Commission on Crystallographic Computing and Commission on Crystallographic Teaching
International Union of Crystallography
http://www.iucr.org/resources/commissions/crystallographic-computing
http://www.iucr.org/resources/commissions/crystallographic-teaching

Computing Newsletter No. 10 / Teaching Newsletter No. 3, November 2009

Computing Addendum (B) to :

Age Concern

http://www.iucr.org/resources/commissions/crystallographic-computing/newsletters
http://www.iucr.org/resources/commissions/crystallographic-teaching/newsletters

Table of Contents

(This issue's editor: Lachlan Cranswick)

(Warning – unless you want to kill 480 pages worth of forest – DO NOT press the "print" button. For hardcopies – you may like to only print out the articles of personal interest.)

Following manuals and reference materials as a separate addendum PDF file to this issue

User's Guide to the CRYRM Crystallographic Computing System (December 1964)

by David J. Duchamp

6209 Litchfield Lane Kalamazoo, Michigan 49009, USA. E-mail: djduchamp@mac.com

Place holder page. Report-proper starts on following page.

USER'S GUIDE TO THE

CRYRM

CRYSTALLOGRAPHIC COMPUTING SYSTEM

BY

DAVID J. DUCHAMP

DECEMBER 1964

GATES AND CRELLIN LABORATORIES OF CHEMISTRY

CALIFORNIA INSTITUTE OF TECHNOLOGY

PASADENA, CALIFORNIA

USER'S GUIDE TO THE

CRYRM CRYSTALLOGRAPHIC COMPUTING SYSTEM

by

David J. Duchamp

This guide is intended to serve as ready reference for users of the CRYRM Crystallographic Computing System. All control card details and instructions for preparing input decks are contained here. Programming details and description of the methods used in various subprograms are elsewhere.

July 1964

Gates and Crellin Laboratories of Chemistry
California Institute of Technology

Revised December 1964

Acknowledgement

CRYRM is the result of a project started in the spring of 1963 by R.E.Marsh, D.Duchamp and N.C.Webb. When N.C.Webb left Caltech, his work was continued by A.P.Kendig. Major contributions were also made by C.M.Gramaccioli and T.A.Beineke. The entire project has been under the supervision of R.E.Marsh, whose continuing support and searching knowledge of crystallographic computing made it possible.

Programming contributions have been made by I.Ambats, T.A.Beineke, D.Duchamp, C.M.Gramaccioli, A.P.Kendig, J.K.Lo, R.E.Marsh, G.Reeke, Mrs.B.Stroll, R.H.Stanford, and N.C.Webb.

The advice and help of K.Hebert, H.Huber, and S.H.Caine of the Caltech Computing Center with problems of system programming and maintenance is gratefully acknowledged.

Table of Contents.

Chapter 1

Introduction

The CRYRM Crystallographic Computing System is a unified program system for doing crystallographic calculations on the IBM 7094 computer. It is designed to make computing as simple as possible for the crystallographer, while at the same time giving him a very great variety of options. The philosophy has been to make the input a minimum, and to provide a maximum amount of communication between the different kinds of calculations. It is hoped that by using CRYRM, crystallographers will be able to spend more time at the heart of the problem--solving structures--and less time doing trivial calculations and worrying about a mountain of programming details.

The System Program:

The heart of the system is the system program. This program operates directly under the IBSYS Basic Monitor. It has been designed with the Caltech 7094-7040 computing setup in mind, but is designed for running on other IBSYS setups with a minimal number of modifications--if any at all are needed.

The system program contains a crystallographic monitor, input-output subroutines, and certain functional subroutines. Under the system program are a number of subprograms, which do the actual calculations, using the subroutines of the system program where needed. The system program and all subprograms are kept in the installation's system library--either tape or disk.

