO.SYSIN DD *
 3.232                5.147                                120.      H.C.P.ZIRCONIUM
3    321       4     4    -4    =4    -4    =4    ØØ   Ø#Ø
 3.615                                                              F.C.C.  COPPER
                     8                                  Ø.5
         ( BLANK CARD - NO FURTHUR CRYSTAL SYSTEM DEFINITION - END OF DATA)
/*

COLUMN   1111111111222222222233333333334444444444555555555566666666667777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

//JGNANG   JOB ('********/########',E4),J.G.NAPIER,
//         CLASS=B,TIME=2
//EXECUTE EXEC PROGEXEC,LIB='MATLIB.JGN.CRYST',MEM=ANGLES
//GO.SYSIN DD *
3#232                5.147                                120.      H.C.P.  ZIRCONIUM.
                    3                          4
 3.615                                                              F.C.C.  COPPER.
 2
         ( BLANK CARD - NO FURTHUR CRYSTAL SYSTEM DEFINITION - END OF DATA)
/*

COLUMN   1111111111222222222233333333334444444444555555555566666666667777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

//JGNSFSB  JOB ('********/########',E4),J.G.NAPIER,
//         CLASS=B,TIME=2
//EXECUTE EXEC PROGEXEC,LIB='MATLIB.JGN.CRYST',MEM=STRFSB
//GO.SYSIN DD *
 3.232                5.147                                120.      H.C.P.  ZIRCONIUM
         ( BLANK CARD - DEFAULT OPTIONS FOR INDICES )
  100.      1
 2 ZR 40    Ø.Ø
 Ø.Ø  Ø.Ø  Ø.Ø       .6667.3333.5ØØØ
4.1Ø5Ø  28.492  3.144Ø  5.277Ø  1.229Ø  Ø.6Ø1Ø                                        S&B ZR4Ø
 3.615                                                              F.C.C.  COPPER
         ( BLANK CARD - DEFAULT OPTIONS FOR INDICES )
  100.      1
 4 CU 29    Ø.56
 Ø.Ø  Ø.Ø  Ø.Ø        Ø.5  Ø.5  Ø.Ø        Ø.5  Ø.Ø  Ø.5        Ø.Ø  Ø.5  Ø.5
3.535Ø  26.48Ø  2.396Ø  4.963Ø  Ø.9ØØØ  Ø.59ØØ  Ø.ØØØØ  Ø.ØØØØ  Ø.ØØØØ  S&B CU29
         ( BLANK CARD - NO FURTHUR CRYSTAL SYSTEM DEFINITION - END OF DATA)
/*
```

(continued)

```
COLUMN   1111111111222222222233333333334444444444555555555566666666667777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

//JGNSB4B  JOB ('********/########',E4),J.G.NAPIER,
//        CLASS=B,TIME=2
//EXECUTE EXEC PROGEXEC,LIB='MATLIB.JGN.CRYST',MEM=SB4BMS
//GO.SYSIN DD *
 3.232                5.147                              120.      H.C.P.  ZIRCONIUM
  100.      1
 2 ZR 40    0.
 0.0  0.0   0.        .6667.3333.5000
4.1050  28.492  3.1440  5.2770  1.2290  0.6010                                    S&B ZR40
  1   0 -1   0
  1   1 -2   0
  1   0 -1   1
  1   0   0  2
  1   0 -1   2
  1   1 -2   2
  2   0 -2   1
          ( BLANK CARD - NO FURTHUR INDICES - NEW CRYSTAL DEFINITION MAY FOLLOW )
 3.615                                                      F.C.C.  COPPER
  100.      1
 4 CU 29    0.56
 0.0  0.0   0.        0.5  0.5  0.0       0.5  0.0  0.5       0.0  0.5  0#5
3.5350  26.48   2.3960  4.0630  0.9000  0.5900  0.3000  0.0000  0.0000  S&B CU29
  1   1   1
  2   0   0
  2   2   0
  3   1   1
  3   3   1
  4   2   0
  4   2   2
          ( BLANK CARD - NO FURTHUR INDICES - NEW CRYSTAL DEFINITION MAY FOLLOW )
          ( BLANK CARD - NO FURTHUR CRYSTAL SYSTEM DEFINITION - END OF DATA)
/*

COLUMN   1111111111222222222233333333334444444444555555555566666666667777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

 /JGNDT4B  JOB ('********/########',E4),J.G.NAPIER,
//        CLASS=B,TIME=2
//EXECUTE EXEC PROGEXEC,LIB='MATLIB.JGN.CRYST',MEM=DT4BMS
//GO.SYSIN DD *
 3.6027                                                     18/8 STAINLESS STEEL
 100.       3
 4 CR 24
 0.0  0.0  0.0 0.192 0.5  0.5  0.0 0.192 0.5  0.0  0.5 0.192 0.0  0.5  0.5 0.192
6.969      6.427      5.291      4.231      3.434      2.849      2.403      D  CR CRD1
2.049      1.761      1.523      1.323      1.014      0.792      0.631      &  CR CRD2
0.514      0.427      0.310      0.237      0.188      0.154      0.127      T  CR CRD3
0.085      0.060      0.044      0.034      0.022      0.015                 *  CR CRD4
 4 FE 26
 0.0  0.0  0.0 0.733 0.5  0.5  0.0 0.733 0.5  0.0  0.5 0.733 0.0  0.5  0.5 0.733
7.165      6.669      5.558      4.436      3.562      2.928      2.461      D  FE CRD1
2.104      1.818      1.584      1.388      1.080      0.854      0.686      &  FE CRD2
```

(continued)

```
 0.561      0.466      0.336      0.255      0.202      0.165      0.136      T  FE CRD3
 0.091      0.065      0.048      0.037      0.024      0.017                 *  FE CRD4
  4 NI 28
  0.0  0.0  0.0 0.075 0.5  0.5  0.0 0.075 0.5  0.0  0.5 0.075 0.0  0.5  0.5 0.075
 6.569      6.169      5.249      4.283      3.500      2.914      2.474      D  NI CRD1
 2.133      1.858      1.631      1.440      1.136      0.909      0.737      &  NI CRD2
 0.605      0.504      0.364      0.275      0.217      0.176      0.146      T  NI CRD3
 0.097      0.069      0.052      0.040      0.026      0.018                 *  NI CRD4
   1  1  1
   2  0  0
   2  2  0
   3  1  1
   3  3  1
   4  2  0
   4  2  2
          ( BLANK CARD - NO FURTHUR INDICES - NEW CRYSTAL DEFINITION MAY FOLLOW )
 2.950                4.686                                  120.0    C.P.H.  TITANIUM
 100.       1
 2 TI 22    0.0
 0.0  0.0   0.0       .6667.3333.5000
 8.776      7.937      6.199      4.643      3.564      2.844      2.341      D  TI CRD1
 1.964      1.668      1.428      1.230      0.930      0.721      0.573      &  TI CRD2
 0.467      0.389      0.285      0.219      0.175      0.143      0.117      T  TI CRD3
 0.078      0.055      0.041      0.031      0.020      0.014                 *  TI CRD4
   1  0 -1  0
   1  1 -2  0
   1  0 -1  1
   0  0  0  2
   1  0 -1  2
   1  1 -2  2
   2  0 -2  1
          ( BLANK CARD - NO FURTHUR INDICES - NEW CRYSTAL DEFINITION MAY FOLLOW )
          ( BLANK CARD - NO FURTHUR CRYSTAL SYSTEM DEFINITION - END OF DATA)
/*

COLUMN   1111111111222222222233333333334444444444555555555566666666667777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890


//JGNSTER  JOB ('********/########',E4),J.G.NAPIER,
//         CLASS=B,TIME=2
//PGM      EXEC  BUFFPROG,PRG=VPLOT
//SYSUT2 DD      SYSOUT=F
//EXECUTE EXEC PROGEXEC,LIB='MATLIB.JGN.CRYST',MEM=STEREO
//GO.AEPLOT DD SYSOUT=F
//GO.SYSIN DD *
 3.230                5.147                                  120.0    C.P.H. ZIRCONIUM.
                   3                        4
    0     0     0     1     2     1
    0     0     0     1     2     2
    0     0     0     1     2     3
    2    -1    -1     0     2     1
    2    -1    -1     0     2     2
    2    -1    -1     0     2     3
          ( BLANK CARD - NO FURTHUR INDICES - NEW CRYSTAL DEFINITION MAY FOLLOW )
          ( BLANK CARD - NO FURTHUR CRYSTAL SYSTEM DEFINITION - END OF DATA)
/*
```

(continued)

```
COLUMN   1111111111222222222233333333334444444444555555555566666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890123456789

