Iowa State University
Digital Repository

Retrospective Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

1972

The crystal structure determinations of trans-dicyano trien cobalt (III) perchlorate and potassium antimony (III) pentachloride and two phase determination techniques based on Patterson deconvolution

Robert Kingsley Wismer
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Part of the Physical Chemistry Commons

Recommended Citation

Wismer, Robert Kingsley, "The crystal structure determinations of trans-dicyano trien cobalt (III) perchlorate and potassium antimony (III) pentachloride and two phase determination techniques based on Patterson deconvolution " (1972). *Retrospective Theses and Dissertations*. 6135.
https://lib.dr.iastate.edu/rtd/6135

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

www.manaraa.com

INFORMATION TO USERS

This dissertation was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

University Microfilms

300 North Zeeb Road
Ann Arbor, Michigan 48106

A Xerox Education Company

73-9500

WISMER, Robert Kingsley, 1945-
THE CRYSTAL STRUCTURE DETERMINATIONS OF
TRANS-DICYANO TRIEN COBALT(III) PERCHLORATE
AND POTASSIUM ANTIMONY(III) PENTACHLORIDE
AND TWO PHASE DETERMINATION TECHNIQUES
BASED ON PATTERSON DECONVOLUTION.

Iowa State University, Ph.D., 1972
Chemistry, physical

University Microfilms, A XEROX Company, Ann Arbor, Michigan

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED.

The crystal structure determinations of trans-dicyano trien cobalt(III) perchlorate and potassium antimony(III) pentachloride and two phase determination techniques based on Patterson deconvolution

by

Robert Kingsley Wismer

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of
The Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Department: Chemistry

Major: Physical Chemistry

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Of Science and Technology
Ames, Iowa

1972

PLEASE NOTE:

Some pages may have

indistinct print.

Filmed as received.

University Microfilms, A Xerox Education Company

TABLE OF CONTENTS

INTRODUCTION

This thesis describes the crystal structures of two compounds and details two methods for the determination of the phases of structure factors. The crystal structures were undertaken to extend the currently available knowledge of cobalt and antimony complexes.

The cobalt compound was believed by its discoverers to be cis-$\alpha(NC)_2$ trien cobalt(III) perchlorate. Thus, it would have been the sole known mononuclear cobalt complex in which the cyanide group was coordinated to cobalt through the nitrogen atom. The structure that was determined is actually trans-$(CN)_2$ trien cobalt(III) perchlorate and is one of the few examples in which a quadradentate ligand, not restricted by its own structure to be planar, is equatorially coordinated to cobalt.

The second structure described is that of K_2SbCl_5. The structure of the $SbCl_5^=$ anion has only been determined once previously, in the compound $(NH_4)_2SbCl_5$. In that instance, the disagreement between observed and calculated structure factors was such that the uncertainty in bond distances was 0.5 Å. Structures of antimony halides are known to be very sensitive to the cation present.

The first of the phase determining methods applies the techniques of pseudo heavy atom superposition and then uses space group symmetry to determine atomic locations in electron density space. All of these operations are performed by a computer program with a minimum of human intervention.

The second phase determining method also uses superposition techniques, to enhance the phase determining properties of a pseudo electron density map which has been computed from origin-defining reflections. The coefficients of the Fourier transform of this modified map are then used to phase those reflections with the largest normalized unitary structure factors. This method greatly simplifies one of the major problems of direct methods, the expansion of the origin-defining set.

THE CRYSTAL AND MOLECULAR STRUCTURE OF TRANS-DICYANO TRIETHYLENETETRAMINE COBALT(III) PERCHLORATE

Introduction

The bonding mode of cyanide to cobalt has been the subject of considerable controversy in recent years. The energy difference between the cyano and isocyano coordinations is small enough to permit either to occur.[1] However, neutron diffraction studies of cyano-cobalt complexes[2,3,4] have regularly shown that the cyanide group bonds through carbon. Since it is ofttimes difficult to distinguish between atoms which differ by but one electron by X-ray diffraction, bonding via the carbon has sometimes been assumed.[5]

Kuroda and Gentile[6] recently have prepared what they characterize as an isocyano cobalt complex, cis-α-isocyanotriethylenetetramine cobalt(III) perchlorate. Their assignment of the isocyano coordination is based on visible and ultraviolet spectroscopy. Because there is some doubt as to the correctness of this assignment,[7] it was decided to carry out an X-ray structure determination of this compound.

Experimental

A sample of the compound, prepared by the method of Kuroda and Gentile,[6] was kindly supplied by Dr. James H.

Espenson. Amber-colored crystals were obtained by recrystallization from an aqueous solution. High resolution infrared spectra of the initial sample and of the recrystallized material, both in KBr pellets, were virtually identical, indicating that isomerization had not occurred during solution or recrystallization. Microscopic examination revealed that the crystals have sharply defined faces and are needle-like in appearance with a distorted hexagonal cross section in which alternate sides are of unequal lengths.

Crystals were selected and mounted on a glass fiber with Duco cement thinned with amyl acetate. Preliminary Weissenberg and precession photographs exhibited 2/m Laue symmetry, indicating a monoclinic space group. The following systematic absences were observed: $h0\ell$ when $h + \ell = 2n + 1$, and $0k0$ when $k = 2n + 1$. These absences are only consistent with the space group $P2_1/n$. The unit cell parameters at 25° C are $\underline{a} = 9.8544 \pm 0.0015$, $\underline{b} = 22.3529 \pm 0.0045$, $\underline{c} = 6.6766 \pm 0.0020$ Å, and $\beta = 100.85 \pm 0.03°$. These parameters and their standard deviations were obtained by a least squares fit[8] to the two-theta values of twelve independent reflections whose centers were determined by top-bottom, left-right beam splitting on a previously aligned Hilger-Watts four-circle diffractometer (MoKα radiation, $\lambda = 0.71069$ Å). Any error in the instrumental zero was eliminated by centering the reflection at both plus two-theta and minus two-theta. A calculated density of 1.62 g/cc for four molecules per unit cell agrees quite well with an observed density of 1.65 ± 0.01 g/cc, which was determined by flotation techniques.

For data collection, a crystal was selected having approximate dimensions 0.1 x 0.1 x 0.9 mm along the a, b, and c crystal axes, respectively, and was mounted such that the c axis coincided with the ϕ axis of the diffractometer. Data were collected at room temperature using a Hilger-Watts four-circle diffractometer interfaced to an SDS 910 computer in a real time mode, equipped with a scintillation counter, and using Zr-filtered MoKα radiation. Within a two-theta sphere of 60° ($\sin\theta/\lambda$ = 0.704 Å^{-1}) all data in the hkℓ and $\bar{h}$kℓ octants were recorded using the θ-2θ scan technique with a take-off angle of 5.5°. Symmetric scan ranges of 1.2° in 2θ at low two-theta values to 2.0° at large two-theta values were used. Stationary-crystal, stationary-counter background counts of half the scan time were taken at the beginning and end of each scan. A counting rate of 0.4096 seconds per step of 0.01° in θ was employed. A total of 5081 reflections were measured in this way.

As a general check on electronic and crystal stability, the intensities of three standard reflections were remeasured periodically during the data collection period. These reflections did not vary to any significant degree during the entire period of data collection.

Based on a linear absorption coefficient of $\mu = 14.982\ \mathrm{cm}^{-1}$, the maximum and minimum transmission factors[9] were 87.86% and 85.96% respectively, and no absorption correction was deemed necessary.

The intensity data were corrected for Lorentz-polarization effects. The estimated error in each intensity was calculated by

$$\sigma_I^2 = C_T + C_B + (0.03\ C_T)^2 + (0.03\ C_B)^2$$

where C_T and C_B are the total count and the background count, respectively. The factor 0.03 represents an estimate of non-statistical errors. The estimated deviations in the structure factors were calculated by the finite difference method.[10] Of the 4678 independent reflections, 3314 were considered observed ($> 3\sigma_F$).

Solution and Refinement

The positions of the cobalt and chlorine atoms were obtained from analysis of a sharpened Patterson function.[11] The remaining non-hydrogen atoms were found by successive structure factor[12] and electron density map calculations.[13,14] These atomic positions were then refined by a full matrix least squares procedure,[12] minimizing the function $\Sigma\omega(|F_o|-|F_c|)^2$, where $\omega = 1/\sigma_F{}^2$, to a conventional discrepancy factor of $R = \Sigma||F_o|-|F_c||/\Sigma|F_o| = 0.170$. The scattering factors used were those of Hanson *et al.*[15] with the cobalt and chlorine scattering factors modified for the real and imaginary parts of anomalous dispersion.[16]

An electron density difference map verified that all the non-hydrogen atoms had been accounted for, but indicated that some anisotropic motion, particularly of the cobalt and chlorine atoms, was present. Accordingly, anisotropic refinement was begun and after six cycles of refinement, values of R and $R\ (= \Sigma\omega(|F_o|-|F_c|)^2/\Sigma\omega|F_o|)^2)^{1/2})$ of 0.154 and 0.169, respectively, were obtained. The positions of the hydrogens bonded to carbons in the triethylenetetramine ligand were calculated with an H-C-C angle of 109.5° and a C-H distance of 1.07 Å. The hydrogen atoms were assigned isotropic thermal parameters of 3.0 $Å^2$. Two cycles of refinement with anisotropic thermal parameters for all non-hydrogen atoms, varying only the positional and thermal parameters of the non-hydrogen atoms and the overall scale factor, followed by a recalculation of the hydrogen atom positions and two further cycles of least squares refinement, yielded values for R and ωR of 0.123

and 0.156, respectively.

A final electron density difference map showed no peaks greater than 1.2 e/Å^3. A final statistical analysis of the F_o and F_c values as a function of the scattering angle and magnitude of F_o revealed no unusual trends and suggests that the relative weighting scheme used is a reasonable one.

The bonding orientation of the two cyanide groups was determined after the final least squares refinement. The atom multipliers of the two cyanide group atoms were allowed to vary under two sets of conditions. In the first of these, carbon scattering factors were assigned to C(7) and C(8) and nitrogen scattering factors were assigned to N(5) and N(6). In the second instance, the scattering factor assignments were reversed. All four atom multipliers were initially set at a value of 1.00. The results of these refinements are shown in Table I and clearly indicate the correct assignments.

The final positional and thermal parameters for the non-hydrogen atoms are listed in Table II. The standard deviations were calculated from the inverse matrix of the final least squares refinement cycle. The calculated hydrogen atom positions are listed in Table III. Bond

Table I. Determination of the bonding mode of cyanide in <u>trans</u>-$(CN)_2$ trien $Co^{III}ClO_4$

Atom	Assigned Scattering Factor Table	Atom Multiplier Actual	Theoretical Co-C-N	Theoretical Co-N-C	R	ωR
C(7)	Carbon	1.033	1.0	1.17		
C(8)	Carbon	1.034	1.0	1.17		
N(5)	Nitrogen	1.026	1.0	0.86		
N(6)	Nitrogen	1.046	1.0	0.86	12.4	15.8
C(7)	Nitrogen	0.823	0.86	1.0		
C(8)	Nitrogen	0.825	0.86	1.0		
N(5)	Carbon	1.293	1.17	1.0		
N(6)	Carbon	1.314	1.17	1.0	12.4	15.8

Table II

Final positional and anisotropic thermal parameters of non-hydrogen atoms in trans-$(NC)_2$-trien $Co^{III}ClO_4$ [a,b]

Atom	x	y	z	$10^4\beta_{11}$	$10^4\beta_{22}$	$10^4\beta_{33}$	$10^4\beta_{12}$	$10^4\beta_{13}$	$10^4\beta_{23}$
Co	0.6629(1)	0.3488(1)	0.6552(2)	55.9(1.4)	9.2(0.2)	169.9(3.6)	1.1(0.5)	13.9(1.6)	0.7(0.9)
Cl	0.6616(3)	0.1289(1)	0.6466(6)	79.7(3.0)	15.6(0.6)	306.3(9.7)	3.6(1.1)	12.7(4.3)	6.1(2.2)
O(1)	0.6601(9)	0.0863(4)	0.4868(15)	139(12)	21(2)	360(32)	10(4)	-26(16)	11(7)
O(2)	0.5498(9)	0.1700(4)	0.5928(16)	121(12)	30(2)	486(39)	32(4)	-79(17)	43(8)
O(3)	0.7893(8)	0.1635(4)	0.6748(15)	92(10)	26(2)	386(32)	-5(4)	-7(14)	13(7)
O(4)	0.6586(11)	0.0988(5)	0.8283(16)	204(17)	34(3)	333(33)	-.2(5.6)	90(19)	-3(9)
N(1)	0.4727(8)	0.3159(4)	0.5898(14)	69(9)	14(2)	198(24)	-9(3)	2(12)	10(6)
N(2)	0.5927(8)	0.4122(3)	0.8075(12)	77(9)	11(2)	148(21)	3(3)	11(11)	4(5)
N(3)	0.8378(8)	0.3891(4)	0.7368(15)	66(9)	14(2)	224(26)	-4(3)	23(13)	-9(6)
N(4)	0.7579(9)	0.2898(4)	0.5051(13)	97(11)	14(2)	197(25)	8(4)	39(13)	7(6)
C(7)	0.6226(10)	0.3942(5)	0.4078(18)	68(11)	12(2)	205(30)	3(4)	17(15)	-.8(6.7)
C(8)	0.6997(10)	0.3013(4)	0.8952(18)	63(11)	9(2)	250(33)	-.7(3.5)	31(15)	-7(7)
C(1)	0.3914(11)	0.3494(6)	0.7122(20)	77(12)	22(3)	274(36)	-7(5)	15(17)	-.5(9)
C(2)	0.4392(11)	0.4144(5)	0.7393(20)	62(11)	20(3)	313(39)	10(4)	39(17)	15(9)
C(3)	0.6694(12)	0.4679(5)	0.7878(19)	93(13)	12(2)	280(37)	-4(4)	3(17)	-11(8)
C(4)	0.8173(13)	0.4546(5)	0.7531(24)	118(17)	16(3)	388(47)	-7(5)	11(22)	-19(9)
C(5)	0.9274(12)	0.3720(6)	0.6003(21)	81(13)	26(3)	289(40)	-5(5)	42(19)	-15(10)
C(6)	0.9102(13)	0.3046(6)	0.5589(23)	105(16)	26(3)	407(50)	18(6)	96(23)	14(11)
N(5)	0.6027(11)	0.4174(5)	0.2488(19)	116(13)	19(2)	349(38)	3(4)	27(18)	9(8)
N(6)	0.7204(10)	0.2762(4)	1.0540(15)	115(12)	16(2)	230(28)	3(4)	28(15)	5(7)

[a] In this and subsequent tables, numbers in parentheses represent standard deviations in the least significant digits.

[b] Anisotropic thermal parameters are defined by: $T = \exp[-(h^2\beta_{11} + k^2\beta_{22} + \ell^2\beta_{33} + 2hk\beta_{12} + 2k\ell\beta_{23} + 2h\ell\beta_{13})]$.

Table III. Calculated hydrogen atomic positions in trans-$(CN)_2$ trien $Co^{III}ClO_4$

Atom	x	y	z
H(AC1)	0.4041	0.3289	0.8624
H(BC1)	0.2834	0.3479	0.6404
H(AC2)	0.3934	0.4374	0.8519
H(BC2)	0.4113	0.4386	0.5960
H(AC3)	0.6762	0.4947	0.9921
H(BC3)	0.6135	0.4932	0.6569
H(AC4)	0.8303	0.4765	0.6143
H(BC4)	0.8906	0.4724	0.8798
H(AC5)	1.0339	0.3827	0.6673
H(BC5)	0.8990	0.3961	0.4566
H(AC6)	0.9606	0.2912	0.4366
H(BC6)	0.9593	0.2799	0.6960

lengths and bond angles, and significant non-bonded distances are listed in Tables IV and V, respectively, along with their standard deviations.[17] The final values of the observed and calculated structure factors are listed in Table VI.

Shortly after the data had been collected and the crystal removed from the diffractometer, the instrument was found to be slightly misaligned. Thus, the data are somewhat poorer than might be expected. However, since all the stereochemical features of interest were well determined and the electron density difference map was rather featureless, the data were not retaken.

Description of the Structure

The results clearly indicate that the bonding of cyanide to the cobalt is through the carbon and not the nitrogen, and that these groups are _trans_ and not _cis_ to one another, contrary to the predictions of Kuroda and Gentile.[6] Hence, the compound is _trans_-dicyanotriethylenetetramine cobalt(III) perchlorate, and is illustrated in Figure 1.[18] This configuration has been anticipated by Konya, Nishikawa, and Shibita[7] on the basis of spectroscopic examination of similar compounds. The _trans_ configuration is also consistent with an exceedingly sharp band for the C-N stretching frequency at 2145 cm^{-1}, with no evidence of splitting.

Table IV. Selected bond distances and angles in angstroms and degrees for cis-$(CN)_2$ trien $Co^{III}ClO_4$

Cl-O(1)	1.427(9)	C(8)-N(6)	1.180(13)
Cl-O(2)	1.424(8)	N(1)-C(1)	1.462(15)
Cl-O(3)	1.456(9)	N(2)-C(2)	1.488(12)
Cl-O(4)	1.392(10)	N(2)-C(3)	1.476(13)
Co-N(1)	1.978(8)	N(3)-C(4)	1.484(14)
Co-N(2)	1.951(8)	N(3)-C(5)	1.444(15)
Co-N(3)	1.922(8)	N(4)-C(6)	1.506(15)
Co-N(4)	2.001(8)	C(1)-C(2)	1.527(16)
Co-C(7)	1.909(12)	C(3)-C(4)	1.551(17)
Co-C(8)	1.894(12)	C(5)-C(6)	1.536(18)
C(7)-N(5)	1.161(14)		

O(1)-Cl-O(2)	110.7(6)	N(4)-Co-C(8)	91.5(4)
O(1)-Cl-O(3)	109.1(6)	C(7)-Co-C(8)	177.9(4)
O(1)-Cl-O(4)	109.3(6)	Co-N(1)-C(1)	106.7(6)
O(2)-Cl-O(3)	107.1(6)	Co-N(2)-C(2)	108.1(7)
O(2)-Cl-O(4)	112.3(7)	Co-N(2)-C(3)	108.9(7)
O(3)-Cl-O(4)	108.4(6)	Co-N(3)-C(4)	110.8(7)
N(1)-Co-N(2)	88.0(4)	Co-N(3)-C(5)	109.0(7)
N(1)-Co-N(3)	172.5(4)	Co-N(4)-C(6)	106.3(7)
N(1)-Co-N(4)	99.4(4)	Co-C(7)-N(5)	173.7(10)
N(1)-Co-C(7)	89.2(4)	Co-C(8)-N(6)	174.1(9)

Table IV (Continued)

N(1)-Co-C(8)	89.2(4)	N(1)-C(1)-C(2)	111.3(9)
N(2)-Co-N(3)	84.5(4)	N(2)-C(2)-C(1)	105.9(8)
N(2)-Co-N(4)	172.6(4)	N(2)-C(3)-C(4)	111.4(8)
N(2)-Co-C(7)	91.8(4)	N(3)-C(4)-C(3)	110.2(9)
N(2)-Co-C(8)	89.6(4)	N(3)-C(5)-C(6)	108.3(10)
N(3)-Co-N(4)	88.1(4)	N(4)-C(6)-C(5)	109.1(9)
N(3)-Co-C(7)	90.7(4)	C(2)-N(2)-C(3)	116.8(8)
N(3)-Co-C(8)	91.0(4)	C(4)-N(3)-C(5)	114.5(10)
N(4)-Co-C(7)	87.3(4)		

Table V. Significant non-bonded distances in *trans*-$(CN)_2$ trien $Co^{III}ClO_4$[a]

C(7)-H(BC3)	2.870(10)
C(7)-H(AC4)	2.884(10)
C(8)-0(2)	2.694(13)
C(8)-0(3)	3.601(13)
N(4)-0(3)	3.034(12)
O(1)-H(AC5)'	2.34(1)
O(4)-H(BC1)'	2.49(1)

[a] Primed atoms indicate atoms in another molecule.

Table VI. Observed and calculated structure factors

[illegible]

Table VI. (Continued)

[illegible]

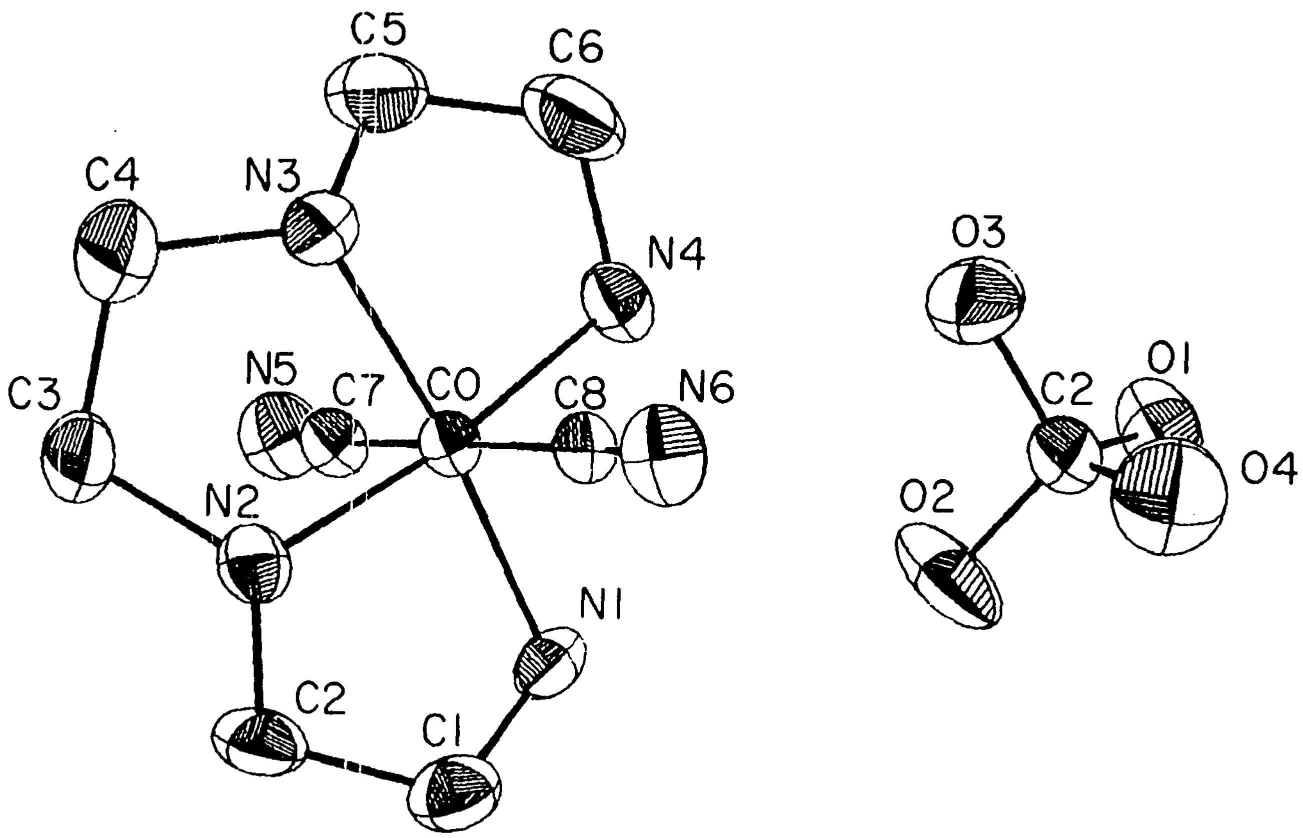

Figure 1. A formula unit of trans-cyano triethylenetetramine cobalt(III) perchlorate

The cyanide groups are within a degree of being perpendicular to the nitrogens of the trien ligand (Table IV), and the variations from an ideal octahedral configuration displayed by the N-Co-N angles appear to simply reflect the restrictions imposed by the quadradentate ligand (Figure 2). This trien ligand coordinated to cobalt in a cis configuration has been investigated by Freemen and Maxwell,[19] and Dwyler and Maxwell;[20] their results have been used as the basis of many of the comparisons which follow.

The N(1)-Co-N(2) and N(3)-Co-N(4) angles of 88.0 and 88.1° are reasonable and consistent with similar results obtained in other investigations of the trien ligand. A slight compression of the Co-N(2)-C(3)-C(4)-N(3) ring due to the equatorial coordination of the ligand adequately explains the somewhat small value of 84.5° for the N(2)-Co-N(3) angle. Hydrogen-hydrogen repulsions would be expected to produce an increase in the N(1)-Co-N(4) angle compared to the nominal 90°, and the 99° angle found is quite reasonable. A similar result was obtained during the investigation of a cobalt compound containing a homologous ligand.[21] The bond angles within the trien ligand, with the exception of the C-N-C angles, do not differ from the expected tetrahedral angles by more than three standard deviations. In addition, they fall within two standard deviations of the range of values of 108.4°

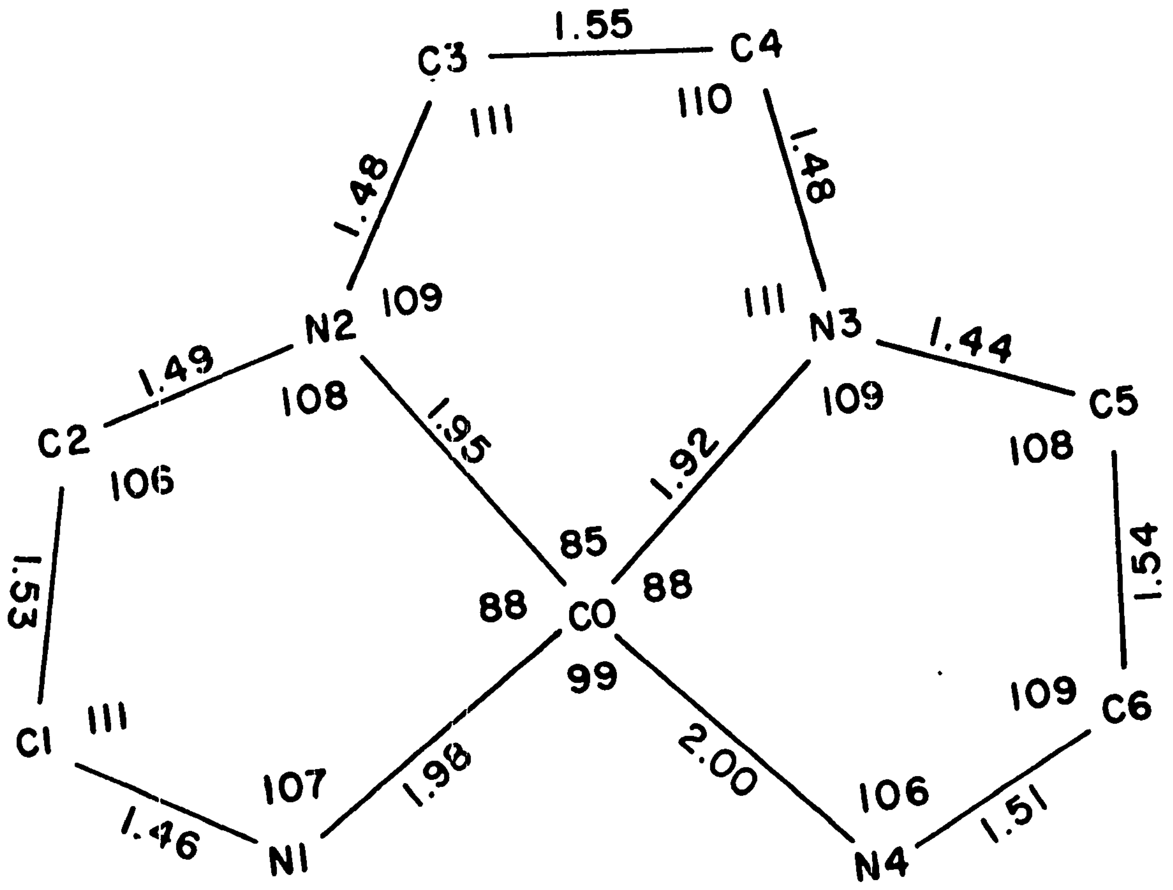

Figure 2. Bond distances and angles within the trien ligand

to 112.3° for Co-N-C bonds, and of 105.2° to 111.4° for N-C-C bonds reported by other investigators of this ligand. The C-N-C bond angles of 116.8° and 114.6° are also reasonable in view of the fact that the Co-N-C angles are, on the average, somewhat less than tetrahedral. These C-N-C angles also agree within three standard deviations with other results obtained in investigations of this ligand.

The bond distances within the trien ligand range from 1.92 to 2.00 Å for Co-N bonds, from 1.44 to 1.51 Å for N-C bonds, and from 1.53 to 1.55 Å for C-C bonds, with average values of 1.96, 1.48, and 1.54 Å, respectively. These distances in no instance differ significantly from previously reported values for other compounds involving the trien ligand coordinated to cobalt.

The four trien nitrogens and the cobalt atom all fall within 0.01 Å of the least squares plane. The three five-membered rings formed by the quadradentate ligand and the cobalt atom are not planar, however, as the carbon atoms are located above and below the N-Co-N planes. Nonetheless, all of the atoms in the trien ligand fall within 0.37 Å of a least squares plane.

As has been described above, the coordination of the cyanide groups to the cobalt is clearly through the carbon. The Co-C average bond distance of 1.90 Å is in good agreement with previously published values. The average C-N distance of 1.17 Å also agrees quite well with the range of previously reported values, 1.15 to 1.18 Å. [2,22-24] The fact that no part of the chain N(6)-C(8)-Co-C(7)-N(5) is linear, with bond angles of N(6)-C(8)-Co, C(8)-Co-C(7), and Co-C(7)-N(5) of 174.1, 177.9, and 173.7°, respectively, is somewhat unusual but not surprising. Curry and Runciman[2] have already observed such a phenomenon. Much of this can be attributed to steric effects both inside the moiety and between moieties (Figure 3). Significant non-bonded distances are given in Table V.

The Cl-O bond distances in the perchlorate group average 1.42 Å and the angles average 109.5°. The relatively large thermal parameters, possibly due to spatial as well as temporal disorder probably account for the large individual deviations from the average values. However, these average values agree reasonably well with previously published values for this anion.[21] A weak hydrogen bond is possible between N(4) and O(3), which are 3.03 Å apart. Such a hydrogen bond would help stabilize the perchlorate group.

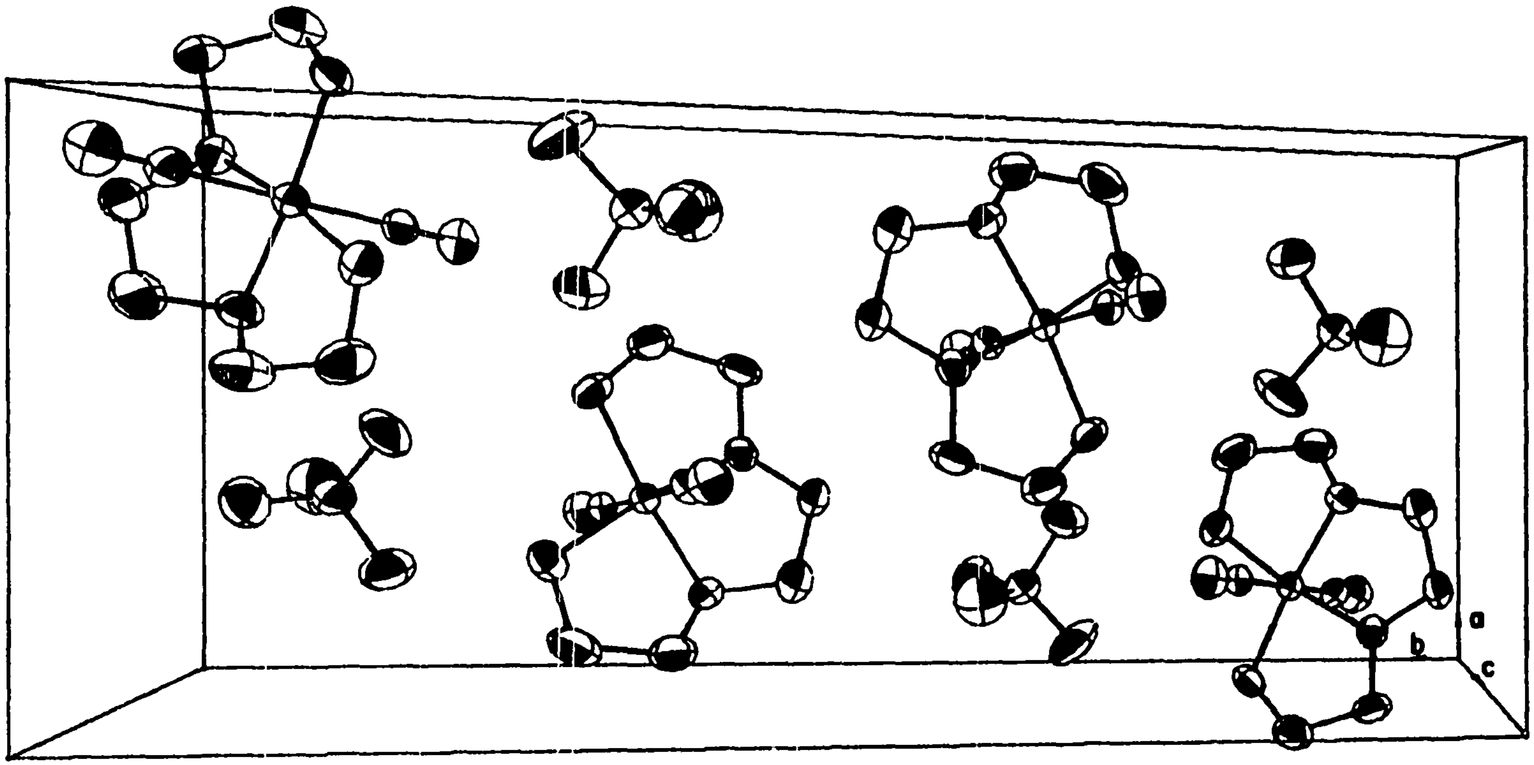

Figure 3. Unit cell showing packing of trans-cyano triethylenetetramine cobalt(III) perchlorate

THE STRUCTURE OF POTASSIUM ANTIMONY(III) PENTACHLORIDE

Introduction

Structural investigations of antimony halide complexes have been actively pursued in this laboratory during recent years, beginning with the work of Lawton and Jacobson.[25] The structures of species containing antimony(III) are of particular interest because of questions concerning the stereochemical role of the lone pair of electrons.