At the beginning of a job, the crystallographic monitor assumes control. It does such chores as accounting, job sign on, loading of special magnetic data tapes, etc. When a control card comes in requesting calculations which require a subprogram, the correct subprogram is loaded into memory from the library, then it is given control of the 7094. After doing its calculations, the subprogram returns control to the system, which then reads more control cards, calling in other subprograms as required. At the end of a job, the system program terminates all output and signs off the accounting.

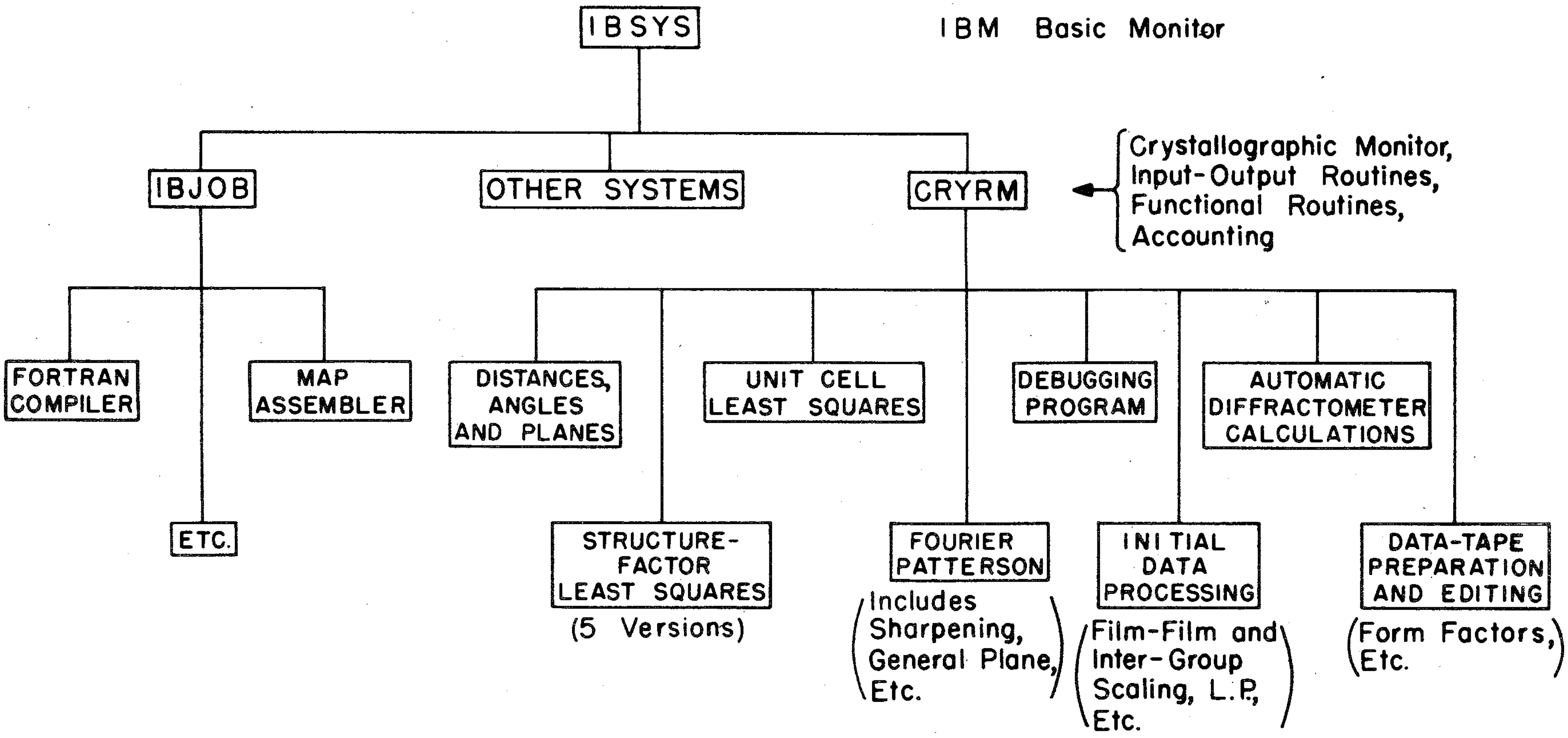
A diagram showing levels of control in the 7094 when CRYRM is operating is shown in Figure 1.