//JGNKIKU  JOB ('********/########',E4),J.G.NAPIER,
//         CLASS=B,TIME=2
//PGM      EXEC  BUFFPROG,PRG=VPLOT
//SYSUT2 DD     SYSOUT=F
//EXECUTE EXEC PROGEXEC,LIB='MATLIB.JGN.CRYST',MEM=KIKU
//GO.AEPLOT DD SYSOUT=F
//GO.SYSIN DD *
 2.866                                                          IRON
9                2                                  4
 0 0 1      0.0   0.0 00001 20. 1.936      KOSSEL PLOT WITH IRON K-ALPHA RADIATION.
 0 1 1      0.0   0.0 00001 20. 1.936      KOSSEL PLOT WITH IRON K-ALPHA RADIATION.
 1 1 1      0.0   0.0 00001 20. 1.936      KOSSEL PLOT WITH IRON K-ALPHA RADIATION.
         ( BLANK CARD - NO FURTHUR INDICES - NEW CRYSTAL DEFINITION MAY FOLLOW )
 2.866                                                          IRON
                 2                                  4
 0 0 1      0.0   0.0 00001 5.0            KIKUCHI PLOT FOR 200 KV. ELECTRON BEAM.
 0 1 1      0.0   0.0 00001 5.0            KIKUCHI PLOT FOR 200 KV. ELECTRON BEAM.
 1 1 1      0.0   0.0 00001 5.0            KIKUCHI PLOT FOR 200 KV. ELECTRON BEAM.
         ( BLANK CARD - NO FURTHUR INDICES - NEW CRYSTAL DEFINITION MAY FOLLOW )
         ( BLANK CARD - NO FURTHUR CRYSTAL SYSTEM DEFINITION - END OF DATA)
/*
```

APPENDIX D

SELECTED EXAMPLES OF PROGRAM OUTPUT

APPENDIX D

SELECTED EXAMPLES OF PROGRAM OUTPUT

PROGRAM DSPACE

H.C.P. ZIRCONIUM
REFLECTION RESTRICTION: P

HEXAGONAL SYSTEM
A= 3.2320 C= 5.1470

H	K	I	L	D	H	K	I	L	D
0	0	0	1	5.147	0	0	0	1	5.147
0	0	0	2	2.574	1	0	-1	0	2.799
0	0	0	3	1.716	0	0	0	2	2.574
0	0	0	4	1.287	1	0	-1	1	2.459
1	0	-1	0	2.799	1	0	-1	2	1.894
1	0	-1	1	2.459	0	0	0	3	1.716
1	0	-1	2	1.894	1	1	-2	0	1.616
1	0	-1	3	1.463	1	1	-2	1	1.542
1	0	-1	4	1.169	1	0	-1	3	1.463
1	1	-2	0	1.616	2	0	-2	0	1.399
1	1	-2	1	1.542	1	1	-2	2	1.369
1	1	-2	2	1.369	2	0	-2	1	1.350
1	1	-2	3	1.176	0	0	0	4	1.287
1	1	-2	4	1.007	2	0	-2	2	1.229
2	0	-2	0	1.399	1	1	-2	3	1.176
2	0	-2	1	1.350	1	0	-1	4	1.169
2	0	-2	2	1.229	2	0	-2	3	1.084
2	0	-2	3	1.084	2	1	-3	0	1.058
2	0	-2	4	0.947	2	1	-3	1	1.036
2	1	-3	0	1.058	1	1	-2	4	1.007
2	1	-3	1	1.036	2	1	-3	2	0.978
2	1	-3	2	0.978	2	0	-2	4	0.947
2	1	-3	3	0.900	3	0	-3	0	0.933
2	1	-3	4	0.817	3	0	-3	1	0.918
2	2	-4	0	0.808	2	1	-3	3	0.900
2	2	-4	1	0.798	3	0	-3	2	0.877
2	2	-4	2	0.771	3	0	-3	3	0.820
2	2	-4	3	0.731	2	1	-3	4	0.817
2	2	-4	4	0.684	2	2	-4	0	0.808
3	0	-3	0	0.933	2	2	-4	1	0.798
3	0	-3	1	0.918	3	1	-4	0	0.776
3	0	-3	2	0.877	2	2	-4	2	0.771
3	0	-3	3	0.820	3	1	-4	1	0.768
3	0	-3	4	0.755	3	0	-3	4	0.755
3	1	-4	0	0.776	3	1	-4	2	0.743
3	1	-4	1	0.768	2	2	-4	3	0.731
3	1	-4	2	0.743	3	1	-4	3	0.707
3	1	-4	3	0.707	4	0	-4	0	0.700
3	1	-4	4	0.665	4	0	-4	1	0.693
3	2	-5	0	0.642	2	2	-4	4	0.684
3	2	-5	1	0.637	4	0	-4	2	0.675
3	2	-5	2	0.623	3	1	-4	4	0.665
3	2	-5	3	0.601	4	0	-4	3	0.648
3	2	-5	4	0.575	3	2	-5	0	0.642
3	3	-6	0	0.539	3	2	-5	1	0.637
3	3	-6	1	0.536	3	2	-5	2	0.623
3	3	-6	2	0.527	4	0	-4	4	0.615
3	3	-6	3	0.514	4	1	-5	0	0.611
3	3	-6	4	0.497	4	1	-5	1	0.607
4	0	-4	0	0.700	3	2	-5	3	0.601

(continued)

PROGRAM DSPACE

CUBIC SYSTEM
A= 3.o150

F.C.C. COPPER
REFLECTION RESTRICTION: P

H	K	L	D	H	K	L	D
1	0	0	3.615	1	0	0	3.615
1	1	0	2.556	1	1	0	2.556
1	1	1	2.087	1	1	1	2.087
2	0	0	1.808	2	0	0	1.808
2	1	0	1.617	2	1	0	1.617
2	1	1	1.476	2	1	1	1.476
2	2	0	1.278	2	2	0	1.278
2	2	1	1.205	2	2	1	1.205
2	2	2	1.044	3	0	0	1.205
3	0	0	1.205	3	1	0	1.143
3	1	0	1.143	3	1	1	1.090
3	1	1	1.090	2	2	2	1.044
3	2	0	1.003	3	2	0	1.003
3	2	1	0.966	3	2	1	0.966
3	2	2	0.377	4	0	0	0.904
3	3	0	0.852	3	2	2	0.877
3	3	1	0.829	4	1	0	0.877
3	3	2	0.771	4	1	1	0.852
3	3	3	0.696	3	3	0	0.852
4	0	0	0.904	3	3	1	0.829
4	1	0	0.877	4	2	0	0.808
4	1	1	0.852	4	2	1	0.789
4	2	0	0.808	3	3	2	0.771
4	2	1	0.789	4	2	2	0.738
4	2	2	0.738	4	3	0	0.723
4	3	0	0.723	5	0	0	0.723
4	3	1	0.709	4	3	1	0.709
4	3	2	0.671	5	1	0	0.709
4	3	3	0.620	3	3	3	0.696
4	4	0	0.639	5	1	1	0.696
4	4	1	0.629	4	3	2	0.671
4	4	2	0.603	5	2	0	0.671
4	4	3	0.565	5	2	1	0.660
4	4	4	0.522	4	4	0	0.639
5	0	0	0.723	4	4	1	0.629
5	1	0	0.709	5	2	2	0.629
5	1	1	0.696	4	3	3	0.620
5	2	0	0.671	5	3	0	0.620
5	2	1	0.660	5	3	1	0.611
5	2	2	0.629	4	4	2	0.603
5	3	0	0.620	6	0	0	0.603
5	3	1	0.611	6	1	0	0.594
5	3	2	0.586	5	3	2	0.586
5	3	3	0.551	6	1	1	0.586
5	4	0	0.565	6	2	0	0.572
5	4	1	0.558	4	4	3	0.565
5	4	2	0.539	6	2	1	0.565
5	4	3	0.511	5	4	0	0.565
5	5	0	0.511	5	4	1	0.558
5	5	1	0.506	5	3	3	0.551

(continued)

PROGRAM ANGLES

H.C.P. ZIRCONIUM.
REFLECTION RESTRICTION: P

HEXAGONAL SYSTEM
A= 3.2320 C= 5.1470

SPECIFIC H	K	I	L	GENERAL H	K	I	L	ANGLES BETWEEN PLANES					
1	0	-1	1	1	0	-1	1	80.93	57.08	52.11	0.0		
1	0	-1	1	1	0	-1	2	86.88	75.94	49.54	18.87		
1	0	-1	1	1	0	-1	3	87.03	79.76	50.44	29.96		
1	0	-1	1	1	1	-2	0	90.00	40.46				
1	0	-1	1	1	1	-2	1	81.77	54.36	29.66			
1	0	-1	1	1	1	-2	2	75.28	67.03	26.06			
1	0	-1	1	2	0	-2	1	72.63	56.69	43.75	13.33		
1	0	-1	1	2	1	-3	0	80.44	48.39	33.89			
1	0	-1	1	2	1	-3	1	86.19	75.00	56.34	44.20	41.70	24.59
1	0	-1	1	3	0	-3	1	69.70	58.84	38.81	18.26		
1	0	-1	1	3	1	-4	0	75.90	52.47	31.48			
1	0	-1	2	1	0	-1	2	85.19	71.77	39.56	0.0		
1	0	-1	2	1	0	-1	3	74.10	63.21	36.44	11.09		
1	0	-1	2	1	1	-2	0	90.00	54.12				
1	0	-1	2	1	1	-2	1	77.26	70.20	38.76			
1	0	-1	2	1	1	-2	2	83.98	66.95	27.39			
1	0	-1	2	2	0	-2	1	82.33	62.61	58.69	32.19		
1	0	-1	2	2	1	-3	0	82.65	59.23	50.24			
1	0	-1	2	2	1	-3	1	88.69	74.13	69.33	61.43	49.51	39.23
1	0	-1	2	3	0	-3	1	78.36	62.34	57.68	37.13		
1	0	-1	2	3	1	-4	0	79.18	62.01	48.93			
1	0	-1	3	1	0	-1	3	63.01	53.82	30.29	0.0		
1	0	-1	3	1	1	-2	0	90.00	63.09				
1	0	-1	3	1	1	-2	1	79.84	75.20	46.59			

(continued)

PROGRAM ANGLES

H.C.P. ZIRCONIUM.
REFLECTION RESTRICTION: P

HEXAGONAL SYSTEM
A= 3.2320 C= 5.1470

SPECIFIC U	V	T	W	GENERAL U	V	T	W	ANGLES BETWEEN DIRECTIONS									
1	0	-1	1	1	0	-1	1	180.00	136.81	100.79	94.81	85.19	79.21	43.19	0.0		
1	0	-1	1	1	0	-1	2	161.13	140.39	114.76	104.06	75.94	65.24	39.61	18.87		
1	0	-1	1	1	0	-1	3	152.52	139.62	120.72	112.67	67.33	59.28	40.38	27.48		
1	0	-1	1	1	1	-2	0	129.61	90.00	50.39							
1	0	-1	1	1	1	-2	1	151.70	108.50	104.23	75.77	71.50	28.30				
1	0	-1	1	1	1	-2	2	158.40	119.52	93.19	86.81	60.48	21.60				
1	0	-1	1	2	0	-2	1	162.09	128.11	112.71	92.96	87.04	67.29	51.89	17.91		
1	0	-1	1	2	1	-3	0	134.07	123.81	98.00	82.00	56.19	45.93				
1	0	-1	1	2	1	-3	1	151.55	138.40	115.78	110.71	107.67	95.21	84.79	72.33	69.29	64.22
1	0	-1	1	3	0	-3	1	154.44	123.38	120.36	98.83	81.17	59.64	56.62	25.56		
1	0	-1	1	3	1	-4	0	135.61	120.69	101.78	78.22	59.31	44.39				
1	0	-1	2	1	0	-1	2	180.00	152.30	131.12	122.92	57.08	48.88	27.64	0.0		
1	0	-1	2	1	0	-1	3	171.39	155.14	138.11	131.53	48.47	41.89	24.86	8.61		
1	0	-1	2	1	1	-2	0	114.44	90.00	65.56							
1	0	-1	2	1	1	-2	1	141.02	114.32	92.66	87.34	65.68	38.98				
1	0	-1	2	1	1	-2	2	157.39	129.75	110.84	69.16	50.25	22.61				
1	0	-1	2	2	0	-2	1	143.23	125.73	98.62	93.85	86.15	81.38	54.27	36.77		
1	0	-1	2	2	1	-3	0	116.83	111.17	95.18	84.82	68.83	63.17				
1	0	-1	2	2	1	-3	1	135.63	129.01	111.94	101.72	97.93	93.02	86.98	82.07	78.28	68.06
1	0	-1	2	3	0	-3	1	135.58	119.07	101.50	91.66	88.34	78.50	60.93	44.42		
1	0	-1	2	3	1	-4	0	117.63	109.34	97.61	82.39	70.66	62.37				
1	0	-1	3	1	0	-1	3	180.00	160.38	145.66	140.14	39.86	34.34	19.62	0.0		
1	0	-1	3	1	1	-2	0	107.17	90.00	72.83							
1	0	-1	3	1	1	-2	1	134.55	116.15	100.37	79.63	63.85	45.45				

(continued)

PROGRAM ANGLES

F.C.C. COPPER.
REFLECTION RESTRICTION: P

CUBIC SYSTEM
A= 3.6150

SPECIFIC H	K	L	GENERAL H	K	L	ANGLES BETWEEN PLANES					
1	1	0	2	1	1	90.00	73.22	54.74	30.00		
1	1	0	2	2	1	90.00	76.37	45.00	19.47		
1	1	0	3	1	0	77.08	63.43	47.87	26.57		
1	1	0	3	1	1	90.00	64.76	31.48			
1	1	0	3	2	0	78.69	66.91	53.96	11.31		
1	1	0	3	2	1	79.11	67.79	55.46	40.89	19.11	
1	1	0	3	2	2	90.00	80.13	46.69	30.96		
1	1	0	3	3	1	90.00	71.07	49.54	13.26		
1	1	0	3	3	2	90.00	81.33	41.08	25.24		
1	1	0	4	1	0	80.13	59.04	46.69	30.96		
1	1	0	4	1	1	90.00	70.53	60.00	33.56		
1	1	0	4	2	1	81.12	72.02	62.42	39.51	22.21	
1	1	0	4	3	0	81.87	64.90	55.55	8.13		
1	1	0	4	3	1	82.03	73.90	65.42	56.31	46.10	13.90
1	1	0	4	3	2	82.45	74.77	48.96	38.02	23.20	
1	1	0	4	3	3	90.00	83.03	43.31	31.91		
1	1	0	4	4	1	90.00	68.33	52.01	10.02		
1	1	0	4	4	3	90.00	83.66	39.37	27.94		
1	1	1	1	1	1	70.53	0.0				
1	1	1	2	1	0	75.04	39.23				
1	1	1	2	1	1	90.00	61.87	19.47			
1	1	1	2	2	1	78.90	54.74	15.79			
1	1	1	3	1	0	68.58	43.09				
1	1	1	3	1	1	79.98	58.52	29.50			
1	1	1	3	2	0	80.79	36.81				

(continued)