Only a very few antimony(III) chloride structures have been reported in the literature. $SbCl_3$ exhibits a trigonal pyramid structure with antimony at the apex,[26] while in $(NH_4)_2SbCl_5$ the chlorine atoms are arranged in a square pyramidal coordination geometry around the antimony.[27] In $[C_5H_5NH][Sb^{III}Cl_4]$, the halogen configuration around the antimony can be described as tetragonally distorted octahedral with two of the equitorial atoms removed,[28] and $SbCl_6^{3-}$ forms perfect octahedra in $Co(NH_3)_6SbCl_6$.[29]

Because of the relatively poor determination (R = 0.24) of the structure of $(NH_4)_2SbCl_5$ and because of the seeming variety of structural geometries possible for antimony(III), it was decided to undertake a structure determination of K_2SbCl_5.

Experimental

A sample of K_2SbCl_5 was kindly supplied by Dr. Donald Macalady, who prepared the compound in the following manner: Two solutions, one of $SbCl_3$ in 3N HCl and one of KCl in 3N HCl, were combined so that the molar ratio of potassium to antimony in the resulting solution was 5:7. This solution was allowed to stand, and crystals formed by slow evaporation. Microscopic examination revealed many different crystal morphologies. Thin plates appeared to be the basic unit of all these morphologies, however.

A crystal was cut from one of these plates and mounted on the end of a glass fiber with Duco cement thinned with amyl acetate. Preliminary Weissenberg and precession photographs exhibited 2/m Laue symmetry, indicating a monoclinic space group. The following systematic absences were observed: $h0\ell$ when $\ell = 2n + 1$, and $0k0$ when $k = 2n + 1$. These absences are only consistent with the space group $P2_1/c$. The unit cell parameters at 25°C are $\underline{a} = 8.8686 \pm 0.0007$, $\underline{b} = 12.4577 \pm 0.0013$, $\underline{c} = 8.9280 \pm 0.0013$ Å and $\beta = 110.512 \pm 0.011°$. These parameters and their standard deviations were obtained by a least squares fit[8] to the 2θ values of sixteen independent reflections whose centers were determined by left-right, top-bottom beam splitting on a previously aligned four-circle diffractometer (MoKα radiation, $\lambda = 0.71069$ Å). A calculated density of 2.711 g/cc for four molecules per unit cell agrees quite well with an observed density of 2.72 ± 0.01 g/cc, determined by flotation techniques.

For data collection the crystal described above, measuring approximately 0.2 mm along each of the crystal axes, was mounted so that the $\underline{c}$ axis coincided with the ϕ axis of the diffractometer. Data were collected at room temperature using an automated four-circle diffractometer designed and built in the Ames Laboratory. The upper full circle was purchased from STOE and is equipped with encoders (Baldwin Optical) and drive motors. The design of the base allows the encoders to be directly connected to the main θ and 2θ shafts, using solid and hollow shaft encoders, respectively. The diffractometer is interfaced to a PDP-15 computer in a real time mode and is equipped with a scintillation counter. Zirconium-filtered MoKα radiation was used for the data collection. A scan rate of 0.1 second per step of 0.01° in θ was employed with a variable scan range of 35 steps plus 1 step per degree theta. Stationary-crystal, stationary-counter background counts of a quarter of the scan time were taken at the beginning and end of each scan. Before the scan was made a peak height measurement was used to determine if the reflection was observed. To be scanned, the reflection had to exceed the background by more than six counts. If the reflection met this criterion, the ω setting was then adjusted slightly, if necessary, to maximize the peak height. Within a two-theta sphere of 50° ($\sin\theta/\lambda = 0.595$ Å^{-1}), all data in the $hk\ell$ and $\bar{h}k\ell$ octants were measured in this manner, using a take-off angle of 4.5°. Of the 1750 reflections examined, 1614 met the peak height criterion and were scanned.

As a general check on electronic and crystal stability, the intensities of three standard reflections were remeasured every twenty-five reflections. These reflections did not vary to any significant degree during the entire period of data collection.

The intensity data were corrected for Lorentz-polarization effects and for effects due to absorption. An absorption correction was made using the Tompa-Alcock absorption correction program[30,31] using a linear absorption coefficient of $\mu = 52.28\ \text{cm}^{-1}$. The maximum and minimum transmission factors were 43.95% and 35.98%, respectively. The estimated error in each intensity was calculated by

$$\sigma_I^2 = C_T + 2C_B + (0.03\ C_T)^2 + (0.03\ C_B)^2 + (0.03\ C_I)^2\ T_a^{\ 2},$$

where C_T, C_B, C_I, and T_a are the total count, the background count, the net count, and the transmission factor, respectively, and the factor 0.03 represents an estimate of non-statistical errors. The estimated deviations in the structure factors were calculated by the finite difference method.[10] Of the 1614 independent reflections, 1543 were considered observed ($> 2.0\sigma_F$).

Solution and Refinement

The position of the antimony atom was obtained from analysis of a sharpened three-dimensional Patterson function.[11] The remaining atoms were found by successive structure factor[12] and electron density map calculations.[14] These atomic positions were refined by a full matrix least squares procedure,[12] minimizing the function $\Sigma\omega(|F_o|-|F_c|)^2$, where $\omega = 1/\sigma_F^2$, to a conventional discrepancy factor of R = 0.169. The scattering factors used were those of Hanson _et_ _al._,[15] modified for the real and imaginary parts of anomalous dispersion.[16]

An electron density difference map verified that all the atoms had been accounted for, but indicated that some anisotropic motion, particularly of the antimony, was present. Accordingly, anisotropic refinement was begun and after four cycles of refinement, values of R and ωR of 0.054 and 0.078, respectively, were obtained.

A final electron density difference map showed no peaks greater than 0.9 e/Å^3. A final statistical analysis of $\overline{\omega\Lambda^2}$, where Λ = ($|F_o| - |F_c|$), as a function of the scattering angle and magnitude of F_o revealed no unusual trends and suggests that the relative weighting scheme used is a reasonable one.

The final positional and thermal parameters are listed in Table VII. The standard deviations were calculated from the inverse matrix of the final least squares cycle. Bond lengths, bond angles, and significant non-bonded distances[17] are listed in Table VIII. The final values of the observed and calculated structure factors are listed in Table IX.

Description and Discussion

The configuration of $SbCl_5^=$ in K_2SbCl_5 is essentially that of a square pyramid which has been distorted by interionic and packing forces. The inversion-related anions are packed approximately base to base, as depicted in Figure 4.[18] The axial antimony-chlorine bond length is 2.385 Å, in good agreement with the axial distance in $(NH_4)_2SbCl_5$ (2.36 Å),[27] the shorter distance (2.38 Å) in in$[C_5H_5NH][Sb^{III}Cl_4]$,[28] and only slightly longer than the distance (2.32 Å) in $SbCl_3$.[26]

Table VII. Final atomic coordinates and thermal parameters (x 10^4) for K_2SbCl_5[a]

Atom	x	y	z	β_{11}	β_{22}	β_{33}	β_{12}	β_{13}	β_{23}
Sb	0.1836(1)	-0.0068(1)	0.2025(1)	75(1)	42(1)	80(1)	-2(.4)	20(.8)	-3(.4)
Cl(1)	0.2414(3)	0.0959(2)	0.0510(3)	141(4)	55(2)	98(3)	-3(2)	47(3)	3(2)
Cl(2)	-0.0097(3)	0.1519(2)	0.2070(3)	90(3)	51(2)	160(4)	6(2)	47(3)	2(2)
Cl(3)	0.4157(3)	-0.1400(2)	0.2121(3)	143(4)	68(2)	115(4)	35(2)	41(3)	.8(2)
Cl(4)	0.3885(3)	0.1137(2)	0.3563(3)	89(3)	61(2)	99(3)	-17(2)	26(3)	-9(2)
Cl(5)	0.1706(3)	-0.0770(2)	0.4611(3)	118(4)	67(2)	100(3)	-11(2)	40(3)	16(2)
K(1)	0.1853(3)	0.1746(2)	0.5939(3)	135(3)	56(1)	112(3)	2(2)	52(3)	-1(2)
K(2)	0.3742(3)	-0.1285(2)	0.8452(3)	106(3)	94(2)	129(4)	5(2)	43(3)	-24(2)

[a] Anisotropic thermal parameters are defined by:

$$T = \exp -(h^2\beta_{11} + k^2\beta_{22} + \ell^2\beta_{33} + hk\beta_{12} + h\ell\beta_{13} + k\ell\beta_{23}) .$$

Table VIII. Bond lengths, bond angles, and significant non-bonded distances in K_2SbCl_5[a]

Sb-Cl(1)	2.799(2)	Sb-Cl(2)	2.625(2)
Sb-Cl(3)	2.622(2)	Sb-Cl(4)	2.385(2)
Sb-Cl(5)	2.509(2)		
Cl(4)-Sb-Cl(1)	81.92(7)	Cl(4)-Sb-Cl(2)	83.54(8)
Cl(4)-Sb-Cl(3)	87.06(8)	Cl(4)-Sb-Cl(5)	87.77(8)
Cl(1)-Sb-Cl(2)	88.50(7)	Cl(2)-Sb-Cl(5)	90.43(8)
Cl(5)-Sb-Cl(3)	92.19(3)	Cl(3)-Sb-Cl(1)	87.22(8)
Cl(1)-Sb-Cl(5)	169.69(8)	Cl(2)-Sb-Cl(3)	170.13(8)
Sb-Sb'	3.932(1)	Sb-Cl(2)'	3.881(3)
Cl(1)-Sb'	3.701(2)	Cl(1)-Cl(2)'	3.699(3)
Cl(2)-Cl(5)$_{i}$	3.768(3)	Cl(1)-Cl(3)$_{ii}$	3.832(4)
Cl(1)-K(1)$_{iii}$	3.187(3)	Cl(1)-K(2)$_{iii}$	3.290(3)
Cl(2)-K(2)$_{iii}$	3.168(3)	Cl(1)-K(1)$_{iv}$	3.247(3)
Cl(2)-K(1)$_{iv}$	3.147(3)	Cl(3)-K(2)$_{v}$	3.190(4)
Cl(2)-K(1)	3.289(3)	Cl(4)-K(1)	3.317(3)

[a] Primed atoms are those related by the center of inversion, as in Figure 4. Other symmetry operations referred to are: (i) -x, ½+y, -z; (ii) x, y, z-1; (iii) x, ½-y, z-1; (iv) x, -½-y, z-1; (v) 1-x, -y, -z.

Table IX. Observed and calculated structure factors

[illegible]

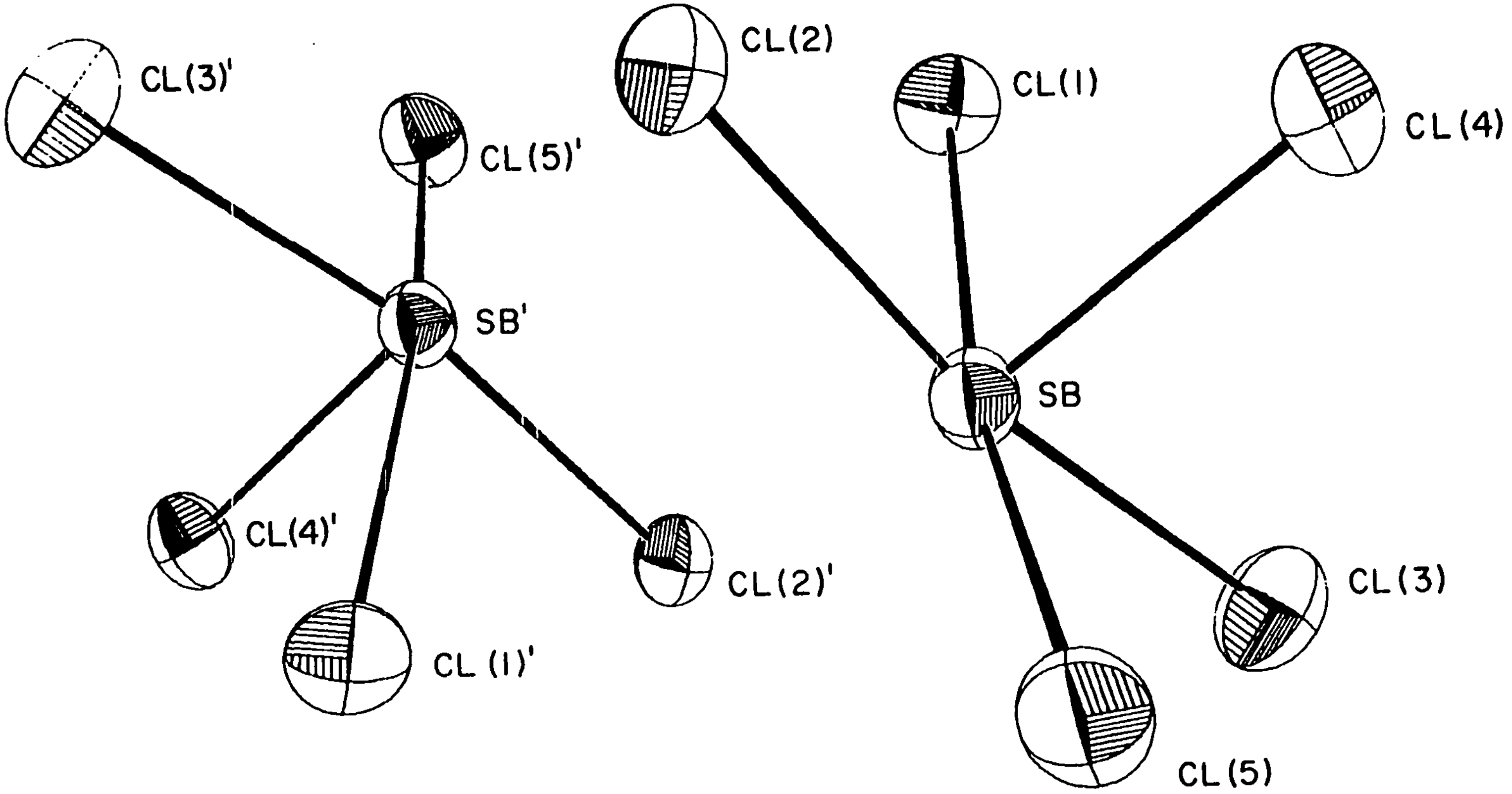

Figure 4. Two inversion-related $SbCl_5^{=}$ anions

The antimony-chlorine bond distances in the base are all considerably longer than the axial distance. Two of these distances are almost identical (2.622, 2.625 Å) and involve chlorines *trans* to each other. The other pair of chlorines form bonds of 2.799 and 2.509 Å length, with an average of 2.654 Å. In $(NH_4)_2SbCl_5$ all basal antimony-chlorine bonds were found to be essentially of equal length (2.62 Å average) while in $[C_5H_5NH][SbCl_4]$ the corresponding bond length for chlorines *trans* to one another is 2.63 Å.

In the present study, the lengthening of the Sb-Cl(1) distance to 2.799 Å appears to be due to electrostatic repulsions and packing effects, as it is this chlorine that is involved in the closest non-bonded contacts; significant non-bonded distances are given in Table VIII. An antimony atom is only 3.701 Å from Cl(1), whereas the sum of the van der Waal's radii is 4.0 Å. Also, the shortest chlorine-chlorine distance found, 3.699 Å, is between this chlorine and Cl(2)'. Both these distances would become significantly shorter if the Sb-Cl(1) bond were shortened. Thus, it can be inferred that this bond can be readily distorted if more efficient packing of the ions results. The shortening of the Sb-Cl(5) bond appears to be a *trans* effect in response to the lengthening of the Sb-Cl(1) bond.

The observed bond lengths of the basal antimony-chlorine bonds described above are consistent with characterizing the antimony bonding orbitals as being composed primarily of p-orbitals as in the three-center four-electron bonding scheme described by Porter and Jacobson.[32] The lone pair appears to have a comparatively small stereochemical effect. This is supported by the relatively short antimony-antimony distance of 3.932 Å, where localized lone pairs, if present, would have to be directed toward one another. Distortions of the Cl-Sb-Cl bond angles from the expected 90° (cf. Table VIII) can be explained by interionic repulsions between inversion-related complexes.

The arrangement of the $SbCl_5^=$ complexes in the unit cell is such that the chlorine atoms approximate a closest packed arrangement in the bc plane, as is evident in Figure 5, with the shortest chlorine-chlorine distances approximating the sum of the van der Waal's radii. None of the potassium ions lie between these planes, but rather are located among the $SbCl_5^=$ ions in the planes. The shortest potassium-chlorine distance is 3.147 Å.

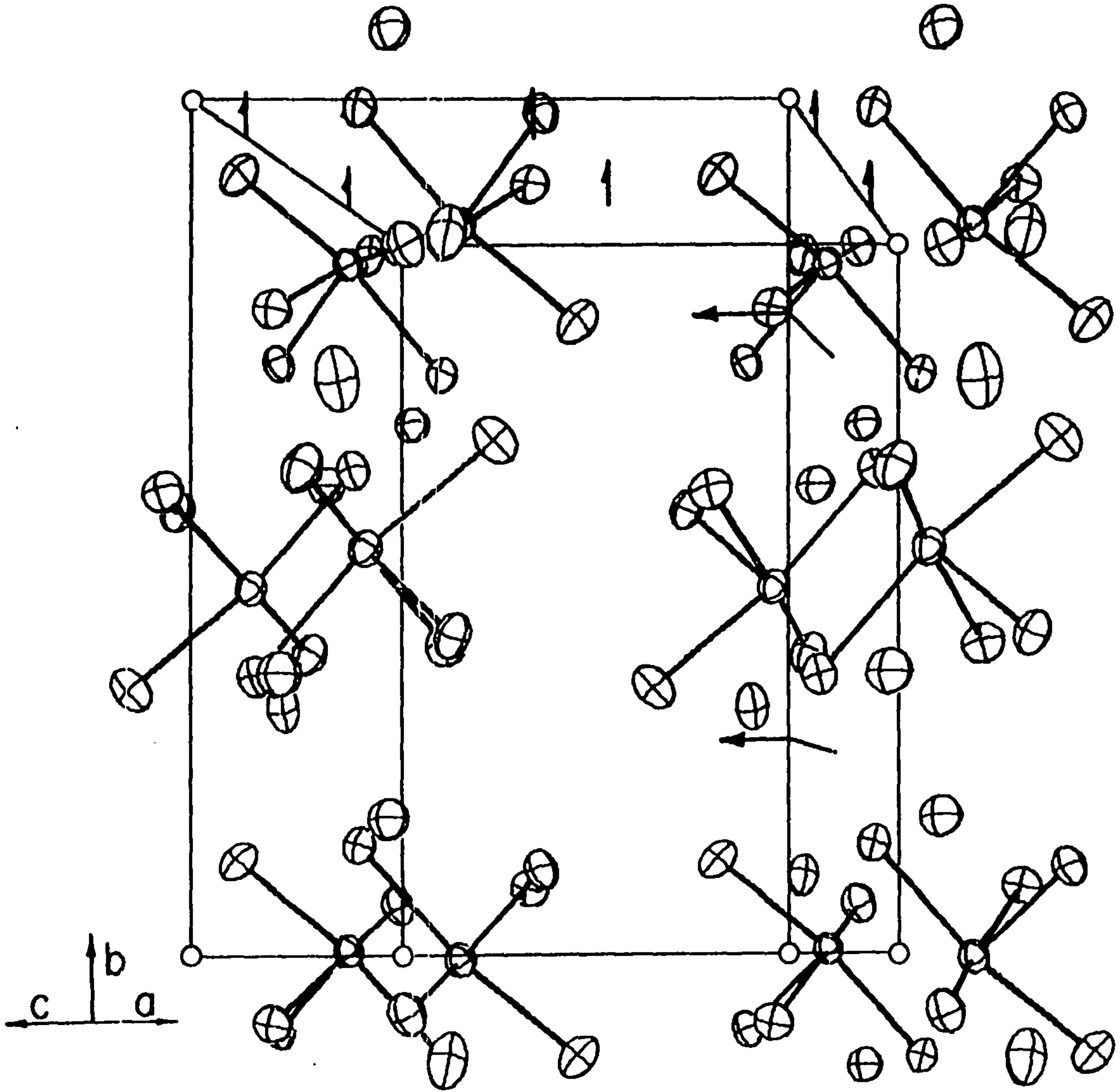

Figure 5. Packing diagram of K_2SbCl_5

ALOP: A PROGRAMMED, SEMI-AUTOMATIC VERSION OF THE PSEUDO HEAVY ATOM TECHNIQUE

Introduction

The increasing availability of high speed digital computers has enabled the crystallographer to make much more frequent use of mathematical techniques which previously were extremely time consuming. This fact has greatly enhanced the development and usage of direct methods, which recently have evolved to such an extent as to permit semi-automatic solution of crystal structures. In contrast, such indirect methods as Patterson superposition techniques have been employed more and more infrequently in recent times. Despite their great power, these latter methods require a great deal of human intervention and are somewhat tedious to perform. Clearly, what is required is a series of programs which would enable one to solve a structure using superposition techniques but requiring a minimum of human intervention.

This series of programs probably would have greatest applicability to, and thus should be specifically designed to solve, the pseudo heavy atom problem, in which the "heavy" atom does not contribute enough to the phasing to make heavy atom methods feasible. This problem occurs with compounds in which there are many light atoms and but few heavy atoms, such as transition metal complexes or heavy atom derivatives of organic compounds.

In what follows, the essential features of this series of programs are detailed. The techniques employed naturally divide into two categories: (1) preparation of a Patterson map suitable for evaluation, and (2) utilization of this map to solve the structure.

Map Preparation

A sharpened Patterson map[11] is created on either tape or disk by the program ALFF[14] or any other similar Fourier series program. However, in this form the map is somewhat cumbersome to analyze. Hence, the program ALPP (Ames Laboratory Peak Picker) finds and outputs the heights and positions of all the peaks above a certain limit on the Patterson map. ALPP is divided into four subprocedures, the function of which is described below, and a main procedure, which controls the order of execution of the subprocedures. The program listing for ALPP is given in Appendix A.

Input to the first subprocedure, SETUP, consists of the size of the map and the minimum height of a peak which should be recognized. This subprocedure then reads and rewrites the map by layers, setting all points less than the minimum limit equal to zero.

The second subprocedure, LAYERS, reads these map layers and outputs layers with all non-peak grid points negated. The essence of this subprocedure is the subroutine PICKER, which is flowcharted in Figure 6. PICKER analyzes a line of the map point by point. The absolute value of each point is compared with that of the point before it. The lesser of the two is negated. LAYERS calls PICKER for each row and each column of the layer, and then compares points located diagonally to each point, negating those with smaller absolute values.

The next subprocedure, NTRLAY, compares the absolute value of a peak with those of the nine peaks adjacent to it in the next layer, again negating the smaller values. The only positive grid points which now remain are those of peaks. Next, these peak locations and heights are output and the modified map is printed, if desired for debugging purposes.

The final subprocedure, PKSORT, is a modified PL/I translation of QSTSRT. PKSORT sorts the peaks in order of decreasing height and outputs the ordered peaks.

The entire program runs in less core and requires somewhat less execution time than does ALFF. Thus, it is feasible to run the two programs back to back as one job.

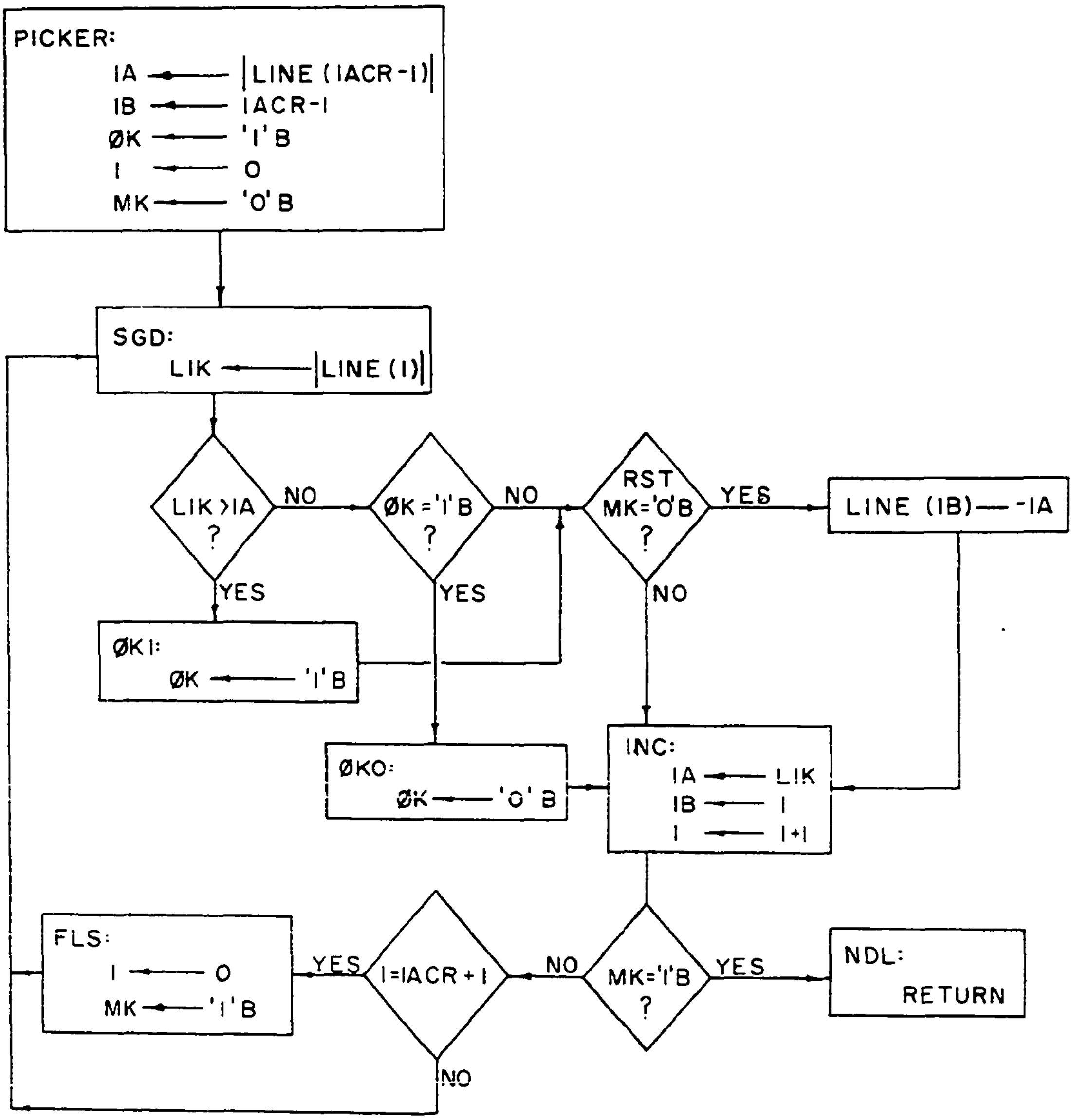

Figure 6. Flowchart of subroutine PICKER

Solution of the Structure

The main program of the series is ALOP (Ames Laboratory Origin Picker), and the program listing is given in Appendix B. ALOP is divided into a main procedure and seven subprocedures.

The main procedure reads in the size of the map, the number of peaks produced by ALPP, and the symmetry operations of the space group under consideration. These are in the form: $p_x x + q_x$, $p_y y + q_y$, $p_z z + q_z$, where p_x, p_y, $p_z = 1$ or -1 and q_x, q_y, $q_z = 0$ or ½. Thus one can represent an equivalent position produced by: an inversion, a two-fold axis, a two-fold screw axis, a mirror plane in one of the three crystal directions or an _a_-, _b_-, or _c_-glide plane. Any of the triclinic, monoclinic, or orthorhombic space groups can be represented by these five symmetry operations (except Fdd2 or Fddd). In addition, any of the other 230 crystallographic space groups can be converted into a space group which only uses these operations, although some loss of symmetry generally occurs.

The first subprocedure, PART1, searches the peaks produced by ALPP for a suitable heavy atom peak. This peak must be among the largest 20% of the peaks. In addition, if any of its coordinates, u, v, w, is equal to 0 or ½ of the unit cell, then it must be at least twice as large as the largest peak which has no one of u, v, w equal to 0 or ½. These criteria assure that the heavy atom peak chosen is not solely a Harker vector, but allow the choice of a heavy atom which has 0 or ½ as one of its three coordinates, such as the antimony atom in K_2SbCl_5.

When PART1 finds a suitable heavy atom peak, it prints its coordinates, u_H, v_H, w_H, and uses it as the shift vector of an in-core algebraic superposition. For a peak u, v, w, to be retained, there must also be a peak at $u' = u + u_H$, $v' = v + v_H$, $w' = w + w_H$ in the list of peaks produced by ALPP. In practice, since ALPP only determines the maximum grid point, and not the center, of each peak, these requirements are somewhat too stringent and thus u', v', w' is permitted to be off by one grid point in each direction. The values of u', v', w' thus found are output for subsequent use.

PART2 assumes that the peaks produced by PART1 represent an electron density map of the compound, with its origin shifted to t_x, t_y, t_z, *i.e.*, the following relations hold: $u_i = x_i + t_x$, $v_i = y_i + t_y$, $w_i = z_i + t_z$, where x_i, y_i, z_i are the coordinates of the peak in electron density space. When the symmetry operations defined above are applied to a peak at x, y, z they produce an equivalent peak at $p_x x + q_x$, $p_y y + q_y$, $p_z z + q_z$. Thus for every peak at $u = x + t_x$, $v = y + t_y$, $w = z + t_z$ there is an equivalent peak at $u' = p_x x + t_x + q_x$, $v' = p_y y + t_y$

$+ q_y$, $w' = p_z z + t_z + q_z$. In Table X the results of subtracting and adding u, v, w and u', v', w' are given for all permutations of p_x, p_y, p_z. The asterisked (*) sums determine values of t_x, t_y, and t_z. PART2 and PART3 accumulate all possible values of t_x, t_y, and t_z for the symmetry operations, as shown in Table XI. PART3 and PART4 write these values, along with the frequency of occurrence of each, on file SORTIN for future use. If any one of t_x, t_y, t_z is not determined, as is the case with all but inversion symmetry, it is set equal to -1. In addition to writing out the exact values of t_x, t_y, and t_z, those which are as much as one grid point off in each direction are also written out. Tables X and XI also demonstrate that values of t_x, t_y, t_z determined by two-folds, screws, mirrors, and glides have to meet criteria before they are acceptable, whereas inversions do not. Therefore, a greater weight is placed on these former values.

At this point, the following items may be present on file SORTIN: t_x, t_y, t_z triples; t_x, t_y; t_y, t_z; or t_x, t_z doubles; or t_x, t_y, or t_z singles. These are sorted by the IBM procedure IHESRTA within their different types in order of decreasing frequency of occurrence. PART5 reads and stores 1/64 of the triples, 1/16 of the doubles, and 1/4 of the singles, all of which have the largest frequencies. These singles, doubles, and triples

Table X. Relationships used in determining t_x, t_y, t_z

Quantity	u	v	w
U	$A = x + t_x$	$B = y + t_y$	$C = z + t_z$
U_0	$A + q_x$	$B + q_y$	$C + q_z$
$U_0 + U$	$2A + q_x$	$2B + q_y$	$2C + q_z$
$U_0 - U$	q_x	q_y	q_z
U_1	$-x + t_x + q_x$	$B + q_y$	$C + q_z$
$U_1 + U$	* $2t_x + q_x$	$2B + q_y$	$2C + q_z$
$U_1 - U$	$-2x + q_x$	q_y	q_z
U_2	$A + q_x$	$-y + t_y + q_y$	$C + q_z$
$U_2 + U$	$2A + q_x$	* $2t_y + q_y$	$2C + q_z$
$U_2 - U$	q_x	$-2y + q_y$	q_z
U_3	$-x + t_x + q_x$	$-y + t_y + q_y$	$C + q_z$
$U_3 + U$	* $2t_x + q_x$	* $2t_y + q_y$	$2C + q_z$
$U_3 - U$	$-2x + q_x$	$-2y + q_y$	q_z
U_4	$A + q_x$	$B + q_y$	$-z + t_z + q_z$
$U_4 + U$	$2A + q_x$	$2B + q_y$	* $2t_z + q_z$
$U_4 - U$	q_x	q_y	$-2z + q_z$
U_5	$-x + t_x + q_x$	$B + q_y$	$-z + t_z + q_z$
$U_5 + U$	* $2t_x + q_x$	$2B + q_y$	* $2t_z + q_z$
$U_5 - U$	$-2x + q_x$	q_y	$-2z + q_z$

Table X. (Continued)

Quantity	u	v	w
U_6	$A + q_x$	$-y + t_y + q_y$	$-z + t_z + q_z$
$U_6 + U$	$2A + q_x$	* $2t_y + q_y$	* $2t_z + q_z$
$U_6 - U$	q_x	$2y + q_y$	$2z + q_z$
U_7	$-x + t_x + q_x$	$-y + t_y + q_y$	$-z + t_z + q_z$
$U_7 + U$	* $2t_x + q_x$	* $2t_y + q_y$	* $2t_z + q_z$
$U_7 - U$	$2x + q_x$	$2y + q_y$	$2z + t_z$

Table XI. Restrictions placed on the determination of t_x, t_y, t_z by the type of symmetry element

Type of symmetry element	Determines	Restrictions
Plane perpendicular to x	t_x	$v_1 - v = q_y$, $w_1 - w = q_z$
Plane perpendicular to y	t_y	$u_2 - u = q_x$, $w_2 - w = q_z$
Plane perpendicular to z	t_z	$u_4 - u = q_x$, $v_4 - v = q_y$
Axis in x	t_y, t_z	$u_6 - u = q_x$
Axis in y	t_x, t_z	$v_5 - v = q_y$
Axis in z	t_x, t_y	$w_3 - w = q_z$
Inversion	t_x, t_y, t_z	None

are then merged with each other as described in Table XII, with a latitude of one grid point allowed in each direction. Thus, if a single in t_x equals the t_x of a t_x, t_y double or is off by one grid point, then the frequency of the t_x, t_y double is increased by the frequency of the single. A record is kept of each such merge.