Communication between Subprograms and the DATA TAPE:

Communication between subprograms is accomplished in several ways. Certain subprograms set indicators in the system for others which may follow. Others leave tables stored in a protected area of memory. The main method of accomplishing communication and for preserving information for future runs is by means of a magnetic DATA TAPE. The DATA TAPE for a structure carries on it a variety of information, including cell parameters, the compound name, the kinds of atoms in it, an atomic information deck, and information for each reflection,such as,indices, observed structure factor, sine squared theta / lambda squared, complete calculated structure factors at the present stage of refinement, form factors, etc.

Each DATA TAPE carries on it the information for one structure only. This information is updated by the Structure-Factor Least Squares subprogram at each entry. It may be used as input to future structure-factor least squares calculations; Fourier-type calculations; distances, angles, and planes calculations; etc. The DATA TAPE is prepared by the Data Tape Preparation and Editing subprogram. Reflection information on it may be edited by this subprogram also.

[illegible] by the Data Tape Preparation and Editing subprogram. Reflection information on it may be edited by this subprogram also.


IBSYS
IBM Basic Monitor
IBJOB
OTHER SYSTEMS
CRYRM
Crystallographic Monitor, Input-Output Routines, Functional Routines, Accounting
FORTRAN COMPILER
MAP ASSEMBLER
ETC.
DISTANCES, ANGLES AND PLANES
STRUCTURE-FACTOR LEAST SQUARES
(5 Versions)
UNIT CELL LEAST SQUARES
FOURIER PATTERSON
(Includes Sharpening, General Plane, Etc.)
DEBUGGING PROGRAM
INITIAL DATA PROCESSING
(Film-Film and Inter-Group Scaling, L.P., Etc.)
AUTOMATIC DIFFRACTOMETER CALCULATIONS
DATA-TAPE PREPARATION AND EDITING
(Form Factors, Etc.)

CRYRM SUBPROGRAMS

Figure 1.

The system also provides for the use of another magnetic data tape designated DATA TAPE EXTRA. This tape carries a more limited amount of information. It is used for such chores as communication between subprograms which reduce raw data and the Data Tape Preparation and Editing subprogram, and storage of Fourier-type maps.

Both DATA TAPE and DATA TAPE EXTRA are special tapes which must be loaded by the computer operator when the job requiring them is to be run. There are provisions in the system program to provide for this.

The Subprograms:

An explanation of why these are called 'subprograms' is in order. This is done to emphasize that they run under the CRYRM system program. As Figure 1 shows, they run on the same level as the Fortran compiler or the IBMAP assembler. They are more complicated and provide a greater variety of options than many 'full-fledged programs' for doing the same type calculations. The CRYRM Crystallographic System can be best understood if it is regarded as one single unified monster program, all of which does not remain in memory at one time.

The capabilities of the subprograms are described in the following chapters. Several features, some of which involve new concepts, will be pointed out here. The Initial Data Processing subprogram can provide for carrying data from raw film readings all the way through inter-group scaling to suitable input for the Data Tape Preparation and Editing Subprogram. The Structure-Factor Least Squares subprograms provide for multiple matrix refinement of parameters; that is, the parameters may be placed in almost any matrix setup that the user desires. The Fourier subprogram provides for the calculation of Fourier-type maps which are on an inches/angstrom scale after being printed, if such is desired. It also provides for Fourier sections through any general plane which the user may desire.

Choosing an Asymmetric Unit of Data:

The following conventions should be used in indexing data for the following types of crystals. Most limitations are due to the Fourier subprogram.

Triclinic: No reflection may have two or more negative indices.

Monoclinic: No negative 'k' or 'l' indices are allowed.

Orthorhombic: All indices must be positive.

Tetragonal $\frac{4}{m}$: No negative indices; this convention is the same as orthorhombic.