```
SUBROUTINE XTLSYS                                F.C.C.  COPPER                                NSYS=  1
                                                 CRYSTAL DATA
   A=  3.61500        B=  3.61500        C=  3.61500        ALFA=  90.00        BETA=  90.00        GAMA=  90.00

                                                 CALCULATION DATA
   CA= 0.00000        CB= 0.00000        CG= 0.00000        SA= 1.00000        SB= 1.00000        SG= 1.00000
   S11=  170.77850    S22=  170.77850    S33=  170.77850    S12=   -0.00000    S23=   -0.00000    S31=   -0.00000
   SQRANG= 1.00000                                  VOLUME OF UNIT CELL=   47.24163


SUBROUTINE INDEX
   HMIN =  0          HMAX =  4          KMIN =  -4         KMAX =  4          LMIN =  -4         LMAX =  4
   IH   =  1          IK   =  0          IL   =  0          INOMAX = 101       NPARM =  0         DMIN =  0.0
   KABS =  F          JABS =  16         NSABS =  1         NMULT =  2         NOPDER =  1        NORD =  3 2 1

SUBROUTINE WAVESP
   ELECTRON ACCELERATING VOLTAGE OF  100.0 KILOVOLTS WAS SPECIFIED, WAVELENGTH = 0.03700 ANGSTROMS
   RELATIVISTIC MASS CORRECTION FACTOR =  1.19578          INC=  1


SUBROUTINE ATMPSB,   VERSION 1 : 12TH JULY,1973
 READS IN  DATA RELEVANT TO ATOMIC POSITIONS IN THE LATTICE, FILLING FACTORS, AND SCATTERING FACTORS.
ENTRY FCALSB  CALCULATES THE STRUCTURE FACTOR & EXTINCTION DISTANCE FOR THE GIVEN INDICES

    NO.     SYMBOL  A.N.          POSITION / FILL.

     1        CU      29   0.0   0.0   0.0   /1.000    0.500 0.500 0.0   /1.000    0.500 0.0   0.500 /1.000
                           0.0   0.500 0.500 /1.000