This record is used in PART6, where all of the possible mergings for each symmetry element present are computed. Consider, as an example, the space group $P2_1/c$ which has an inversion producing a triple t_x, t_y, t_z, a two-fold screw axis producing a double in t_x, t_z and a c-glide producing a single in t_y. PART5 should have merged a single in t_y and a double in t_x, t_z with each triple and left a record of having done so. The values of the triples which were not suitably merged are eliminated, and the triple remaining with the highest frequency of occurrence is then output for use in PART7.

Sometimes, however, a space group may not have an inversion. If it has three two-fold or screw axes, these will produce three doubles: t_{x1} t_{y1}, t_{x2} t_{z2}, t_{y3} t_{z3}. In the cases where $t_{x1} = t_{x2}$, $t_{y1} = t_{y3}$, and $t_{z2} = t_{z3}$, a triple is created with a frequency equal to the sum of the frequencies of the three doubles. The highest such triple is then taken as the origin. If any one of t_x, t_y, or t_z is not determined by the above procedures, it

Table XII. Merging of singles and doubles into doubles and triples

Single or double	Double or triple merged into
t_x	t_x t_y, t_x t_z, t_x t_y t_z
t_y	t_x t_y, t_y t_z, t_x t_y t_z
t_z	t_x t_z, t_y t_z, t_x t_y t_z
t_x t_y	t_x t_z, t_y t_z, t_x t_y t_z
t_x t_z	t_x t_y, t_y t_z, t_x t_y t_z
t_y t_z	t_x t_y, t_x t_z, t_x t_y t_z

is set to zero, as the origin may have any value in that direction.

PART7 computes the electron density space coordinates of all the peaks found in PART1 by using the equations: $x_i = u_i - t_x$, $y_i = v_i - t_y$, $z_i = w_i - t_z$. For each peak, this procedure then attempts to find the symmetry equivalent peaks. If at least half of the equivalent peaks are present, a peak is output with the appropriate fractional coordinates.

ALOP uses less than 32K words of main core storage and runs in less than five minutes of CPU time for a 32 x 32 x 32 map with less than 500 peaks input from ALPP.

Results

The programs ALPP and ALOP were tried on two different problems and the technique was successful in both cases. In the first case, the structure of D-(1,5)-glucono lactone (space group $P2_12_12_1$)[33] was "chlorinated" by substituting chlorines for two of the hydrogens, yielding a molecule with two chlorines, six oxygens, and six carbons. The heavy atom ratio for this compound was $\Sigma Z^2_{heavy} / \Sigma (Z_{light})^2 = 0.963$. All of the non-hydrogen atoms were used in a structure factor calculation and a sharpened Patterson map was computed from these structure factors. ALPP found 502 off-origin peaks in this map with heights ranging from 13 to 234. A heavy atom peak, which proved to be an oxygen-chlorine vector, was found at u = 30, v = 26, w = 3 with a height of 99 on the 32 x 32 x 32 map. In-core superposition retained 187 peaks with a minimum height of 23. This is a good demonstration of how the superposition process tends to eliminate spurious peaks. Multiple merging produced eleven possible origins, of which the one with the highest frequency of occurrence was $t_x = 14$, $t_y = 13$, $t_z = 12$. When this value was used as an origin, 33 symmetry independent peaks were retained. Subsequent least square refinement of the positional parameters and atom multipliers of these 33 positions through twelve least squares cycles resulted in a conventional discrepancy factor of R = 0.206. The scattering factor curve for this and the subsequent test case is defined by:

$$\hat{f}_i = \sum_{j=1}^{N} f_{ij} \Big/ \sum_{j=1}^{N} Z_j,$$

where f_{ij} is the normal scattering factor for the jth atom over a range of $\sin\theta/\lambda$,[15] Z_j is the atomic number of the jth atom, and the sum is over all the atoms in the unit cell.

The second test case was that of K_2SbCl_5 which crystallizes in space group $P2_1/c$ with four eight-atom moieties per unit cell. The heavy atom ratio for this compound is 1.200. ALPP found 258 off-origin peaks on the sharpened

Patterson map, ranging in height from 17 to 508. A heavy atom peak, which was produced by an antimony-antimony vector, was found at u = 20, v = 16, w = 3 with a height of 282 on the 32 x 32 x 32 map. In-core superposition retained 112 peaks. The origin with the highest frequency of occurrence was at $t_x = 10$, $t_y = 0$, $t_z = 9$, and its use yielded 20 symmetry independent peaks. Sixteen cycles of full matrix least squares refinement of the positional parameters and atom multipliers of these positions led to the removal of eleven of them as their atom multipliers dropped to less than 0.40. The remaining nine positions refined to yield a conventional discrepancy factor of R = 0.240.

PSST: A COMBINATION OF PATTERSON SUPERPOSITION AND SIGMA-2 TECHNIQUES FOR PHASE DETERMINATION

Introduction

At present there are two methods commonly used for solving structures of moderate complexity. Direct methods, such as those using the Σ_2 relation,[34] are primarily reciprocal space methods, while methods based on the deconvolution of the Patterson function (for a detailed bibliography see Buerger[35]) are primarily real space methods.

There is a certain parallelism between these two methods and both have their advantages and disadvantages. Both involve some initial choices which can greatly influence the success of the method. The Σ_2 relation can be readily programmed and many automatic or semi-automatic procedures have been developed based on this relation. However, it is usually difficult to make use of known structural information in this approach. Furthermore, if no reasonable structure is produced, the investigator has little recourse but to modify his choice of origin-determining reflections or to closely examine those phases selected at the early and most critial part of this phase determination procedure. With Patterson methods, on the other hand, there are usually a greater number of options initially available corresponding to differing modes of selection of peaks for superposition. Also, known structural information can be more readily introduced, since a real space representation is being employed. For complex structures, however, large numbers of superpositions are generally required and severe degradation of the structural image can result due to the accumulation of errors in the atom position selections.

The advent of the fast Fourier transform technique and its application to crystallography[36] allows rapid real space - reciprocal space conversions. Thus, it is appropriate to consider whether a hybrid direct method - Patterson method approach having greater power than either method alone can be developed. It is just such a hybrid that is described here.

Theoretical Basis of the Method

In direct methods employing the Σ_2 relation, one starts with E's of large magnitude so that phases will be determined with reasonably high probabilities. Due to the fact that the initial contributors are few in number, erroneous phase indications may still result which then propagate in the phase determination procedure. Consider the map produced by a Fourier transform using as non-zero coefficients only those reflections input into the Σ_2 relation (henceforth this map will be referred to as the Σ_2 map). For the phase determination procedure to be successful, it is necessary that the square of this Σ_2-map better resemble the true electron density function than does the Σ_2-map itself. If so, the phases of the transformed coefficients with larger magnitudes should be good approximations to the phases of the corresponding large $|E|$'s.

Since the initial set of E's is very small in number, possibly only the three origin-determining reflections, it is the characteristics of the Σ_2-map which should be closely examined. First, if large $|E|$'s are used, there is a high probability that atoms will lie on, or at least near, some of the maxima occurring on this map. Second, there will be a large number of extraneous maxima on the square of the Σ_2-map -- a number far exceeding the number of atoms in the cell. The first is an asset and the second a liability for successful phase determination. These extraneous peaks come from either maxima or minima on the Σ_2-map. It is the presence of these extra peaks that greatly inhibits the production of additional phase information.

One would like to modify the square of the Σ_2-map so that it better resembles the electron density function. An obvious such modification is to eliminate all negative regions (or possibly regions less than a certain lower bound) by setting these regions equal to zero before squaring. Indeed, this technique has been applied by Barrett and Zwick[37] to some extent in the extension and refinement of crystallographic protein phases.

In the initial application of the Σ_2 relation, there are so few phased E's that even with the modification described above, there are large positive regions in many areas of the map which do not correspond to atomic locations. However, the Patterson function contains images of the structure with each atom in turn at the origin. Therefore, if an atomic position can be selected on the Σ_2-map, a superposition of the appropriately scaled Patterson map on this position should produce a map with considerably fewer extraneous peaks and with much better resemblance to the electron density function. If such a map is then squared to emphasize the higher peaks and transformed, better and more extensive phase information will result. Once a sufficiently large set of phased E's has been obtained, the remaining $|E|$'s can be efficiently phased by normal application of phase extension and refinement techniques of the Σ_2 type.

Application of the Method

Centric case

The compound β-picoline-N-oxide fumaric acid adduct[38] was chosen as a centrosymmetric test case. We felt this would be a good test case as the structure had previously been solved only with difficulty by use of a roving model --

vector verification technique after usual superposition and symbolic addition procedures had not proved successful.

This adduct crystallizes in the space group $P2_1/c$ with $\underline{a}$ = 3.888, $\underline{b}$ = 14.194, $\underline{c}$ = 14.666 Å and β = 98.85°. There are two adducts of formula $C_{16}H_{18}N_2O_{12}$ per unit cell, i.e., the adduct has a center of symmetry. Values of $|E|$ were computed in the ordinary way and there were 270 $|E|$'s greater than 1.0. The three selected origin-determining reflections are given in Table XIII. These three, along with the symmetry-related reflections, were used in a Σ_2-map calculation (32 x 32 x 32 grid points). Some typical sections of this map are shown in Figure 7.

The positions of the five highest peaks on this map were noted and in each case checked by examining the Patterson for the presence of vectors between symmetry-related atoms. In each case, such vectors were found and indeed these peaks fell approximately (within 0.5 Å) at the positions of four carbons and a nitrogen in the structure.

All of the negative regions in the Σ_2-map were then set equal to zero and five independent superpositions were carried out by placing the origin of the Patterson at each of the five peak positions and the symmetry-related points. A sharpened Patterson function was used and the minimum procedure employed.[39] The scale of the Patterson was

Table XIII. Origin-determining reflections for the centric test case (β-picoline-N-oxide fumaric acid adduct)

h	k	l	|E|	Sign
1	0	2	4.17	+
3	1	3	2.53	-
1	14	-5	2.14	+

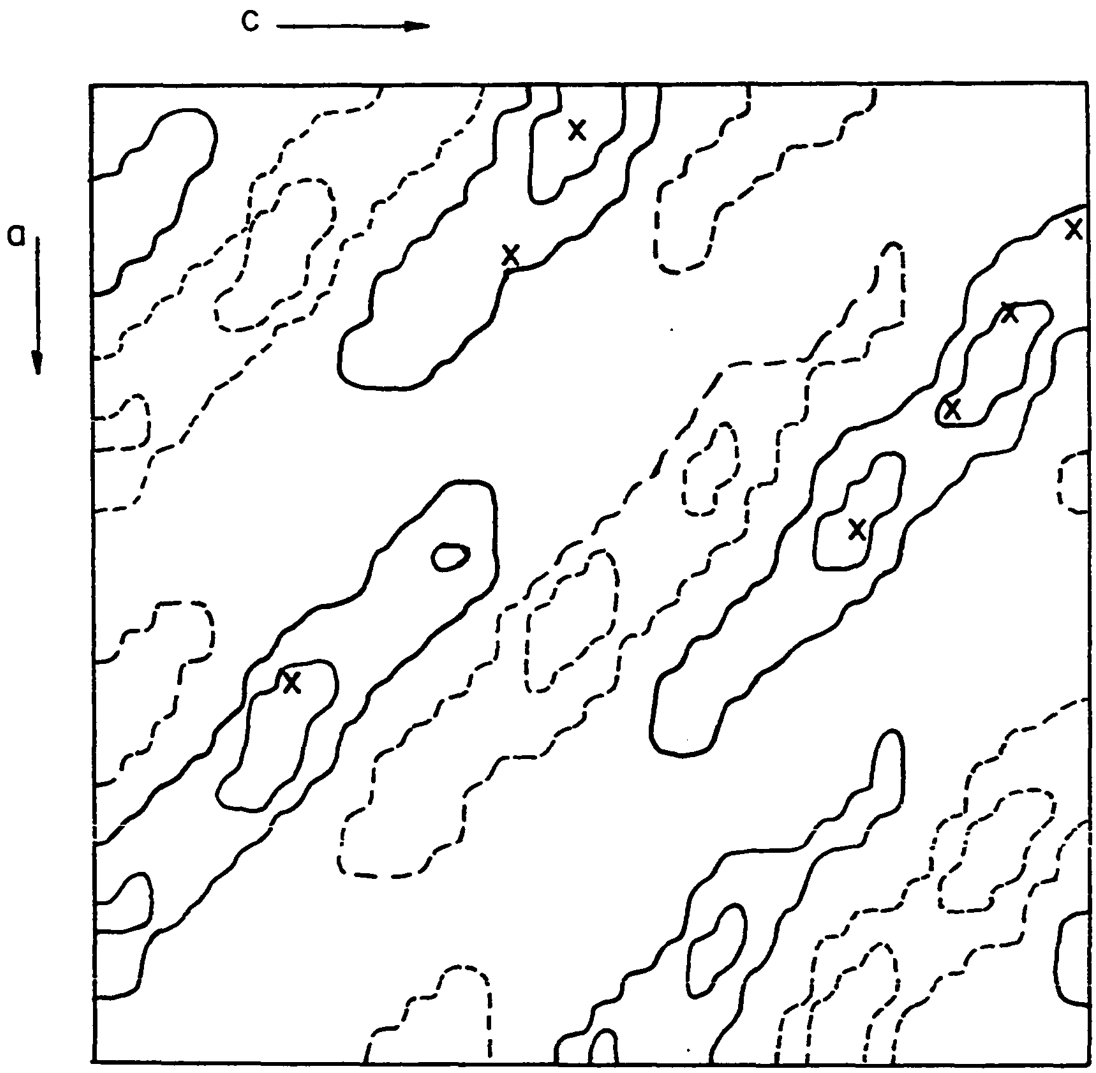

Figure 7a. Composite of layers 3,4,5 of a Σ_2-map for β-picoline-N-oxide fumaric acid adduct. —— contoured at 40 and 80. ---- contoured at -40 and -80. Atom positions indicated by X.

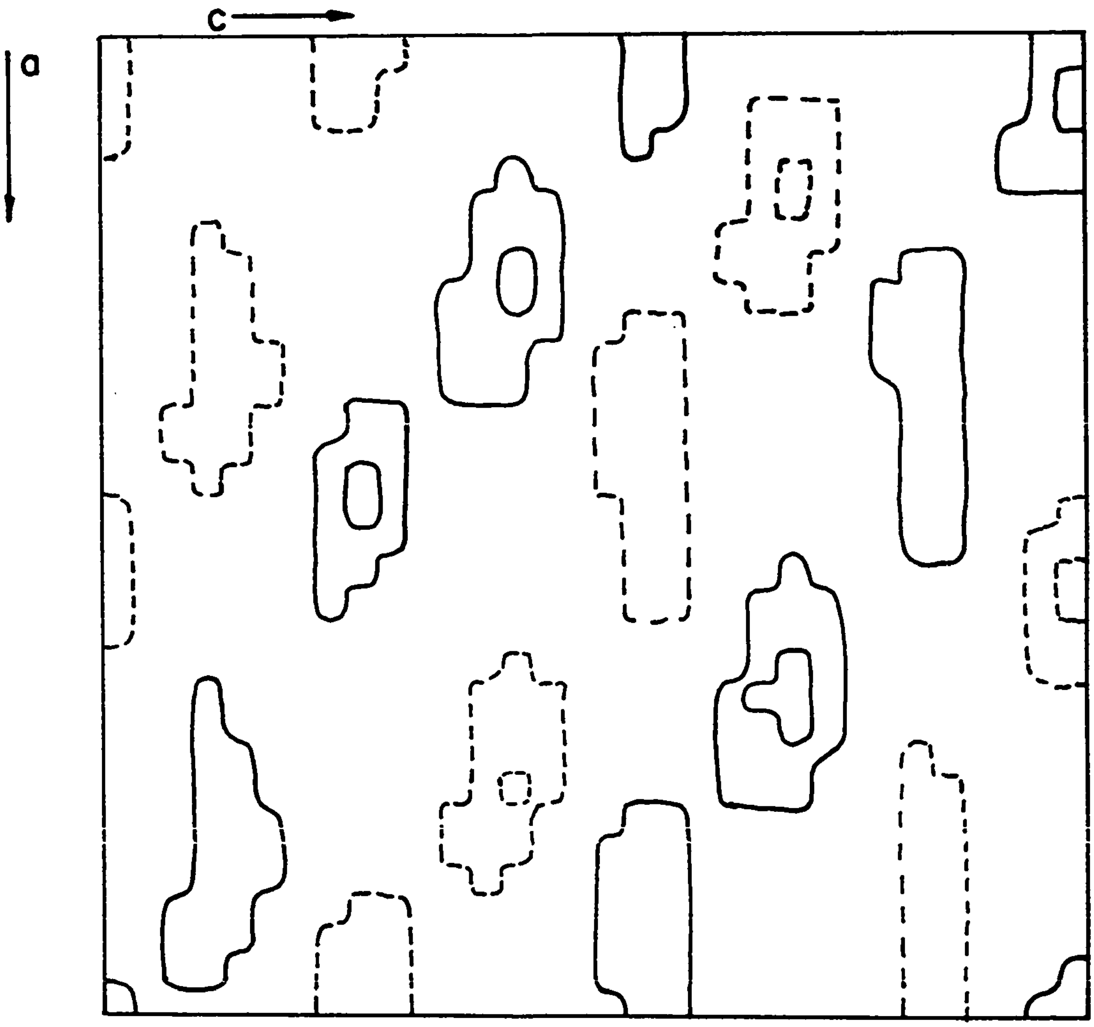

Figure 7b. Layer 9 of a Σ_2-map for β-picoline-N-oxide fumaric acid adduct. —— contoured at 15 and 25. ---- contoured at -15 and -25. No atoms in or near this section.

adjusted so that the value of the highest off-origin peak on the Patterson map was approximately equal to the value of the highest off-origin peak on the Σ_2-map. Figure 8 shows some typical results using the peak corresponding to the nitrogen atom position as the superposition point.

Each of the five maps was then squared and the resulting transform coefficients (G_{hkl}) calculated using a fast Fourier algorithm.[14] Only the 50 reflections with the largest values of the product $|E^*G|$ and with $|E| > 1.50$ were kept. Some typical results are shown in Table XIV. The signs determined for each hkl were then averaged. (In some cases more than 80% of the 50 largest $|E|$'s were signed correctly in the individual sets.) These signed reflections were then used to compute a new Σ_2-type map, and negative regions were set equal to zero. The transform of the square of this map was used to obtain signs (92% of which were correct) for input at the 90% confidence level into a phase extending program.[40] The resulting complete set of 270 phases was used to compute a map from which the positions of all the non-hydrogen atoms in the structure were readily determined. There were no significant spurious peaks on this map. This method has recently been used to solve the structure of diisopropyl-(2,3,4,5-tetraphenyl-cyclopenta-2,4-diene)-phosphate, which crystallizes in space group $P2_1/c$ with 160 non-

Table XIV. Representative results[a] obtained for a centric test case (β-picoline-N-oxide fumaric acid adduct)

				Calculated signs					Known
h	k	l	|E|	1	2	3	4	5	sign
-2	10	4	3.25	-	+	+	+	+	+
-1	11	7	3.13	-	+	+		+	+
0	11	9	3.00	-		+			+
0	12	4	2.94				+		-
-1	10	12	2.80	-	-	-	-	-	-
1	10	8	2.73	+	-	-		-	-
-2	4	12	2.63				+		+
1	11	7	2.55	+	-			-	-

[a] The absence of a calculated sign indicates that the value of | E*G | for that reflection was smaller than the limit used.

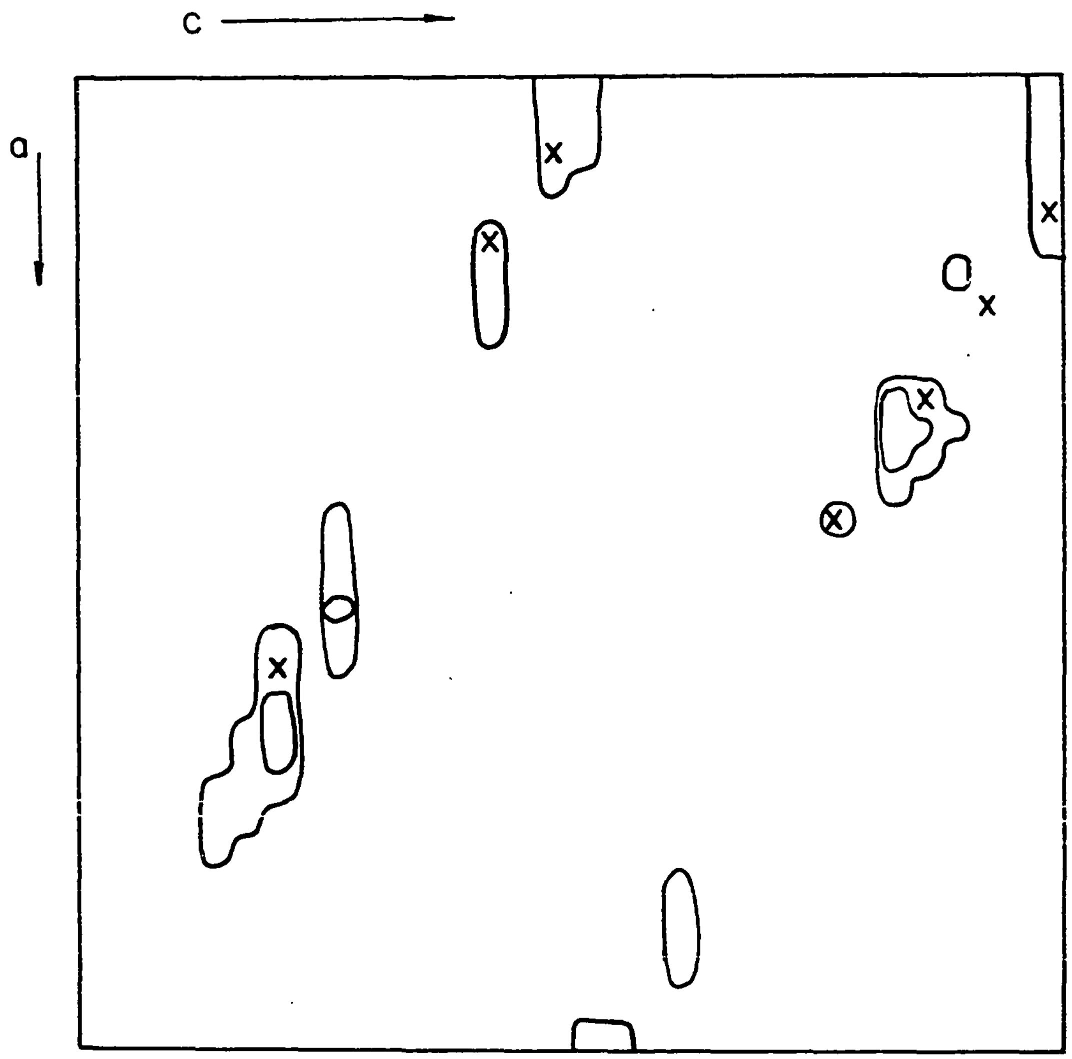

Figure 8a. Results of Patterson map superimposed on the Σ_2-map of Figure 7 at the four nitrogen equivalent positions. Composite of layers 3,4,5. —— contoured at 9 and 18. Atom positions indicated by X.

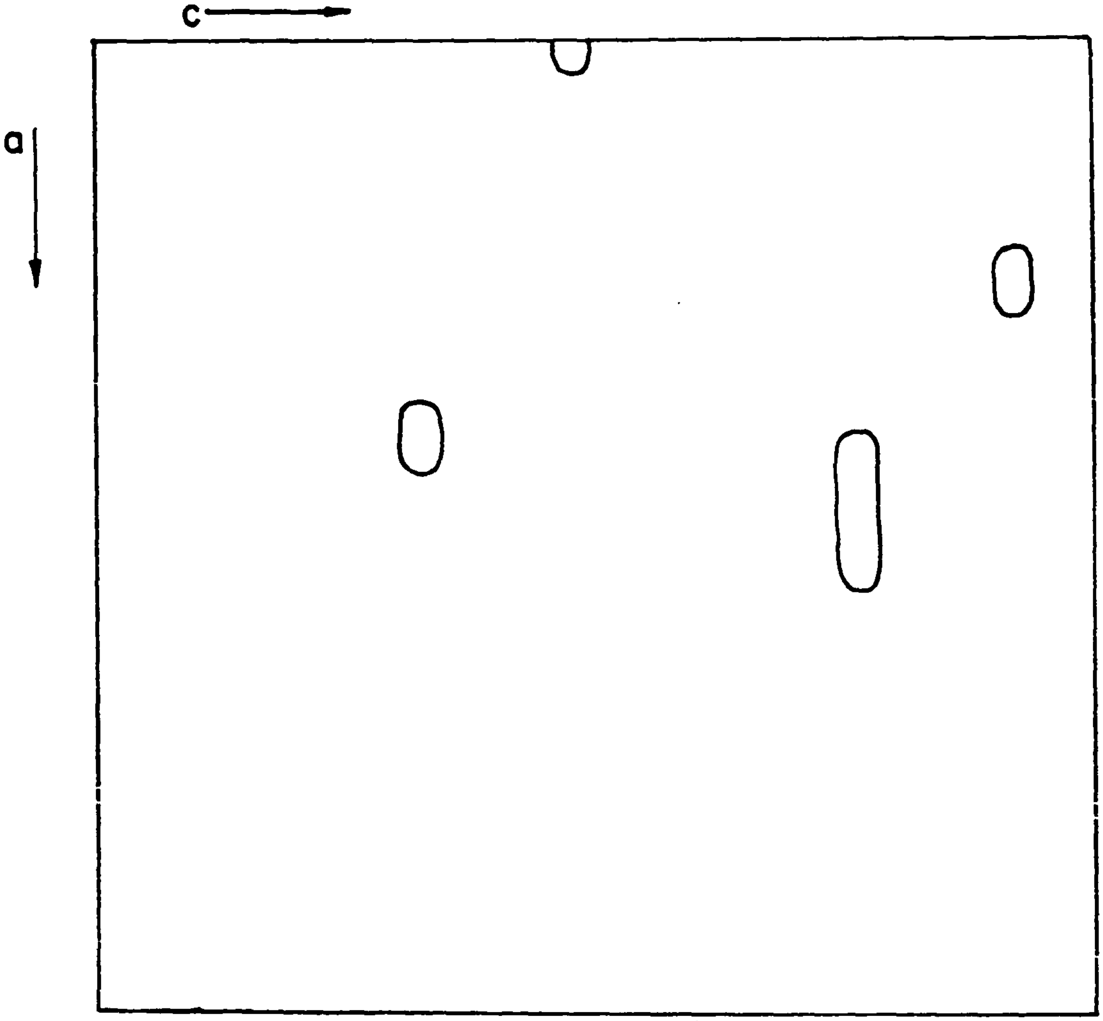

Figure 8b. Results of Patterson map superimposed on the Σ_2-map of Figure 7 at the four nitrogen equivalent positions. Layer 9. —— contoured at 5. No atoms in or near this section.

hydrogen atoms per cell.[41]

Acentric case

The compound chosen as a test for the acentric case was D-(1,5)-glucono lactone,[33] which crystallizes in the space group $P2_12_12_1$ with axial lengths $\underline{a}$ = 7.838, $\underline{b}$ = 12.322, $\underline{c}$ = 7.544 Å. Of the 1259 unique observed reflections, 435 have $|E|$ greater than 1.0. Three origin-determining reflections were selected ($|E|$ = 2.39, 2.51, 2.04), two of which were non-zonal reflections. To make the test a reasonably severe one, no enantiomorph-selecting reflection was included; the enantiomorph was chosen as part of the superposition procedure.

The Σ_2-map produced from these and their symmetry-related reflections showed more than 25 large peaks, all of which had relative heights between 120 and 185. The relative height of the other peaks on this map was less than 90. However, only eight of these 25 peaks generated appropriate Harker vectors on the Patterson. Of these eight, four were clearly in one enantiomorph, since they remained when the superposition point was a member of the set.

Using these four peaks and proceeding in a manner similar to the centric case above, the average phases of the 30 largest $|E^*G|$'s with $|E| > 1.40$ were obtained. Some typical results for this acentric structure are given in

Table XV. The transform of the square of the expanded Σ_2-map yielded initial phases for the 50 largest $|E*G|$'s with $|E| > 1.40$. These phased E's were then input into a phase-extending program at the 90% confidence level. In the resulting set of 410 phased E's, 390 were within 40° of the published phases and their transform readily revealed all the non-hydrogen atomic positions. As in the centric case, there were no significant spurious peaks on this final map.

Extensions of the Method

Since a reasonably severe test of the method was desired, only three origin-determining reflections were used for the initial Σ_2-map calculation in each case. Other reflections, phased by either Σ_2 or Σ_1 relations (and perhaps with appropriate weights), can also be included to produce peaks of greater reliability. In the centric case, those Σ_1 reflections which were determined with greater than 90% probability were included in the calculation of the Σ_2-map. Their addition improved the phase agreement to a small extent. The frequency check procedure[42] can also be applied to the Σ_2-map.

It is also possible to include an additional strong reflection in the origin-determining reflections, thus producing two Σ_2-maps in the centric case, for example. The resulting two sets of averaged phases could then be

Table XV. Representative results[a] obtained for an acentric test case (D-(1,5)-glucono lactone)

				Calculated phases (°)				Known
h	k	l	|E|	1	2	3	4	phase
0	3	6	3.48	270	270	270	270	270
0	4	3	2.67	0	180	180	180	180
0	7	9	2.58		90	90	90	90
0	11	2	2.36	89				90
2	2	6	2.22		277		256	296
1	7	5	2.19			276	284	293
4	-9	-1	2.04	230	243	220	256	252
0	4	0	1.95	179				0
3	-3	-7	1.77	179	171	182	177	197
3	4	2	1.64		261		278	316
1	-3	-2	1.59		83		90	81
0	2	3	1.58	179	0		0	180
2	4	6	1.56	84	88		91	114

[a] The absence of a calculated phase indicates that the value of |E*G| for that reflection was smaller than the limit used.

handled in a similar way as in those direct methods programs which produce multiple solutions.

Lower bounds different from zero can be used for modification of the resultant superposition map. Indeed, some calculations for the centric example have shown that use of a lower bound of 10% of the maximum peak height results in somewhat improved agreement.

From Tables XIV and XV, it is obvious that a straight average is a poor procedure compared to a more judicious method of averaging. Preliminary computations indicate that in the centric case a requirement of a net of two like signs produces a significant improvement in the percentage of correct signs selected.

The phases are also somewhat sensitive to grid point resolution of the maps. In the centric structure, calculations were run using a grid of 0.12 x 0.45 x 0.45 Å resolution. Results improve if a resolution of approximately 0.25 Å is used in all three directions.

Squaring the resultant superposition map appears to give the best phases compared with other powers which could be employed. All exponents from 0.2 to 6.6 in steps of 0.2 were examined for a typical run, with 2.0 appearing the most satisfactory.

RESEARCH PROPOSALS

The proposals for research given below suggested themselves during the course of the research described in this thesis. No claim of an exhaustive literature search for these ideas is implied, although it is the author's belief that they are not presently under investigation.

Due to the fact that more investigators without formal training are employing crystallography as a research tool, there is a desperate need for the development of computer programs which can make many of the routine decisions now requiring human intervention. ALPP eliminates, or at least makes much less tedious, the contouring of a map, and is an example of this type of program. Such a program should be developed to implement the PSST method. One could write a program to implement the heavy atom technique, which would work in conjunction with ALOP, by cycling structure factor and Fourier series calculations, at the same time deciding whether or not possible atomic positions should be retained. As well as being easy for the initiate to use, the above programs should also permit enough intervention so that the course of their execution can be altered by the more experienced user.

The prime difficulties in implementing direct methods are in choosing between various sets of phases produced and in discriminating spurious from real atomic peaks. The first problem is being extensively researched and such methods as the comparison of absolute figures of merit and the psi zero test of MULTAN,[40] the absolute value of rho test reported by W. Ozbirn,[43] and a plot of psi zero vs. the absolute figure of merit as described by Cochran and Douglas[44] are all examples of possible solutions. A promising approach to the problem of peak discrimination should be the superposition of the Patterson on a likely atomic position, similar to the technique used in the PSST method. Another approach would be the development of a program which makes use of known interatomic vectors, such as those within a benzene ring, to choose peaks of chemical sense. A third would be the development of a more exact and easily used technique for producing three-dimensional models. A model constructed of wooden spheres suspended by fine lines from a perforated plate would satisfy these requirements.

The relative ease in determining the bonding mode of cyanide in trans-$(CN)_2$ trien cobalt (III) perchlorate indicates that X-ray diffraction data may be sensitive enough to distinguish between carbon and nitrogen. A more intensive effort on the part of investigators on this problem and perhaps even a reexamination of published data could be initiated.

The number of antimony (III) structures which have been determined is exceedingly small. The investigation of a number of different compounds of formula $M_2Sb^{III}X_5$ (M = Na, K, Rb, Cs; X = Cl, Br, I) would be of interest in determining the influence of the M^+ and the X^- on the configuration of the $SbX_5^=$ complex.

The Σ_2-map has a much wider applicability than is described in the PSST method. In combination with the discriminator function[45] to test possible atomic positions, it should have great power. In addition, using the expanded starting set of phased E's to produce a map which then would be used in place of the Σ_2-map, is an extension which has promise.