Tetragonal $\frac{4}{m}mm$: All indices must be positive, and $h \geqq k$.

Cubic $\frac{2}{m}\bar{3}$: All indices must be positive and $h \leqq l$, $k \leqq l$.

Cubic m3m : All indices must be positive and $h \leqq k \leqq l$.

Hexagonal

Input to CRYRM:

The next chapter describes how one should prepare a card input deck for CRYRM. In the chapters which follow it the capabilities of the different subprograms and the format for the different control and data cards is discussed. In cataloging the different control cards, a certain scheme has been followed. Each control card is listed as two lines--the first line shows what must appear on the card, excluding certain constants which may be required; the second line consists of a series of "===="'s, these being replaced by appropriate characters when a constant or optional field may or must be placed in the corresponding position on the card. Where an = sign appears in the second row, exactly what appears in the first row (whether it is a character or blanks) must appear on the card. An example follows:

```
DISPERSION CORRECTION
===========================fff======△△△△△△======△△△△△△==========================
```

This card is described in Chapter 5. DISPERSION CORRECTION must appear on this kind of card. A form factor identifier must appear in col. 28-30. A constant must appear in col. 37-42. A constant may appear in col. 49-54. Unless otherwise specified, any non-integral constant may be entered by actually punching a decimal point somewhere in the alloted field, for example, 2.534, .005, 31. , etc. In most cases, the constant may also be punched with the decimal point assumed to be between two specified columns; for example, if the second row of the card description shows XXXXXX for the field of a constant, six columns are allowed for this constant, the decimal point being assumed between the second and third columns if none is punched (the . below the XXX's denotes the assumed decimal point.)

Programming for CRYRM:

All programs for CRYRM are written in the MAP Symbolic Programming Language and are assembled in the absolute mode. Whereas this type of programming requires more time and effort on the part of the programmer than, say, Fortran, many more things can be done in the computer; and the programs which are produced are very much more efficient in both space and time than programs written in some compiler language. For programs written for CRYRM there is available, in the CRYRM library, a subprogram which will do such things as tracing through a certain section of a subprogram or printing out certain sections of memory at different times for aid in finding errors in the subprogram.

Chapter 2

The CRYRM Running Deck

The preparation and arrangement of cards for a CRYRM running deck is discussed here. The deck described here is suitable for card input through the present Caltech setup-- cards are read by the 7040 and placed on disk for the 7094. The $ENDJOB card is not required for jobs submitted from remote consoles. Minor modifications (see Appendix 1) are needed if input is to be from a magnetic tape loaded by some other computer. No provisions have been made for input directly through the slow on-line card reader into the 7094.

Beginning and End of a Deck:

The first card must always be a $ID card. This card causes the accounting to be signed on and signals the beginning of a new job. The second card must always be $EXECUTE CRYRM. This card causes the CRYRM system to be given control of the 7094 if it is not already in control. (Actually the order of these two cards is reversed by the 7040, so the $EXECUTE CRYRM card reaches the 7094 before the $ID.) The last card in all decks must be a $ENDJOB card. This signals the system that the input for this job has been finished, so after any pending calculations are finished, the job will be signed off, and the next job (if there is one) will be read in.

The In-Between Cards:

Between the $EXECUTE CRYRM and the $ENDJOB cards are placed the system and subprogram control and data cards which cause the required calculations to be accomplished. None of these cards may begin with a $ character.

The arrangement of the deck is diagrammed in Figure 2. The system control cards may be divided into two groups: A. those which do not cause the loading of and transfer to a subprogram, and B. those which do. The NAME and COMMENT cards, for example, are Group A system control cards. They do not require a subprogram to complete the action which they specify. The STRUCTURE FACTOR CALCULATION and the FOURIER cards, on the other hand, are examples of Group B system control cards. They cause subprograms to be called into memory to complete the action called for.