          SCATTERING EQUATION CONSTANTS

ATOM NO.  SYMBOL  DEBYE        A1        B1        A2        B2        A3        B3        A4        B4        C

    1       CU    0.0        3.5350   26.4800    2.3960    4.9630    0.9000    0.5900    0.0       0.0       0.0
```


(continued)

PROGRAM STRFSB F.C.C. COPPER REFLECTION RESTRICTION= P

SPECIFIED BEAM KV. = 100.0 WAVELENGTH = 0.0370 RELATIVISTIC MASS = 1.19578
CUBIC SYSTEM
A= 3.6150

H	K	L	D	STR.FAC	EXT.DIST.
1	1	1	2.0671	16.480	243.4
2	0	0	1.8075	14.183	282.8
2	2	0	1.2781	9.589	418.3
2	2	2	1.0436	7.466	537.2
3	1	1	1.0900	7.899	507.7
3	3	1	0.8293	5.362	747.9
3	3	3	0.6957	4.057	988.4
4	0	0	0.9037	6.107	656.6
4	2	0	0.8083	5.152	778.4
4	2	2	0.7379	4.457	899.6
4	4	0	0.6390	3.549	1129.7
4	4	2	0.6025	3.243	1236.2
4	4	4	0.5218	2.624	1527.4

THERE ARE 13 ENTRIES IN THE TABLE

THE (0,0,0) REFLECTION HAS A STRUCTURE FACTOR OF 32.673 AND AN EXTINCTION DISTANCE OF 122.3 ANGSTROMS
FOR A DBYE TEMPERATURE CORRECTION FACTOR OF 0.0

(continued)

SUBROUTINE XTLSYS

H.C.P. ZIRCONIUM NSYS= 4

CRYSTAL DATA

A= 3.23200 B= 3.23200 C= 5.14700 ALFA= 90.00 BETA= 90.00 GAMA= 120.00

CALCULATION DATA

CA= 0.00000 CB= 0.00000 CG= -.50000 SA= 1.00000 SB= 1.00000 SG= 0.86603
S11= 276.72669 S22= 276.72669 S33= 81.83643 S12= 138.36334 S23= -0.00000 S31= -0.00000
SCRANG= 0.86603 VOLUME OF UNIT CELL= 46.56156

SUBROUTINE INDEX

HMIN =	0	HMAX =	4	KMIN =	-4	KMAX =	4	LMIN =	-4	LMAX =	4
IH =	0	IK =	0	IL =	1	INDMAX =	100	NPARM =	0	DMIN =	0.0
KABS =	P	JABS =	10	NSABS =	1	NMULT =	2	NORDER =	1	NORD =	3 2 1

SUBROUTINE WAVESP

ELECTRON ACCELERATING VOLTAGE OF 100.0 KILOVOLTS WAS SPECIFIED, WAVELENGTH = 0.03700 ANGSTROMS
RELATIVISTIC MASS CORRECTION FACTOR = 1.19578 IND= 1

SUBROUTINE ATMPSB, VERSION 1 : 12TH JULY,1973
READS IN DATA RELEVANT TO ATOMIC POSITIONS IN THE LATTICE, FILLING FACTORS, AND SCATTERING FACTORS.
ENTRY FCALS8 CALCULATES THE STRUCTURE FACTOR & EXTINCTION DISTANCE FOR THE GIVEN INDICES

NO.	SYMBOL	A.N.	POSITION / FILL.						
1	ZR	40	0.0	0.0	0.0	/1.000	0.667	0.333	0.500 /1.000

SCATTERING EQUATION CONSTANTS

ATOM NO.	SYMBOL	DEBYE	A1	B1	A2	B2	A3	B3	A4	B4	C
1	ZR	0.0	4.1050	28.4920	3.1440	5.2770	1.2290	0.6010	0.0	0.0	0.0

(continued)

PROGRAM STRFSB H.C.P. ZIRCONIUM REFLECTION RESTRICTION= P

SPECIFIED BEAM KV. = 100.0 WAVELENGTH = 0.0370 RELATIVISTIC MASS = 1.19578

HEXAGONAL SYSTEM
A= 3.2320 C= 5.1470

H	K	I	L	D	STR.FAC.	EXT.DIST.
0	0	0	2	2.5735	12.383	319.2
0	0	0	4	1.2868	6.207	636.9
1	0	-1	0	2.7990	6.597	599.2
1	0	-1	1	2.4589	10.335	382.5
1	0	-1	2	1.8944	4.688	843.3
1	0	-1	3	1.4627	6.192	638.4
1	0	-1	4	1.1691	2.776	1424.0
1	1	-2	0	1.6160	7.954	497.0
1	1	-2	2	1.3686	6.649	594.5
1	1	-2	4	1.0066	4.588	861.5
2	0	-2	0	1.3995	3.406	1160.6
2	0	-2	1	1.3505	5.675	696.6
2	0	-2	2	1.2295	2.944	1342.6
2	0	-2	3	1.0845	4.382	902.2
2	0	-2	4	0.9472	2.108	1875.0
2	1	-3	0	1.0579	2.451	1613.0
2	1	-3	1	1.0363	4.130	957.1
2	1	-3	2	0.9785	2.207	1790.8
2	1	-3	3	0.9005	3.396	1164.0
2	1	-3	4	0.8172	1.695	2331.3
2	2	-4	0	0.8080	3.332	1186.2
2	2	-4	2	0.7709	3.099	1275.2
2	2	-4	4	0.6843	2.582	1530.7
3	0	-3	0	0.9330	4.128	957.5
3	0	-3	2	0.8771	3.772	1047.8
3	0	-3	4	0.7553	3.003	1315.9
3	1	-4	0	0.7763	1.566	2523.9
3	1	-4	1	0.7676	2.667	1482.0
3	1	-4	2	0.7432	1.464	2699.2
3	1	-4	3	0.7073	2.351	1680.8
3	1	-4	4	0.6647	1.235	3199.0
3	2	-5	0	0.6421	1.174	3365.0
3	2	-5	1	0.6372	2.011	1965.4
3	2	-5	2	0.6230	1.124	3516.4
3	2	-5	3	0.6014	1.850	2136.3
3	2	-5	4	0.5746	1.002	3945.1
3	3	-6	0	0.5387	1.831	2157.9
3	3	-6	2	0.5272	1.777	2223.0
3	3	-6	4	0.4969	1.635	2415.9
4	0	-4	0	0.6997	1.336	2957.3
4	0	-4	1	0.6934	2.280	1733.0
4	0	-4	2	0.6752	1.266	3121.1
4	0	-4	3	0.6479	2.060	1918.1
4	0	-4	4	0.6147	1.103	3583.1
4	1	-5	0	0.6108	2.184	1809.5
4	1	-5	2	0.5943	2.100	1881.5
4	1	-5	4	0.5518	1.893	2087.1
4	2	-6	0	0.5290	0.892	4427.5
4	2	-6	1	0.5262	1.535	2573.7
4	2	-6	2	0.5181	0.867	4556.8

(continued)

PROGRAM SB4BMS H.C.P. ZIRCONIUM WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,IA,LA= 0 0 0 2

H	K	I	L	D SPACE	STR.FAC.	SI(G)	U
0	0	0	2	2.5735	12.383	319.2	0.08466
0	0	0	4	1.2868	6.207	636.9	0.04243
0	0	0	6	0.8578	3.649	1083.2	0.02495

NEW VALUE FOR H,K,I,L= 0 0 0 2 U=0.10118 SI(G)= 267.1

PROGRAM SB4BMS H.C.P. ZIRCONIUM WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,IA,LA= 1 0 -1 0

H	K	I	L	D SPACE	STR.FAC.	SI(G)	U
1	0	-1	0	2.7990	6.597	599.2	0.04510
2	0	-2	0	1.3995	3.406	1160.6	0.02329
3	0	-3	0	0.9330	4.128	957.5	0.02823

NEW VALUE FOR H,K,I,L= 1 0 -1 0 U=0.05222 SI(G)= 517.5

PROGRAM SB4BMS H.C.P. ZIRCONIUM WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,IA,LA= 1 0 -1 1

H	K	I	L	D SPACE	STR.FAC.	SI(G)	U
1	0	-1	1	2.4589	10.335	382.5	0.07066
2	0	-2	2	1.2295	2.944	1342.6	0.02013
3	0	-3	3	0.8196	0.001	3694103.2	0.00001

NEW VALUE FOR H,K,I,L= 1 0 -1 1 U=0.07571 SI(G)= 357.0

(continued)

PROGRAM SB4BMS F.C.C. COPPER WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,LA= 1 1 1

H	K	L	D SPACE	STR.FAC.	SI(G)	U
1	1	1	2.0871	15.959	251.3	0.10754
2	2	2	1.0436	6.565	610.9	0.04424
3	3	3	0.6957	3.038	1319.9	0.02048

NEW VALUE FOR H,K,L= 1 1 1 U=0.12197 SI(G)= 221.6

PROGRAM SB4BMS F.C.C. COPPER WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,LA= 2 0 0

H	K	L	D SPACE	STR.FAC.	SI(G)	U
2	0	0	1.8075	13.588	295.2	0.09156
4	0	0	0.9037	5.145	779.4	0.03468
6	0	0	0.6025	2.205	1818.0	0.01487

NEW VALUE FOR H,K,L= 2 0 0 U=0.09977 SI(G)= 270.9

PROGRAM SB4BMS F.C.C. COPPER WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,LA= 2 2 0

H	K	L	D SPACE	STR.FAC.	SI(G)	U
2	2	0	1.2781	8.801	455.7	0.05931
4	4	0	0.6390	2.519	1591.7	0.01698
6	6	0	0.4260	0.889	4508.5	0.00599

NEW VALUE FOR H,K,L= 2 2 0 U=0.06081 SI(G)= 444.4

(continued)

SUBROUTINE XTLSYS 18/8 STAINLESS STEEL NSYS= 1

CRYSTAL DATA

A= 3.60270 B= 3.60270 C= 3.60270 ALFA= 90.00 BETA= 90.00 GAMA= 90.00

CALCULATION DATA

CA= 0.00000 CB= 0.00000 CG= 0.00000 SA= 1.00000 SB= 1.00000 SG= 1.00000
S11= 168.46605 S22= 168.46605 S33= 168.46605 S12= -0.00000 S23= -0.00000 S31= -0.00000
SCRANG= 1.00000 VOLUME OF UNIT CELL= 46.76105

SUBROUTINE WAVESP
ELECTRON ACCELERATING VOLTAGE OF 100.0 KILOVOLTS WAS SPECIFIED, WAVELENGTH = 0.03700 ANGSTROMS
RELATIVISTIC MASS CORRECTION FACTOR = 1.19578 IND= 1

SUBROUTINE ATMPDT, VERSION 1 : 21ST. NOV.1974

READS IN DATA RELEVANT TO ATOMIC POSITIONS IN THE LATTICE, FILLING FACTORS, AND SCATTERING FACTORS.
ENTRY FCALDT CALCULATES THE STRUCTURE FACTOR & EXTINCTION DISTANCE FOR THE GIVEN INDICES

NO.	SYMBOL	A.N.	POSITION / FILL.										
1	CR	24	0.0	0.0	0.0	/0.192	0.500	0.500	0.0	/0.192	0.500	0.0	0.500 /0.192
			0.0	0.500	0.500	/0.192							
2	FE	26	0.0	0.0	0.0	/0.733	0.500	0.500	0.0	/0.733	0.500	0.0	0.500 /0.733
			0.0	0.500	0.500	/0.733							
3	NI	28	0.0	0.0	0.0	/0.075	0.500	0.500	0.0	/0.075	0.500	0.0	0.500 /0.075
			0.0	0.500	0.500	/0.075							

SCATTERING EQUATION CONSTANTS

ATOM NO.	SYMBOL	DEBYE	(THETA/WAVE) : SCATTERING FACTOR									
1	CR	0.0										
	0.0 :	6.9690	0.05 :	6.4270	0.10 :	5.2910	0.15 :	4.2310	0.20 :	3.4340	0.25 :	2.8490
	0.30 :	2.4030	0.35 :	2.0490	0.40 :	1.7610	0.45 :	1.5230	0.50 :	1.3230	0.60 :	1.0140
	0.70 :	0.7920	0.80 :	0.6310	0.90 :	0.5140	1.00 :	0.4270	1.20 :	0.3100	1.40 :	0.2370
	1.60 :	0.1880	1.80 :	0.1540	2.00 :	0.1270	2.50 :	0.0850	3.00 :	0.0600	3.50 :	0.0440
	4.00 :	0.0340	5.00 :	0.0220	6.00 :	0.0150						
2	FE	0.0										
	0.0 :	7.1650	0.05 :	6.6690	0.10 :	5.5580	0.15 :	4.4360	0.20 :	3.5620	0.25 :	2.9680
	0.30 :	2.4610	0.35 :	2.1040	0.40 :	1.8180	0.45 :	1.5840	0.50 :	1.3880	0.60 :	1.0800
	0.70 :	0.8540	0.80 :	0.6860	0.90 :	0.5610	1.00 :	0.4660	1.20 :	0.3360	1.40 :	0.2550
	1.60 :	0.2020	1.80 :	0.1650	2.00 :	0.1360	2.50 :	0.0910	3.00 :	0.0650	3.50 :	0.0480
	4.00 :	0.0370	5.00 :	0.0240	6.00 :	0.0170						
3	NI	0.0										
	0.0 :	6.5690	0.05 :	6.1690	0.10 :	5.2490	0.15 :	4.2830	0.20 :	3.5000	0.25 :	2.9140
	0.30 :	2.4740	0.35 :	2.1330	0.40 :	1.8580	0.45 :	1.6310	0.50 :	1.4400	0.60 :	1.1360
	0.70 :	0.9090	0.80 :	0.7370	0.90 :	0.6050	1.00 :	0.5040	1.20 :	0.3640	1.40 :	0.2750
	1.60 :	0.2170	1.80 :	0.1760	2.00 :	0.1460	2.50 :	0.0970	3.00 :	0.0690	3.50 :	0.0520
	4.00 :	0.0400	5.00 :	0.0260	6.00 :	0.0180						

(continued)

PROGRAM DT4BMS 18/8 STAINLESS STEEL WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,LA= 1 1 1

H	K	L	D SPACE	STR.FAC.	SI(G)	U
1	1	1	2.0800	14.574	272.4	0.09921
2	2	2	1.0400	6.943	571.7	0.04727
3	3	3	0.6933	3.855	1029.4	0.02625

NEW VALUE FOR H,K,L= 1 1 1 U=0.11454 SI(G)= 236.0

PROGRAM DT4BMS 18/8 STAINLESS STEEL WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,LA= 2 0 0

H	K	L	D SPACE	STR.FAC.	SI(G)	U
2	0	0	1.8014	12.695	312.7	0.08642
4	0	0	0.9007	5.737	691.9	0.03906
6	0	0	0.6005	3.032	1308.7	0.02065

NEW VALUE FOR H,K,L= 2 0 0 U=0.09557 SI(G)= 282.8

PROGRAM DT4BMS 18/8 STAINLESS STEEL WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,LA= 2 2 0

H	K	L	D SPACE	STR.FAC.	SI(G)	U
2	2	0	1.2737	8.846	448.8	0.06022
4	4	0	0.6369	3.355	1182.9	0.02285
6	6	0	0.4246	1.650	2403.3	0.01125

NEW VALUE FOR H,K,L= 2 2 0 U=0.06231 SI(G)= 433.7

(continued)

```
SUBROUTINE XTLSYS                                C.P.H.  TITANIUM                                NSYS=  4
                                                 CRYSTAL DATA
   A=  2.95000        B=  2.95000        C=  4.68600        ALFA=  90.00        BETA=  90.00        GAMA= 120.00

                                                 CALCULATION DATA
   CA= 0.00000        CB= 0.00000        CG= -.50000        SA= 1.00000        SB= 1.00000        SG= 0.86603
   S11=  191.09468    S22=  191.09468    S33=   56.80013    S12=   95.54734    S23=   -0.00000    S31=   -0.00000
   SCRANG= 0.86603                                  VOLUME OF UNIT CELL=   35.31644

SUBROUTINE WAVESP
   ELECTRON ACCELERATING VOLTAGE OF  200.0 KILOVOLTS WAS SPECIFIED, WAVELENGTH = 0.02507 ANGSTROMS
   RELATIVISTIC MASS CORRECTION FACTOR =  1.39156          IND=  1


 SUBROUTINE ATMPOT,   VERSION 1 : 21ST. NOV.1974

 READS IN  DATA RELEVANT TO ATOMIC POSITIONS IN THE LATTICE, F ILLING FACTORS, AND SCATTERING FACTORS.
ENTRY FCALCT  CALCULATES THE STRUCTURE FACTOR & EXTINCTION DISTANCE FOR THE GIVEN INDICES

    NO.      SYMBOL  A.N.          POSITION / FILL.