APPENDIX A

PROGRAM LISTING OF ALPP

```
//A375ALPP  JOB  A0310,WISMER,MSGLEVEL=(1,1)                            JCL-0010
//PICKPEAK  EXEC  PL1LFCL,PARM.PL1L='ATR,XREF,NOSTMT,OPT=02',           JCL-0020
//     TIME.PL1L=(0,45),REGION.PL1L=128K,                               JCL-0030
//     TIME.LKED=(0,10),PARM.LKED='LIST,LET,OVLY'                       JCL-0040
//PL1L.SYSIN  DD  *                                                     JCL-0050
 /*  PICK PEAKS FROM AN ELECTRON DENSITY OR PATTERSON MAP            */ ALPP0010
 CONTOUR: PROCEDURE OPTIONS (MAIN);                                     ALPP0020
 /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ALPP0030
          THE PURPOSE OF THIS PROGRAM IS TO PICK PEAKS FROM EITHER AN   ALPP0040
      ELECTRON-DENSITY MAP OR A PATTERSON MAP.  THE MAP MAY BE PRODUCED ALPP0050
      BY THE PROGRAM "FOUR" (ROGERS & JACOBSON, IS-2155), BY THE        ALPP0060
      PROGRAM "ALFF" (HUBBARD, QUICKSALL & JACOBSON, IS-2625), OR BY ANYALPP0070
      OTHER SUITABLE MAP-GENERATING PROGRAM.  THE MAP IS INPUT LINE BY  ALPP0080
      LINE FROM THE FILE "MAPIN".  FILE "MPOUT" IS A DIRECT-ACCESS      ALPP0090
      VERSION OF THE MAP, USED FOR INTERMEDIATE STORAGE.  FILE "PEAKS"  ALPP0100
      CONTAINS ALL OF THE PEAKS PICKED FROM THE MAP, IN ORDER OF        ALPP0110
      DESCENDING PEAK HEIGHT, IN THE FORM: IU, IV, IW, IHEIGHT, WHERE   ALPP0120
      ALL VARIABLES ARE FULL-WORD INTEGERS.                             ALPP0130
          THIS PROGRAM WAS WRITTEN IN FEBRUARY, 1972 BY R.K.WISMER.     ALPP0140
                                                                        ALPP0150
  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ALPP0160
 CARD COLUMN VARIABLE FORMAT  EXPLANATION                               ALPP0170
  1    1-80    TITLE    A(80)  ANY ALPHANUMERIC TITLE                   ALPP0180
  2    1- 4    PGM      A(4)   PROGRAM PRODUCING MAP, "ALFF" OR "FOUR"  ALPP0190
       5- 8    IACR     F(4)   NUMBER OF GRIDS ACROSS FULL LINE.        ALPP0200
       9-12    IDWN     F(4)   NUMBER OF LINES DOWN FULL LAYER.         ALPP0210
      13-16    ILAY     F(4)   NUMBER OF LAYERS IN FULL MAP.            ALPP0220
      17-20    IPRT     F(4)   =1 PRINT THE MAP; =0 DO NOT PRINT THE MAP.ALPP0230
      21-24    LMT      F(4)   LOWER LIMIT, BELOW WHICH PEAK NOT ACCEPTEDALPP0240
      25-28    FW       F(4)   FIELD WIDTH FOR PRINTED MAP.             ALPP0250
      29-32    SP       F(4)   NUMBER OF SPACES BETWEEN LINES.          ALPP0260
 ***   NO PEAK ON THE MAP MAY EXCEED THE VALUE 2**14.                   ALPP0270
 ***   THE FULL CELL MAP IN ALL THREE DIRECTIONS MUST BE INPUT.         ALPP0280
 ***   THE INPUT MAP MAY HAVE ANY ORIENTATION.                          ALPP0290
  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */ALPP0300
          DECLARE  (IACR, IDWN, ILAY, IPRT, LMT, FW, SP, IACS, PMIN,    ALPP0310
               PMAX, #PEAK)        FIXED BINARY (15);                   ALPP0320
 /*                                                                     ALPP0330
          READ IN THE CONTROL CARDS AND INITIALIZE THE MAP              ALPP0340
                                                                     */ ALPP0350
          CALL SETUP (IACR,IDWN,ILAY,IPRT,LMT,FW,SP,IACS,PMIN,PMAX);    ALPP0360
 /*                                                                     ALPP0370
          PICK PEAKS FROM EACH LAYER                                    ALPP0380
                                                                     */ ALPP0390
          CALL LAYERS (IACR,IDWN,ILAY,FW,SP,IACS,PMIN,PMAX);            ALPP0400
 /*                                                                     ALPP0410
          PICK PEAKS BETWEEN LAYERS AND OUTPUT PEAKS                    ALPP0420
                                                                     */ ALPP0430
          CALL NTRLAY (IACR,IDWN,ILAY,FW,SP,IACS,PMIN,PMAX,#PEAK,IPRT); ALPP0440
 /*                                                                     ALPP0450
```

```
          SORT PEAKS INTO DESCENDING ORDER BY HEIGHT AND OUTPUT         ALPP0460
                                                                     */ ALPP0470
          CALL PKSORT(#PEAK);                                           ALPP0480
          PUT EDIT ('NORMAL PROGRAM TERMINATION')(PAGE,A);              ALPP0490
          END CONTOUR;                                                  ALPP0500



* PROCESS('ATR,XREF,NOSTMT,OPT=02');                                    STUP0010
 /*   READ IN THE CONTROL CARDS AND INITIALIZE THE MAP               */ STUP0020
 SETUP:   PROCEDURE (IACR,IDWN,ILAY,IPRT,LMT,FW,SP,IACS,PMIN,PMAX);     STUP0030
          DECLARE  (IACR, IDWN, ILAY, IPRT, LMT, FW, SP, IACS, PMIN,    STUP0040
                    PMAX, ISTP)         FIXED BINARY (15);              STUP0050
          DECLARE   LNALFF (0:IACR-1)   CONTROLLED FLOAT BINARY,        STUP0060
                   (LNFOUR(0:IACR), LAYER (0:IACR,0:IDWN))              STUP0070
                                        CONTROLLED FIXED BINARY (15);   STUP0080
          DECLARE   PGM                 CHARACTER (4);                  STUP0090
          DECLARE   XLM                 FLOAT BINARY;                   STUP0100
          DECLARE   TITLE               CHARACTER (80);                 STUP0110
          DECLARE   MPOUT          FILE ENVIRONMENT (REGIONAL(1)) KEYED;STUP0120
          DECLARE   STAMP               CHARACTER (54) EXTERNAL;        STUP0130
          CALL TM_DT;                                                   STUP0140
 /*                                                                     STUP0150
          READ AND INTERPRET THE CONTROL CARDS                          STUP0160
                                                                     */ STUP0170
          GET EDIT (TITLE,PGM,IACR,IDWN,ILAY,IPRT,LMT,FW,SP,IMUD)       STUP0180
                   (A(80),A(4),7 F(4),X(44),F(4));                      STUP0190
          CLOSE FILE (SYSIN);                                           STUP0200
          PUT EDIT (TITLE, STAMP,                                       STUP0210
                    'NUMBER OF GRIDS ACROSS A FULL LINE =',IACR,        STUP0220
                    'NUMBER OF LINES DOWN A FULL LAYER = ',IDWN,        STUP0230
                    'NUMBER OF LAYERS IN A FULL MAP =',ILAY,            STUP0240
                    'LOWER LIMIT OF A RECOGNIZABLE PEAK = ',LMT)        STUP0250
                   (A(80),SKIP(2),X(20),A(54),4 (SKIP(2),X(25),A,       STUP0260
                    COL(65),F(4))));                                    STUP0270
          PUT EDIT ('THE GENERATING PROGRAM WAS "',PGM,'".')            STUP0280
                   (SKIP(2),X(20),A,A(4),A);                            STUP0290
          IACS = 0;                                                     STUP0300
          IF (IPRT=1)  THEN PRTR: DO;                                   STUP0310
               PUT EDIT ('THE MAP WILL BE PRINTED')(SKIP(2),X(25),A);   STUP0320
          PUT EDIT ('FIELD WIDTH =',FW,'SPACES BETWEEN LINES =',SP)     STUP0330
                   (2 (SKIP(2),X(25),A,F(4)));                          STUP0340
          CALL PRSETUP (IACS,IACR,FW,SP,PMIN,PMAX,ISTP);                STUP0350
          IF (ISTP=1)  THEN GO TO ABND;                    ELSE;        STUP0360
               END PRTR;  ELSE  PUT EDIT ('THE MAP WILL NOT BE PRINTED')STUP0370
                                         (SKIP(2),X(25),A);             STUP0380
          IF (PGM='ALFF')  THEN ALLOCATE LNALFF;                ELSE;   STUP0390
          ALLOCATE  LNFOUR, LAYER;                                      STUP0400
          PUT PAGE;                                                     STUP0410
```

```
         OPEN FILE (MAPIN) INPUT RECORD SEQUENTIAL,                     STUP0420
              FILE (MPOUT) OUTPUT RECORD SEQUENTIAL;                    STUP0430
/*                                                                      STUP0440
         READ MAP LINES, ZERO POINTS LESS THAN LMT, & WRITE MAP LAYERS STUP0450
                                                                   */  STUP0460
         IF (PGM='ALFF')  THEN DO;     XLM = LMT;                       STUP0470
              NLAY = ILAY-1;         NDWN = IDWN-1;        END;         STUP0480
                          ELSE DO;                                      STUP0490
              NLAY = ILAY - 1;       NDWN = IDWN;          END;         STUP0500
LYLUP:   DO L = 0 TO NLAY BY 1;                                         STUP0510
LNLUP:   DO K = 0 TO NDWN BY 1;                                         STUP0520
         IF (PGM='ALFF')  THEN AL: DO;                                  STUP0530
              READ FILE (MAPIN) INTO (LNALFF);                          STUP0540
GDLUP:        DO J = 0 TO IACR-1 BY 1;                                  STUP0550
              IF (LNALFF(J)<=XLM)  THEN LAYER (J,K) = 0;                STUP0560
                                   ELSE LAYER (J,K) = LNALFF(J);        STUP0570
              END GDLUP;                                                STUP0580
              LAYER(IACR,K) = LAYER(0,K);                               STUP0590
              END AL;                                                   STUP0600
                          ELSE FR: DO;                                  STUP0610
              READ FILE (MAPIN) INTO (LNFOUR);                          STUP0620
PTLUP:        DO J = 0 TO IACR BY 1;                                    STUP0630
              IF (LNFOUR(J)<=LMT)  THEN LAYER (J,K) = 0;                STUP0640
                                   ELSE LAYER (J,K) = LNFOUR (J);       STUP0650
              END PTLUP;                                                STUP0660
              END FR;                                                   STUP0670
         END LNLUP;                                                     STUP0680
         LAYER(*,IACR) = LAYER(*,0);                                    STUP0690
         WRITE FILE (MPOUT) FROM (LAYER) KEYFROM(L);                    STUP0700
         PUT EDIT ('LAYER NUMBER ',L,' WRITTEN OUT.')                   STUP0710
                  (SKIP,A,F(3),A);                                      STUP0720
         END LYLUP;                                                     STUP0730
         FREE LNALFF, LNFOUR, LAYER;                                    STUP0740
         CLOSE FILE (MAPIN),  FILE (MPOUT);                             STUP0750
         RETURN;                                                        STUP0760
ABND:    END SETUP;                                                     STUP0770


* PROCESS('ATR,XREF,NOSTMT,OPT=02');                                    TMDT0010
 /* CREATES A TIME - DATE STAMP FOR FUTURE USE                     */  TMDT0020
 (SUBRG,STRG):  TM_DT:  PROCEDURE;                                      TMDT0030
 DECLARE STAMP CHARACTER(54) EXTERNAL,  AMPM CHARACTER(4);              TMDT0040
 DECLARE MOS(12) CHARACTER(9) VARYING INITIAL ('JANUARY','FEBRUARY',    TMDT0050
         'MARCH','APRIL','MAY',"JUNE','JULY','AUGUST','SEPTEMBER',      TMDT0060
         'OCTOBER','NOVEMBER','DECEMBER');                              TMDT0070
 DECLARE TM CHARACTER(9), TMS(2) CHARACTER(2) DEFINED TM;  TM = TIME;   TMDT0080
 DECLARE DT CHARACTER(6), DS (3) CHARACTER(2) DEFINED DT;  DT = DATE;   TMDT0090
 IF (TMS(1) > 11)  THEN AMPM = "P.M.';   ELSE AMPM = 'A.M.';            TMDT0100
 STAMP = 'RUN ON ' || MOS(DS(2)) || ' ' || DS(3) || ', 19' || DS(1) ||   TMDT0110
     ', AT ' || TMS(1) || ' HOURS ' || TMS(2) || ' MINUTES ' || AMPM;    TMDT0120
 END TM_DT;                                                               TMDT0130


* PROCESS('ATR,XREF,NOSTMT,OPT=02');                                      PRSU0010
 /*   A PROCEDURE TO CHECK THE PRINT PARAMETERS AND SET UP PRINTING  */   PRSU0020
 PRSETUP: PROCEDURE (ACRS,ACRE,FW,SP,PMIN,PMAX,ISTP);                     PRSU0030
          DECLARE  (ACRS, ACRE,     /* INITIAL & FINAL ACROSS GRIDS   */  PRSU0040
                    FW, SP,         /* FIELD WIDTH, SPACES BTWN LINES */  PRSU0050
                    PMIN, PMAX,     /* MINIMUM AND MAXIMUM PEAK HEIGHT*/  PRSU0060
                    ISTP, PMN, PMX)     FIXED BINARY (15);                PRSU0070
          DECLARE   XXX             CHARACTER (9);                        PRSU0080
          DECLARE   MN(2)           FIXED BIN INIT (-9,-99),              PRSU0090
                    MX(2)           FIXED BIN INIT (99,999);              PRSU0100
          DECLARE   NUMS(*)         CHAR(*) EXTERNAL CONTROLLED,          PRSU0110
                    HEAD            CHAR (132) STATIC EXTERNAL;           PRSU0120
          IF (FW*(ACRE-ACRS+1)>128)  THEN DO;                             PRSU0130
               PUT EDIT ('***** FW IS TOO LARGE FOR THE NUMBER OF GRID',PRSU0140
                         ' POINTS ACROSS THE PAGE * TERMINATING *****')   PRSU0150
                        (SKIP(4),COL(20),A,A);                            PRSU0160
               GO TO TERM;                                   END;         PRSU0170
          IF (FW=2) THEN I=1;                                ELSE I=2;    PRSU0180
          PMIN = MN(I);                                                   PRSU0190
          PMAX = MX(I);                                                   PRSU0200
          IF ((PMIN<MN(I)) | (PMAX>MX(I)))  THEN DO;                      PRSU0210
               PUT EDIT ('***** PMIN OR PMAX OUT OF RANGE * DEFAULTS ',   PRSU0220
                          'ASSUMED *****')                                PRSU0230
                        (SKIP(4),COL(20),A,A);                            PRSU0240
               IF (PMIN<MN(I))  THEN PMIN = MN(I);                        PRSU0250
               IF (PMAX>MX(I))  THEN PMAX = MX(I);           END;         PRSU0260
          IF (PMIN>0)  THEN PMIN=0;                                       PRSU0270
          ALLOCATE  NUMS (PMIN-1:PMAX+1) CHAR(FW);                        PRSU0280
 /*                                                                       PRSU0290
          NUMS ARRAY CONTAINS CHAR REPRESENTATION OF ALL NUMBERS FROM     PRSU0300
          PMIN TO PMAX.  HEAD IS THE HEADING FOR EACH LAYER               PRSU0310
                                                                    */    PRSU0320
 BL:      DO J = 1 TO 132 BY 1;                                           PRSU0330
          SUBSTR(HEAD,J,1) = ' ';                            END BL;      PRSU0340
 NM:      DO I = PMIN TO PMAX;                                            PRSU0350
          XXX = I;                                                        PRSU0360
          NUMS(I) = SUBSTR(XXX,10-FW,FW);                    END NM;      PRSU0370
          PMN = PMIN - 1;                                                 PRSU0380
          PMX = PMAX + 1;                                                 PRSU0390
          NUMS(PMN) = SUBSTR('     ',6-FW,FW);                            PRSU0400
          NUMS(PMX) = SUBSTR('*****',6-FW,FW);                            PRSU0410
          J = 5 - FW;                                                     PRSU0420
 HD:      DO I = ACRS TO ACRE;                                            PRSU0430
```

```
         J = J + FW;
         SUBSTR(HEAD,J,FW) = NUMS(I);                           END HD;
         ISTP = 0;
RTN:     RETURN;
TERM:    ISTP = 1;
         GO TO RTN;
         END PRSETUP;



/*    PICK PEAKS FROM EACH LAYER AND WRITE OUT THE REVISED LAYERS    */
LAYERS:  PROCEDURE (IACR,IDWN,ILAY,FW,SP,IACS,PMIN,PMAX);
         DECLARE   (IACR,IDWN,ILAY,FW,SP,IACS,PMIN,PMAX)
                                     FIXED BINARY (15);
         DECLARE   MPOUT        FILE ENVIRONMENT (REGIONAL(1)) KEYED;
         DECLARE  (LNACR(0:IACR), LNDWN(0:IDWN), LAYER(0:IACR,0:IDWN))
                                     CONTROLLED FIXED BINARY;
         OPEN FILE (MPOUT) UPDATE RECORD DIRECT;
         ALLOCATE  LAYER, LNDWN, LNACR;
         PUT PAGE;
/*
         FIND THE PEAKS LAYER BY LAYER
                                                                     */
LAYLP:   DO L = 0 TO ILAY-1 BY 1;
         READ FILE (MPOUT) INTO (LAYER) KEY (L);
/*
         GO THROUGH THE LAYERS ROW BY ROW
                                                                     */
LINLP:   DO K = 0 TO IDWN BY 1;
         LNACR(*) = LAYER (*,K);
         CALL PICKER (LNACR,IACR);
         LAYER(*,K) = LNACR;
         END LINLP;
/*
         GO THROUGH THE LAYERS COLUMN BY COLUMN
                                                                     */
COLUP:   DO J = 0 TO IACR BY 1;
         LNDWN = LAYER (J,*);
         CALL PICKER (LNDWN,IDWN);
         LAYER(J,*) = LNDWN;
         END COLUP;
/*
         CHECK DIAGONAL ELEMENTS
                                                                     */
KLUP:    DO K = 0 TO IDWN BY 1;
         IF (K=0)  THEN IDM = IDWN - 1;
                   ELSE IDM = K - 1;
         IF (K=IDWN)  THEN IDP = 1;
```

```
                       ELSE IDP = K + 1;
 JLUP:    DO J = 0 TO IACR BY 1;
          IF (LAYER(J,K) = 0)  THEN GO TO NDJL;                 ELSE;
          IF (J=0)  THEN IAM = IACR - 1;
                    ELSE IAM = J - 1;
          IF (J=IACR)  THEN IAP = 1;
                       ELSE IAP = J + 1;
          LJK = LAYER(J,K);
          LAJK = ABS(LAYER(J,K));
          IF (ABS(LAYER(IAP,IDP)) > LAJK)  THEN LJK = -LAJK;
                    ELSE LAYER(IAP,IDP) = -ABS(LAYER(IAP,IDP));
          IF (ABS(LAYER(IAP,IDM)) > LAJK)  THEN LJK = -LAJK;
                    ELSE LAYER(IAP,IDM) = -ABS(LAYER(IAP,IDM));
          IF (ABS(LAYER(IAM,IDP)) > LAJK)  THEN LJK = -LAJK;
                    ELSE LAYER(IAM,IDP) = -ABS(LAYER(IAM,IDP));
          IF (ABS(LAYER(IAM,IDM)) > LAJK)  THEN LJK = -LAJK;
                    ELSE LAYER(IAM,IDM) = -ABS(LAYER(IAM,IDM));
          LAYER(J,K) = LJK;
 NDJL:    END JLUP;
          END KLUP;
 /*
          OUTPUT THE CONTOURED MAP LAYER
                                                                    */
          REWRITE FILE (MPOUT) FROM (LAYER) KEY (L);
          PUT EDIT ('PEAK-PICKED LAYER ',L,' HAS BEEN REWRITTEN.')
                   (SKIP,A,F(3),A);
          END LAYLP;
          RETURN;
          END LAYERS;



* PROCESS('ATR,XREF,NOSTMT,OPT=02');
 /*  PROCEDURE TO CONTOUR A LINE                                    */
 PICKER:  PROCEDURE (LINE,IACR);
          DECLARE   LINE (*)            FIXED BINARY (15);
          DECLARE  (IACR,IA,IB,I,LIK)   FIXED BINARY (15);
          DECLARE  (OK, MK)             BIT (1);
          IA = ABS (LINE(IACR-1));
          IB = IACR - 1;
          OK = '1'B;
          I = 0;
          MK = '0'B;
 SGD:     LIK = ABS (LINE(I));
          IF (LIK > IA)  THEN GO TO OK1;                        ELSE
          IF (OK)  THEN GO TO OK0;                ELSE GO TO RST;
 OK0:     OK = '0'B;
          GO TO INC;
 OK1:     OK = '1'B;
 RST:     IF (¬MK)  THEN LINE(IB) = -IA;                     ELSE;
 INC:     IA = LIK;
          IB = I;
          I = I + 1;
          IF (MK)  THEN GO TO NDL;                           ELSE;
          IF (I = IACR+1)  THEN GO TO FLS;       ELSE GO TO SGD;
 FLS:     I = 0;
          MK = '1'B;
          GO TO SGD;
 NDL:     RETURN;
          END PICKER;
```

```
* PROCESS('ATR,XREF,NOSTMT,OPT=02');
 /*    PICK PEAKS BETWEEN LAYERS AND OUTPUT THEIR POSITION AND HEIGHT */
 NTRLAY:  PROCEDURE (IACR,IDWN,ILAY,FW,SP,IACS,PMIN,PMAX,#PEAK,IPRT);
          DECLARE   (IACR,IDWN,ILAY,FW,SP,IACS,PMIN,PMAX,#PEAK,IPRT)
                                        FIXED BINARY (15);
          DECLARE  (LAYER(0:IACR,0:IDWN), LYRB4(0:IACR,0:IDWN))
                                        CONTROLLED FIXED BINARY (15);
          DECLARE   HEAD                CHARACTER (132) STATIC EXT;
          DECLARE 1 PEAK,
                    2 (U,V,W,VALU)      FIXED BINARY (31);
          DECLARE   MPOUT          FILE ENVIRONMENT(REGIONAL(1)) KEYED,
                    PEAKS          FILE RECORD SEQUENTIAL;
          DECLARE   LNACR(0:IACR)       CONTROLLED FIXED BINARY(15);
          ALLOCATE  LNACR;
          #PEAK = 0;
          PUT PAGE;
          OPEN FILE (MPOUT) UPDATE RECORD DIRECT;
          ALLOCATE  LAYER, LYRB4;
          READ FILE (MPOUT) INTO (LYRB4) KEY (ILAY-1);
 /*
          GO THROUGH THE MAP LAYER BY LAYER
                                                                    */
 LAYLUP:  DO L = 0 TO ILAY-1 BY 1;
          READ FILE (MPOUT) INTO (LAYER) KEY(L);
          IF (L=0)  THEN LB4 = ILAY - 1;
                    ELSE LB4 = L - 1;
 /*
          GO THROUGH EACH LAYER LINE BY LINE
                                                                    */
 DWNLUP:  DO K = 0 TO IDWN BY 1;
          IF (K=0)  THEN IDM = IDWN - 1;
                    ELSE IDM = K - 1;
          IF (K=IDWN)  THEN IDP = 1;
                       ELSE IDP = K + 1;
 /*
```

```
         GO THROUGH EACH LINE POINT BY POINT
                                                                   */
ACRLUP:  DO J = 0 TO IACR BY 1;
         IF (LAYER(J,K) = 0)  THEN GO TO NDAL;                 ELSE;
         IF (J=0)  THEN IAM = IACR - 1;
                   ELSE IAM = J - 1;
         IF (J=IACR)  THEN IAP = 1;
                      ELSE IAP = J + 1;
         LJK = LAYER(J,K);
         LAJK = ABS(LAYER(J,K));
/*
         DETERMINE IF THE PEAK SHOULD BE KEPT
                                                                   */
         IF (ABS(LYRB4(IAP,IDP)) > LAJK)  THEN LJK = -LAJK;
                   ELSE LYRB4(IAP,IDP) = -ABS(LYRB4(IAP,IDP));
         IF (ABS(LYRB4(IAP,  K)) > LAJK)  THEN LJK = -LAJK;
                   ELSE LYRB4(IAP,  K) = -ABS(LYRB4(IAP,  K));
         IF (ABS(LYRB4(IAP,IDM)) > LAJK)  THEN LJK = -LAJK;
                   ELSE LYRB4(IAP,IDM) = -ABS(LYRB4(IAP,IDM));
         IF (ABS(LYRB4(  J,IDP)) > LAJK)  THEN LJK = -LAJK;
                   ELSE LYRB4(  J,IDP) = -ABS(LYRB4(  J,IDP));
         IF (ABS(LYRB4(  J,  K)) > LAJK)  THEN LJK = -LAJK;
                   ELSE LYRB4(  J,  K) = -ABS(LYRB4(  J,  K));
         IF (ABS(LYRB4(  J,IDM)) > LAJK)  THEN LJK = -LAJK;
                   ELSE LYRB4(  J,IDM) = -ABS(LYRB4(  J,IDM));
         IF (ABS(LYRB4(IAM,IDP)) > LAJK)  THEN LJK = -LAJK;
                   ELSE LYRB4(IAM,IDP) = -ABS(LYRB4(IAM,IDP));
         IF (ABS(LYRB4(IAM,  K)) > LAJK)  THEN LJK = -LAJK;
                   ELSE LYRB4(IAM,  K) = -ABS(LYRB4(IAM,  K));
         IF (ABS(LYRB4(IAM,IDM)) > LAJK)  THEN LJK = -LAJK;
                   ELSE LYRB4(IAM,IDM) = -ABS(LYRB4(IAM,IDM));
         LAYER(J,K) = LJK;
NDAL:    END ACRLUP;
         END DWNLUP;
         REWRITE FILE (MPOUT) FROM (LYRB4) KEY (LB4);
         IF (IPRT=1)  THEN CALL OUTP;                          ELSE;
         PUT EDIT ('MAP LAYER ',L,' HAS BEEN REWRITTEN')
                 (SKIP,A,F(3),A);
         LYRB4 = LAYER;
         END LAYLUP;
         REWRITE FILE (MPOUT) FROM (LAYER) KEY (ILAY-1);
         FREE LYRB4;
         PUT PAGE;
         OPEN FILE (PEAKS) OUTPUT;
/*
         FIND AND WRITE OUT ALL THE PEAKS
                                                                   */
LUPLAY:  DO L = 0 TO ILAY-1 BY 1;
         READ FILE (MPOUT) INTO (LAYER) KEY (L);
         W = L;
```

```
 /*                                                                     NRLY0860
         LOOP THROUGH LINES AND GRID POINTS                             NRLY0870
                                                                     */ NRLY0880
 LUPDWN: DO K = 0 TO IDWN-1 BY 1;                                       NRLY0890
         V = K;                                                         NRLY0900
 LUPACR: DO J = 0 TO IACR-1 BY 1;                                       NRLY0910
         IF (LAYER(J,K) <= 0)  THEN GO TO NDA;                  ELSE;   NRLY0920
         U = J;                                                         NRLY0930
         VALU = LAYER(J,K);                                             NRLY0940
         PUT EDIT ('PEAK FOUND AT:  U=',U,'V =',V,'W =',W,'HEIGHT =',   NRLY0950
                   VALU,' ') (4 (A,F(4),X(4)),X(1),A);                  NRLY0960
         WRITE FILE (PEAKS) FROM (PEAK);                                NRLY0970
         #PEAK = #PEAK + 1;                                             NRLY0980
 NDA:    END LUPACR;                                                    NRLY0990
         END LUPDWN;                                                    NRLY1000
         PUT SKIP(2);                                                   NRLY1010
         END LUPLAY;                                                    NRLY1020
         CLOSE FILE (MPOUT), FILE (PEAKS);                              NRLY1030
         FREE LAYER;                                                    NRLY1040
         PUT EDIT ('#PEAKS FOUND =',#PEAK)    (SKIP(3),A,F(6));         NRLY1050
 /*                                                                     NRLY1060
         PROCEDURE TO OUTPUT A MAP LAYER                                NRLY1070
                                                                     */ NRLY1080
 OUTP:   PROCEDURE;                                                     NRLY1090
         PUT EDIT ('LAYER NUMBER ',L,' OF ',ILAY,HEAD,' ')              NRLY1100
                  (PAGE,2 (A,F(4)),SKIP(2),A(132),SKIP,A);              NRLY1110
 LOOP:   DO LUPI = 0 TO IDWN BY 1;                                      NRLY1120
         LNACR(*) = LAYER(*,LUPI);                                      NRLY1130
         CALL PLINE (FW,SP,PMIN,PMAX,LUPI,IACS,IACR,LNACR);             NRLY1140
         END LOOP;                                                      NRLY1150
         PUT EDIT (HEAD) (SKIP(3),A(132));                              NRLY1160
         RETURN;                                                        NRLY1170
         END OUTP;                                                      NRLY1180
         RETURN;                                                        NRLY1190
         END NTRLAY;                                                    NRLY1200


* PROCESS('ATR,XREF,NOSTMT,OPT=02');                                    PKST0010
 /*   SORT THE PEAKS INTO DESCENDING ORDER BY HEIGHT                 */ PKST0020
 PKSORT:  PROCEDURE (#PEAK);                                            PKST0030
 /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * PKST0040
 METHOD:                                                                PKST0050
         A MECIAN VALUE IS PICKED FROM THE LIST.  THE LIST IS DIVIDED   PKST0060
         INTO TWO PARTS, WITH ALL THE VALUES LARGER THAN THE MEDIAN IN  PKST0070
         THE UPPER PART AND ALL THE SMALLER VALUES IN THE LOWER PART.   PKST0080
         EACH PART IS FURTHER DIVIDED IN THE SAME MANNER UNTIL ALL      PKST0090
         PARTS ARE OF LENGTH ONE, WHEREUPON THE SORT IS COMPLETE.       PKST0100
         THE ARRAYS IL AND IU STORE POINTERS TO EACH OF THE PARTS.      PKST0110
```

```
                                                                                   PKST0120
         A SORT IN  ASCENDING ORDER IS ACCOMPLISHED BY REVERSING THE               PKST0130
         COMPARISONS ON THE SEVEN CARDS INVOLVING LOGICAL COMPARISONS              PKST0140
         WITH THE EG ELEMENT IN THE B STRUCTURAL ARRAY.  THESE                     PKST0150
         COMPARISONS ARE THE STATEMENTS LABELED: REV1, REV2, REV3,                 PKST0160
         REV4, REV5, REV6, REV7.    THEY ARE REVERSED IN THE FOLLOWING             PKST0170
         MANNER:  >=  -->  <=   AND  <=  -->  >= .                                 PKST0180
                                                                                   PKST0190
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */ PKST0200
         DECLARE  (IU(16), IL(16))     FIXED BINARY;                               PKST0210
         DECLARE   #PEAK               FIXED BINARY (15);                          PKST0220
         DECLARE 1 PEAK,                                                           PKST0230
                   2 (U,V,W,VALU)      FIXED BINARY (31);                          PKST0240
         DECLARE   PEAKS               FILE RECORD SEQUENTIAL;                     PKST0250
         DECLARE 1 T,                                                              PKST0260
                   2 (U,V,W,EG)        FIXED BINARY (15);                          PKST0270
         DECLARE 1 B (#PEAK)           CONTROLLED,                                 PKST0280
                   2 (U,V,W,EG)        FIXED BINARY (15);                          PKST0290
         DECLARE 1 TT LIKE T;                                                      PKST0300
         ALLOCATE B;                                                               PKST0310
         OPEN FILE (PEAKS);                                                        PKST0320
/*                                                                                 PKST0330
         READ IN THE PEAKS PICKED FROM THE MAP                                     PKST0340
                                                                                */ PKST0350
ILUP:    DO I = 1 TO #PEAK BY 1;                                                   PKST0360
         READ FILE (PEAKS) INTO (PEAK);                                            PKST0370
         B.U(I) = PEAK.U;                                                          PKST0380
         B.V(I) = PEAK.V;                                                          PKST0390
         B.W(I) = PEAK.W;                                                          PKST0400
         B.EG(I) = PEAK.VALU;                                                      PKST0410
         END ILUP;                                                                 PKST0420
         CLOSE FILE (PEAKS);                                                       PKST0430
/*                                                                                 PKST0440
         DO THE ACTUAL SORT                                                        PKST0450
                                                                                */ PKST0460
         M = 1;                                                                    PKST0470
         II, I = 1;                                                                PKST0480
         J = #PEAK;                                                                PKST0490
L05:     IF (I >= J) THEN              GO TO L70;                          ELSE;   PKST0500
L10:     K = I;                                                                    PKST0510
         IJ = (J+I)/2;                                                             PKST0520
         T = B(IJ);                                                                PKST0530
REV1:    IF (B.EG(I) >= T.EG)  THEN    GO TO L20;                          ELSE;   PKST0540
         B(IJ) = B(I);                                                             PKST0550
         B(I) = T;                                                                 PKST0560
         T = B(IJ);                                                                PKST0570
L20:     L=J;                                                                      PKST0580
REV2:    IF (B.EG(J) <= T.EG)  THEN    GO TO L40;                          ELSE;   PKST0590
         B(IJ) = B(J);                                                             PKST0600
         B(J) = T;                                                                 PKST0610
         T = B(IJ);
REV3:    IF (B.EG(I) >= T.EG) THEN      GO TO L40;               ELSE;
         B(IJ) = B(I);
         B(I) = T;
         T = B(IJ);                     GO TO L40;
L30:     B(L) = B(K);
         B(K) = TT;
L40:     L = L-1;
REV4:    IF (B.EG(L) < T.EG) THEN       GO TO L40;               ELSE;
         TT = B(L);
L50:     K = K+1;
REV5:    IF (B.EG(K) > T.EG) THEN       GO TO L50;               ELSE;
         IF (K <= L) THEN               GO TO L30;               ELSE;
         IF (L-I <= J-K) THEN           GO TO L60;               ELSE;
         IL(M) = I;
         IU(M) = L;
         I = K;
         M = M+1;                       GO TO L80;
L60:     IL(M) = K;
         IU(M) = J;
         J = L;
0.615661 M = M+1;                       GO TO L80;
L70:     M = M-1;
         IF (M = 0)  THEN GO TO NDSRT;                           ELSE;
         I = IL(M);
         J = IU(M);
L80:     IF (J-I >= II) THEN            GO TO L10;               ELSE;
         IF (I = II) THEN               GO TO L05;               ELSE;
         I = I-1;
L90:     I = I+1;
         IF (I = J) THEN                GO TO L70;               ELSE;
         T = B(I+1);
REV6:    IF (B.EG(I) >= T.EG) THEN      GO TO L90;               ELSE;
         K = I;
L95:     B(K+1) = B(K);
         K = K-1;
REV7:    IF (T.EG > B.EG(K)) THEN       GO TO L95;               ELSE;
         B(K+1) = T;                    GO TO L90;
/*
         WRITE OUT THE SORTED PEAKS
                                                                     */
NDSRT:   PUT EDIT ('LIST OF SORTED PEAKS',
                   '         IU        IV        IW   IHEIGHT     ',
                   '         IU        IV        IW   IHEIGHT     ',
                   '         IU        IV        IW   IHEIGHT     ')
                  (PAGE,A,SKIP,3 A);
         PUT SKIP;
         OPEN FILE (PEAKS) OUTPUT;
LUPI:    DO I = 1 TO #PEAK BY 1;
         PEAK.U = B.U(I);
```

```
          PEAK.V = B.V(I);                                               PKST1120
          PEAK.W = B.W(I);                                               PKST1130
          PEAK.VALU = B.EG(I);                                           PKST1140
          WRITE FILE (PEAKS) FROM (PEAK);                                PKST1150
          PUT EDIT (PEAK,' ') (4 F(10),X(3),A);                          PKST1160
          PUT FILE (PUNCH) EDIT (PEAK) (4 F(5));                         PKST1170
          END LUPI;                                                      PKST1180
          CLOSE FILE (PEAKS);                                            PKST1190
          FREE B;                                                        PKST1200
          PUT EDIT ('THE SORTED PEAKS HAVE BEEN WRITTEN OUT')            PKST1210
                   (SKIP(4),A);                                          PKST1220
          RETURN;                                                        PKST1230
          END PKSORT;                                                    PKST1240
```

```
* PROCESS('ATR,XREF');                                                   PLIN0010
 /*   PROCEDURE TO PRINT A LINE OF THE MAP                            */ PLIN0020
 (SUBRG,STRG):                                                           PLIN0030
 PLINE:   PROCEDURE (FW, SP, PMN, PMX, LN, ACRS, ACRE, LIN);             PLIN0040
          DECLARE   NUMS(*)        CHAR(*) EXTERNAL CONTROLLED,          PLIN0050
                    HEAD           CHAR (132) STATIC EXTERNAL,           PLIN0060
                    LINE           CHAR (132) STATIC,                    PLIN0070
                    LIN(*)         CONTROLLED FIXED BINARY,              PLIN0080
                   (FW, SP, PMN, PMX, LN, ACRS, ACRE, LL)                PLIN0090
                                   FIXED BINARY;                         PLIN0200
          LINE = HEAD;                                                   PLIN0210
          SUBSTR(LINE,1,2) = SUBSTR(NUMS(LN),FW-1,2);                    PLIN0220
          J = 5 - FW;                                                    PLIN0230
 LOOP:    DO I = ACRS TO ACRE;                                           PLIN0240
          J = J + FW;                                                    PLIN0250
          LL = LIN(I);                                                   PLIN0260
          IF (LL<PMN)  THEN LL=PMN - 1;                                  PLIN0270
          IF (LL>PMX)  THEN LL=PMX + 1;                                  PLIN0280
          SUBSTR(LINE,J,FW) = NUMS(LL);                    END LOOP;     PLIN0290
          J = J + FW + 2;                                                PLIN0300
          IF (J>130)  THEN GO TO PRINT;                                  PLIN0310
          SUBSTR(LINE,J,2) = SUBSTR(NUMS(LN),FW-1,2);                    PLIN0320
 PRINT:   PUT SKIP(SP+1) EDIT (LINE) (A(132));                           PLIN0330
          RETURN;                                                        PLIN0340
          END PLINE;                                                     PLIN0350
```

```
//LKED.SYSLMOD  DD  DSNAME=PROG.CRYST,                                   JCL-0060
//    UNIT=DISK,VOLUME=SER=PROGPK,DISP=(OLD,KEEP)                        JCL-0070
//LKED.SYSIN  DD  *                                                      JCL-0080
  OVERLAY ALPHA                                                          OVLY0010
```

```
 INSERT SETUP                                                          OVLY0020
 OVERLAY BETA                                                          OVLY0030
 INSERT TM_DT                                                          OVLY0040
 OVERLAY BETA                                                          OVLY0050
 INSERT PRSETUP                                                        OVLY0060
 OVERLAY ALPHA                                                         OVLY0070
 INSERT LAYERS                                                         OVLY0080
 INSERT PICKER                                                         OVLY0090
 OVERLAY ALPHA                                                         OVLY0100
 INSERT NTRLAY                                                         OVLY0110
 INSERT PLINE                                                          OVLY0120
 OVERLAY ALPHA                                                         OVLY0130
 INSERT PKSORT                                                         OVLY0140
 NAME  ALPP(R)                                                         JCL-0090
```