After the leading cards, any number of Group A system control cards may be placed, or if none are required, they may be skipped. Next comes a single Group B control card, asking for calculations which require a subprogram. This causes the system to load the required subprogram into memory from the library. An overlay deck may follow the Group B card. If so, the system processes this overlay deck before giving control of the 7094 to the subprogram, but after loading the subprogram into memory. (Overlay decks are not usually required). Since at this point the subprogram has control, any subprogram control cards,or data cards,which may be needed by the subprogram must be placed next in the deck. When the subprogram has completed its calculations it will return control to the system; therefore, the card following the subprogram deck must be a system control card. This may be more Group A cards, if desired; a Group B card, in which case the previous sequence is repeated, except for different calculations; or if nothing else is required, the $ENDJOB card.

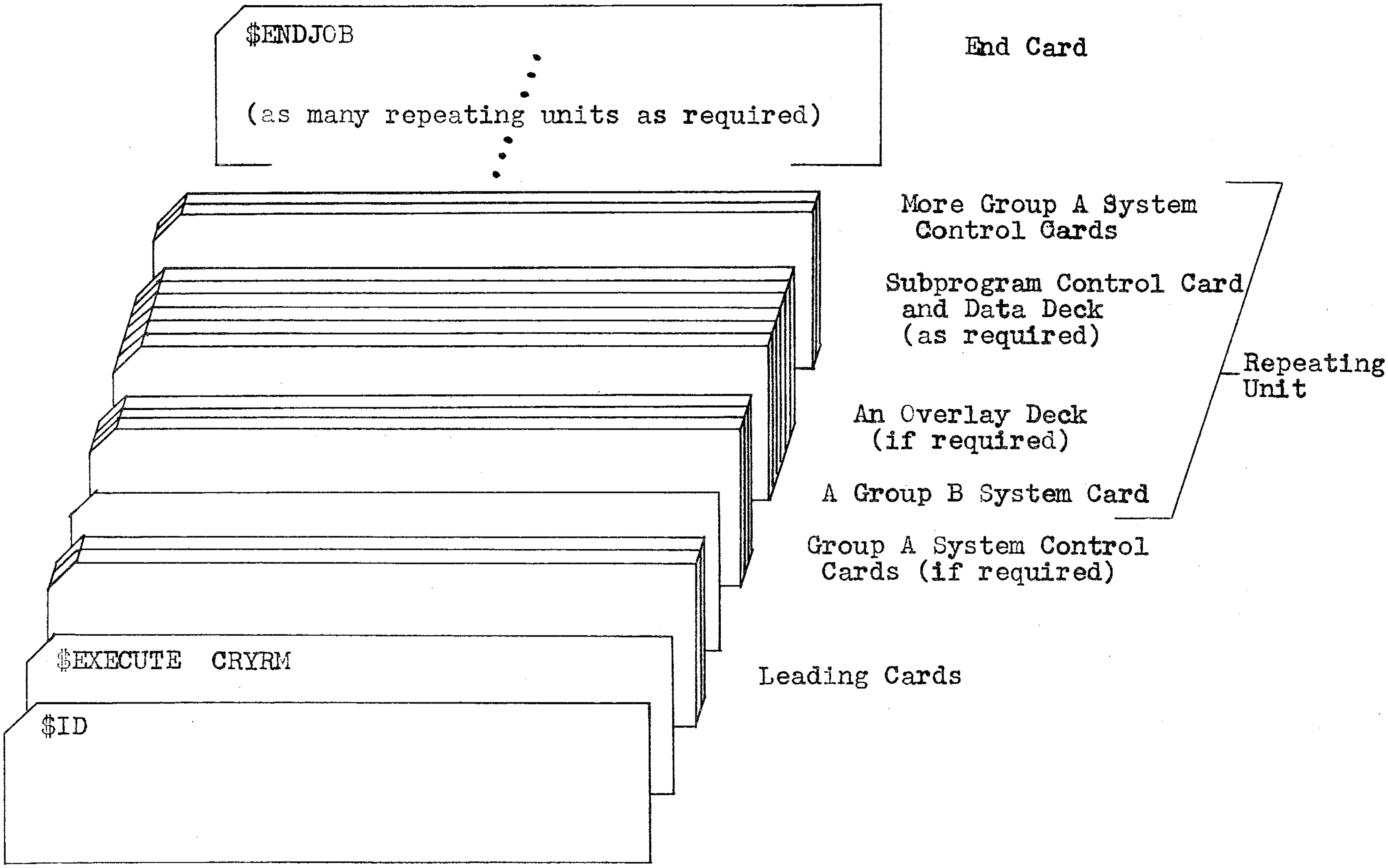

$ENDJOB
(as many repeating units as required)
End Card
More Group A System Control Cards
Subprogram Control Card and Data Deck (as required)
Repeating Unit
An Overlay Deck (if required)
A Group B System Card
Group A System Control Cards (if required)
$EXECUTE CRYRM
Leading Cards
$ID

Figure 2.

The Color Code:

The color of the cards which are loaded into the computer in no way affects the calculations which are done, but a certain color code has been adopted to help the operators and to make the assembly and interpretation of a running deck easier.

BROWN--------	Reserved for $ID cards only.
RED SOLID----	Reserved for $ENDJOB cards only.
RED STRIPE---	Reserved for all other Group A and Group B system cards.
NO COLOR-----	Reserved for data cards.

All users should follow the above code. The first two items pertain to all jobs run at the Caltech computing center, not only CRYRM jobs. Cards of colors other than those specified above are suitable for subprogram control cards. Certain recommendations can be made which the user should follow if it is convenient.