     1         TI     22   0.0   0.0   0.0   /1.000    0.667 0.333 0.500 /1.000

          SCATTERING EQUATION CONSTANTS

ATOM NO.  SYMBOL  DEBYE                              (THETA/WAVE) : SCATTERING FACTOR

     1       TI    0.0
          0.0  :  8.7760    0.05 :  7.9370    0.10 :  6.1990    0.15 :  4.6430    0.20 :  3.5640    0.25 :  2.8440
          0.30 :  2.3410    0.35 :  1.9640    0.40 :  1.6680    0.45 :  1.4280    0.50 :  1.2300    0.60 :  0.9300
          0.70 :  0.7210    0.80 :  0.5730    0.90 :  0.4670    1.00 :  0.3890    1.20 :  0.2850    1.40 :  0.2190
          1.60 :  0.1750    1.80 :  0.1430    2.00 :  0.1170    2.50 :  0.0780    3.00 :  0.0550    3.50 :  0.0410
          4.00 :  0.0310    5.00 :  0.0200    6.00 :  0.0140
```


(continued)

PROGRAM DT4BMS C.P.H. TITANIUM WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,IA,LA= 0 0 0 2

H	K	I	L	D SPACE	STR.FAC.	SI(G)	U
C	0	0	2	2.3430	7.978	375.9	0.07191
C	0	0	4	1.1715	3.669	817.2	0.03307
C	0	0	6	0.7810	1.997	1501.1	0.01800

NEW VALUE FOR H,K,I,L= 0 0 0 2 U=0.08195 SI(G)= 329.8

PROGRAM DT4BMS C.P.H. TITANIUM WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,IA,LA= 1 0 -1 0

H	K	I	L	D SPACE	STR.FAC.	SI(G)	U
1	0	-1	0	2.5548	4.356	688.3	0.03927
2	0	-2	0	1.2774	2.050	1462.7	0.01848
3	0	-3	0	0.8516	2.304	1301.0	0.02077

NEW VALUE FOR H,K,I,L= 1 0 -1 0 U=0.04350 SI(G)= 621.3

PROGRAM DT4BMS C.P.H. TITANIUM WAVELENGTH= 0.0370 RELMAS= 1.196

INPUT DATA: HA,KA,IA,LA= 1 0 -1 1

H	K	I	L	D SPACE	STR.FAC.	SI(G)	U
1	0	-1	1	2.2431	6.606	453.9	0.05954
2	0	-2	2	1.1215	1.729	1734.0	0.01559
3	0	-3	3	0.7477	0.001	5116395.6	0.00001

NEW VALUE FOR H,K,I,L= 1 0 -1 1 U=0.06274 SI(G)= 430.7

(continued)

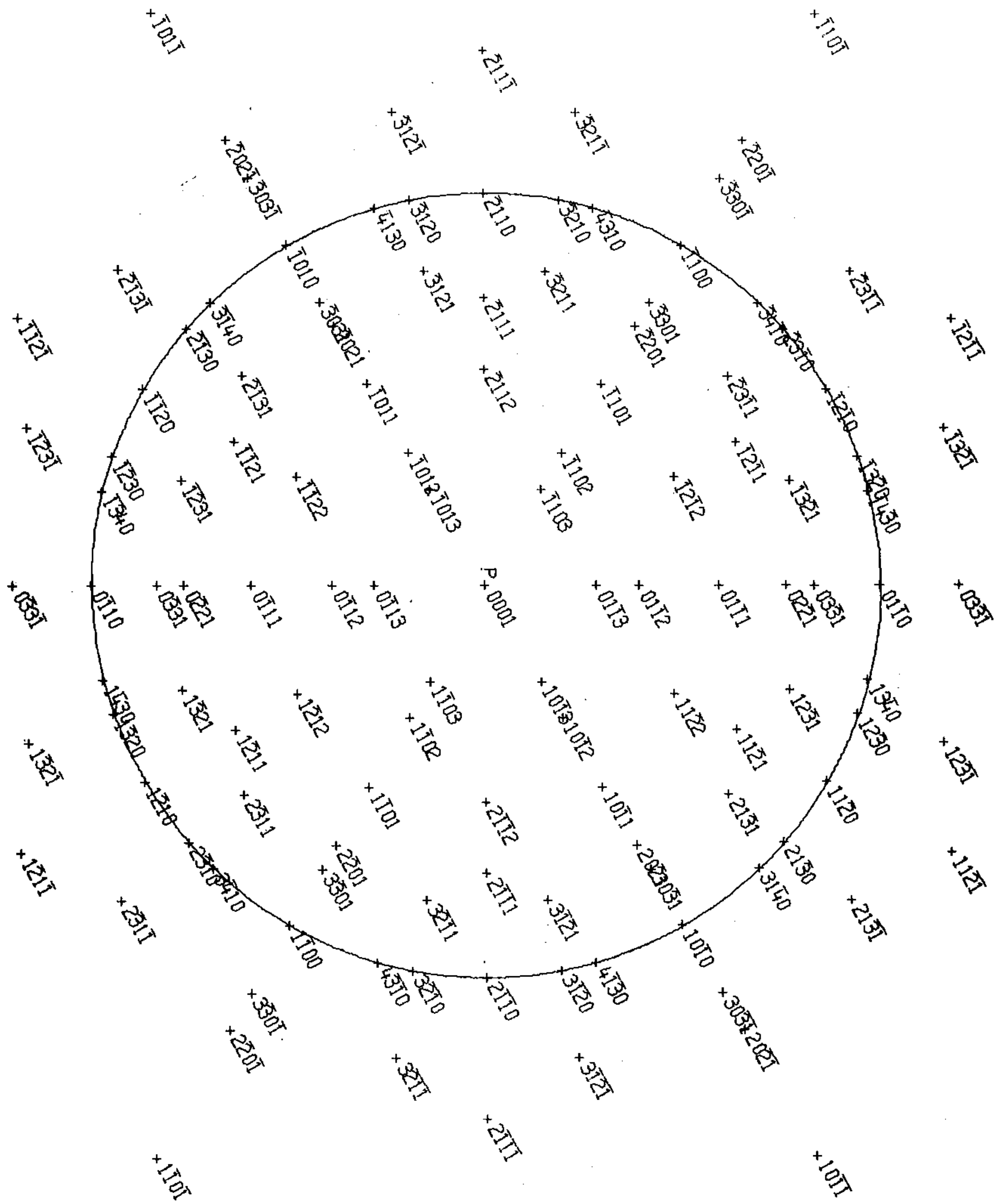
PROGRAM STEREO

POLE= 2 -1 -1 0 NDP= 2 NGSO= 2 C.P.H. ZIRCONIUM.

THETA= 0.0 RADIUS= 10.00000 ZPLN= 10.00000

H	K	I	L	COS(1)	COS(2)	COS(3)	ANG(1)	ANG(2)	ANG(3)	X	Y	CONST	LABEL
0	0	0	1	0.0	1.00000	0.0	90.000	0.0	90.000	0.0	10.000	10.000	+
0	0	0	-1	0.0	-1.00000	0.0	90.000	180.000	90.000	0.0	-10.000	10.000	+
1	0	-1	0	0.50000	0.0	0.86603	60.000	90.000	30.000	2.679	0.0	5.359	+
0	-1	1	0	-1.00000	0.0	0.0	180.000	90.000	90.000	-10.000	0.0	10.000	+
0	1	-1	0	1.00000	0.0	0.0	0.0	90.000	90.000	10.000	0.0	10.000	+
1	-1	0	0	-0.50000	0.0	0.86603	120.000	90.000	30.000	-2.679	0.0	5.359	+
1	0	-1	1	0.43931	0.47751	0.76091	63.940	61.477	40.455	2.495	2.712	5.679	+
0	-1	1	1	-0.87863	0.47751	0.0	151.477	61.477	90.000	-8.786	4.775	10.000	+
0	1	-1	1	0.87863	0.47751	0.0	28.523	61.477	90.000	8.786	4.775	10.000	+
1	-1	0	1	-0.43931	0.47751	0.76091	116.060	61.477	40.455	-2.495	2.712	5.679	+
1	0	-1	-1	0.43931	-0.47751	0.76091	63.940	118.523	40.455	2.495	-2.712	5.679	+
0	-1	1	-1	-0.87863	-0.47751	0.0	151.477	118.523	90.000	-8.786	-4.775	10.000	+
0	1	-1	-1	0.87863	-0.47751	0.0	28.523	118.523	90.000	8.786	-4.775	10.000	+
1	-1	0	-1	-0.43931	-0.47751	0.76091	116.060	118.523	40.455	-2.495	-2.712	5.679	+
1	0	-1	2	0.33853	0.73593	0.58635	70.213	42.614	54.102	2.134	4.639	6.304	+
0	-1	1	2	-0.67706	0.73593	0.0	132.614	42.614	90.000	-6.771	7.359	10.000	+
0	1	-1	2	0.67706	0.73593	0.0	47.386	42.614	90.000	6.771	7.359	10.000	+
1	-1	0	2	-0.33853	0.73593	0.58635	109.787	42.614	54.102	-2.134	4.639	6.304	+
1	0	-1	-2	0.33853	-0.73593	0.58635	70.213	137.386	54.102	2.134	-4.639	6.304	+
0	-1	1	-2	-0.67706	-0.73593	0.0	132.614	137.386	90.000	-6.771	-7.359	10.000	+
0	1	-1	-2	0.67706	-0.73593	0.0	47.386	137.386	90.000	6.771	-7.359	10.000	+
1	-1	0	-2	-0.33853	-0.73593	0.58635	109.787	137.386	54.102	-2.134	-4.639	6.304	+
1	0	-1	3	0.26142	0.85244	0.45279	74.846	31.522	63.077	1.799	5.868	6.883	+
0	-1	1	3	-0.52283	0.85244	0.0	121.522	31.522	90.000	-5.228	8.524	10.000	+
0	1	-1	3	0.52283	0.85244	0.0	58.478	31.522	90.000	5.228	8.524	10.000	+
1	-1	0	3	-0.26142	0.85244	0.45279	105.154	31.522	63.077	-1.799	5.868	6.883	+
1	0	-1	-3	0.26142	-0.85244	0.45279	74.846	148.478	63.077	1.799	-5.868	6.883	+
0	-1	1	-3	-0.52283	-0.85244	0.0	121.522	148.478	90.000	-5.228	-8.524	10.000	+
0	1	-1	-3	0.52283	-0.85244	0.0	58.478	148.478	90.000	5.228	-8.524	10.000	+
1	-1	0	-3	-0.26142	-0.85244	0.45279	105.154	148.478	63.077	-1.799	-5.868	6.883	+