APPENDIX B

PROGRAM LISTING OF ALOP

```
//A375ALOP  JOP  'A0310,RE=128K,TI=04,LI=5',WISMER,MSGLEVEL=(1,1)
//PATTERSN  EXEC   PL1LFCL,PARM.PL1L='ATR,XREF,NOSTMT,OPT=02',
//     TIME.PL1L=(2,40),REGION.PL1L=128K,
//     PARM.LKED='LIST,LET,OVLY',TIME.LKED=(0,50)
//PL1L.SYSIN  DD  *
 /*    FIND THE BEST SET OF PATTERSON PEAKS                              */
 ALOP:     PROCEDURE (SDBG) OPTIONS (MAIN);
 /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
          THIS PROGRAM IS DESIGNED TO PICK A LIKELY HEAVY ATOM PEAK
      FROM A LIST OF PATTERSON PEAKS, PREFERABLY NOT ON A HARKER LINE
      OR PLANE, WITH COORDINATES UH, VH, WH.  IT THEN SEARCHES THE REST
      OF THE LIST.  FOR EACH PEAK IN THE LIST, WITH COORDINATES U, V, W,
      THERE MUST ALSO BE A PEAK AT (U-UH,V-VH,W-WH) OR, IF NOT, AT MOST
      1 GRID PCINT AWAY IN ANY DIRECTION.  IF THERE IS SUCH A PEAK,
      U,V,W, IS RETAINED AND GIVEN THE VALUE SQRT(P(U,V,W)*P(U-UH,V-VH,
      W-WH)),  WEIGHTED IF THE PEAK IS NOT RIGHT ON.
          THE SECOND PART OF THE PROGRAM MAKES USE OF THE LIST OF PEAKS
      RETAINED BY PART ONE.  FOR EVERY PEAK IN ELECTRON-DENSITY SPACE
      AT (X,Y,Z) THERE SHOULD ALSO BE A PEAK AT (PX*X+QX, PY*Y+QY,
      PZ*Z+QZ) WHERE PX,PY,PZ = 1,-1 AND QX,QY,QZ = 0,1/2,  THE COORDIN-
      ATES OF PATTERSON SPACE (U,V,W) ARE RELATED TO THOSE OF ELECTRON-
      DENSITY SPACE (X,Y,Z) BY THE RELATION: (X,Y,Z)=(U-TX,V-TY,W-TZ).
      THE ELECTRON-DENSITY COORDINATE OF THE SYMMETRY-RELATED PEAK IS
      THUS GIVEN BY: (PX(U-TX)+QX,PY(V-TY)+QY,PZ(W-TZ)+QZ) AND THE
      PATTERSON COORDINATES OF THIS PEAK ARE: U'=PX*U+QX+(1-PX)*TX,
      V'=PY*V+QY+(1-PY)*TY, W'=PZ*W+QZ+(1-PZ)*TZ.  PART TWO SEARCHES
      THE LIST OF RETAINED PEAKS FOR THOSE PEAKS WHICH OCCUR BOTH AT
      U,V,W, AND AT U',V',W' FOR VARIOUS VALUES OF TX,TY,TZ, EXCLUDING
      TX,TY,TZ=0.  FOR EACH VALUE OF TX,TY,TZ BOTH THE NUMBER OF
      COINCIDENCES AND THE AVERAGE PEAK HEIGHT IS COMPUTED, AND THOSE
      TEN SETS OF TX,TY,TZ GIVING THE LARGEST VALUE FOR EACH OF THESE
      TWO QUANTITIES ARE OUTPUT.
          THIS PROGRAM WAS WRITTEN IN FEBRUARY, 1972 BY R.K.WISMER.

  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 CARD COLUMN FCRMAT VARIABLE  EXPLANATION
  1     1-80  A(80)   TITLE   ANY ALPHANUMERIC TITLE.
  2     1- 4  F(4)   #PEAKS   NUMBER OF PEAKS TO BE INPUT.
        5- 8  F(4)    IACR    NUMBER OF GRIDS ACROSS A FULL LINE.
        9-12  F(4)    IDWN    NUMBER OF LINES DOWN A FULL LAYER.
       13-16  F(4)    ILAY    NUMBER OF LAYERS IN A FULL MAP.
       17-20  F(4)   #SYMM    NUMBER OF SYMMETRY CARDS TO FOLLOW.
 3,ETC. 1- 4  F(4)    PX      =1,-1
        5- 8  F(4)    XQ      =0,1. QX = XQ/2
        9-12  F(4)    PY      =1,-1
       13-16  F(4)    YQ      =0,1. QY = YQ/2
       17-20  F(4)    PZ      =1,-1
       21-24  F(4)    ZQ      =0,1. QZ = ZQ/2
  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
          DECLARE   TITLE                CHARACTER (80);
```

```
         DECLARE   STAMP                    CHARACTER (54) EXTERNAL;        ALP00460
         DECLARE 1 SYMMPQ (#SYMM)           CONTROLLED EXTERNAL,            ALP00470
                   2 (PX,QX,PY,QY,PZ,QZ)         FIXED BINARY (15);         ALP00480
         DECLARE  (#PEAKS,#SYMM,#RTND,IACR,IDWN,ILAY)                       ALP00490
                                            FIXED BINARY (15);              ALP00500
         DECLARE  (SGNL, NCNT)              BIT (1) EXTERNAL;               ALP00510
/*                                                                          ALP00520
         SGNL='1'B IF THERE IS A SYMMETRY ELEMENT WITH PX=PY=PZ=-1.         ALP00530
         NCNT='1'B IF THERE ARE SYMMETRY ELEMENTS WITH ONLY:  PX=-1,        ALP00540
                   PY=-1, PZ=-1, PX=PY=-1, PY=PZ=-1, PZ=PX=-1.              ALP00550
                                                                       */   ALP00560
         DECLARE  (ITXYZ,LNG,IERR)          FIXED BINARY (31);              ALP00570
         DECLARE   IHESRTA   ENTRY          (CHAR(53), CHAR(39),            ALP00580
                                             FIXED BIN(31), FIXED BIN(31)); ALP00590
         DECLARE   SFLD                     CHARACTER (53),                 ALP00600
                   RFLD                     CHARACTER (39);                 ALP00610
         DECLARE   SORTIN                   FILE RECORD SEQUENTIAL;         ALP00620
         DECLARE   SDBG                     CHARACTER (100) VARYING;        ALP00630
         DECLARE   DBG                      EXTERNAL BIT (1);               ALP00640
         DECLARE   DNV                      BIT (1) EXTERNAL;               ALP00650
         IF (LENGTH(SDBG) > 1)  THEN DNV = '1'B;      ELSE DNV = '0'B;      ALP00660
         IF (LENGTH(SDBG) = 0)  THEN DBG = '0'B;      ELSE DBG = '1'B;      ALP00670
         CALL TM_DT;                                                        ALP00680
         LNG = 24000;                                                       ALP00690
         RFLD=' RECORD TYPE=V,LENGTH=(24,24,24,24,24) ';                    ALP00700
         ON SUBSCRIPTRANGE SNAP BEGIN; PUT DATA (I);  EXIT;  END;           ALP00710
/*                                                                          ALP00720
         READ AND INTERPRET THE CONTROL CARDS                               ALP00730
                                                                       */   ALP00740
         IF (DBG)  THEN PUT EDIT ('DEBUGGING OUTPUT WILL BE PRINTED.')      ALP00750
                                 (SKIP(4),A);                   ELSE;       ALP00760
         GET EDIT (TITLE,#PEAKS,IACR,IDWN,ILAY,#SYMM)                       ALP00770
                  (A(80),5 F(4));                                           ALP00780
         PUT EDIT (TITLE, STAMP,                                            ALP00790
                   'NUMBER OF POINTS ACROSS A FULL LINE =',IACR,            ALP00800
                   'NUMBER OF LINES DOWN A FULL LAYER =',IDWN,              ALP00810
                   'NUMBER OF LAYERS IN A FULL MAP =',ILAY,                 ALP00820
                   'NUMBER OF PEAKS IN INPUT DATA SET =',#PEAKS,            ALP00830
                   'NUMBER OF SYMMETRY CARDS TO BE READ =',#SYMM)           ALP00840
                  (A(80),SKIP(3),A(54),5 (SKIP(3),X(20),A,F(4)));           ALP00850
         ALLOCATE SYMMPQ;                                                   ALP00860
SLUP:    DO I = 1 TO #SYMM BY 1;                                            ALP00870
         GET EDIT (PX(I),QX(I),PY(I),QY(I),PZ(I),QZ(I))                     ALP00880
                  (SKIP,6 F(4));                                            ALP00890
         PUT EDIT ('SYMMETRY OPERATION',I,PX(I),'*X +',QX(I),'/2',          ALP00900
                   PY(I),'*Y +',QY(I),'/2',PZ(I),'*Z +',QZ(I),'/2')         ALP00910
                  (SKIP(2),X(25),A,F(3),3 (F(7),A,F(2),A));                 ALP00920
         END SLUP;                                                          ALP00930
         GET SKIP;                                                          ALP00940
/*                                                                          ALP00950
```

```
          CLOSE FILE (SYSIN);                                                  ALP00960
          PICK AN APPROPRIATE HEAVY-ATOM PEAK & TEST ALL OTHER PEAKS           ALP00970
                                                                           */  ALP00980
          CALL PART1 (#PEAKS,IACR,IDWN,ILAY,#RTND);                            ALP00990
 /*                                                                            ALP01000
          USING SYMMETRY, FIND ALL POSSIBLE VALUES OF TX, TY, TZ               ALP01010
                                                                           */  ALP01020
          CALL PART2 (#RTND,IACR,IDWN,ILAY,#SYMM);                             ALP01030
          IF (¬SGNL)  THEN GO TO PT4;                                   ELSE;  ALP01040
 /*                                                                            ALP01050
 /*                                                                            ALP01060
          GROUP ALL TX, TY, TZ AND OUTPUT THOSE WITH FREQUENCY > 5             ALP01070
                                                                           */  ALP01080
 PT3:     OPEN FILE (SORTIN) OUTPUT;                                           ALP01090
          CALL PART3 (IACR,IDWN,ILAY,#RTND);                                   ALP01100
          IF (¬NCNT)  THEN GO TO SRT;                                   ELSE;  ALP01110
 /*                                                                            ALP01120
          PICK ALL NON-INVERSION TX,TY,TZ > 5 IN # AND OUTPUT W/ SYM#          ALP01130
                                                                           */  ALP01140
 PT4:     CALL PART4 (IACR, IDWN, ILAY);                                       ALP01150
 /*                                                                            ALP01160
          SORT THE FILE OF TX,TY,TZ BY TX, TY, TZ & SYM#                       ALP01170
                                                                           */  ALP01180
 SRT:     CLOSE FILE (SORTIN);                                                 ALP01190
          SFLD=' SORT FIELDS=(11,02,FI,D,21,4,FL,D),SIZE=E7000             ';  ALP01200
          CALL IHESRTA (SFLD,RFLD,LNG,IERR);                                   ALP01210
          IF (IERR=0)  THEN GO TO PT5;                                  ELSE;  ALP01220
          PUT EDIT ('THE SECOND SORT WAS UNSUCCESSFUL -- RETRY')               ALP01230
                   (SKIP(2),A);                                                ALP01240
          EXIT;                                                                ALP01250
 /*                                                                            ALP01260
          MAP THE SYMMETRY ELEMENTS INTO EACH OTHER                            ALP01270
                                                                           */  ALP01280
 PT5:     CALL PART5 (IACR, IDWN, ILAY);                                       ALP01290
 /*                                                                            ALP01300
          CHOOSE THE MOST FREQUENT VALUE OF TX, TY, AND TZ                     ALP01310
                                                                           */  ALP01320
          CALL PART6 (IACR, IDWN, ILAY);                                       ALP01330
 /*                                                                            ALP01340
          DETERMINE WHICH PEAKS ARE TO BE KEPT                                 ALP01350
                                                                           */  ALP01360
          CALL PART7 (IACR, IDWN, ILAY, #RTND, #SYMM);                         ALP01370
          END ALOP;                                                            ALP01380


* PROCESS('ATR,XREF,NOSTMT,OPT=02');                                           TMDT0010
 /* CREATES A TIME - DATE STAMP FOR FUTURE USE                             */  TMDT0020
 (SUBRG,STRG):  TM_DT:  PROCEDURE;                                             TMDT0030
 DECLARE STAMP CHARACTER(54) EXTERNAL,   AMPM CHARACTER(4);                 TMDT0040
 DECLARE MOS(12) CHARACTER(9) VARYING INITIAL ('JANUARY','FEBRUARY',        TMDT0050
         'MARCH','APRIL','MAY','JUNE','JULY','AUGUST','SEPTEMBER',          TMDT0060
         'OCTOBER','NOVEMBER','DECEMBER');                                  TMDT0070
 DECLARE TM CHARACTER(9), TMS(2) CHARACTER(2) DEFINED TM;  TM = TIME;       TMDT0080
 DECLARE DT CHARACTER(6), DS (3) CHARACTER(2) DEFINED DT;  DT = DATE;       TMDT0090
 IF (TMS(1) > 11)  THEN AMPM = 'P.M.';   ELSE AMPM = 'A.M.';                TMDT0100
 STAMP = 'RUN ON ' || MOS(DS(2)) || ' ' || DS(3) || ', 19' || DS(1) ||      TMDT0110
     ', AT ' || TMS(1) || ' HOURS ' || TMS(2)  || ' MINUTES ' || AMPM;      TMDT0120
 END TM_DT;                                                                 TMDT0130


* PROCESS('ATR,XREF,NOSTMT,OPT=02');                                        -P1-0010
 /*   PICK AN APPROPRIATE HEAVY-ATOM PEAK & TEST ALL OTHER PEAKS         */ -P1-0020
 /*   IN-CORE HEAVY-ATOM SINGLE SUPERPOSITION                            */ -P1-0030
 PART1:   PROCEDURE (#PEAKS,IACR,IDWN,ILAY,#RTND);                          -P1-0040
          DECLARE  (#PEAKS,IACR,IDWN,ILAY,#RTND)  FIXED BINARY (15);        -P1-0050
          DECLARE 1 PEAKS (#PEAKS)        CONTROLLED,                       -P1-0060
                    2 (U,V,W,VALU,MEAN)   FIXED BINARY (15);                -P1-0070
          DECLARE   PKFILE                FILE RECORD INPUT SEQUENTIAL;     -P1-0080
          DECLARE   RETAIN                FILE RECORD SEQUENTIAL;           -P1-0090
          DECLARE 1 PEAKR,                                                  -P1-0100
                    2 (RU,RV,RW,VAL,AVG)      FIXED BINARY (31);            -P1-0110
          DECLARE 1 SPEAK,                                                  -P1-0120
                    2 (SU,SV,SW,VLU)      FIXED BINARY (31);                -P1-0130
          DECLARE   DIF                   LABEL;                            -P1-0140
          DECLARE  (BU, BV, BW)           BIT (1);                          -P1-0150
          DECLARE  (DELU,DELV,DELW)       FIXED BINARY (15);                -P1-0160
          DECLARE  (P1, P2, P3)           FLOAT BINARY;                     -P1-0170
          DECLARE  (IDU,IDV,IDW)          FIXED BINARY (15);                -P1-0180
          DECLARE   DBG                   EXTERNAL BIT (1);                 -P1-0190
          ON SUBSCRIPTRANGE SNAP BEGIN; PUT DATA (I,J,K,L); EXIT; END;      -P1-0200
 /*                                                                         -P1-0210
          SET UP CONSTANTS FOR CHECKING                                     -P1-0220
                                                                         */ -P1-0230
          IA2 = IACR / 2;                                                   -P1-0240
          ID2 = IDWN / 2;                                                   -P1-0250
          IL2 = ILAY / 2;                                                   -P1-0260
          P1 = 0.800;                                                       -P1-0270
          P2 = 0.650;                                                       -P1-0280
          P3 = 0.500;                                                       -P1-0290
 /*                                                                         -P1-0300
          READ IN THE PEAKS PICKED FROM THE MAP BY ALPPKR                   -P1-0310
                                                                         */ -P1-0320
          ALLOCATE PEAKS;                                                   -P1-0330
 PLUP:    DO I = 1 TO #PEAKS BY 1;                                          -P1-0340
          GET EDIT (SPEAK) (4 F(5));                                        -P1-0350
          U(I) = SU;                                                        -P1-0360
         V(I) = SV;
         W(I) = SW;
         VALU(I) = VLU;
         MEAN(I) = 0;
         END PLUP;
/*
         FIND THE LIKELY HEAVY ATOM PEAK FROM THE FIRST 20%
                                                                   */

         IF (#PEAKS<100)  THEN ITOP = MIN(20,#PEAKS);
                          ELSE ITOP = #PEAKS / 5;
         IA2 = IACR / 2;
         ID2 = IDWN / 2;
         IL2 = ILAY / 2;
         DIF = HARD;
         ITIME = 1;
         JTIME = 0;
TLUP:    DO I = 1 TO ITOP BY 1;
         IU = U(I);
         IV = V(I);
         IW = W(I);
         KVAL = VALU(I);
         GO TO DIF;
/*
         LARGEST PEAK ACCEPTED AS HEAVY ATOM WHICH HASN'T U,V,W=0,1/2
                                                                   */
HARD:    IF ((IU=0)|(IU=IACR)|(IU=IA2)|(IV=0)|(IV=IDWN)|(IV=ID2)|
             (IW=0)|(IW=ILAY)|(IW=IL2))  THEN GO TO NOPE;
                                         ELSE GO TO YEP;
/*
         LARGEST ACCEPTED AS HEAVY ATOM HAVING ONLY ONE OF U,V,W=0,1/2
                                                                   */
MEDM:    BU = ¬((IU=0)|(IU=IACR)|(IU=IA2));
         BV = ¬((IV=0)|(IV=IDWN)|(IV=ID2));
         BW = ¬((IW=0)|(IW=ILAY)|(IW=IL2));
         IF ((BU&BV) | (BU&BW) | (BV&BW))  THEN GO TO YEP;
                                           ELSE GO TO NOPE;
/*
         LARGEST ACCEPTED AS HEAVY ATOM HAVING TWO ONLY OF U,V,W=0,1/2
                                                                   */
EASY:    BU = ¬((IU=0)|(IU=IACR)|(IU=IA2));
         BV = ¬((IV=0)|(IV=IDWN)|(IV=ID2));
         BW = ¬((IW=0)|(IW=ILAY)|(IW=IL2));
         IF (BU | BV | BW)  THEN GO TO YEP;
                            ELSE GO TO NOPE;
NOPE:    END TLUP;
         IF (ITIME=1)  THEN GO TO FIRST;
                       ELSE IF (ITIME=2)  THEN GO TO SCND;
                                          ELSE GO TO FAIL;
FIRST:   DIF = MEDM;
         ITIME = 2;
         GO TC TLUP;                                                    -P1-0870
SCND:    DIF = EASY;                                                    -P1-0880
         ITIME = 3;                                                     -P1-0890
         GO TC TLUP;                                                    -P1-0900
/*                                                                      -P1-0910
         UNABLE TO FIND A SUITABLE HEAVY ATOM                           -P1-0920
                                                                     */ -P1-0930
FAIL:    PUT EDIT ('***** UNABLE TO PICK A HEAVY-ATOM PEAK *****')      -P1-0940
                  (SKIP(4),A);                                          -P1-0950
         EXIT;                                                          -P1-0960
/*                                                                      -P1-0970
         SUITABLE HEAVY ATOM PEAK FOUND                                 -P1-0980
                                                                     */ -P1-0990
YEP:     KU = IU;                                                       -P1-1000
         KV = IV;                                                       -P1-1010
         KW = IW;                                                       -P1-1020
         KP = I;                                                        -P1-1030
         PUT EDIT ('HEAVY ATOM PEAK FOUND AT U =',KU,'V =',KV,'W =',    -P1-1040
                   KW,'VALUE =',KVAL)                                   -P1-1050
                  (SKIP(3),X(20), 4 (A,F(4),X(3)));                     -P1-1060
/*                                                                      -P1-1070
         DECICE ON WHETHER TO USE A HARKER OR A GENERAL PEAK            -P1-1080
                                                                     */ -P1-1090
         IF ((ITIME=1) & (JTIME=0))  THEN GO TO TAGN;           ELSE;   -P1-1100
         IF ((ITIME¬=1) & (JTIME=1))  THEN GO TO COMPR;         ELSE;   -P1-1110
         GO TO RSTR;                                                    -P1-1120
TAGN:    GKP = KP;           GKVL = KVAL;           JTIME = 1;          -P1-1130
         GKU = KU;           GKV = KV;              GKW = KW;           -P1-1140
         GO TO FIRST;                                                   -P1-1150
COMPR:   GNRL = FLOAT (GKVL);                                           -P1-1160
         HRKR = FLOAT (KVAL) / 2.2000;                                  -P1-1170
         IF (HRKR > GNRL)  THEN GO TO TYPE;                     ELSE;   -P1-1180
         KU = GKU;           KV = GKV;              KW = GKW;           -P1-1190
         KP = GKP;           KVAL = GKVL;                               -P1-1200
TYPE:    PUT EDIT ('##### PEAK FINALLY CHOSEN AT:   #####',             -P1-1210
                   'U =',KU,'V =',KV,'W =',KW,'VALUE =',KVAL,           -P1-1220
                  'INDEX =',KP) (SKIP(2),X(10),A,SKIP,3 (X(20),A,F(5)),-P1-1230
                   SKIP,2 (X(20),A,F(5)));                              -P1-1240
/*                                                                      -P1-1250
         USING THIS PEAK, DETERMINE WHICH PEAKS TO RETAIN               -P1-1260
                                                                     */ -P1-1270
RSTR:    PUT EDIT ('PEAKS FOUND, USING HEAVY-ATOM PEAK',                -P1-1280
                   '    U    V    W   HEIGHT     MEAN',                 -P1-1290
                   '    U    V    W   HEIGHT     MEAN',                 -P1-1300
                   '    U    V    W   HEIGHT     MEAN',                 -P1-1310
                   '    U    V    W   HEIGHT     MEAN',' ')             -P1-1320
                  (PAGE,A,SKIP,A,A,A,A,SKIP,A);                         -P1-1330
         PUT SKIP;                                                      -P1-1340
ILUP:    DO I = 1 TO #PEAKS BY 1;                                       -P1-1350
/*                                                                      -P1-1360
```

```
         CALCULATE THE PEAK POSITION TO LOOK FOR
                                                                    */

         K = I;
         IDU = KU + U(I);
         IDV = KV + V(I);
         IDW = KW + W(I);
         IF (IDU<0)  THEN IDU = IDU + IACR;
                     ELSE IF (IDU>IACR)  THEN IDU = IDU - IACR; ELSE;
         IF (IDU<0)  THEN IDU = IDU + IACR;
                     ELSE IF (IDU>IACR)  THEN IDU = IDU - IACR; ELSE;
         IF (IDV<0)  THEN IDV = IDV + IDWN;
                     ELSE IF (IDV>IDWN)  THEN IDV = IDV - IDWN; ELSE;
         IF (IDV<0)  THEN IDV = IDV + IDWN;
                     ELSE IF (IDV>IDWN)  THEN IDV = IDV - IDWN; ELSE;
         IF (IDW<0)  THEN IDW = IDW + ILAY;
                     ELSE IF (IDW>ILAY)  THEN IDW = IDW - ILAY; ELSE;
         IF (IDW<0)  THEN IDW = IDW + ILAY;
                     ELSE IF (IDW>ILAY)  THEN IDW = IDW - ILAY; ELSE;

/*
         DETERMINE IF ANY OF THE OTHER PEAKS FITS EXACTLY
         TO THIS CALCULATED POSITION, OR IF IT IS OFF ONLY SLIGHTLY
                                                                    */
JLUP:    DO J = 1 TO #PEAKS BY 1;
         L = J;
         DELU = U(J) - IDU;
         DELV = V(J) - IDV;
         DELW = W(J) - IDW;
         IF (DELU >  IA2)  THEN DELU = DELU - IACR;                ELSE;
         IF (DELU >  IA2)  THEN DELU = DELU - IACR;                ELSE;
         IF (DELU < -IA2)  THEN DELU = DELU + IACR;                ELSE;
         IF (DELU < -IA2)  THEN DELU = DELU + IACR;                ELSE;
         IF (DELV >  ID2)  THEN DELV = DELV - IDWN;                ELSE;
         IF (DELV >  ID2)  THEN DELV = DELV - IDWN;                ELSE;
         IF (DELV < -ID2)  THEN DELV = DELV + IDWN;                ELSE;
         IF (DELV < -ID2)  THEN DELV = DELV + IDWN;                ELSE;
         IF (DELW >  IL2)  THEN DELW = DELW - ILAY;                ELSE;
         IF (DELW >  IL2)  THEN DELW = DELW - ILAY;                ELSE;
         IF (DELW < -IL2)  THEN DELW = DELW + ILAY;                ELSE;
         IF (DELW < -IL2)  THEN DELW = DELW + ILAY;                ELSE;
         IF ((DELU=0) & (DELV=0) & (DELW=0))  THEN GO TO OFF0;     ELSE;
         DELU = ABS (DELU);
         DELV = ABS (DELV);
         DELW = ABS (DELW);
         IF(((DELU=1) & (DELV=0) & (DELW=0)) |
            ((DELU=0) & (DELV=1) & (DELW=0)) |
            ((DELU=0) & (DELV=0) & (DELW=1)))  THEN GO TO OFF1; ELSE;
         IF(((DELU=1) & (DELV=1) & (DELW=0)) |
            ((DELU=0) & (DELV=1) & (DELW=1)) |
            ((DELU=1) & (DELV=0) & (DELW=1)))  THEN GO TO OFF2; ELSE;
         IF ((DELU=1) & (DELV=1) & (DELW=1))  THEN GO TO OFF3;     ELSE;
```

```
        END JLUP;
        GO TC NDI;
/*
        PEAK IS RIGHT ON
                                                                    */
OFFO:   MEAN(J) = SQRT (FLOAT (VALU(I) * VALU(J)));
        PUT EDIT (U(J),V(J),W(J),VALU(J),MEAN(J)) (3 F(5),2 F(9));
        GO TO NDI;
/*
        PEAK IS OFF 1 GRID IN ONE DIRECTION
                                                                    */
OFF1:   MEAN(J) = P1 * SQRT (FLOAT (VALU(I) * VALU(J)));
        PUT EDIT (U(J),V(J),W(J),VALU(J),MEAN(J)) (3 F(5),2 F(9));
        GO TC NDI;
/*
        THE PEAK IS OFF 1 GRID POINT IN EACH OF TWO DIRECTIONS
                                                                    */
OFF2:   MEAN(J) = P2 * SQRT (FLOAT (VALU(I) * VALU(J)));
        PUT EDIT (U(J),V(J),W(J),VALU(J),MEAN(J)) (3 F(5),2 F(9));
        GO TO NDI;
/*
        THE PEAK IS OFF 1 GRID POINT IN ALL THREE DIRECTIONS
                                                                    */
OFF3:   MEAN(J) = P3 * SQRT (FLOAT (VALU(I) * VALU(J)));
        PUT EDIT (U(J),V(J),W(J),VALU(J),MEAN(J)) (3 F(5),2 F(9));
NDI:    END ILUP;
/*
        WRITE OUT THE PEAKS WHICH WERE FOUND
                                                                    */
        OPEN FILE (RETAIN) OUTPUT;
        #RTND = 0;
LUPI:   DO I = 1 TO #PEAKS BY 1;
        IF (MEAN(I)=0)  THEN GO TO FNLI;                         ELSE;
        RU = U(I);
        RV = V(I);
        RW = W(I);
        VAL = VALU(I);
        AVG = MEAN(I);
        #RTND = #RTND + 1;
        WRITE FILE (RETAIN) FROM (PEAKR);
FNLI:   END LUPI;
        CLOSE FILE (RETAIN);
        FREE PEAKS;
        RETURN;
        END PART1;


* PROCESS('ATR,XREF,NOSTMT,OPT=02');
```

```
/*   USING SYMMETRY, FIND ALL THE VALUES OF TX, TY, TZ                   */ -P2-0020
/*   FROM THE PEAKS OBTAINED FROM THE SUPERPOSITION OF PART 1             */ -P2-0030
PART2:   PROCEDURE (#RTND,IACR,IDWN,ILAY,#SYMM);                              -P2-0040
/*                                                                            -P2-0050
         THIS SECTION COMPUTES ALL OF THE POSSIBLE VALUES OF TX,TY, &         -P2-0060
         TZ FROM THE PEAKS LEFT BY THE SUPERPOSITION AS FOLLOWS:              -P2-0070
         LET U,V,W & U',V',W' REPRESENT THE TWO PEAKS.  IF THESE PEAKS        -P2-0080
         ARE SYMMETRY RELATED BUT THE SUPERPOSITION ORIGIN IS OFF             -P2-0090
         BY TX,TY,TZ, THEN U = X+TX, V = Y+TY, AND W = Z+TZ.  DEPENDING-P2-0100
         ON THE VALUES OF PX,PY,PZ THERE ARE THEN 18 DIFFERENT COMBINA--P2-0110
         TIONS OF THE TWO PEAKS (DESIGNATED AS P1 AND P2):                    -P2-0120
                                                                              -P2-0130
 QUANTITY     U OF QUANTITY        V OF QUANTITY        W OF QUANTITY         -P2-0140
----------   ---------------      ---------------      ---------------        -P2-0150
    P1          X + TX               Y + TY               Z + TZ              -P2-0160
    P2(0) ---   QX + X + TX          QY + Y + TY          QZ + Z + TZ         -P2-0170
 P2(0)+P1       QX + 2X + 2TX        QY + 2Y + 2TY        QZ + 2Z + TZ        -P2-0180
 P2(0)-P1       QX                   QY                   QZ                  -P2-0190
    P2(1) ---   QX - X + TX          QY + Y + TY          QZ + Z + TZ         -P2-0200
 P2(1)+P1     * QX + 2TX             QY + 2Y + 2TY        QZ + 2Z + TZ        -P2-0210
 P2(1)-P1       QX - 2X            # QY                 # QZ                  -P2-0220
    P2(2) ---   QX + X + TX          QY - Y + TY          QZ + Z + TZ         -P2-0230
 P2(2)+P1       QX + 2X + 2TX      * QY + 2TY             QZ + 2Z + 2TZ       -P2-0240
 P2(2)-P1     # QX                   QY - 2Y            # QZ                  -P2-0250
    P2(3) ---   QX - X + TX          QY - Y + TY          QZ + Z + TZ         -P2-0260
 P2(3)+P1     * QX + 2TX           * QY + 2TY             QZ + 2Z + 2TZ       -P2-0270
 P2(3)-P1       QX - 2X              QY - 2Y            # QZ                  -P2-0280
    P2(4) ---   QX + X + TX          QY + Y + TY          QZ - Z + TZ         -P2-0290
 P2(4)+P1       QX + 2X + 2TX        QY + 2Y + 2TY      * QZ + 2TZ            -P2-0300
 P2(4)-P1     # QX                 # QY                   QZ - 2Z             -P2-0310
    P2(5) ---   QX - X + TX          QY + Y + TY          QZ - Z + TZ         -P2-0320
 P2(5)+P1     * QX + 2TX             QY + 2Y + 2TY      * QZ + 2TZ            -P2-0330
 P2(5)-P1       QX - 2X            # QY                   QZ - 2Z             -P2-0340
    P2(6) ---   QX + X + TX          QY - Y + TY          QZ - Z + TZ         -P2-0350
 P2(6)+P1       QX + 2X + 2TX      * QY + 2TY           * QZ + 2TZ            -P2-0360
 P2(6)-P1     # QX                   QY - 2Y              QZ - 2Z             -P2-0370
    P2(7) ---   QX - X + TX          QY - Y + TY          QZ - Z + TZ         -P2-0380
 P2(7)+P1     * QX + 2TX           * QY + 2TY           * QZ + 2TZ            -P2-0390
 P2(7)-P1       QX - 2X              QY - 2Y              QZ - 2Z             -P2-0400
                                                                              -P2-0410
         AS QX,QY,QZ ARE EITHER 0 OR 1/2 FOR ANY SYMMETRY ELEMENT AND         -P2-0420
         ARE KNOWN, VALUES OF  TX; TY; TX,TY; TZ; TX,TZ; TY,TZ OR             -P2-0430
         TX,TY,TZ ARE THEN COMPUTED.  FOR ALL BUT THE INVERSION, THE          -P2-0440
         FREQUENCY OF OCCURRENCE OF THESE VALUES IS STORED IN CORE.           -P2-0450
         FOR THE INVERSION, THE TX,TY,TZ TRIPLES ARE WRITTEN OUT,             -P2-0460
         SORTED INTO ORDER, AND THEN GROUPED TOGETHER ON REAPING IN            -P2-0470
         PART 3.  A LATITUDE OF ONE GRID POINT IN EACH DIRECTION IS           -P2-0480
         ALLOWED FOR THE VALUES OF TX, TY, TZ AND APPROPRIATE WEIGHTS         -P2-0490
         ASSIGNED.  THE ASTERISKED (*) RELATIONS ABOVE GIVE VALUES            -P2-0500
         OF TX,TY,TZ, PROVIDED THE NUMBERED (#) RELATIONS ARE OBEYED.         -P2-0510
```

```
                                                                    */
          DECLARE (#RTND,IACR,IDWN,ILAY,#SYMM)    FIXED BINARY (15);
          DECLARE   IS (#SYMM)            FIXED BINARY (15) CONTROLLED;
          DECLARE 1 PEAKS (#RTND)         CONTROLLED EXTERNAL,
                  2 (U,V,W,VALU,MEAN) FIXED BINARY (15);
          DECLARE 1 SYMMPQ (#SYMM)        CONTROLLED EXTERNAL,
                  2 (PX,QX,PY,QY,PZ,QZ)     FIXED BINARY (15);
          DECLARE   RETAIN                FILE RECORD SEQUENTIAL;
          DECLARE 1 PEAKR,
                  2 (RU,RV,RW,VAL,AVG)      FIXED BINARY (31);
          DECLARE 1 TXYZ,
                  2 (TX, TY, TZ, KS)  FIXED BINARY (15),
                  2 (VIJ, MIJ, FREQ)  FLOAT BINARY;
          DECLARE   SYM(7)                LABEL;
          DECLARE   ITXYZ                 FIXED BINARY (31);
          DECLARE   (XLINE#(O:IACR), YLINE#(O:IDWN), ZLINE#(O:ILAY),
               XYLAY#(O:IA,O:ID), YZLAY#(O:ID,O:IL), ZXLAY#(O:IL,O:IA))
                                       EXTERNAL CONTROLLED FLOAT BINARY;
/*     THE SEVEN-MEMBERED BIT(1) ARRAY "ION" SHOWS THE TYPE OF SYMMETRY
       ELEMENTS BEING USED.  THE BIT IS SET IF THE SYMMETRY ELEMENT IS
       PRESENT.
       ELEMENT               NO     PX     PY     PZ   DETERMINE TX,TY,TZ?
       PLANE PERP. TO X       1     -1      1      1   TX
       PLANE PERP. TO Y       2      1     -1      1   TY
       AXIS IN Z              3     -1     -1      1   TX, TY
       PLANE PERP. TO Z       4      1      1     -1   TZ
       AXIS IN Y              5     -1      1     -1   TX, TZ
       AXIS IN X              6      1     -1     -1   TY, TZ
       INVERSION              7     -1     -1     -1   TX, TY, TZ
       SGNL='1'B IF THERE IS AN INVERSION;  NCNT='1'B IF THERE ARE OTHERS
                                                                    */
          DECLARE   (ION(7),SGNL,NCNT)    BIT (1) EXTERNAL;
          DECLARE    AL(7)                LABEL;
          DECLARE    DBG                  EXTERNAL BIT (1);
          DECLARE   (IA2,ID2,IL2,I4A,I4D,I4L,JPIU,JPIV,JPIW,JMIU,JMIV,
                     JMIW,JMIUP,JMIVP,JMIWP,JMIUM,JMIVM,JMIWM)
                                          FIXED BINARY (15);
          DECLARE    LAY# (O:ILAY)        EXTERNAL CTL BIT(1);
          DECLARE   (XQ, YQ, ZQ)          EXTERNAL FIXED BIN;
          IA2 = IACR / 2;
          ID2 = IDWN / 2;
          IL2 = ILAY / 2;
          I4A = IACR * 4;
          I4D = IDWN * 4;
          I4L = ILAY * 4;
/*
          I1 : MIRRORS AND PLANES RIGHT ON (TX|TY|TZ).
          I2 : TWO-FOLDS OR SCREWS RIGHT ON (TX,TY|TY,TZ|TX,TZ).
          I3 : MIRRORS AND PLANES OFF ONE GRID POINT.
          I4 : TWO-FOLDS OR SCREWS OFF ONE IN ONE DIRECTION.
```

```
         I6 : TWO-FOLDS OR SCREWS OFF ONE IN BOTH DIRECTIONS.
                                                                   */
         DECLARE  (I1,I2,I3,I4,I5,I6,TH)          FLOAT BINARY;
         DECLARE  (J1,J2,J3,J4,J5,J6,MEANIJ)      FLOAT BINARY;
         TH = 10.000;                        J1 = 18;
         J2 = 10;                            J3 = 9;
         J4 = 6;                             J5 = 4;
         J6 = 2;
         ON SUBSCRIPTRANGE SNAP BEGIN; PUT DATA
                  (I,KS,J,K,TX,TY,TZ);            EXIT;          END;
/*
         READ IN THE PEAKS, SORTED BY LAYER
                                                                   */
         OPEN FILE (RETAIN) INPUT;
         ALLOCATE  PEAKS, IS;
ILUP:    DO I = 1 TO #RTND BY 1;
         READ FILE (RETAIN) INTO (PEAKR);
         U(I) = RU;
         V(I) = RV;
         W(I) = RW;
         VALU(I) = VAL;
         MEAN(I) = AVG;
         END ILUP;
         CLOSE FILE (RETAIN);
         ION = '0'B;
         NCNT = '0'B;
         IA = IACR;
         ID = IDWN;
         IL = ILAY;
/*
         TRANSFORM THE SYMMETRY ELEMENTS INTO MORE USEABLE FORM
                                                                   */
SLUP:    DO I = 1 TO #SYMM BY 1;
         IS(I) = 0;
         IF (PX(I) = -1)  THEN IS(I) = IS(I) + 1;                ELSE;
         IF (PY(I) = -1)  THEN IS(I) = IS(I) + 2;                ELSE;
         IF (PZ(I) = -1)  THEN IS(I) = IS(I) + 4;                ELSE;
         IF (QX(I) = 1)  THEN QX(I) = IACR / 2;                  ELSE;
         IF (QY(I) = 1)  THEN QY(I) = IDWN / 2;                  ELSE;
         IF (QZ(I) = 1)  THEN QZ(I) = ILAY / 2;                  ELSE;
         KS = IS(I);
         IF (KS = 7)  THEN DO;
         XQ = QX(I);
         YQ = QY(I);
         ZQ = QZ(I);                              END;           ELSE;
         ION(KS) = '1'B;
         END SLUP;
         PUT EDIT (#RTND,' PEAKS WERE RETAINED') (SKIP(2),F(5),A);
         IF (DBG)  THEN PUT EDIT('ION =',ION) (SKIP,A,7 (B(1),X(1)));
                   ELSE;
```

```
          IF (DBG)  THEN PUT PAGE;                            ELSE;
/*
          ALLOCATE ARRAYS FOR MIRRCRS, GLIDES, TWO-FOLDS, AND SCREWS
                                                                 */
ALLOC:    DO I = 1 TO 7;
          IF (ION(I))  THEN GO TO AL(I);        ELSE GO TO NDA;
AL(1):    ALLOCATE XLINE#;
          XLINE# = 0.000;
          NCNT = '1'B;
          GO TC NDA;
AL(2):    ALLOCATE YLINE#;
          YLINE# = 0.000;
          NCNT = '1'B;
          GO TC NDA;
AL(3):    ALLOCATE XYLAY#;
          XYLAY# = 0.000;
          NCNT = '1'B;
          GO TC NDA;
AL(4):    ALLOCATE ZLINE#;
          ZLINE# = 0.000;
          NCNT = '1'B;
          GO TO NDA;
AL(5):    ALLOCATE ZXLAY#;
          ZXLAY# = 0.000;
          NCNT = '1'B;
          GO TC NDA;
AL(6):    ALLOCATE YZLAY#;
          YZLAY# = 0.000;
          NCNT = '1'B;
          GO TC NDA;
AL(7):    ALLOCATE  LAY#;
          LAY# = '0'B;
NDA:      END ALLOC;
          SGNL = ION(7);
/*
          SEARCH THROUGH ALL PEAKS
                                                                 */
OUTER:    DO J = 1 TO #RTND BY 1;
          JU = U(J);
          JV = V(J);
          JW = W(J);                MEANJ = MEAN(J);
INNER:    DO I = J TO #RTND BY 1;   MEANIJ=FLOAT(MEANJ+MEAN(I))/TH;
          IU = U(I);         I1 = J1 * MEANIJ;   I2 = J2 * MEANIJ;
          IV = V(I);         I3 = J3 * MEANIJ;   I4 = J4 * MEANIJ;
          IW = W(I);         I5 = J5 * MEANIJ;   I6 = J6 * MEANIJ;
/*
          COMPUTE THE SUM (P) AND THE DIFFERENCE (M) BETWEEN PEAKS
          JU,JV,JW AND IU,IV,IW ALLOWING ONE GRID POINT POSITIVE (P)
          AND NEGATIVE (M) ERROR
                                                                 */
```

```
         JPIU = JU + IU;                                              -P2-2020
         JPIV = JV + IV;                                              -P2-2030
         JPIW = JW + IW;                                              -P2-2040
         JMIU = MOD (JU - IU + I4A, IACR);                            -P2-2050
         JMIV = MOD (JV - IV + I4D, IDWN);                            -P2-2060
         JMIW = MOD (JW - IW + I4L, ILAY);                            -P2-2070
         JMIUP = MOD (JMIU + 1 + I4A,  IACR);                         -P2-2080
         JMIUM = MOD (JMIU - 1 + I4A,  IACR);                         -P2-2090
         JMIVP = MOD (JMIV + 1 + I4D,  IDWN);                         -P2-2100
         JMIVM = MOD (JMIV - 1 + I4D,  IDWN);                         -P2-2110
         JMIWP = MOD (JMIW + 1 + I4L,  ILAY);                         -P2-2120
         JMIWM = MOD (JMIW - 1 + I4L,  ILAY);                         -P2-2130