1. Put all least-squares or structure factor subprogram decks on GREEN cards.
2. Put Fourier-type subprogram input on GRAY STRIPE cards.
3. Put Inital Data Processing input, except data cards, on YELLOW STRIPE cards.

Hints and Warnings:

1. Put at least two or more cards between the $EXECUTE CRYRM card and the $ENDJOB card. Only one may sometimes cause trouble.
2. Be careful with system control cards which print a message to the operator asking for loading of tapes. Don't use a sequence which prints him the same message twice.
3. If a dump of 7094 core storage is desired, even if the job finishes with no apparent error, this may be forced by requesting a dump at the beginning of the deck, and then making a deliberate error (for example, a PATTERSON card without a space group number) on the card before the $ENDJOB card.

requesting a dump at the beginning of the deck, and then making a deliberate error (for example, a PATTERSON card without a space group number) on the card before the $ENDJOB card.

4. When estimating the time required for a set of calculations, remember that the job will be terminated with an error if this time (times a small factor--slightly greater than one) is exceeded. It is not possible to give a false low estimate of time in order to get a higher running priority.

5. When a DATA TAPE or DATA TAPE EXTRA are to be loaded, the operator of the computer should be warned beforehand so that he may have the correct reel mounted when CRYRM asks him for it. DATA TAPE will be loaded on Channel 'A' and DATA TAPE EXTRA on Channel 'B'. For example, if a DATA TAPE whose name is CRY-210 is to be loaded, tell the operator "Tape CRY-210 on Channel 'A' ".

Examples of Running Decks:

Example 1.

```
$ID                DJD109A,CH98057,1
$EXECUTE           CRYRM
DATA TAPE             CRY-206
PATTERSON                                 SPACE GROUP NO.    12
THREE-DIMENSIONAL
$ENDJOB
```

The above cards will cause a three-dimensional Patterson map to be calculated for the structure whose cell size and observed structure factors are on tape CRY-206.

The DATA TAPE card will cause a message to the operator asking for the loading of tape CRY-206.