1	1	-2	0	0.86603	0.0	0.50000	30.000	90.000	60.000	5.774	0.0	6.667	+
1	-2	1	0	-0.86603	0.0	0.50000	150.000	90.000	60.000	-5.774	0.0	6.667	+
2	-1	-1	0	0.0	0.0	1.00000	90.000	90.000	0.0	0.0	0.0	5.000	+
1	1	-2	1	0.82630	0.29938	0.47707	34.279	72.579	61.506	5.594	2.027	6.770	+
1	-2	1	1	-0.82630	0.29938	0.47707	145.721	72.579	61.506	-5.594	2.027	6.770	+
2	-1	-1	1	0.0	0.29938	0.95413	90.000	72.579	17.421	0.0	1.532	5.117	+
1	1	-2	-1	0.82630	-0.29938	0.47707	34.279	107.421	61.506	5.594	-2.027	6.770	+
1	-2	1	-1	-0.82630	-0.29938	0.47707	145.721	107.421	61.506	-5.594	-2.027	6.770	+
2	-1	-1	-1	0.0	-0.29938	0.95413	90.000	107.421	17.421	0.0	-1.532	5.117	+
1	1	-2	2	0.73355	0.53155	0.42351	42.815	57.890	64.943	5.153	3.734	7.025	+
1	-2	1	2	-0.73355	0.53155	0.42351	137.185	57.890	64.943	-5.153	3.734	7.025	+
-1	-1	2	2	-0.73355	0.53155	-0.42351	137.185	57.890	115.057	-12.724	9.221	17.346	+
-1	2	-1	2	0.73355	0.53155	-0.42351	42.815	57.890	115.057	12.724	9.221	17.346	+
2	-1	-1	2	0.0	0.53155	0.84703	90.000	57.890	32.110	0.0	2.878	5.414	+
1	1	-2	-2	0.73355	-0.53155	0.42351	42.815	122.110	64.943	5.153	-3.734	7.025	+
1	-2	1	-2	-0.73355	-0.53155	0.42351	137.185	122.110	64.943	-5.153	-3.734	7.025	+
-1	-1	2	-2	-0.73355	-0.53155	-0.42351	137.185	122.110	115.057	-12.724	-9.221	17.346	+
-1	2	-1	-2	0.73355	-0.53155	-0.42351	42.815	122.110	115.057	12.724	-9.221	17.346	+
2	-1	-1	-2	0.0	-0.53155	0.84703	90.000	122.110	32.110	0.0	-2.878	5.414	+
2	0	-2	1	0.48250	0.26223	0.83572	61.151	74.798	33.309	2.628	1.428	5.447	+
0	-2	2	1	-0.96501	0.26223	0.0	164.798	74.798	90.000	-9.650	2.622	10.000	+
0	2	-2	1	0.96501	0.26223	0.0	15.202	74.798	90.000	9.650	2.622	10.000	+
2	-2	0	1	-0.48250	0.26223	0.83572	118.849	74.798	33.309	-2.628	1.428	5.447	+

(continued)



STEREOGRAPHIC PROJECTION OF PLANES OF C.P.H. ZIRCONIUM.

POLE = 0001	SCALE: RADIUS= 10.00 CMS.	
A= 3.2300	B= 3.2300	C= 5.1470
ALFA= 90.00	BETA= 90.00	GAMA= 120.00

(continued)

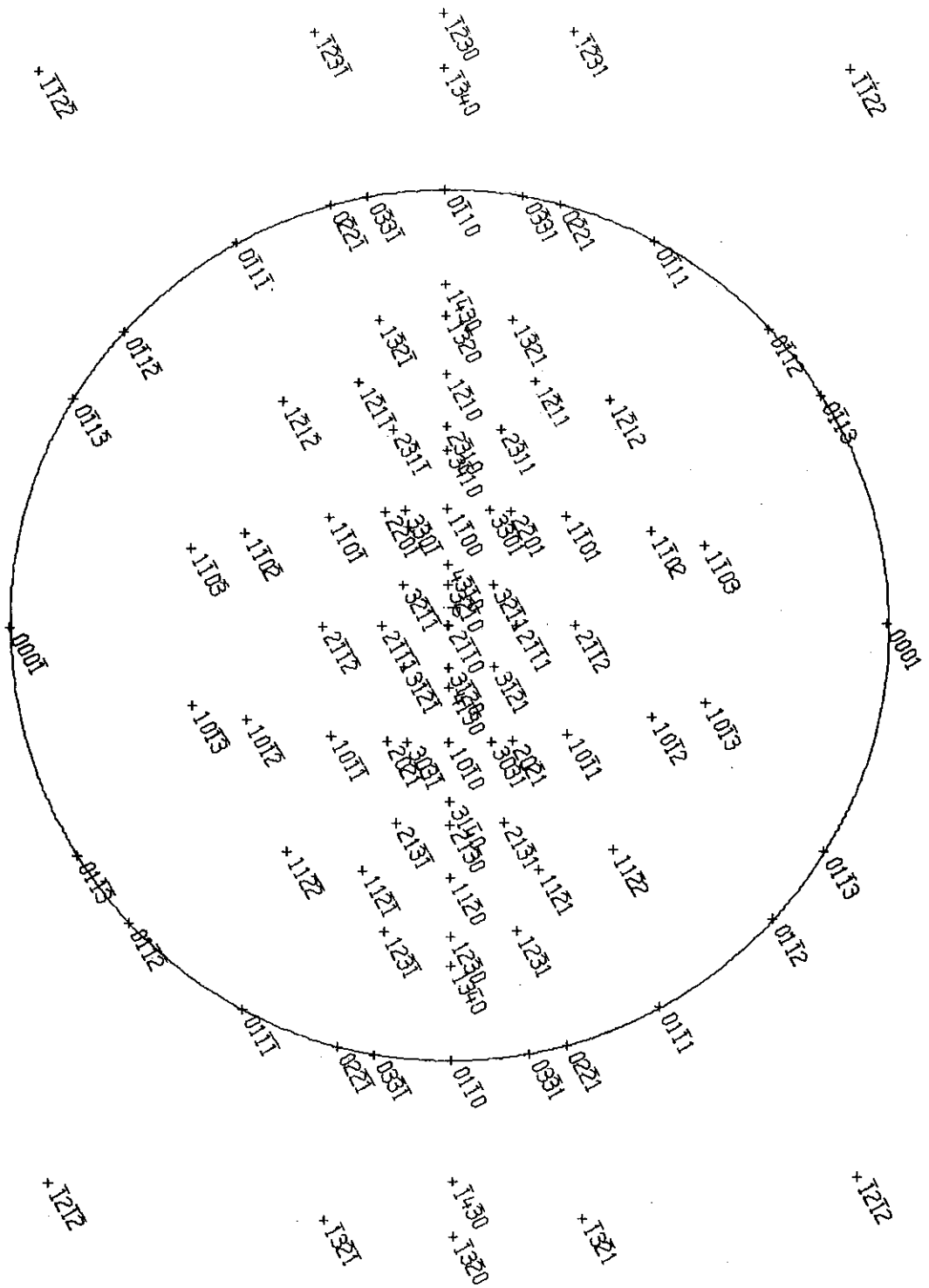
PROGRAM STEREO

FCLE= 0 0 0 1 NDP= 2 NGSO= 2 C.P.H. ZIRCONIUM.

THETA= 0.0 RADIUS= 10.00000 ZPLN= 10.00000

H	K	I	L	COS(1)	COS(2)	COS(3)	ANG(1)	ANG(2)	ANG(3)	X	Y	CONST	LABEL
0	0	0	1	0.0	0.0	1.00000	90.000	90.000	0.0	0.0	0.0	5.000	+
1	0	-1	0	0.86603	0.50000	0.0	30.000	60.000	90.000	8.660	5.000	10.000	+
0	-1	1	0	0.0	-1.00000	0.0	90.000	180.000	90.000	0.0	-10.000	10.000	+
-1	1	0	0	-0.86603	0.50000	0.0	150.000	60.000	90.000	-8.660	5.000	10.000	+
-1	0	1	0	-0.86603	-0.50000	0.0	150.000	120.000	90.000	-8.660	-5.000	10.000	+
0	1	-1	0	0.0	1.00000	0.0	90.000	0.0	90.000	0.0	10.000	10.000	+
1	-1	0	0	0.86603	-0.50000	0.0	30.000	120.000	90.000	8.660	-5.000	10.000	+
1	0	-1	1	0.76091	0.43931	0.47751	40.455	63.940	61.477	5.150	2.973	6.768	+
0	-1	1	1	0.0	-0.87863	0.47751	90.000	151.477	61.477	0.0	-5.947	6.768	+
-1	1	0	1	-0.76091	0.43931	0.47751	139.545	63.940	61.477	-5.150	2.973	6.768	+
-1	0	1	1	-0.76091	-0.43931	0.47751	139.545	116.060	61.477	-5.150	-2.973	6.768	+
0	1	-1	1	0.0	0.87863	0.47751	90.000	28.523	61.477	0.0	5.947	6.768	+
1	-1	0	1	0.76091	-0.43931	0.47751	40.455	116.060	61.477	5.150	-2.973	6.768	+
1	0	-1	-1	0.76091	0.43931	-0.47751	40.455	63.940	118.523	14.563	8.408	19.139	+
-1	1	0	-1	-0.76091	0.43931	-0.47751	139.545	63.940	118.523	-14.563	8.408	19.139	+
-1	0	1	-1	-0.76091	-0.43931	-0.47751	139.545	116.060	118.523	-14.563	-8.408	19.139	+
1	-1	0	-1	0.76091	-0.43931	-0.47751	40.455	116.060	118.523	14.563	-8.408	19.139	+
1	0	-1	2	0.58635	0.33853	0.73593	54.102	70.213	42.614	3.378	1.950	5.761	+
0	-1	1	2	0.0	-0.67706	0.73593	90.000	132.614	42.614	0.0	-3.900	5.761	+
-1	1	0	2	-0.58635	0.33853	0.73593	125.898	70.213	42.614	-3.378	1.950	5.761	+
-1	0	1	2	-0.58635	-0.33853	0.73593	125.898	109.787	42.614	-3.378	-1.950	5.761	+
0	1	-1	2	0.0	0.67706	0.73593	90.000	47.386	42.614	0.0	3.900	5.761	+
1	-1	0	2	0.58635	-0.33853	0.73593	54.102	109.787	42.614	3.378	-1.950	5.761	+
1	0	-1	3	0.45279	0.26142	0.85244	63.077	74.846	31.522	2.444	1.411	5.398	+
0	-1	1	3	0.0	-0.52283	0.85244	90.000	121.522	31.522	0.0	-2.822	5.398	+
-1	1	0	3	-0.45279	0.26142	0.85244	116.923	74.846	31.522	-2.444	1.411	5.398	+
-1	0	1	3	-0.45279	-0.26142	0.85244	116.923	105.154	31.522	-2.444	-1.411	5.398	+