/*                                                                    -P2-2140
         LOOP THROUGH ALL SYMMETRY ELEMENTS                           -P2-2150
                                                                   */ -P2-2160
SYML:    DO K = 1 TO #SYMM BY 1;                                      -P2-2170
         KS = IS (K);                                                 -P2-2180
         GO TC SYM(KS);                                               -P2-2190
/*                                                                    -P2-2200
         MIRRCR OR GLIDE PLANE PERPENDICULAR TO X; DETERMINE TX       -P2-2210
         NOTE P2(1) ABOVE THAT V2-V1 = QY+/-1 AND W2-W1 = QZ+/-1      -P2-2220
         BEFORE U2+U1 = QX+2TX HOLDS.                                 -P2-2230
                                                                   */ -P2-2240
SYM(1):  IF ((JMIVP ¬= QY(K)) & (JMIV ¬= QY(K)) & (JMIVM ¬= QY(K)))    -P2-2250
                        THEN GO TO NDSY;                       ELSE;  -P2-2260
         IF ((JMIWP ¬= QZ(K)) & (JMIW ¬= QZ(K)) & (JMIWM ¬= QZ(K)))    -P2-2270
                        THEN GO TO NDSY;                       ELSE;  -P2-2280
         TX = MOD (((JPIU - QX(K)) / 2) + I4A,  IACR);                -P2-2290
         XLINE#(TX) = XLINE#(TX) + I1;                                -P2-2300
         TX = MOD (TX + 1,  IACR);                                    -P2-2310
         XLINE#(TX) = XLINE#(TX) + I3;                                -P2-2320
         TX = MOD (TX - 2 + IACR,  IACR);                             -P2-2330
         XLINE#(TX) = XLINE#(TX) + I3;                                -P2-2340
         GO TC NDSY;                                                  -P2-2350
/*                                                                    -P2-2360
         MIRROR OR GLIDE PLANE PERPENDICULAR TO Y; DETERMINE TY       -P2-2370
         NOTE P2(2) ABOVE THAT U2-U1 = QX+/-1 AND W2-W1 = QZ+/-1      -P2-2380
         BEFORE V2+V1 = GY+2TY HOLDS.                                 -P2-2390
                                                                   */ -P2-2400
SYM(2):  IF ((JMIUP ¬= QX(K)) & (JMIU ¬= QX(K)) & (JMIUM ¬= QX(K)))    -P2-2410
                        THEN GO TO NDSY;                       ELSE;  -P2-2420
         IF ((JMIWP ¬= QZ(K)) & (JMIW ¬= QZ(K)) & (JMIWM ¬= QZ(K)))    -P2-2430
                        THEN GO TO NDSY;                       ELSE;  -P2-2440
         TY = MOD (((JPIV - QY(K)) / 2) + I4D,  IDWN);                -P2-2450
         YLINE#(TY) = YLINE#(TY) + I1;                                -P2-2460
         TY = MOD (TY + 1,  IDWN);                                    -P2-2470
         YLINE#(TY) = YLINE#(TY) + I3;                                -P2-2480
         TY = MOD (TY - 2 + IDWN,  IDWN);                             -P2-2490
         YLINE#(TY) = YLINE#(TY) + I3;                                -P2-2500
         GO TC NDSY;                                                  -P2-2510
```

```
/*
         TWO-FOLD OR TWO-FOLD SCREW AXIS IN Z; DETERMINE TX & TY
         NOTE P2(3) ABOVE THAT W2-W1 = QZ+/-1 BEFORE U2+U1 = QX+2TX
         AND V2+V1 = QY+2TY HCLDS.
                                                                  */
SYM(3):  IF ((JMIWP ¬= QZ(K)) & (JMIW ¬= QZ(K)) & (JMIWM ¬= QZ(K)))
                      THEN GO TO NDSY;                         ELSE;
         TX = MOD (((JPIU - QX(K)) / 2) + I4A,  IACR);
         TY = MOD (((JPIV - QY(K)) / 2) + I4D,  IDWN);
         XYLAY#(TX,TY) = XYLAY#(TX,TY) + I2;
         TXP = MOD (TX + 1, IACR);
         TYP = MOD (TY + 1, IDWN);
         TXM = MOD (TX - 1 + IACR, IACR);
         TYM = MOD (TY - 1 + IDWN, IDWN);
         XYLAY#(TX,TYP) = XYLAY#(TX,TYP) + I4;
         XYLAY#(TX,TYM) = XYLAY#(TX,TYM) + I4;
         XYLAY#(TXP,TY) = XYLAY#(TXP,TY) + I4;
         XYLAY#(TXM,TY) = XYLAY#(TXM,TY) + I4;
         XYLAY#(TXM,TYP) = XYLAY#(TXM,TYP) + I6;
         XYLAY#(TXM,TYM) = XYLAY#(TXM,TYM) + I6;
         XYLAY#(TXP,TYM) = XYLAY#(TXP,TYM) + I6;
         XYLAY#(TXP,TYP) = XYLAY#(TXP,TYP) + I6;
         GO TC NDSY;
/*
         MIRRCR OR GLIDE PLANE PERPENDICULAR TO Z; DETERMINE TZ
         NOTE P2(4) ABOVE THAT U2-U1 = QX+/-1 AND V2-V1 = QY+/-1
         BEFORE W2+W1 = QZ+2TZ HOLDS
                                                                  */
SYM(4):  IF ((JMIUP ¬= QX(K)) & (JMIU ¬= QX(K)) & (JMIUM ¬= QX(K)))
                      THEN GO TO NDSY;                         ELSE;
         IF ((JMIVP ¬= QY(K)) & (JMIV ¬= QY(K)) & (JMIVM ¬= QY(K)))
                      THEN GO TO NDSY;                         ELSE;
         TZ = MOD (((JPIW - QZ(K)) / 2) + I4L,  ILAY);
         ZLINE#(TZ) = ZLINE#(TZ) + I1;
         TZ = M3D (TZ + 1,  ILAY);
         ZLINE#(TZ) = ZLINE#(TZ) + I3;
         TZ = MOD (TZ - 2 + ILAY,  ILAY);
         ZLINE#(TZ) = ZLINE#(TZ) + I3;
         GO TC NDSY;
/*
         TWO-FOLD OR TWO-FOLD SCREW AXIS IN Y; DETERMINE TX & TZ
         NOTE P2(5) ABOVE THAT V2-V1 = QY+/-1 BEFORE U2+U1 = QX+2TX
         AND W2+W1 = QZ+2TZ HOLDS.
                                                                  */
SYM(5):  IF ((JMIVP ¬= QY(K)) & (JMIV ¬= QY(K)) & (JMIVM ¬= QY(K)))
                      THEN GO TO NDSY;                         ELSE;
         TX = MOD (((JPIU - QX(K)) / 2) + I4A,  IACR);
         TZ = MOD (((JPIW - QZ(K)) / 2) + I4L,  ILAY);
         ZXLAY#(TZ,TX) = ZXLAY#(TZ,TX) + I2;
         TZP = MOD (TZ + 1, ILAY);
         TXP = MOD (TX + 1, IACR);                                      -P2-3020
         TZM = MOD (TZ - 1 + ILAY, ILAY);                               -P2-3030
         TXM = MOD (TX - 1 + IACR, IACR);                               -P2-3040
         ZXLAY#(TZ,TXP) = ZXLAY#(TZ,TXP) + I4;                          -P2-3050
         ZXLAY#(TZ,TXM) = ZXLAY#(TZ,TXM) + I4;                          -P2-3060
         ZXLAY#(TZP,TX) = ZXLAY#(TZP,TX) + I4;                          -P2-3070
         ZXLAY#(TZM,TX) = ZXLAY#(TZM,TX) + I4;                          -P2-3080
         ZXLAY#(TZM,TXP) = ZXLAY#(TZM,TXP) + I6;                        -P2-3090
         ZXLAY#(TZM,TXM) = ZXLAY#(TZM,TXM) + I6;                        -P2-3100
         ZXLAY#(TZP,TXM) = ZXLAY#(TZP,TXM) + I6;                        -P2-3110
         ZXLAY#(TZP,TXP) = ZXLAY#(TZP,TXP) + I6;                        -P2-3120
         GO TC NDSY;                                                    -P2-3130
/*                                                                      -P2-3140
         TWO-FOLD OR TWO-FOLD SCREW AXIS IN X; DETERMINE TY & TZ        -P2-3150
         NOTE P2(6) ABOVE THAT U2-U1 = QX+/-1 BEFORE V2+V1 = QY+2TY     -P2-3160
         AND W2+W1 = QZ+2TZ                                             -P2-3170
                                                                     */ -P2-3180
SYM(6):  IF ((JMIUP ¬= QX(K)) & (JMIU ¬= QX(K)) & (JMIUM ¬= QX(K)))     -P2-3190
                        THEN GO TO NDSY;                         ELSE;  -P2-3200
         TZ = MOD (((JPIW - QZ(K)) / 2) + I4L,  ILAY);                  -P2-3210
         TY = MOD (((JPIV - QY(K)) / 2) + I4D,  IDWN);                  -P2-3220
         YZLAY#(TY,TZ) = YZLAY#(TY,TZ) + I2;                            -P2-3230
         TYP = MOD (TY + 1, IDWN);                                      -P2-3240
         TZP = MOD (TZ + 1, ILAY);                                      -P2-3250
         TYM = MOD (TY - 1 + IDWN, IDWN);                               -P2-3260
         TZM = MOD (TZ - 1 + ILAY, ILAY);                               -P2-3270
         YZLAY#(TY,TZP) = YZLAY#(TY,TZP) + I4;                          -P2-3280
         YZLAY#(TY,TZM) = YZLAY#(TY,TZM) + I4;                          -P2-3290
         YZLAY#(TYP,TZ) = YZLAY#(TYP,TZ) + I4;                          -P2-3300
         YZLAY#(TYM,TZ) = YZLAY#(TYM,TZ) + I4;                          -P2-3310
         YZLAY#(TYM,TZP) = YZLAY#(TYM,TZP) + I6;                        -P2-3320
         YZLAY#(TYM,TZM) = YZLAY#(TYM,TZM) + I6;                        -P2-3330
         YZLAY#(TYP,TZM) = YZLAY#(TYP,TZM) + I6;                        -P2-3340
         YZLAY#(TYP,TZP) = YZLAY#(TYP,TZP) + I6;                        -P2-3350
         GO TC NDSY;                                                    -P2-3360
/*                                                                      -P2-3370
         CENTER OF INVERSION; DETERMINE TX, TY, & TZ                    -P2-3380
                                                                     */ -P2-3390
SYM(7):  TZ = MOD (((JPIW -  ZQ   ) / 2) + I4L,  ILAY);                 -P2-3400
/*                                                                      -P2-3410
         DETERMINE THE LAYERS WITH TX.TY.TZ                             -P2-3420
                                                                     */ -P2-3430
OUTP:    DO IZ = -1 TC 1 BY 1;                                          -P2-3440
         JZ = MOD (TZ + IZ + ILAY, ILAY);                               -P2-3450
         LAY#(JZ) = '1'B;                                               -P2-3460
         END OUTP;                                                      -P2-3470
NDSY:    END SYML;                                                      -P2-3480
         END INNER;                                                     -P2-3490
         IF (CBG)  THEN PUT EDIT (' PEAK# =',J) (A,F(4));        ELSE;  -P2-3500
         END CUTER;                                                     -P2-3510
```

```
              RETURN;                                                     -P2-3520
              END PART2;                                                  -P2-3530


* PROCESS('ATR,XREF,NOSTMT,OPT=02');                                      -P3-0010
 /*    OUTPUT ALL VALUES OF TX.TY.TZ WITH FREQUENCY > 5 TO BE SORTED  */ -P3-0020
 PART3:   PROCEDURE (IACR,IDWN,ILAY,#RTND);                               -P3-0030
 /*                                                                       -P3-0040
              IN PART2, ALL OF THE POSSIBLE VALUES OF TZ WERE FLAGGED AS  -P3-0050
          TO WHETHER OR NOT THEY OCCURRED.  IN THIS SUBROUTINE, A LAYER OF -P3-0060
          TX.TY IS FILLED FOR EACH VALUE OF TZ, WITH THE FREQUENCY OF     -P3-0070
          CCCURRENCE OF EACH VALUE OF TX.TY.TZ.  AFTER THE LAYER HAS BEEN -P3-0080
          CREATED, ALL OF THE TX.TY.TZ TRIPLES WITH FREQUENCY > 5 ARE OUTPUT-P3-0090
                                                                      */ -P3-0100
              DECLARE   LAYER (0:IACR,0:IDWN)    FLOAT BIN CTL;           -P3-0110
              DECLARE   LAY#(0:ILAY)             EXTERNAL BIT(1) CTL;     -P3-0120
              DECLARE  (IACR,IDWN,ILAY,#RTND)    FIXED BINARY (15);       -P3-0130
              DECLARE   SORTIN                   FILE RECORD SEQUENTIAL;  -P3-0140
              DECLARE  (DBG, DNV)                EXTERNAL BIT (1);        -P3-0150
              DECLARE 1 PEAKS (#RTND)            CONTROLLED EXTERNAL,     -P3-0160
                        2 (U,V,W,VALU,MEAN)      FIXED BINARY (15);       -P3-0170
                                                                          -P3-0180
              DECLARE 1 SXYZ,                                             -P3-0190
                        2 (SX, SY, SZ, SK)       FIXED BINARY (15),       -P3-0190
                        2 (VAL, AVG, FREQ)       FLCAT BINARY;            -P3-0200
              DECLARE  (O2,O3,O4,O5,M1,M2,M3,TH) FLOAT BINARY;            -P3-0210
              DECLARE  (XQ, YQ, ZQ)              EXTERNAL FIXED BIN;      -P3-0220
              DECLARE   #INV                     FIXED BIN;               -P3-0230
              #INV = 0;                      TH = 50.000;                 -P3-0240
              O2 = 2.00;                                                  -P3-0250
              O3 = 3.00;                                                  -P3-0260
              O4 = 4.00;                                                  -P3-0270
              O5 = 5.00;                                                  -P3-0280
              X5 = O5 * 5.0000;                                           -P3-0290
              VAL, AVG = 0.0000;                                          -P3-0300
              SK = 7;                                                     -P3-0310
              I4A = IACR * 4;                                             -P3-0320
              I4D = IDWN * 4;                                             -P3-0330
              I4L = ILAY * 4;                                             -P3-0340
              ON ERROR SNAP EXIT;                                         -P3-0350
              IF (DBG)  THEN PUT EDIT ('XQ =',XQ,'YQ =',YQ,'ZQ =',ZQ)     -P3-0360
                        (SKIP(4),3 (A,F(5),X(2)));                ELSE;  -P3-0370
              IF (CBG)  THEN PUT EDIT ('LAY# =  ',LAY#)                   -P3-0380
                        (SKIP(2),A,(ILAY+1) (X(2),B(1)));         ELSE;  -P3-0390
              IF (DNV)  THEN DO;                                          -P3-0400
              PUT EDIT ('THE TX, TY, TZ PAIRS THAT HAVE MORE THAN 5',     -P3-0410
                         ' TX TY TZ SY   VALU   MEAN   FREQ',             -P3-0420
                         ' TX TY TZ SY   VALU   MEAN   FREQ',             -P3-0430
                         ' TX TY TZ SY   VALU   MEAN   FREQ',             -P3-0440
```

```
                   ' TX TY TZ SY   VALU   MEAN   FREQ')                  -P3-0450
                   (PAGE,A,SKIP,A,A,A,A);   PUT SKIP;      END; ELSE;   -P3-0460
         OPEN FILE (SORTIN) OUTPUT;                                     -P3-0470
         ALLOCATE LAYER;                                                -P3-0480
/*                                                                      -P3-0490
         LOOP THROUGH THE LAYERS, ONE AT A TIME                         -P3-0500
         IF (¬LAY#( LAY))  THEN GO TO NDLAY;                     ELSE;  -P3-0510
                                                                    */  -P3-0520
LAYLUP:  DO LAY = 0 TC ILAY BY 1;                                       -P3-0530
         LAYER = 0.000;                                                 -P3-0540
/*                                                                      -P3-0550
         OUTER LOOP THROUGH ALL THE PEAKS                               -P3-0560
                                                                    */  -P3-0570
CUTER:   DO J = 1 TO #RTND BY 1;       MEANJ = MEAN(J);                 -P3-0580
         K = J + 1;                                                     -P3-0590
         IF (K > #RTND)  THEN GO TO NDOTR;                       ELSE;  -P3-0600
         JU = U(J);                                                     -P3-0610
         JV = V(J);                                                     -P3-0620
         JW = W(J);                                                     -P3-0630
/*                                                                      -P3-0640
         INNER LOOP THROUGH THE REMAINDER OF THE PEAKS                  -P3-0650
                                                                    */  -P3-0660
INNER:   DO I = K TO #RTND BY 1;       MEANIJ=FLOAT(MEANJ+MEAN(I))/TH; -P3-0670
         JPIW = MOD (((JW + W(I) - ZQ) / 2) + I4L, ILAY);               -P3-0680
         JPIWP = MOD (JPIW + 1, ILAY);                                  -P3-0690
         JPIWM = MOD (JPIW - 1 + ILAY, ILAY);                           -P3-0700
         IF (JPIW =  LAY)  THEN GO TO RTON;                      ELSE;  -P3-0710
         IF ((JPIWP =  LAY)|(JPIWM =  LAY))  THEN GO TO ZOFF1;          -P3-0720
                                             ELSE GO TO NDINR;          -P3-0730
/*                                                                      -P3-0740
         THE VALUE OF ILAY = TZ                                         -P3-0750
                                                                    */  -P3-0760
RTON:    M1 = 05 * MEANIJ;                                              -P3-0770
         M2 = 04 * MEANIJ;                                              -P3-0780
         M3 = 03 * MEANIJ;                                              -P3-0790
         GO TC OK;                                                      -P3-0800
/*                                                                      -P3-0810
         THE VALUE OF ILAY = TZ+1 OR TZ-1                               -P3-0820
                                                                    */  -P3-0830
ZOFF1:   M1 = 04 * MEANIJ;                                              -P3-0840
         M2 = 03 * MEANIJ;                                              -P3-0850
         M3 = 02 * MEANIJ;                                              -P3-0860
/*                                                                      -P3-0870
         SET UP THE VALUES OF TX AND TY OFF ONE GRID POINT              -P3-0880
                                                                    */  -P3-0890
OK:      JPIU = MOD (((JU + U(I) - XQ) / 2) + I4A, IACR);               -P3-0900
         JPIV = MOD (((JV + V(I) - YQ) / 2) + I4D, IDWN);               -P3-0910
         JPIUP = MOD (JPIU + 1, IACR);                                  -P3-0920
         JPIVP = MOD (JPIV + 1, IDWN);                                  -P3-0930
         JPIUM = MOD (JPIU - 1 + IACR, IACR);                           -P3-0940
```

```
          JPIVM = MOD (JPIV - 1 + IDWN, IDWN);
/*
          FILL THE ARRAY WITH THE FREQUENCIES
                                                                      */
FILL:     LAYER (JPIU ,JPIV )  =  LAYER (JPIU ,JPIV ) + M1;
          LAYER (JPIU ,JPIVP)  =  LAYER (JPIU ,JPIVP) + M2;
          LAYER (JPIU ,JPIVM)  =  LAYER (JPIU ,JPIVM) + M2;
          LAYER (JPIUP,JPIV )  =  LAYER (JPIUP,JPIV ) + M2;
          LAYER (JPIUP,JPIVP)  =  LAYER (JPIUP,JPIVP) + M3;
          LAYER (JPIUP,JPIVM)  =  LAYER (JPIUP,JPIVM) + M3;
          LAYER (JPIUM,JPIV )  =  LAYER (JPIUM,JPIV ) + M2;
          LAYER (JPIUM,JPIVP)  =  LAYER (JPIUM,JPIVP) + M3;
          LAYER (JPIUM,JPIVM)  =  LAYER (JPIUM,JPIVM) + M3;
NDINR:    END INNER;
NDOTR:    END OUTER;
/*
          WRITE OUT THE TX.TY.TZ TRIPLES FROM THE LAYER
                                                                      */
          SZ = LAY;
SYLP:     DO SY = 0 TO IDWN BY 1;
SXLP:     DO SX = 0 TO IACR BY 1;
          FREQ = LAYER (SX,SY);
          IF (FREQ > X5)  THEN GO TO OUTP;         ELSE GO TO NDSX;
OUTP:     WRITE FILE (SORTIN) FROM (SXYZ);
          #INV = #INV + 1;
          IF (CNV)  THEN PUT EDIT (SXYZ) (4 F(3),3 F(7));        ELSE;
NDSX:     END SXLP;
          END SYLP;
NDLAY:    END LAYLUP;
          FREE LAYER, LAY#, PEAKS;
          PUT EDIT('THERE ARE',#INV,'TX.TY.TZ TRIPLES FROM INVERSIONS.')
                  (SKIP(2),A,F(10),X(1),A);
          PUT EDIT ('***** ALL INVERSIONS HAVE BEEN OUTPUT *****')
                   (SKIP(2),A);
          RETURN;
          END PART3;


* PROCESS('ATR,XREF,NOSTMT,OPT=02');
 /*   OUTPUT TX, TY, TZ, TX.TY, TY.TZ, TZ.TX WITH FREQUENCY > 10      */
 PART4:   PROCEDURE (IACR, IDWN, ILAY);
 /*
          IN PART2 THE TX,TY,TZ,TX.TY,TY.TZ,TX.TZ GROUP(S) ARE PROCUCED
          FROM THE NON-INVERSION SYMMETRY ELEMENTS.  THESE VALUES OF
          THE TRANSLATIONS ARE STORED IN THE EXTERNAL ARRAYS XLINE#,
          YLINE#, ZLINE#, XYLAY#, YZLAY#, ZXLAY#.  THESE VALUES ARE
          WRITTEN OUT ON THE FILE "SORTIN", WITH THE UNDETERMINED VALUES
          SET EQUAL TO -1.
```

```
                                                                    */ -P4-0110
         DECLARE  (IACR,IA,IDWN,ID,ILAY,IL) FIXED BINARY (15);         -P4-0120
         DECLARE   ION(7)              BIT (1) EXTERNAL;               -P4-0130
         DECLARE   SORTIN              FILE RECORD SEQUENTIAL;         -P4-0140
         DECLARE   SLB(6)              LABEL;                          -P4-0150
         DECLARE  (XLINE#(0:IACR), YLINE#(0:IDWN), ZLINE#(0:ILAY),     -P4-0160
              XYLAY#(0:IA,0:ID), YZLAY#(0:ID,0:IL), ZXLAY#(0:IL,0:IA)) -P4-0170
                                  EXTERNAL CONTROLLED FLOAT BINARY;    -P4-0180
         DECLARE 1 SXYZ,                                               -P4-0190
                   2 (SX, SY, SZ, SK)  FIXED BINARY (15),              -P4-0200
                   2 (VALU,MEAN,FREQ)  FLOAT BINARY;                   -P4-0210
         DECLARE   TEN                 FLOAT BINARY;                   -P4-0220
         DECLARE   DBG                 EXTERNAL BIT (1);               -P4-0230
         TEN = 10.000;                                                 -P4-0240
         IA = IACR;                                                    -P4-0250
         ID = IDWN;                                                    -P4-0260
         IL = ILAY;                                                    -P4-0270
         VALU, MEAN = 1.000;                                           -P4-0280
         ON SUBSCRIPTRANGE SNAP BEGIN; PUT DATA                        -P4-0290
                  (SK,SX,SY,SZ);                 EXIT;          END;   -P4-0300
         IF (DBG)  THEN DO;                                            -P4-0310
         PUT EDIT ('PEAKS FROM MIRRORS, GLIDES, TWO-FOLDS, & SCREWS',  -P4-0320
                   ' SX SY SZ TY  VALUE   MEAN   FREQ',                -P4-0330
                   ' SX SY SZ TY  VALUE   MEAN   FREQ',                -P4-0340
                   ' SX SY SZ TY  VALUE   MEAN   FREQ',                -P4-0350
                   ' SX SY SZ TY  VALUE   MEAN   FREQ')                -P4-0360
                  (PAGE,A,SKIP,A,A,A,A);                               -P4-0370
         PUT SKIP;                     END;                     ELSE;  -P4-0380
/*                                                                     -P4-0390
         GO THROUGH ALL OF THE SIX NON-INVERSION SYMMETRY TYPES        -P4-0400
                                                                    */ -P4-0410
STYP:    DO SK = 1 TO 6 BY 1;                                          -P4-0420
         IF (ION(SK))  THEN GO TO SLB(SK);            ELSE GO TO NDS;  -P4-0430
/*                                                                     -P4-0440
         OUTPUT ALL TX WITH # > 10 FOR A PLANE NORMAL TO X             -P4-0450
                                                                    */ -P4-0460
SLB(1):  SY, SZ = -1;                                                  -P4-0470
XLUP:    DO SX = 0 TO IACR BY 1;                                       -P4-0480
         IF (XLINE#(SX) < TEN)  THEN GO TO NDX;                 ELSE;  -P4-0490
         FREQ = XLINE#(SX);                                            -P4-0500
         IF (DBG)  THEN PUT EDIT (SXYZ) (4 F(3),3 F(7,0));      ELSE;  -P4-0510
         WRITE FILE (SORTIN) FROM (SXYZ);                              -P4-0520
NDX:     END XLUP;                                                     -P4-0530
         FREE XLINE#;                                                  -P4-0540
         GO TC NDS;                                                    -P4-0550
/*                                                                     -P4-0560
         OUTPUT ALL TY WITH # > 10 FOR A PLANE NORMAL TO Y             -P4-0570
                                                                    */ -P4-0580
SLB(2):  SX, SZ = -1;                                                  -P4-0590
YLUP:    DO SY = 0 TO IDWN BY 1;                                       -P4-0600
```

```
         IF (YLINE#(SY) < TEN)  THEN GO TO NDY;                  ELSE;  -P4-0610
         FREQ = YLINE#(SY);                                             -P4-0620
         IF (DBG)  THEN PUT EDIT (SXYZ) (4 F(3),3 F(7,0));       ELSE;  -P4-0630
         WRITE FILE (SORTIN) FROM (SXYZ);                               -P4-0640
NDY:     END YLUP;                                                      -P4-0650
         FREE YLINE#;                                                   -P4-0660
         GO TC NDS;                                                     -P4-0670
/*                                                                      -P4-0680
         OUTPUT ALL TX.TY WITH # > 10 FOR AN AXIS IN Z                  -P4-0690
                                                                     */ -P4-0700
SLB(3):  SZ = -1;                                                       -P4-0710
XLXY:    DO SX = 0 TO IACR BY 1;                                        -P4-0720
YLXY:    DO SY = 0 TO IDWN BY 1;                                        -P4-0730
         IF (XYLAY#(SX,SY) < TEN)  THEN GO TO NDXY;              ELSE;  -P4-0740
         FREQ = XYLAY#(SX,SY);                                          -P4-0750
         IF (DBG)  THEN PUT EDIT (SXYZ) (4 F(3),3 F(7,0));       ELSE;  -P4-0760
         WRITE FILE (SORTIN) FROM (SXYZ);                               -P4-0770
NDXY:    END YLXY;                                                      -P4-0780
         END XLXY;                                                      -P4-0790
         FREE XYLAY#;                                                   -P4-0800
         GO TC NDS;                                                     -P4-0810
/*                                                                      -P4-0820
         OUTPUT ALL TZ WITH # > 10 FOR A PLANE NORMAL TO Z              -P4-0830
                                                                     */ -P4-0840
SLB(4):  SX, SY = -1;                                                   -P4-0850
ZLUP:    DO SZ = 0 TO ILAY BY 1;                                        -P4-0860
         IF (ZLINE#(SZ) < TEN)  THEN GO TO NDZ;                  ELSE;  -P4-0870
         FREQ = ZLINE#(SZ);                                             -P4-0880
         IF (DBG)  THEN PUT EDIT (SXYZ) (4 F(3),3 F(7,0));       ELSE;  -P4-0890
         WRITE FILE (SORTIN) FROM (SXYZ);                               -P4-0900
NDZ:     END ZLUP;                                                      -P4-0910
         FREE ZLINE#;                                                   -P4-0920
         GO TC NDS;                                                     -P4-0930
/*                                                                      -P4-0940
         OUTPUT ALL TZ.TX WITH # > 10 FOR AN AXIS IN Y                  -P4-0950
                                                                     */ -P4-0960
SLB(5):  SY = -1;                                                       -P4-0970
ZLZX:    DO SZ = 0 TO ILAY BY 1;                                        -P4-0980
XLZX:    DO SX = 0 TO IACR BY 1;                                        -P4-0990
         IF (ZXLAY#(SZ,SX) < TEN)  THEN GO TO NDZX;              ELSE;  -P4-1000
         FREQ = ZXLAY#(SZ,SX);                                          -P4-1010
         IF (DBG)  THEN PUT EDIT (SXYZ) (4 F(3),3 F(7,0));       ELSE;  -P4-1020
         WRITE FILE (SORTIN) FROM (SXYZ);                               -P4-1030
NDZX:    END XLZX;                                                      -P4-1040
         END ZLZX;                                                      -P4-1050
         FREE ZXLAY#;                                                   -P4-1060
         GO TO NDS;                                                     -P4-1070
/*                                                                      -P4-1080
         OUTPUT ALL TY.TZ WITH # > 10 FOR AN AXIS IN X                  -P4-1090
                                                                     */ -P4-1100
```

```
SLB(6):   SX = -1;
YLYZ:     DO SY = 0 TO IDWN BY 1;
ZLYZ:     DO SZ = 0 TO ILAY BY 1;
          IF (YZLAY#(SY,SZ) < TEN)  THEN GO TO NDYZ;            ELSE;
          FREQ = YZLAY#(SY,SZ);
          IF (DBG)  THEN PUT EDIT (SXYZ) (4 F(3),3 F(7,0));      ELSE;
          WRITE FILE (SORTIN) FROM (SXYZ);
NDYZ:     END ZLYZ;
          END YLYZ;
          FREE YZLAY#;
NDS:      END STYP;
          RETURN;
          END PART4;


* PROCESS('ATR,XREF,NOSTMT,OPT=02');
 /*   MAP THE SYMMETRY ELEMENTS INTO EACH OTHER                        */
PART5:    PROCEDURE (IACR, IDWN, ILAY);
 /*
          BEFORE THIS PART, THE VALUE OF TX,TY,TZ,TX.TY,TX.TZ,TY.TZ. &
          TX.TY.TZ WRITTEN OUT BY PART3 AND PART4, WERE SORTED BY
          DECREASING FREQUENCY (SO THAT THE MOST FREQUENT ARE FIRST) &
          BY DECREASING SYMMETRY (SO THAT THE INVERSIONS ARE FIRST).
          A NUMBER OF EACH OF THESE, PROPORTIONAL TO THE SIZE OF THE
          MAP AND TO THE TOTAL NUMBER OF EACH SYMMETRY ELEMENT POSSIBLE,
          IS NOW READ AND STORED.  AT THIS POINT, THE SYMMETRY ELEMENTS
          ARE MAPPED INTO EACH OTHER, AS SHOWN BELOW, WITH A LATITUDE
          OF ONE GRID POINT ALLOWED IN EACH DIRECTION:

           SYMMETRY ELEMENT        SYMMETRY ELEMENT MAPPED INTO
                 TX                TX.TY, TX.TZ, TX.TY.TZ
                 TY                TX.TY, TY.TZ, TX.TY.TZ
                 TZ                TX.TZ, TY.TZ, TX.TY.TZ
               TX.TY               TY.TZ, TX.TZ, TX.TY.TZ
               TX.TZ               TX.TY, TY.TZ, TX.TY.TZ
               TY.TZ               TX.TY, TX.TZ, TX.TY.TZ

          A RECORD IS KEPT AS EACH MAPPING IS DONE.  ALSO, THE SUMS OF
          THE FREQUENCIES ENCOUNTERED BY EACH MAPPING ARE COMPUTED AND
          STORED.
                                                                        */
          DECLARE   (IACR, IDWN, ILAY)  FIXED BINARY (15);
          DECLARE   (IA8,ID8,IL8,#XYZ)  FIXED BINARY (15) EXTERNAL;
          DECLARE 1 TXYZ,
                    2 (TX, TY, TZ, SK)  FIXED BINARY (15),
                    2 (VALU,MEAN,FRQ)   FLOAT BINARY;
          DECLARE   SORTOUT             FILE RECORD SEQUENTIAL;
          DECLARE  (ION(7),SGNL,NCNT)   BIT (1) EXTERNAL;
```

```
         DECLARE 1 STORE (#XYZ)         CONTROLLED,                        -P5-0340
                   2 (SX, SY, SZ, KS)   FIXED BINARY (15),                 -P5-0350
                   2 FREQ               FLOAT BINARY;                      -P5-0360
         DECLARE  (CST(7), TST(7))      FIXED BINARY (15);                 -P5-0370
         DECLARE 1 RESULT (#XYZ)        CONTROLLED EXTERNAL,               -P5-0380
                   2 (RX, RY, RZ, RS)   FIXED BINARY (15),                 -P5-0390
                  2 (BX, BY, BXY, BZ, BXZ, BYZ)  BIT (1),                  -P5-0400
                   2 OK                 FLOAT BINARY;                      -P5-0410
         DECLARE  (FREQI, P1, P2, P0)   FLOAT BINARY;                      -P5-0420
         DECLARE   CK(7)                LABEL;                             -P5-0430
         DECLARE   DBG                  EXTERNAL BIT (1);                  -P5-0440
         ON SUBSCRIPTRANGE SNAP BEGIN;  PUT DATA                           -P5-0450
                  (SK,I,J,KSI,IS,KSJ);            EXIT;            END;    -P5-0460
/*                                                                         -P5-0470
         P0:  RIGHT ON                                                     -P5-0480
         P1:  OFF ONE GRID POINT IN ONE DIRECTION                          -P5-0490
         P2:  OFF ONE GRID POINT IN EACH OF TWO DIRECTIONS                 -P5-0500
                                                                        */ -P5-0510
         P1 = 0.8500;                                                      -P5-0520
         P2 = 0.7000;                                                      -P5-0530
         P0 = 1.0000;                                                      -P5-0540
/*                                                                         -P5-0550
         ALLOCATE ARRAY TO HOLD THE LARGEST TX, TY, TZ / SYMM TYPE         -P5-0560
                                                                        */ -P5-0570
         IA8 = IACR / 4;                                                   -P5-0580
         ID8 = IDWN / 4;                                                   -P5-0590
         IL8 = ILAY / 4;                                                   -P5-0600
         TST(1) = ION(1) * IA8 * 2;                                        -P5-0610
         TST(2) = ION(2) * ID8 * 2;                                        -P5-0620
         TST(4) = ION(4) * IL8 * 2;                                        -P5-0630
         TST(3) = ION(3) * IA8 * ID8 * 2;                                  -P5-0640
         TST(5) = ION(5) * IL8 * IA8 * 2;                                  -P5-0650
         TST(6) = ION(6) * ID8 * IL8 * 2;                                  -P5-0660
         TST(7) = ION(7) * IA8 * ID8 * IL8 * 2;                            -P5-0670
         #XYZ = TST(1)+TST(2)+TST(3)+TST(4)+TST(5)+TST(6)+TST(7);          -P5-0680
         IF (IDBG)  THEN PUT EDIT ('TST =',TST,'ION =',ION)                -P5-0690
                                  (SKIP,A,7 F(4),X(5),A,7 B(1)); ELSE;     -P5-0700
         ALLOCATE  STORE;                                                  -P5-0710
         FREQ = 0.000;                                                     -P5-0720
         CST = 0;                                                          -P5-0730
/*                                                                         -P5-0740
         READ IN THE HIGHEST VALUES OF TX, TY, TZ                          -P5-0750
                                                                        */ -P5-0760
         ON ENDFILE (SORTOUT) GO TO FIND;                                  -P5-0770
         I = 0;                                                            -P5-0780
RRD:     READ FILE (SORTOUT) INTO (TXYZ);                                  -P5-0790
         IF (CST(SK) >= TST(SK))  THEN GO TO RRD;                  ELSE;   -P5-0800
         CST(SK) = CST(SK) + 1;                                            -P5-0810
         I = I + 1;                                                        -P5-0820
         SX(I) = TX;                                                       -P5-0830
         SY(I) = TY;
         SZ(I) = TZ;
         KS(I) = SK;
         FREQ(I) = FREQ(I) + FRQ;
         IF (I < #XYZ)  THEN GO TO RRD;                         ELSE;
/*
         MERGE THE LOWER SYMMETRY ELEMENTS INTO THE HIGHER
                                                                    */
FIND:    CLOSE FILE (SORTOUT);
         ALLOCATE  RESULT;
         BX, BY, BXY, BZ, BXZ, BYZ = '0'B;
         RX = SX;
         RY = SY;
         RZ = SZ;
         RS = KS;
         OK = 0.000;
         IF (DBG)  THEN DO;
         PUT EDIT ('SAVED TX, TY, TZ -- BEFORE MAPPING')(PAGE,A);
         PUT EDIT (' TX TY TZ SY      FREQ TX TY TZ SY      FREQ',
                   ' TX TY TZ SY      FREQ TX TY TZ SY      FREQ',
                   ' TX TY TZ SY      FREQ TX TY TZ SY      FREQ')
                  (SKIP(2),A,A,A);     PUT SKIP(2);
         PUT EDIT ((STORE (I) DO I = 1 TO #XYZ)) (4 F(3),E(10,3,4));
                                       END;                     ELSE;
ILUP:    DO I = 1 TO #XYZ BY 1;
         IS = KS(I);
         ISX = SX(I);
         ISY = SY(I);
         ISZ = SZ(I);
         FREQI = FREQ(I);
JLUP:    DO J = 1 TO #XYZ BY 1;
         KSJ = KS(J);
         GO TO CK(KSJ);
/*
         MERGE MIRRORS AND GLIDES WITH SCREWS, TWO-FOLDS AND INVERSIONS
         MERGE TX WITH TX.TY, TX.TZ, OR TX.TY.TZ
                                                                    */
CK(1):   IF ((IS=1)|(IS=2)|(IS=4)|(IS=6))  THEN GO TO NDJ;      ELSE;
         IDX = ABS (ISX - SX(J));
         IF (IDX < 2)  THEN BX(I),BX(J) = '1'B;                 ELSE;
CMPX:    IF (IDX=0)  THEN OK(I) = OK(I) + (FREQI * FREQ(J));    ELSE;
         IF (IDX=1)  THEN OK(I) = OK(I) + (P1 * FREQI*FREQ(J)); ELSE;
         GO TO NDJ;
/*
         MERGE TY WITH TX.TY, TY.TZ OR TX.TY.TZ
                                                                    */
CK(2):   IF ((IS=1)|(IS=2)|(IS=4)|(IS=5))  THEN GO TO NDJ;      ELSE;
         IDY = ABS (ISY - SY(J));
         IF (IDY < 2)  THEN BY(I),BY(J) = '1'B;                 ELSE;
CMPY:    IF (IDY=0)  THEN OK(I) = OK(I) + (FREQI * FREQ(J));    ELSE;
```

```
           IF (IDY=1)  THEN OK(I) = OK(I) + (P1 * FREQI*FREQ(J)); ELSE;   -P5-1340
           GO TO NDJ;                                                       -P5-1350
/*                                                                          -P5-1360
           MERGE TZ WITH TX.TZ, TY.TZ OR TX.TY.TZ                           -P5-1370
                                                                         */ -P5-1380
CK(4):     IF ((IS=1)|(IS=2)|(IS=3)|(IS=4))  THEN GO TO NDJ;      ELSE;   -P5-1390
           IDZ = ABS (ISZ - SZ(J));                                         -P5-1400
           IF (IDZ < 2)  THEN BZ(I),BZ(J) = '1'B;                 ELSE;   -P5-1410
CMPZ:      IF (IDZ=0)  THEN OK(I) = OK(I) + (FREQI * FREQ(J));    ELSE;   -P5-1420
           IF (IDZ=1)  THEN OK(I) = OK(I) + (P1 * FREQI*FREQ(J)); ELSE;   -P5-1430
           GO TC NDJ;                                                       -P5-1440
/*                                                                          -P5-1450
           MAP MIRRORS & GLIDES INTO INVERSIONS & OTHER MIRRORS & GLIDES -P5-1460
           MERGE TX.TY WITH TX.TZ, TY.TZ OR TX.TY.TZ                        -P5-147C
                                                                         */ -P5-1480
CK(3):     IF ((IS=1)|(IS=2)|(IS=3)|(IS=4))  THEN GO TO NDJ;      ELSE;   -P5-1490
           IDX = ABS (ISX - SX(J));                                         -P5-1500
           IDY = ABS (ISY - SY(J));                                         -P5-1510
           IF ((IDX<2) & (IDY<2))  THEN BXY(I),BXY(J)='1'B;       ELSE;   -P5-1520
           ID1 = IDX;                                                       -P5-1530
           ID2 = IDY;                                                       -P5-1540
           IF (IS=5)  THEN IF (IDX>1)  THEN GO TO NDJ;                      -P5-1550
                                       ELSE DO;    BXY(I) = '1'B;           -P5-1560
                      BXY(J)='1'B;          GO TO CMPX;      END; ELSE;   -P5-1570
           IF (IS=6)  THEN IF (IDY>1)  THEN GO TO NDJ;                      -P5-1580
                                       ELSE DO;    BXY(I) = '1'B;           -P5-1590
                      BXY(J)='1'B;          GO TO CMPY;      END; ELSE;   -P5-1600
           IF ((IS=7)&(IDY<2)&(IDX<2))  THEN GO TO CMPXYZ;                  -P5-1610
                                        ELSE GO TO NDJ;                     -P5-1620
/*                                                                          -P5-1630
           MERGE TX.TZ WITH TX.TY, TY.TZ OR TX.TY.TZ                        -P5-1640
                                                                         */ -P5-1650
CK(5):     IF ((IS=1)|(IS=2)|(IS=4)|(IS=5))  THEN GO TO NDJ;      ELSE;   -P5-1660
           IDZ = ABS (ISZ - SZ(J));                                         -P5-1670
           IDX = ABS (ISX - SX(J));                                         -P5-1680
           IF ((IDX<2) & (IDZ<2))  THEN BXZ(I),BXZ(J)='1'B;       ELSE;   -P5-1690
           ID1 = IDZ;                                                       -P5-1700
           ID2 = IDX;                                                       -P5-1710
           IF (IS=3)  THEN IF (IDX>1)  THEN GO TO NDJ;                      -P5-1720
                                       ELSE DO;    BXZ(I) = '1'B;           -P5-1730
                      BXZ(J)='1'B;          GO TO CMPX;      END; ELSE;   -P5-1740
           IF (IS=6)  THEN IF (IDZ>1)  THEN GO TO NDJ;                      -P5-1750
                                       ELSE DO;    BXZ(I) = '1'B;           -P5-1760
                      BXZ(J)='1'B;          GO TO CMPZ;      END; ELSE;   -P5-1770
           IF ((IS=7)&(IDX<2)&(IDZ<2))  THEN GO TO CMPXYZ;                  -P5-1780
                                        ELSE GO TO NDJ;                     -P5-1790
/*                                                                          -P5-1800
           MERGE TY.TZ WITH TX.TY, TX.TZ OR TX.TY.TZ                        -P5-1810
                                                                         */ -P5-1820
CK(6):     IF ((IS=1)|(IS=2)|(IS=4)|(IS=6))  THEN GO TO NDJ;      ELSE;  --P5-1830
```

```
          IDY = ABS (ISY - SY(J));                                    -P5-1840
          IDZ = ABS (ISZ - SZ(J));                                    -P5-1850
          IF ((IDY<2) & (IDZ<2))  THEN BYZ(I),BYZ(J)='1'B;     ELSE;  -P5-1860
          ID1 = IDY;                                                  -P5-1870
          ID2 = IDZ;                                                  -P5-1880
          IF (IS=3)  THEN IF (IDY>1)  THEN GO TO NDJ;                 -P5-1890
                                      ELSE DO;    BYZ(I) = '1'B;      -P5-1900
                         BYZ(J)='1'B;      GO TO CMPY;    END; ELSE;  -P5-1910
          IF (IS=5)  THEN IF (IDZ>1)  THEN GO TO NDJ;                 -P5-1920
                                      ELSE DO;    BYZ(I) = '1'B;      -P5-1930
                         BYZ(J)='1';       GO TO CMPZ;    END; ELSE;  -P5-1940
          IF ((IS=7)&(IDY<2)&(IDZ<2))  THEN GO TO CMPXYZ;             -P5-1950
                                       ELSE GO TO NDJ;                -P5-1960
 CMPXYZ:  IF (ID1=0)  THEN IF (ID2=0)  THEN P = P0;                   -P5-1970
                                       ELSE P = P1;                   -P5-1980
                      ELSE IF (ID2=0)  THEN P = P1;                   -P5-1990
                                       ELSE P = P2;                   -P5-2000
          OK(J) = OK(J) + (P * FREQI * FREQ(J));                      -P5-2010
 CK(7):   NDJ:     END JLUP;                                          -P5-2020
          END ILUP;                                                   -P5-2030
          OK = OK * ((1000.000 * FLOAT(#XYZ)) / SUM(OK));             -P5-2040
          IF (DBG)  THEN DO;                                          -P5-2050
          PUT EDIT ('SAVED TX, TY, TZ --- AFTER MAPPING')(SKIP(3),A); -P5-2060
          PUT EDIT (' TX TY TZ SY     FREQ TX TY TZ SY      FREQ',    -P5-2070
                    ' TX TY TZ SY     FREQ TX TY TZ SY      FREQ',    -P5-2080
                    ' TX TY TZ SY     FREQ TX TY TZ SY      FREQ')    -P5-2090
                   (SKIP(2),A,A,A);     PUT SKIP(2);                  -P5-2100
          PUT EDIT ((RESULT(I) DO I = 1 TO #XYZ))(4 F(3),6 B(1),F(4,0));-P5-2110
                                        END;                   ELSE;  -P5-2120
          FREE STORE;                                                 -P5-2130
          RETURN;                                                     -P5-2140
          END PART5;                                                  -P5-2150


* PROCESS('ATR,XREF,NOSTMT,OPT=02');                                  -P6-0010
 /*   CHOOSE THE MOST FREQUENT VALUE OF TX, TY, AND TZ             */ -P6-0020
 PART6:   PROCEDURE (IACR, IDWN, ILAY);                               -P6-0030
 /*                                                                   -P6-0040
      THIS SUBROUTINE CHECKS THROUGH ALL OF THE PEAKS MAPPED INTO EACH -P6-0050
      OTHER TO FIND THE HIGHEST SYMMETRY ELEMENT INTO WHICH ALL THE   -P6-0060
      LOWER SYMMETRY ELEMENTS HAVE BEEN MAPPED, WHICH HIGHER SYMMETRY -P6-0070
      ELEMENT ALSO HAS THE HIGHEST FREQUENCY.  IF THIS SYMMETRY ELEMENT -P6-0080
      IS SUCH THAT ALL OF THE OTHER SYMMETRY ELEMENT TYPES HAVE BEEN  -P6-0090
      MAPPED INTO IT, THE PROGRAM SETS THE APPROPRIATE VALUES OF TX,TY, -P6-0100
      & TZ AND EXITS.  IF THIS LEAVES ANY ONE OF TX, TY, OR TZ UNDETER- -P6-0110
      MINED, IT IS SET TO ZERO, AS THE ORIGIN IN THAT DIRECTION MAY BE -P6-0120
      SET TO ANY VALUE.  IF THE SYMMETRY ELEMENTS DETERMINE ALL THREE -P6-0130
      OF TX,TY,TZ BUT DO NOT OVERLAP, SUCH AS TX.TZ AND TY, THE HIGHEST -P6-0140
```

```
FREQUENCY OF EACH IS USED TO DETERMINE THE POSITION OF THE ORIGIN.
IF THE SYMMETRY ELEMENTS OVERLAP, SUCH AS TX.TY, TY.TZ, AND TY.TZ,
THEN THE DIFFERENT SYMMETRY TYPES ARE SEPARATED INTO THREE ARRAYS,
ONE FOR TX.TY, ONE FOR TY.TZ, AND ONE FOR TX.TZ, SORTED BY DECREA-
SING FREQUENCY.  THE THREE ARRAYS ARE THEN MERGED, IN THE FOLLOW-
ING MANNER: LET TX1.TY1, TY2.TZ2, & TX3.TZ3 BE THE THREE DOUBLES.
THEN, IF TX1=TX3 & TY1=TY2 & TZ2=TZ3, THE TRIPLE TX1.TY2.TZ3 IS
THEN SAVED WITH THE SUM OF THE THREE FREQUENCIES.  THE TRIPLE
WITH THE HIGHEST FREQUENCY SUM IS THEN USED AS THE ORIGIN.
                                                               */

    DECLARE   DBG                EXTERNAL BIT (1);
    DECLARE  (ION(7),SGNL,NCNT)  BIT (1) EXTERNAL;
    DECLARE   (IA8,ID8,IL8,#XYZ) FIXED BINARY (15) EXTERNAL;
    DECLARE 1 RESULT (#XYZ)      CONTROLLED EXTERNAL,
              2 (RX, RY, RZ, RS) FIXED BINARY (15),
             2 (BX, BY, BXY, BZ, BXZ, BYZ)  BIT (1),
              2 CK               FLOAT BINARY;
    DECLARE  (CKR (7), CKDB)     BIT (6);
    DECLARE   CKOK (7)           FLOAT BINARY;
    DECLARE   JOK (7)            FIXED BINARY (15);
    DECLARE 1 S2XY3 (IAD8)       CONTROLLED,
              2 (X3, Y3)         FIXED BINARY (15),
              2 OK3              FLOAT BINARY;
    DECLARE 1 S2XZ5 (IAL8)       CONTROLLED,
              2 (X5, Z5)         FIXED BINARY (15),
              2 OK5              FLOAT BINARY;
    DECLARE 1 S2YZ6 (IDL8)       CONTROLLED,
              2 (Y6, Z6)         FIXED BINARY (15),
              2 OK6              FLOAT BINARY;
    DECLARE  (J3,K3,L5,J5,K6,L6) FIXED BINARY (15);
    DECLARE   CHR                CHARACTER (1);
    DECLARE  (IFX, IFY, IFZ)     FIXED BINARY (15) EXTERNAL;
    DECLARE   FOK                FLOAT BINARY;
    DECLARE 1 FXYZ (#OK)         CONTROLLED EXTERNAL,
              2 (XR, YR, ZR, SR) FIXED BINARY (15),
              2 BSTRG            BIT (6),
              2 GOK              FLOAT BINARY;
    DECLARE   #OK                FIXED BINARY EXTERNAL;
    ON SUBSCRIPTRANGE SNAP BEGIN; PUT DATA
 (#OK,INDX,II,J,JRS,IRS,JFNL,I3,I3,I6,J3,J5,J6,I,L,K,I356);
                                 EXIT;                END;

/*
    "CKR" HOLDS BIT STRINGS TO BE CHECKED.  FOR EXAMPLE IF THERE
    ARE THE FOLLOWING SYMMETRY ELEMENTS TY, TX.TZ, TX.TY.TZ,
    AS IN P21/C,  THEN ONE WOULD HAVE ION(1),ION(3),ION(4),ION(6)
    =0 AND ION(2),ION(5),ION(7) = 1.  PART5 SHOULD HAVE MAPPED
    TY INTO TX.TY.TZ AND TX.TZ INTO TX.TY.TZ, AND WOULD HAVE LEFT
    A RECORD OF SUCH MAPPING.  IF, FOR A PARTICULAR TRIPLE OF
    TX.TY.TZ THERE IS NO INDICATION OF BOTH TY AND TX.TZ BEING
    MAPPED IN, THEN THAT TRIPLE IS ASSUMED TO BE FALSE.
```

```
                                                                           */ -P6-0650
          CKR(7) = ION(1) ||ION(2) ||ION(3) ||ION(4) ||ION(5) ||ION(6);  -P6-0660
          CKR(6) = '0'B      ||ION(2) ||ION(3) ||ION(4) ||ION(5) ||'1'B;    -P6-0670
          CKR(5) = ION(1) ||'0'B     ||ION(3) ||ION(4) ||'1'B    ||ION(6);  -P6-0680
          CKR(3) = ION(1) ||ION(2) ||'1'B    ||'0'B    ||ION(5) ||ION(6);  -P6-0690
          CKR(4) = '000'B ||  ION(4)  || '00'B;                              -P6-0700
          CKR(2) = '0'B || ION(2) || '0000'B;                                -P6-0710
          CKR(1) = ION(1) || '00000'B;                                       -P6-0720
          IF (DBG)  THEN PUT EDIT('CKR =',CKR) (SKIP,A,7 (B(7),X(1)));    -P6-0730
                    ELSE;                                                    -P6-0740
          CKOK = 0.000;                                                      -P6-0750
          JOK = 0;                            #OK = 0;                       -P6-0760
          IRS = 0;                            IDL8 = ID8 * IL8 * 2;          -P6-0770
          IAD8 = IA8 * ID8 * 2;               IAL8 = IA8 * IL8 * 2;          -P6-0780
                                                                             -P6-0790
 /*                                                                          
          DETERMINE THE HIGHEST SYMMETRY ELEMENT TO BE CONSIDERED            -P6-0800
                                                                           */ -P6-0810
LUPJ:     DO J = 1 TO #XYZ BY 1;                                             -P6-0820
          CKDB = BX(J) || BY(J) || BXY(J) || BZ(J) || BXZ(J) || BYZ(J); -P6-0830
          JRS = RS(J);                                                       -P6-0840
          IF (CKDB ¬= CKR(JRS))  THEN DO;                                    -P6-0850
               OK(J) = 0.000;                                                -P6-0860
              GO TO NDJL;                END;                       ELSE;    -P6-0870
          IF (OK(J) > CKOK(JRS))  THEN DO;                                   -P6-0880
              CKOK(JRS) = OK(J);                                             -P6-0890
              JOK(JRS) = J;              END;                       ELSE;    -P6-0900
          IF (JRS = 4)  THEN JRS = 3;                               ELSE     -P6-0910
          IF (JRS = 3)  THEN JRS = 4;                               ELSE;    -P6-0920
          IF (JRS >= IRS)  THEN IRS = JRS;                          ELSE;    -P6-0930
NDJL:     IF (OK(J) > 0.0)  THEN #OK = #OK + 1;                     ELSE;    -P6-0940
          END LUPJ;                                                          -P6-0950
          IF (DBG)  THEN DO;                                                 -P6-0960
          PUT EDIT ('SAVED TX, TY, TZ --- BEFORE MERGING & SORTING')        -P6-0970
                   (PAGE,A);                                                 -P6-0980
          PUT EDIT (' TX TY TZ SY      FREQ TX TY TZ SY      FREQ',          -P6-0990
                    ' TX TY TZ SY      FREQ TX TY TZ SY      FREQ',          -P6-1000
                    ' TX TY TZ SY      FREQ TX TY TZ SY      FREQ')          -P6-1010
                   (SKIP(2),A,A,A);      PUT SKIP(2);                        -P6-1020
          PUT EDIT ((RESULT(I) DO I = 1 TO #XYZ))(4 F(3),6 B(1),F(4,0));-P6-1030
          PUT EDIT ('JOK =',JOK,'CKOK =',CKOK)                               -P6-1040
                   (SKIP,A,7 F(5),X(10),A,7 F(10,2));                        -P6-1050
                                         END;                       ELSE;    -P6-1060
                                                                             -P6-1070
 /*                                                                          
          BRANCH, DEPENDING ON THE TYPE OF THE HIGHEST SYMM. ELEMENT         -P6-1080
                                                                           */ -P6-1090
          IF (IRS = 3)  THEN IRS = 4;                               ELSE     -P6-1100
          IF (IRS = 4)  THEN IRS = 3;                               ELSE;    -P6-1110
          JFNL = JOK(IRS);                                                   -P6-1120
          IFX = RX(JFNL);                                                    -P6-1130
          IFY = RY(JFNL);                                                    -P6-1140
```

```
         IFZ = RZ(JFNL);
         FOK = OK(JFNL);
         IF (IRS=7)  THEN GO TO CMP7;                                 ELSE;
         IF ((IRS=6) & (¬ION(5)) & (¬ION(3))) THEN GO TO CMP6;        ELSE;
         IF ((IRS=5) & (¬ION(6)) & (¬ION(3))) THEN GO TO CMP5;        ELSE;
         IF ((IRS=3) & (¬ION(6)) & (¬ION(5))) THEN GO TO CMP3;        ELSE;
         IF ((IRS=4) | (IRS=2) | (IRS=1))  THEN GO TO CMPL;           ELSE;
/*
         THERE ARE THREE TWO-FOLD AXES OR TWO-FOLD SCREW AXES
                                                                   */
         ALLOCATE  S2XY3, S2XZ5, S2YZ6;
         X3, Y3 = 0;
         OK3 = 0.000;
         X5, Z5 = 0;
         OK5 = 0.000;
         Y6, Z6 = 0;
         OK6 = 0.000;
         I3, I5, I6 = 1;
         J3 = JOK(3);
         J5 = JOK(5);
         J6 = JOK(6);
         I356 = 3;
         K356, L356 = 0;
         OK3, OK5, OK6 = 0.000;
A3:      IF (I3 > (IAC8))      THEN GO TO A5;                         ELSE;
         IF (J3 = 0)  THEN GO TO A5;                                  ELSE;
         X3(I3) = RX(J3);
         Y3(I3) = RY(J3);
         OK3(I3) = OK(J3);
         OK(J3) = 0.000;
A5:      IF (I5 > (IAL8))      THEN GO TO A6;                         ELSE;
         IF (J5 = 0)  THEN GO TO A6;                                  ELSE;
         X5(I5) = RX(J5);
         Z5(I5) = RZ(J5);
         OK5(I5) = OK(J5);
         OK(J5) = 0.000;
A6:      IF (I6 > (IDL8))      THEN GO TO Z356;                       ELSE;
         IF (J6 = 0)  THEN GO TO Z356;                                ELSE;
         Y6(I6) = RY(J6);
         Z6(I6) = RZ(J6);
         OK6(I6) = OK(J6);
         OK(J6) = 0.000;
/*
         PLACE ALL OF THE TWO-FOLDS IN ARRAYS IN DESCENDING ORDER
                                                                   */
Z356:    K356 = L356;
         L356 = I356;
         I356 = I3 + I5 + I6;
         IF (K356=I356)  THEN GO TO PKAL;                             ELSE;
         IF (DBG)  THEN PUT EDIT (I356) (F(6));                       ELSE;
         J3, J5, J6 = 0;                               -P6-1650
         B3, B5, B6 = 0.001;                           -P6-1660
         IF (I356 > #XYZ)  THEN GO TO PKAL;            -P6-1670
DLUP:    DO I = 1 TO #XYZ BY 1;                        -P6-1680
         OK356 = OK(I);                                -P6-1690
         J356 = RS(I);                                 -P6-1700
         IF ((OK356 > B3) & (J356 = 3))  THEN DO;      -P6-1710
              B3 = OK356;                              -P6-1720
              J3 = I;                 END;       ELSE; -P6-1730
         IF ((OK356 > B5) & (J356 = 5))  THEN DO;      -P6-1740
              B5 = OK356;                              -P6-1750
              J5 = I;                 END;       ELSE; -P6-1760
         IF ((OK356 > B6) & (J356 = 6))  THEN DO;      -P6-1770
              B6 = OK356;                              -P6-1780
              J6 = I;                 END;       ELSE; -P6-1790
         END DLUP;                                     -P6-1800
         IF (J3 ¬= 0)  THEN I3 = I3 + 1;         ELSE; -P6-1810
         IF (J5 ¬= 0)  THEN I5 = I5 + 1;               -P6-1820
         IF (J6 ¬= 0)  THEN I6 = I6 + 1;               -P6-1830
         IF ((I3>(IAD8))    & (I5>(IAL8))    & (I6>(IDL8)))            -P6-1840
                   THEN GO TO PKAL;              ELSE GO TO A3;        -P6-1850
/*                                                     -P6-1860
         FIND THE MOST FREQUENT AND CONSISTANT TX.TY.TZ                -P6-1870
                                                                    */ -P6-1880
PKAL:    I356 = 0;                                     -P6-1890
         OK = 0.000;                                   -P6-1900
         IF (DBG)  THEN DO;                            -P6-1910
         PUT EDIT ('SORTED TX.TY, TY.TZ, TX.TZ -- PART 6')             -P6-1920
                  (SKIP(4),A);                         -P6-1930
         PUT EDIT (' TX TY TZ SY      FREQ TX TY TZ SY      FREQ',     -P6-1940
                   ' TX TY TZ SY      FREQ TX TY TZ SY      FREQ',     -P6-1950
                   ' TX TY TZ SY      FREQ TX TY TZ SY      FREQ')     -P6-1960
                  (SKIP(2),A,A,A);     PUT SKIP(2);                    -P6-1970
         PUT EDIT ((S2XY3(I) DO I = 1 TO IAD8))                        -P6-1980
                  (2 F(3),X(9),F(7,0));                                -P6-1990
         PUT EDIT ((S2XZ5(I) DO I = 1 TO IAL8))                        -P6-2000
                  (F(3),X(3),F(3),X(6),F(7,0));                        -P6-2010
         PUT EDIT ((S2YZ6(I) DO I = 1 TO IDL8))                        -P6-2020
                  (X(3),2 F(3),X(6),F(7,0));                           -P6-2030
                                       END;                     ELSE;  -P6-2040
LP6:     DO L = 1 TO IDL8 BY 1;                                        -P6-2050
         IF (OK6(L) = 0.000)  THEN GO TO FNL2S;                 ELSE;  -P6-2060
         K6 = Y6(L);                                                   -P6-2070
         L6 = Z6(L);                                                   -P6-2080
LP5:     DO K = 1 TO IAL8 BY 1;                                        -P6-2090
         IF (OK5(K) = 0.000)  THEN GO TO NDL5;                  ELSE;  -P6-2100
         J5 = X5(K);                                                   -P6-2110
         L5 = Z5(K);                                                   -P6-2120
LP3:     DO J = 1 TO IAD8 BY 1;                                        -P6-2130
         IF (OK3(J) = 0.000)  THEN GO TO NDL3;                  ELSE;  -P6-2140
         J3 = X3(J);
         K3 = Y3(J);
         IF ((J3=J5)&(K3=K6) & (L5=L6))  THEN GOOD: DO;
              I356 = I356 + 1;
              IF (I356 > #XYZ)  THEN GO TO FNL2S;                         ELSE;
              RX(I356) = J3;
              RY(I356) = K3;
              RZ(I356) = L5;
              RS(I356) = 7;
              OK(I356) = OK6(L) + OK5(K) + OK3(J);
NDL3:    END LP3;
NDL5:    END LP5;
         END LP6;
FNL2S:   IF (DBG)  THEN DO;
         PUT EDIT ('MERGED TX.TY, TY.TZ, TX.TZ -- PART 6')
                   (SKIP(4),A);
         PUT EDIT (' TX TY TZ SY      FREQ TX TY TZ SY      FREQ',
                   ' TX TY TZ SY      FREQ TX TY TZ SY      FREQ',
                   ' TX TY TZ SY      FREQ TX TY TZ SY      FREQ')
                   (SKIP(2),A,A,A);    PUT SKIP(2);
         PUT EDIT ((RX(I),RY(I),RZ(I),RS(I),OK(I) DC I = 1 TO #XYZ))
                   (3 F(3),F(2),F(11,0));
                                         END;                     ELSE;
         FOK = 0.000;
FLUP:    DO M = 1 TO #XYZ BY 1;
         IF (OK(M) > FOK)  THEN DO;
              MF = M;
              FOK = OK(M);       END;                             ELSE;
         END FLUP;
         IFX = RX(MF);
         IFY = RY(MF);
         IFZ = RZ(MF);
         GO TO WXYZ;
/*
         THERE IS AN INVERSION CENTER
                                                                     */
CMP7:    PUT EDIT ('THE HIGHEST SYMMETRY ELEMENT IS AN INVERSION.')
                   (SKIP(4),A);
         GO TC WXYZ;
/*
         THE HIGHEST SYMMETRY ELEMENT IS A TWO-FOLD OR SCREW
                                                                     */
CMP6:    CHR = 'X';
         IF (JOK(1) ¬= 0)  THEN DO;
              IFX = RX(JOK(1));
              FOK = FOK + OK(JOK(1));
              GO TO P2G;        END;                              ELSE;
         IFX = 0;
         GO TO P2B;
CMP5:    CHR = 'Y';
```

```
          IF (JOK(2) ¬= 0)  THEN DO;                                           -P6-2650
               IFY = RY(JOK(2));                                                -P6-2660
               FCK = FOK + OK(JOK(2));                                          -P6-2670
               GO TO P2G;         END;                                 ELSE;   -P6-2680
          IFY = 0;                                                              -P6-2690
          GO TO P2B;                                                            -P6-2700
CMP3:     CHR = 'Z';                                                            -P6-2710
          IF (JOK(4) ¬= 0)  THEN DO;                                           -P6-2720
               IFZ = RZ(JOK(4));                                                -P6-2730
               FOK = FCK + OK(JOK(4));                                          -P6-2740
               GO TO P2G;         END;                                 ELSE;   -P6-2750
          IFZ = 0;                                                              -P6-2760
P2B:      PUT EDIT (' THE HIGHEST SYMMETRY ELEMENT IS AN AXIS IN ',CHR, -P6-2770
                    'UNABLE TO DETERMINE T',CHR,'.  THUS, T',CHR,              -P6-2780
                    ' SET EQUAL TO ZERO.')                                      -P6-2790
                    (SKIP(4),X(10),A,A(1),SKIP(1),X(20),3 (A,A(1)));           -P6-2800
          GO TO WXYZ;                                                           -P6-2810
P2G:      PUT EDIT (' THE HIGHEST SYMMETRY ELEMENT IS AN AXIS IN ',CHR, -P6-2820
                    'HOWEVER, THERE IS A PLANE NORMAL TO ',CHR)                -P6-2830
                    (SKIP(4),X(10),A,A(1),SKIP(1),X(20),A,A(1));               -P6-2840
          GO TC WXYZ;                                                           -P6-2850
/*                                                                              -P6-2860
          THERE IS ONLY A MIRROR OR GLIDE PLANE NORMAL TO ONE AXIS             -P6-2870
                                                                           */ -P6-2880
CMPL:     PUT EDIT ('THERE IS ONLY A PLANE NORMAL TO ONE AXIS.',              -P6-2890
                    'THE T''S IN THE OTHER DIRECTIONS ARE SET =0.')            -P6-2900
                    (SKIP(4),X(10),A,SKIP(1),X(20),A);                         -P6-2910
          IFX, IFY, IFZ = 0;                                                    -P6-2920
          IF (IRS = 4)  THEN IFZ = RZ(JFNL);                           ELSE;   -P6-2930
          IF (IRS = 2)  THEN IFY = RY(JFNL);                           ELSE;   -P6-2940
          IF (IRS = 1)  THEN IFX = RX(JFNL);                           ELSE;   -P6-2950
WXYZ:     PUT EDIT ('**********  TX =',IFX,'TY =',IFY,'TZ =',IFZ)             -P6-2960
                    (SKIP(2),3 (A,F(5),X(5)));                                 -P6-2970
          FREE  S2YZ6, S2XY3, S2XZ5;                                            -P6-2980
          ALLOCATE  FXYZ;                                                       -P6-2990
          INDX = 0;                                                             -P6-3000
          PUT EDIT ('SAVED TRIPLES, DOUBLES, AND SINGLES') (PAGE,A);           -P6-3010
          PUT EDIT (' TX TY TZ SY      FREQ TX TY TZ SY      FREQ',            -P6-3020
                    ' TX TY TZ SY      FREQ TX TY TZ SY      FREQ',            -P6-3030
                    ' TX TY TZ SY      FREQ TX TY TZ SY      FREQ')            -P6-3040
                    (SKIP(2),A,A,A);     PUT SKIP(2);                          -P6-3050
IILP:     DO II = 1 TO #XYZ BY 1;                                               -P6-3060
          IF (OK(II) > 1)  THEN GD: DO;                                         -P6-3070
          INDX = INDX + 1;                                                      -P6-3080
          XR(INDX) = RX(II);                                                    -P6-3090
          YR(INDX) = RY(II);                                                    -P6-3100
          ZR(INDX) = RZ(II);                                                    -P6-3110
          SR(INDX) = RS(II);                                                    -P6-3120
          GOK(INDX) = OK(II);                                                   -P6-3130
          BSTRG(INDX) = BX(II) || BY(II) || BXY(II) || BZ(II)                  -P6-3140
```

```
                          || BXZ(II) || BYZ(II);                        -P6-3150
               PUT EDIT (FXYZ(INDX)) (4 F(3),B(6),F(4,0));              -P6-3160
          END GD;                  ELSE;               END IILP;        -P6-3170
          FREE RESULT;                                                  -P6-3180
          RETURN;                                                       -P6-3190
          END PART6;                                                    -P6-3200



* PROCESS('ATR,XREF,NOSTMT,OPT=02');                                    -P7-0010
 /*   DETERMINE WHICH PEAKS ARE TO BE KEPT                           */ -P7-0020
 PART7:   PROCEDURE (IACR, IDWN, ILAY, #RTND, #SYMM);                   -P7-0030
 /*                                                                     -P7-0040
      THIS SUBROUTINE MAKES USE OF THE ORIGIN TRANSLATION TX.TY.TZ AND  -P7-0050
      OF THE SYMMETRY ELEMENTS TO DETERMINE WHICH PEAKS FIT IN E.D.     -P7-0060
      SPACE.  IF AT LEAST HALF OF THE SYMMETRY-RELATED PEAKS ARE PRESENT-P7-0070
      WHEN THE ENTIRE MAP HAS BEEN SHIFTED BY TX.TY.TZ, THEN THAT SET   -P7-0080
      OF PEAKS IS OUTPUT IN THE SHIFTED COORDINATES                     -P7-0090
                                                                     */ -P7-0100
          DECLARE  (IACR,IDWN,ILAY,#RTND,#SYMM)   FIXED BINARY (15);    -P7-0110
          DECLARE  (IFX, IFY, IFZ)       FIXED BINARY (15) EXTERNAL;    -P7-0120
          DECLARE 1 SYMMPQ (#SYMM)       CONTROLLED EXTERNAL,          --P7-0130
                    2 (PX,QX,PY,QY,PZ,QZ)    FIXED BINARY (15);         -P7-0140
          DECLARE   #OK                  FIXED BINARY EXTERNAL;         -P7-0150
          DECLARE 1 FXYZ (#OK)           CONTROLLED EXTERNAL,           -P7-0160
                    2 (XR, YR, ZR, SR)   FIXED BINARY (15),             -P7-0170
                    2 BSTRG              BIT (6),                       -P7-0180
                    2 GOK                FLOAT BINARY;                  -P7-0190
          DECLARE 1 PEAKS (#RTND)        CONTROLLED,                    -P7-0200
                    2 (U,V,W,VLU,X,Y,Z)  FIXED BINARY;                  -P7-0210
          DECLARE 1 PEAKR,                                              -P7-0220
                    2 (RU,RV,RW,VAL,AVG)     FIXED BINARY (31);         -P7-0230
          DECLARE  (RETAIN, FINAL)       FILE RECORD SEQUENTIAL;        -P7-0240
          DECLARE  (BX, BY, BZ)          BIT (1);                       -P7-0250
          DECLARE   DBG                  BIT (1) EXTERNAL;              -P7-0260
          ON SUBSCRIPTRANGE SNAP BEGIN;  PUT DATA (I,J,K); EXIT;  END;  -P7-0270
          DECLARE   JAVG (#SYMM)         CONTROLLED FIXED BINARY;       -P7-0280
          DECLARE   PVALUE (16)          FIXED BINARY (15);             -P7-0290
          DECLARE  (L,M,N)               FIXED BINARY (15);             -P7-0300
          DECLARE  (KX,KY,KZ,NSY,CL(24))  FIXED BINARY (15);            -P7-0310
          DECLARE 1 FPEAK,                                              -P7-0320
                    2 (HITE,FX,FY,FZ,#)  FIXED BINARY (31);             -P7-0330
          DECLARE   ONE                  FLOAT BINARY;                  -P7-0340
          ONE = 1.0000;                                                 -P7-0350
          CL(1), CL( 7), CL(13), CL(19) =  40;                          -P7-0360
          CL(2), CL( 8), CL(14), CL(20) =  56;                          -P7-0370
          CL(3), CL( 9), CL(15), CL(21) =  72;                          -P7-0380
          CL(4), CL(10), CL(16), CL(22) =  88;                          -P7-0390
          CL(5), CL(11), CL(17), CL(23) = 104;                          -P7-0400
```

```
         CL(6), CL(12), CL(18), CL(24) = 120;
         L = ROUND (FLOAT (IACR) / 16.000, 0);
         M = ROUND (FLOAT (IDWN) / 16.000, 0);
         N = ROUND (FLOAT (ILAY) / 16.000, 0);
         ISYM = MAX (#SYMM/2 +1, 1);
         IPEAK = 0;
         QACR = FLOAT (IACR);
         QDWN = FLOAT (IDWN);
         QLAY = FLOAT (ILAY);
         ALLOCATE PEAKS;
         ALLOCATE JAVG;
         IF (DBG)  THEN PUT EDIT ('$$$$ TX =',IFX,'TY =',IFY,'TZ =',
                   IFZ) (SKIP(4),3 (A,F(5),X(5)));                 ELSE;
/*
         READ AND STORE THE PEAKS, TRANSFORMING THEM TO E.D. COORDS.
                                                                   */
         OPEN FILE (RETAIN) INPUT;
RLUP:    DO I = 1 TO #RTND BY 1;
         READ FILE (RETAIN) INTO (PEAKR);
         U(I) = RU;
         V(I) = RV;
         W(I) = RW;
         VLU(I) = AVG;
         X(I) = MOD (RU - IFX, IACR);
         Y(I) = MOD (RV - IFY, IDWN);
         Z(I) = MOD (RW - IFZ, ILAY);
         END RLUP;
         CLOSE FILE (RETAIN);
         OPEN FILE (FINAL) OUTPUT;
         NPEAK = 0;
         PUT EDIT ('PEAK BEING CHECKED (PART7) -- #, OR',
                  'IGINAL (UVW) & TRANSFORMED (XYZ) COORDS., EQUIV. P',
                  'OSITIONS (EX,EY,EZ), & # OF MATCHING PEAK (#MP)',
                  '  #    U    V    W VALUE    X    Y    Z    ',
                  '     EX EY EZ #MP    EX EY EZ #MP    EX EY EZ #MP',
                  '     EX EY EZ #MP    EX EY EZ #MP    EX EY EZ #MP')
                  (PAGE,A,A,A,SKIP,A,A,A);
         PUT SKIP;
/*
         LOOP THROUGH ALL RETAINED PEAKS TO DETERMINE WHICH ONES ARE
         ALLOWED BY SYMMETRY AND OUTPUTTING THE ALLOWED PEAKS.
                                                                   */
ILUP:    DO I = 1 TO #RTND BY 1;
         IF (VLU(I) < 1.00)  THEN GO TO NDIL;                      ELSE;
         RU = U(I);
         RV = V(I);
         RW = W(I);
         VAL = VLU(I);
         IVAL = VAL;
         AVG = 0;
         RX = X(I);
         RY = Y(I);
         RZ = Z(I);
         PUT EDIT (I,RU,RV,RW,VAL,RX,RY,RZ)
                  (SKIP,F(3),3 F(4),F(6),3 F(4));
/*
         CREATE EQUIVALENT POSITIONS FROM SYMMETRY ELEMENTS
                                                                     */
SLUP:    DO K = 1 TO #SYMM BY 1;
         KX = MOD ((RX * PX(K)) + QX(K), IACR);
         KY = MOD ((RY * PY(K)) + QY(K), IDWN);
         KZ = MOD ((RZ * PZ(K)) + QZ(K), ILAY);
         KXI = KX + IACR;
         KYI = KY + IDWN;
         KZI = KZ + ILAY;
         NSY = CL(K);
         PUT EDIT (KX,KY,KZ) (COL(NSY), 3 F(3));
/*
         LOOP THROUGH ALL PEAKS - LOOKING FOR SYMMETRY RELATED PEAKS
                                                                     */
JLUP:    DO J = 1 TO #RTND BY 1;
         IF (VLU(J) < 1.00)  THEN GO TO NDJL;                    ELSE;
         OX = MIN (ABS(KX-X(J)),ABS(KXI-X(J)));
         OY = MIN (ABS(KY-Y(J)),ABS(KYI-Y(J)));
         OZ = MIN (ABS(KZ-Z(J)),ABS(KZI-Z(J)));
         BX = (CX <=L);
         BY = (OY <=M);
         BZ = (CZ <=N);
         IF (BX & BY & BZ)  THEN GO TO PNDS;                      ELSE;
NDJL:    END JLUP;
         GO TC NDSL;
PNDS:    AVG = AVG + 1;
         JAVG(AVG) = J;
         PUT EDIT (J) (F(4));
         IVAL = IVAL + VLU(J);
NDSL:    END SLUP;
/*
         OUTPUT THE GOOD PEAKS
                                                                     */
         IF (AVG < ISYM)  THEN GO TO NDIL;                        ELSE;
         NPEAK = NPEAK + AVG + 1;
         HITE = IVAL / (AVG + 1);
         # = AVG;
         FX = RX;
         FY = RY;
         FZ = RZ;
         VLU(I) = 0;
         WRITE FILE (FINAL) FROM (FPEAK);
         IPEAK = IPEAK + 1;
LUPJ:    DO J = 1 TO AVG BY 1;
```

```
         JS = JAVG(J);                                               -P7-1410
         VLU(JS) = 0;                                                -P7-1420
NDLJ:    END LUPJ;                                                   -P7-1430
NDIL:    END ILUP;                                                   -P7-1440
         PUT EDIT ('# PEAKS RETAINED =',NPEAK) (SKIP(2),A,F(5));     -P7-1450
         CLOSE FILE (FINAL);                                         -P7-1460
         OPEN FILE (FINAL) INPUT;                                    -P7-1470
         PUT EDIT ('NUMBER OF INDEPENDENT PEAKS =',IPEAK)            -P7-1480
                  (SKIP(2),A,F(5));                                  -P7-1490
         PUT EDIT ('INDEPENDENT PEAKS AFTER SYMMETRY CHECK',         -P7-1500
                   '     X    Y    Z (FOUR.)     X         Y       Z',-P7-1510
                   '(FRACT.)     HEIGHT      #SYMM',' ')             -P7-1520
                  (PAGE,A,SKIP,A,A,SKIP(2),A);                       -P7-1530
/*                                                                   -P7-1540
         OUTPUT PEAKS REMAINIAG AFTER SYMMETRY CHECK ON "ORFLS" CARDS -P7-1550
         WRITE OUT FOUND PEAKS AND PUNCH LEAST SQUARES CARDS         -P7-1560
                                                                  */ -P7-1570
LUPI:    DO I = 1 TO IPEAK BY 1;                                     -P7-1580
         READ FILE (FINAL) INTO (FPEAK);                             -P7-1590
         XFR = FLOAT (FX) / QACR;                                    -P7-1600
         YFR = FLOAT (FY) / QDWN;                                    -P7-1610
         ZFR = FLOAT (FZ) / QLAY;                                    -P7-1620
         PUT EDIT (FX,FY,FZ,XFR,YFR,ZFR,HITE,#)                      -P7-1630
                  (SKIP,3 F(5),X(4),3 F(10,5),X(9),2 F(10));         -P7-1640
         PUT FILE (PUNCH) EDIT (HITE,ONE,ONE,XFR,YFR,ZFR)            -P7-1650
                  (SKIP,F(5),X(4),5 F(9,5));                         -P7-1660
         END LUPI;                                                   -P7-1670
         CLOSE FILE (FINAL), FILE (PUNCH);                           -P7-1680
         RETURN;                                                     -P7-1690
         END PART7;                                                  -P7-1700
```

```
//LKED.SYSLMOD  DD  DSNAME=PROG.CRYST,                                JCL-0060
//     UNIT=DISK,VOLUME=SER=PROGPK,CISP=(OLD,KEEP)                    JCL-0070
//LKED.SYSIN  DD  *                                                   JCL-0080
  OVERLAY ALPHA                                                       OVLY0010
  INSERT TM_DT                                                        OVLY0020
  OVERLAY ALPHA                                                       OVLY0030
  INSERT PART1                                                        OVLY0040
  OVERLAY ALPHA                                                       OVLY0050
  INSERT PART2                                                        OVLY0060
  OVERLAY ALPHA                                                       OVLY0070
  INSERT IHESRTA                                                      OVLY0080
  OVERLAY ALPHA                                                       OVLY0090
  INSERT PART3                                                        OVLY0100
  OVERLAY ALPHA                                                       OVLY0110
  INSERT PART4                                                        OVLY0120
  OVERLAY ALPHA                                                       OVLY0130
```

```
 INSERT PART5                                                          OVLY0140
 OVERLAY ALPHA                                                         OVLY0150
 INSERT PART6                                                          OVLY0160
 OVERLAY ALPHA                                                         OVLY0170
 INSERT PART7                                                          OVLY0180
 NAME ALOP2(R)                                                         JCL-0090
```