0	1	-1	3	0.0	0.52283	0.85244	90.000	58.478	31.522	0.0	2.822	5.398	+
1	-1	0	3	0.45279	-0.26142	0.85244	63.077	105.154	31.522	2.444	-1.411	5.398	+
1	1	-2	0	0.50000	0.86603	0.0	60.000	30.000	90.000	5.000	8.660	10.000	+
1	-2	1	0	0.50000	-0.86603	0.0	60.000	150.000	90.000	5.000	-8.660	10.000	+
-2	1	1	0	-1.00000	0.0	0.0	180.000	90.000	90.000	-10.000	0.0	10.000	+
-1	-1	2	0	-0.50000	-0.86603	0.0	120.000	150.000	90.000	-5.000	-8.660	10.000	+
-1	2	-1	0	-0.50000	0.86603	0.0	120.000	30.000	90.000	-5.000	8.660	10.000	+
2	-1	-1	0	1.00000	0.0	0.0	0.0	90.000	90.000	10.000	0.0	10.000	+
1	1	-2	1	0.47707	0.82630	0.29938	61.506	34.279	72.579	3.671	6.359	7.696	+
1	-2	1	1	0.47707	-0.82630	0.29938	61.506	145.721	72.579	3.671	-6.359	7.696	+
-2	1	1	1	-0.95413	0.0	0.29938	162.579	90.000	72.579	-7.343	0.0	7.696	+
-1	-1	2	1	-0.47707	-0.82630	0.29938	118.494	145.721	72.579	-3.671	-6.359	7.696	+
-1	2	-1	1	-0.47707	0.82630	0.29938	118.494	34.279	72.579	-3.671	6.359	7.696	+
2	-1	-1	1	0.95413	0.0	0.29938	17.421	90.000	72.579	7.343	0.0	7.696	+
1	1	-2	-1	0.47707	0.82630	-0.29938	61.506	34.279	107.421	6.809	11.794	14.273	+
1	-2	1	-1	0.47707	-0.82630	-0.29938	61.506	145.721	107.421	6.809	-11.794	14.273	+
-2	1	1	-1	-0.95413	0.0	-0.29938	162.579	90.000	107.421	-13.618	0.0	14.273	+
-1	-1	2	-1	-0.47707	-0.82630	-0.29938	118.494	145.721	107.421	-6.809	-11.794	14.273	+
-1	2	-1	-1	-0.47707	0.82630	-0.29938	118.494	34.279	107.421	-6.809	11.794	14.273	+
2	-1	-1	-1	0.95413	0.0	-0.29938	17.421	90.000	107.421	13.618	0.0	14.273	+
1	1	-2	2	0.42351	0.73355	0.53155	64.943	42.815	57.890	2.765	4.790	6.529	+
1	-2	1	2	0.42351	-0.73355	0.53155	64.943	137.185	57.890	2.765	-4.790	6.529	+
-2	1	1	2	-0.84703	0.0	0.53155	147.890	90.000	57.890	-5.531	0.0	6.529	+
-1	-1	2	2	-0.42351	-0.73355	0.53155	115.057	137.185	57.890	-2.765	-4.790	6.529	+
-1	2	-1	2	-0.42351	0.73355	0.53155	115.057	42.815	57.890	-2.765	4.790	6.529	+
2	-1	-1	2	0.84703	0.0	0.53155	32.110	90.000	57.890	5.531	0.0	6.529	+

(continued)



STEREOGRAPHIC PROJECTION OF PLANES OF C.P.H. ZIRCONIUM.

POLE = $\bar{2}110$ SCALE: RADIUS= 10.00 CMS.

A= 3.2300 B= 3.2300 C= 5.1470

ALFA= 90.00 BETA= 90.00 GAMA= 120.00

(continued)

SUBROUTINE XTLSYS IRON NSYS= 1

CRYSTAL DATA

A= 2.86600 B= 2.86600 C= 2.86600 ALFA= 90.00 BETA= 90.00 GAMA= 90.00

CALCULATION DATA

CA= 0.00000 CB= 0.00000 CG= 0.00000 SA= 1.00000 SB= 1.00000 SG= 1.00000
S11= 67.46907 S22= 67.46907 S33= 67.46907 S12= -0.00000 S23= -0.00000 S31= -0.00000
SCRANG= 1.00000 VOLUME OF UNIT CELL= 23.54120

SUBROUTINE INDEX

HMIN = 0 HMAX = 4 KMIN = -4 KMAX = 4 LMIN = -4 LMAX = 4
IH = 1 IK = 1 IL = 0 INDMAX = 4 NPARM = 0 DMIN = 0.0
KABS = 1 JABS = 5 NSABS = 1 NMULT = 2 NCRDER = 1 NORD = 3 2 1

SUBROUTINE WAVESP

X-RAY WAVELENGTH OF 1.93600 ANGSTROMS WAS SPECIFIED, IND= 2

PROGRAM KIKU KOSSEL PLOT WITH IRON K-ALPHA RADIATION.

CX,CY= 6.000 4.000 NLABEL= 1 SCALE= 20.00 DEGREES FOR THE FIRST INCH FROM THE POLE
Z (THE PROJECTION DISTANCE) = 2.75 INCHES THETAM= 69.14 DEGREES FROM THE POLE TO THE FURTHUREST POINT

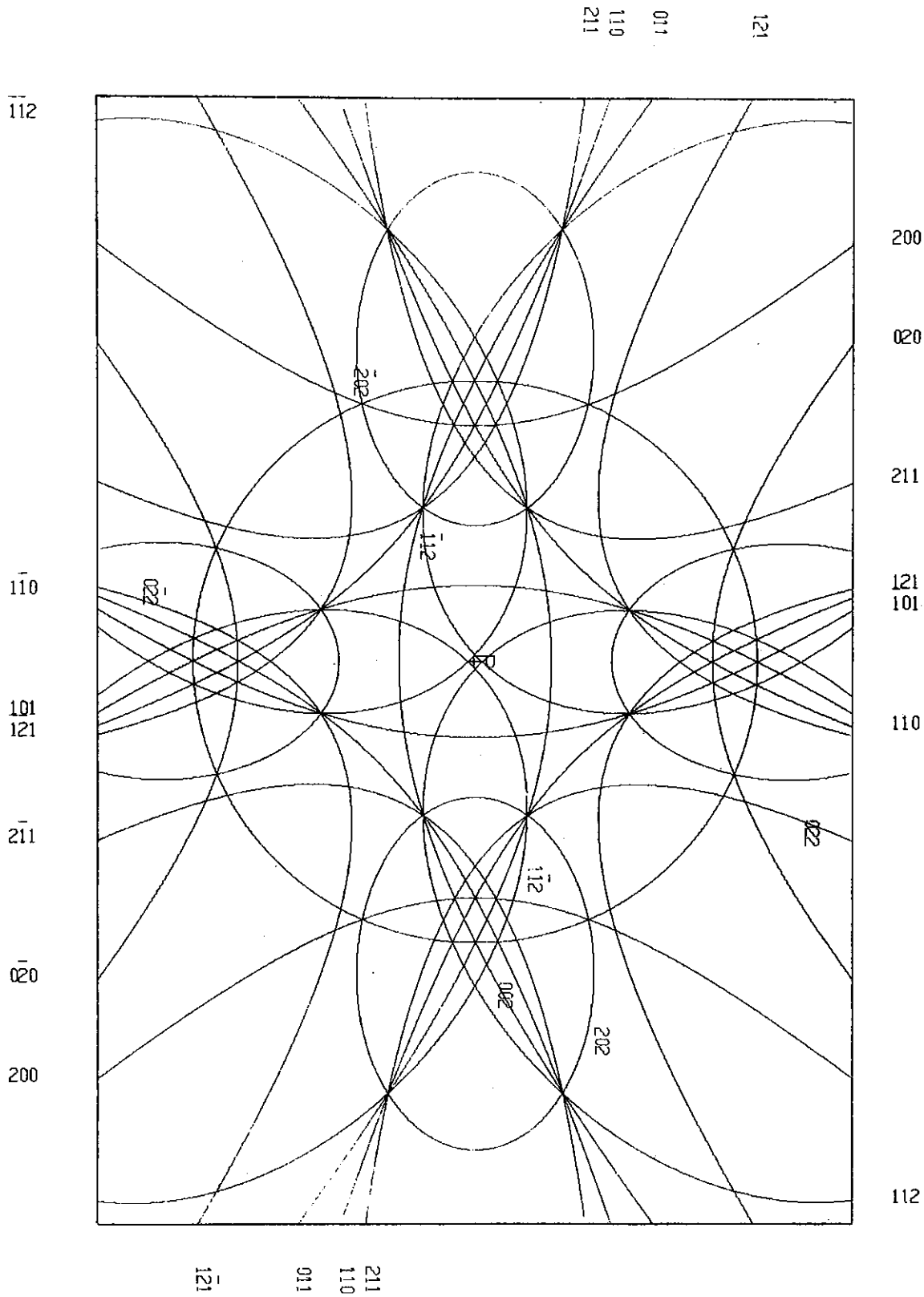
X AXIS= 1.000 0.0 0.0 Y AXIS= -0.000 1.000 -0.000 Z AXIS= 0.0 0.0 1.000

INDICES			BRAGG	D	FIRST LINE				SECOND LINE			
					START		FINISH		START		FINISH	
H	K	L	ANGLE	SPACING	X	Y	X	Y	X	Y	X	Y
1	1	0	28.532	2.027	-0.69	4.00	6.00	-1.43				
1	0	1	28.532	2.027	0.55	4.00	0.55	-4.00				
0	1	1	28.532	2.027	6.00	1.89	-6.00	1.89				
1	-1	0	28.532	2.027	5.98	1.43	-0.69	-4.00				
-1	0	1	28.532	2.027	-0.55	-4.00	-0.55	4.00				
0	-1	1	28.532	2.027	-6.00	-1.89	6.00	-1.89				
-1	1	0	28.532	2.027	-5.98	-1.43	0.69	4.00				
-1	-1	0	28.532	2.027	0.69	-4.00	-6.00	1.43				

(continued)

INDICES H	K	L	BRAGG ANGLE	D SPACING	FIRST LINE START X	START Y	FINISH X	FINISH Y	SECOND LINE START X	START Y	FINISH X	FINISH Y
2	0	0	42.493	1.433	4.45	4.00	4.45	-4.00				
0	2	0	42.493	1.433	3.39	4.00	-3.39	4.00				
0	0	2	42.493	1.433	2.99	-0.23	-2.99	-0.23	-2.99	0.23	2.99	0.23
-2	0	0	42.493	1.433	-4.45	-4.00	-4.45	4.00				
0	-2	0	42.493	1.433	-3.39	-4.00	3.39	-4.00				
2	1	1	55.825	1.170	1.91	4.00	6.00	-1.16				
1	2	1	55.825	1.170	6.00	2.95	-0.78	4.00				
1	1	2	55.825	1.170	0.37	4.00	1.43	-0.55	1.75	-0.55	5.75	4.00
-1	1	2	55.825	1.170	-1.43	-0.55	-0.37	4.00	-5.75	4.00	-1.75	-0.55
2	-1	1	55.825	1.170	5.95	1.16	1.91	-4.00				
-1	2	1	55.825	1.170	0.78	3.98	-6.00	2.95				
1	-1	2	55.825	1.170	1.43	0.55	0.37	-4.00	5.75	-4.00	1.75	0.55
-1	-1	2	55.825	1.170	-0.37	-4.00	-1.43	0.55	-1.75	0.55	-5.75	-4.00
-2	1	1	55.825	1.170	-5.95	-1.16	-1.91	4.00				
1	-2	1	55.825	1.170	-0.78	-3.98	6.00	-2.95				
-2	-1	1	55.825	1.170	-1.91	-4.00	-6.00	1.16				
-1	-2	1	55.825	1.170	-6.00	-2.95	0.78	-4.00				
2	0	2	72.806	1.013	3.17	1.26	3.17	-1.26	3.49	-1.26	3.49	1.26
0	2	2	72.806	1.013	1.26 1.19	3.17 3.96	-1.26 1.26	3.17 3.49	-1.26	3.49	-1.19	4.00
-2	0	2	72.806	1.013	-3.17	-1.26	-3.17	1.26	-3.49	1.26	-3.49	-1.26
0	-2	2	72.806	1.013	-1.26 -1.19	-3.17 -3.96	1.26 -1.26	-3.17 -3.49	1.26	-3.49	1.19	-4.00

(continued)



KOSSEL PLOT WİTH IRON K-ALPHA RADIATION.
IRON WAVELENGTH= 1.93600
POLE = 001
SCALE= 20.00 DEGREES FOR THE FIRST INCH FROM POLE
A= 2.8660 B= 2.8660 C= 2.8660
ALFA= 90.00 BETA= 90.00 GAMA= 90.00

(continued)

SUBROUTINE XTLSYS — IRON — NSYS= 1

CRYSTAL DATA

A= 2.86600 B= 2.86600 C= 2.86600 ALFA= 90.00 BETA= 90.00 GAMA= 90.00

CALCULATION DATA

CA= 0.00000 CB= 0.00000 CG= 0.00000 SA= 1.00000 SB= 1.00000 SG= 1.00000
S11= 67.46907 S22= 67.46907 S33= 67.46907 S12= -0.00000 S23= -0.00000 S31= -0.00000
SQRANG= 1.00000 VOLUME OF UNIT CELL= 23.54120

SUBROUTINE INDEX

HMIN = 0 HMAX = 4 KMIN = -4 KMAX = 4 LMIN = -4 LMAX = 4
IH = 1 IK = 0 IL = 0 INDMAX = 4 NPARM = 0 DMIN = 0.0
KABS = P JABS = 10 NSABS = 1 NMULT = 2 NORDER = 1 NORD = 3 2 1

SUBROUTINE WAVESP

ELECTRON ACCELERATING VOLTAGE OF 200.0 KILOVOLTS WAS SPECIFIED, WAVELENGTH = 0.02507 ANGSTROMS
RELATIVISTIC MASS CORRECTION FACTOR = 1.39156 IND= 1

PROGRAM KIKU — KIKUCHI PLOT FOR 200 KV. ELECTRON BEAM.

CX,CY= 6.000 4.000 NLABEL= 1 SCALE= 5.00 DEGREES FOR THE FIRST INCH FROM THE POLE
Z (THE PROJECTION DISTANCE) = 11.43 INCHES THETAM= 32.25 DEGREES FROM THE POLE TO THE FURTHUREST POINT

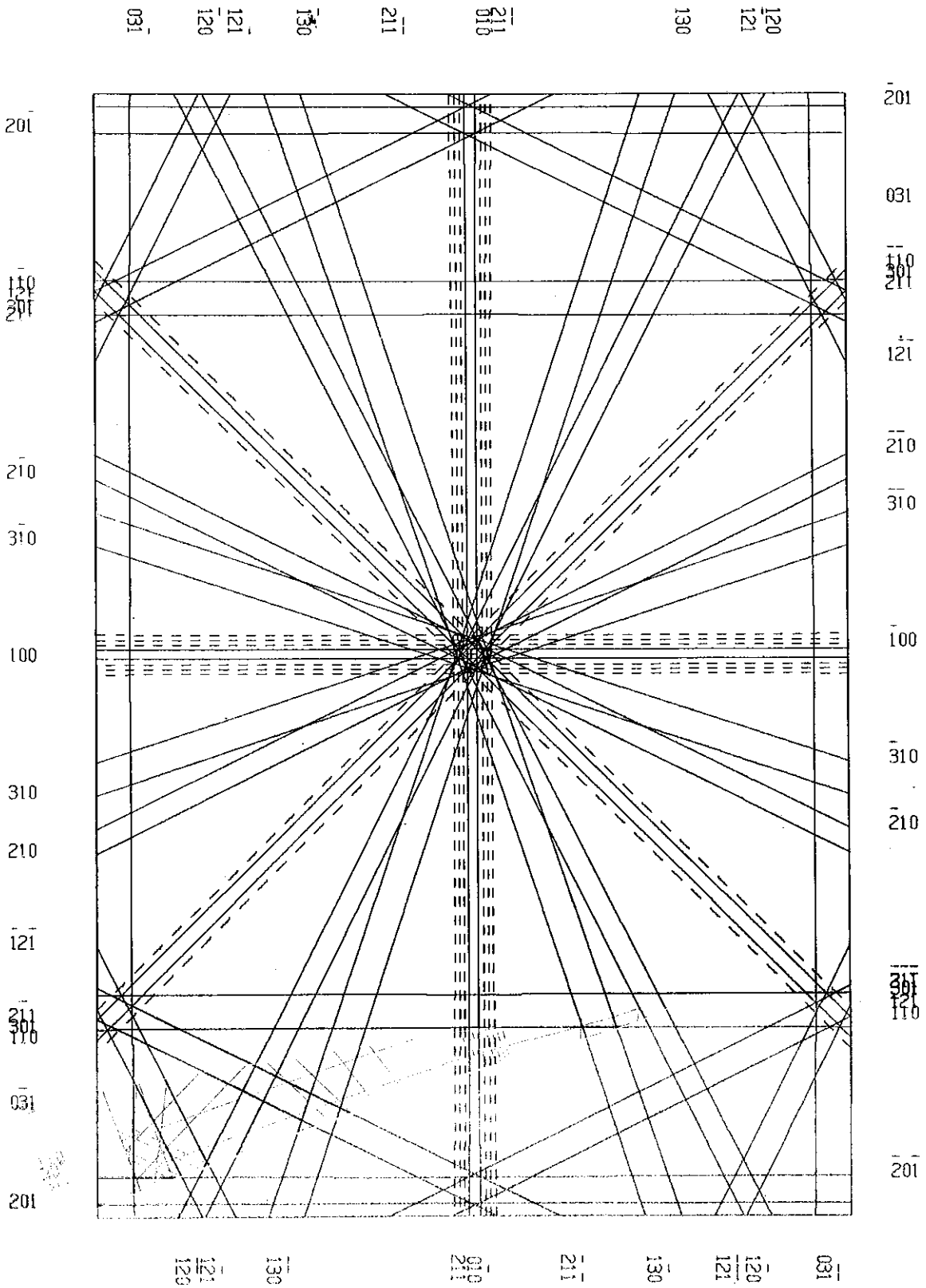
X AXIS= 1.000 0.0 0.0 Y AXIS= -0.000 1.000 -0.000 Z AXIS= 0.0 0.0 1.000

INDICES			BRAGG	D	FIRST LINE START		FIRST LINE FINISH		SECOND LINE START		SECOND LINE FINISH	
H	K	L	ANGLE	SPACING	X	Y	X	Y	X	Y	X	Y
1	0	0	0.251	2.866	0.05	4.00	0.05	-4.00				
0	1	0	0.251	2.866	6.00	0.06	-6.00	0.06				
-1	0	0	0.251	2.866	-0.05	-4.00	-0.05	4.00				
0	-1	0	0.251	2.866	-6.00	-0.06	6.00	-0.06				
1	1	0	0.354	2.027	-3.89	4.00	4.11	-4.00				
1	-1	0	0.354	2.027	4.11	4.00	-3.89	-4.00				
-1	1	0	0.354	2.027	-4.11	-4.00	3.89	4.00				
-1	-1	0	0.354	2.027	3.89	-4.00	-4.11	4.00				

(continued)

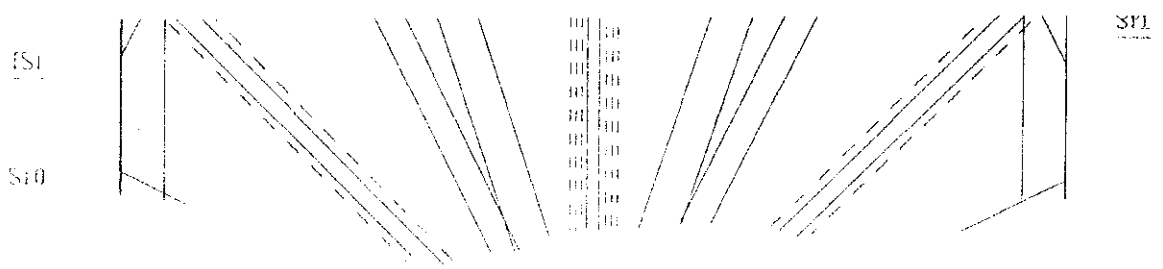
INDICES H	K	L	BRAGG ANGLE	D SPACING	FIRST LINE START X	START Y	FINISH X	FINISH Y	SECOND LINE START X	START Y	FINISH X	FINISH Y
2	0	0	0.501	1.433	0.11	4.00	0.11	-4.00				
0	2	0	0.501	1.433	6.00	0.11	-6.00	0.11				
-2	0	0	0.501	1.433	-0.11	-4.00	-0.11	4.00				
0	-2	0	0.501	1.433	-6.00	-0.11	6.00	-0.11				
2	1	0	0.560	1.282	-1.87	4.00	2.13	-4.00				
2	0	1	0.560	1.282	-5.57	4.00	-5.57	-4.00				
1	2	0	0.560	1.282	6.00	-2.86	-6.00	3.15				
2	-1	0	0.560	1.282	2.13	4.00	-1.87	-4.00				
2	0	-1	0.560	1.282	5.86	4.00	5.86	-4.00				
-1	2	0	0.560	1.282	6.00	3.15	-6.00	-2.86				
-2	1	0	0.560	1.282	-2.13	-4.00	1.87	4.00				
-2	0	1	0.560	1.282	5.57	-4.00	5.57	4.00				
1	-2	0	0.560	1.282	-6.00	-3.15	6.00	2.86				
-2	-1	0	0.560	1.282	1.87	-4.00	-2.13	4.00				
-2	0	-1	0.560	1.282	-5.86	-4.00	-5.86	4.00				
-1	-2	0	0.560	1.282	-6.00	2.86	6.00	-3.15				
2	1	1	0.614	1.170	-6.00	0.90	-3.55	-4.00				
1	2	1	0.614	1.170	-3.11	-4.00	-6.00	-2.54				
2	1	-1	0.614	1.170	3.88	4.00	6.00	-0.23				
1	2	-1	0.614	1.170	6.00	2.89	3.77	4.00				
2	-1	1	0.614	1.170	-3.55	4.00	-6.00	-0.90				
-1	2	1	0.614	1.170	6.00	-2.54	3.11	-4.00				
2	-1	-1	0.614	1.170	6.00	0.23	3.88	-4.00				
-1	2	-1	0.614	1.170	-3.77	4.00	-6.00	2.89				
-2	1	1	0.614	1.170	3.55	-4.00	6.00	0.90				
1	-2	1	0.614	1.170	-6.00	2.54	-3.11	4.00				

(continued)



KIKUCHI PLOT FOR 200 KV. ELECTRON BEAM.
IRON WAVELENGTH= 0.02507

POLE = 001
SCALE= 5.00 DEGREES FOR THE FIRST INCH FROM POLE
A= 2.8660 B= 2.8660 C= 2.8660
ALFA= 90.00 BETA= 90.00 GAMA= 90.00



APPENDIX E

INDEX OF SUBROUTINES AND ENTRY POINTS