LITERATURE CITED

1. J. P. Birk and J. H. Espenson, J. Amer. Chem. Soc., 90, 2266 - 2277 (1968).

2. N. A. Curry and W. A. Runciman, Acta Cryst., 12, 674 - 678 (1959).

3. H. U. Guedel, A. Ludi, P. Fischer, and W. Haelg, J. Chem. Phys., 53(5), 1917 - 1923 (1970).

4. W. Ozbirn and R. A. Jacobson, Inorg. Chim. Acta, 4, 377 - 382 (1970).

5. D. Britton and C. Mahon, Inorg. Chem., 10(3), 586 - 589 (1971).

6. K. Kuroda and P. Gentile, Inorg. Nucl. Chem. Letters, 3, 151 - 156 (1967).

7. K. Konya, H. Nishikawa, and M. Shibita, Inorg. Chem., 7, 1165 - 1168 (1968).

8. D. E. Williams, "LCR-2, A Fortran Lattice Constant Refinement Program," U. S. Atomic Energy Commission Report IS-1052 (Iowa State University and Institute for Atomic Research, Ames, Iowa). 1964.

9. D. J. Wehe, W. R. Busing, and H. A. Levy, "OR ABS, A Fortran Program for Calculating Single Crystal Absorption Corrections," U. S. Atomic Energy Commission Report ORNL-TM-229 (Oak Ridge National Laboratory, Oak Ridge, Tennessee). 1962.

10. S. L. Lawton and R. A. Jacobson, Inorg. Chem., 7, 2124 - 2134 (1968).

11. R. A. Jacobson, J. A. Wunderlich, and W. N. Lipscomb, Acta Cryst., 14, 598 - 607 (1961).

12. W. R. Busing, K. O. Martin, and H. A. Levy, "OR FLS, A Fortran Crystallographic Least Squares Program," U. S. Atomic Energy Commission Report ORNL-TM-305 (Oak Ridge National Laboratory, Oak Ridge, Tennessee). 1962.

13. J. Rodgers and R. A. Jacobson, "ALF: A General Fourier Program in PL I for Triclinic, Monoclinic, and Orthorhombic Space Groups," U. S. Atomic Energy Commission Report IS-2155 (Iowa State University and Institute for Atomic Research, Ames, Iowa). 1969.

14. C. R. Hubbard, C. O. Quicksall, and R. A. Jacobson, "The Fast Fourier Algorithm and the Programs ALFF, ALFFDP, ALFFPROJ, ALFFT, and FRIEDEL," U. S. Atomic Energy Commission Report IS-2625 (Iowa State University and Institute for Atomic Research, Ames, Iowa). 1971.

15. H. P. Hanson, F. Herman, J. D. Lea, and S. Skillman, Acta Cryst., 17, 1040 - 1044 (1960).

16. D. H. Templeton in "International Tables for X-ray Crystallography," Vol. III, pp. 215 - 216, Table 3.3.2c, The Knoch Press, Birmingham, England (1962).

17. W. R. Busing, K. O. Martin, and H. A. Levy, "OR FFE, A Fortran Crystallographic Function and Error Program," U. S. Atomic Energy Commission Report ORNL-TM-306 (Oak Ridge National laboratory, Oak Ridge, Tennessee). 1964.

18. C. A. Johnson, "OR TEP: A Fortran Thermal - Ellipsoid Plot Program for Crystal Structure Illustrations," U. S. Atomic Energy Commission Report ORNL-3794 (2nd Revision) (Oak Ridge National Laboratory, Oak Ridge, Tennessee). 1970.

19. H. C. Freeman and I. E. Maxwell, Inorg. Chem., 8(6), 1293 - 1298 (1969).

20. M. Dwyler and I. E. Maxwell, Inorg. Chem., 9(6), 1459 - 1464 (1970).

21. J. W. Turley and R. G. Asperger, Inorg. Chem., 10(3), 558 - 565 (1971).

22. B. M. Chadwick and A. G. Sharpe, Adv. Inorg. Chem. Radiochem., 8, 84 - 176 (1966).

23. W. P. Shaefer, B. Wang, and R. E. Marsh, Inorg. Chem., 10(7), 1492 - 1497 (1971).

24. A. Wolberg, Acta Cryst., B25(Part 1), 161 - 162 (1969).

25. S. L. Lawton and R. A. Jacobson, J. Amer. Chem. Soc., 88, 616 - 618 (1966).

26. I. Lindqvist and A. Niggli, J. Inorg. Nucl. Chem., 2, 345 - 347 (1956).

27. M. Edstrand, M. Inge, and N. Ingri, Acta Chem. Scand., 9, 122 - 136, (1955).

28. S. K. Porter and R. A. Jacobson, J. Chem. Soc., Sect. A, 1356 - 1359 (1970).

29. D. R. Schroeder and R. A. Jacobson, submitted for publication in Inorg. Chem.

30. N. W. Alcock, presented in part at the International Summer School on Crystallographic Computing, Ottawa, Canada, August, 1969.

31. J. D. Scott, Queen's University, Kingston, Ontario, Canada, personal communication, 1971.

32. S. K. Porter and R. A. Jacobson, J. Chem. Soc., Sect. A, 1359 - 1362 (1970).

33. M. L. Hackert and R. A. Jacobson, Acta Cryst., B27, 203 - 209 (1971).

34. I. L. Karle and J. Karle, Acta Cryst., 21, 849 - 859 (1966).

35. M. J. Buerger, "Vector Space and Its Application in Crystal Structure Investigation," John Wiley, New York, New York, c. 1959.

36. C. R. Hubbard, C. O. Quicksall, and R. A. Jacobson, J. Appl. Cryst., 5, 234 - 236 (1972).

37. A. N. Barrett and M. Zwick, Acta Cryst., A27, 6 - 11 (1971).

38. B. T. Gorres, Ph. D. Thesis, Princeton University, Princeton, New Jersey, 1964.

39. C. R. Hubbard and R. A. Jacobson, "A Fortran IV Crystallographic System of Programs for Generalized Superpositions," U. S. Atomic Energy Commission Report IS-2210 (Iowa State University and Institute for Atomic Research, Ames, Iowa). 1970.

40. P. Main, M. M. Woolfson, and G. Germain, "MULTAN: A Computer Programme for the Automatic Solution of Crystal Structures," York, England and Leuven, Belgium, 1971.

41. N. Krishnamachari and R. A Jacobson, Iowa State University, Ames, Iowa, personal communication, 1972.

42. B. T. Gorres and R. A. Jacobson, Acta Cryst., 17, 1599 - 1603 (1964).

43. W. P. Ozbirn, Ph. D. Thesis, Iowa State University, Ames, Iowa, 1971.

44. W. Cochran and A. S. Douglas, Proc. Roy. Soc., A243, 281 - 288 (1957).

45. M. L. Hackert and R. A. Jacobson, Acta Cryst., B26(11), 1682 - 1685 (1970).

ACKNOWLEDGEMENTS

The sources of aid and encouragement received during the course of the research described in this thesis are much too numerous to mention in completeness. However, the author especially would like to express his deep and sincere gratitude to:

Prof. Robert A. Jacobson for his unhesitating willingness to share his experience and extensive theoretical and practical knowledge, his seemingly inexhaustible words of advice and encouragement, and his constantly high expectations of and confidence in the author's abilities;

the State of Iowa and the United States government for financial assistance in the form of teaching and research assistantships;

the faculty and graduate students for their helpful discussions and repeated assistance, and a very special acknowledgement to the members, both past and present, of X-ray Group I;

James Benson for giving so willingly and patiently of his time, knowledge and experience;

Gary Dubois, summer research trainee, and Susan Scott, research helper, for their assistance with the PSST method and the structure of potassium antimony chloride;

Fred Hollenbeck for his day-to-day technical aid;

the faculty and staff of the Iowa State University Computation Center for their able assistance;

the compound phenyl-tropylium tetrafluoroboron for its data, the analysis of which has encouraged the author to fully explore the available methods for solving crystal structures and to develop new techniques of his own and which has taught the author the true meaning of frustration;

and my wife Debbie, for her unwavering faith and steady encouragement.