Example 2.

```
$ID           DJD504C,CH98057,7
$EXECUTE      CRYRM
DATA TAPE        CRY-210                                                  System Control Cards
LEAST  SQUARES  CYCLE  A10       SPACE GROUP NO.    33
WEIGHTING FUNCTION 1.000 00000 2.445-.6398 .2127 00000 00000     EXT WT
FULL MATRIX      SHIFT FACTOR= 1.0
SCALE FACTOR=  1.0
ANISO     1    .0060183   .0121941   .1089064 -.0004748 -.0010085 -.0067519

ANISO     9    .0556236   .0096826   .0873284 -.0017957 -.0014676 -.0091156
BLOCK MATRIX     SHIFT FACTOR= .5
COORD     1  N     .003025   .195415   .043551
                                                                          Structure-Factor
COORD    13  H     .241900   .078522 -.344734                             Least Squares
NO REFINEMENT                                                             Subprogram
ISO      10   3.5                                                         Control Deck


ISO      13   3.5
END ATOMIC INFORMATION
RECYCLE
DISTANCES, ANGLES, AND PLANES                                     A Group B System Card
COMPLETE DISTANCES AND ANGLES  3.5
PLANE 1,2,3,4,5,6,7,8,9,10,11,12,13                               Distances, Angles and
WEIGHT1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,0.0,0.0,0.0,0.0         Planes Subprogram Deck
$ENDJOB
```

This input deck will do two cycles of least squares, getting its reflection data from the DATA TAPE CRY-210; then it will calculate complete distances and angles and a least-squares plane using the coordinates generated by the last cycle of least squares.

input deck will do two cycles of least squares, getting its reflection data from the DATA TAPE CRY-210; then it will calculate complete distances and angles and a least-squares plane using the coordinates generated by the last cycle of least squares.

Example 3.

```
$ID             DJD505A,CH98057,4
$EXECUTE        CRYRM
EDIT DATA ON TAPE CRY-210
CELL  15.5507 9.3751  3.6626
KIND FC  N  H
SUBSTITUTIONS
         0     2     0   55.62
         0     2     0   55.62   0.0
DELETIONS
         7    12     1   3.66
END OF DATA
LEAST  SQUARES  CYCLE             SPACE GROUP NO.    33
WEIGHTING FUNCTION HUGHES 1/FOBS SQUARED 4FMIN=   4.0
ATOMIC PARAMETERS ARE ON DATA TAPE.
DISTANCES, ANGLES, AND PLANES
COMPLETE DISTANCES AND ANGLES  3.5
COMMENT          FINAL MOLECULAR PLANE FOURIER.
FOURIER                           SPACE GROUP NO.    33
CENTER POINT .1374 .2629 .0088       6.5   A. HORIZ.   6.5   A. VERTICAL
GENERAL PLANE, DIRECTION COSINES .1839    -.3451    .9203    ,Z CONSTANT HORIZ.
$ENDJOB
```

This input deck calls in four subprograms. First comes an edit of the DATA TAPE,and then it does a cycle of least squares using the atomic coordinates which are on the DATA TAPE as input. Next it calculates distances and angles, then does a general plane Fourier, using the calculated structure factors which were put on the DATA TAPE by the previous least squares cycle for phasing purposes.

Chapter 3

System Control Cards.

Discussion:

The CRYRM system control cards may be divided into two groups--A. those which do not cause the loading of any subprogram, and B. those which cause the loading of and transfer of control to some subprogram.

Since Group A control cards do not cause the system to give up control, they should be followed by other system control cards. Group B control cards, however, do cause the system to give control to a subprogram. If a PROGRAM CHANGES FOLLOW or a SYSTEM CHANGES FOLLOW control card immediately follows a Group B control card, the overlays which follow will be placed in memory after the subprogram is loaded in from the library. Overlays may be in the form of OCTAL cards and/or absolute-mode column binary cards. Immediately after such changes are processed (or immediately after the Group B control card if no changes are present) the subprogram is given control; therefore, the next cards in the deck should be subprogram control cards if they are required. A system control card should follow the subprogram cards, since the subprogram will return control to the system when it has finished its calculations.

All jobs must begin and end with system control cards.

Space Group Designation:

The space group numbers used in CRYRM are the same as in the International Tables for X-ray Crystallography, Volume 1. Where there is a choice of origin, the origin is taken at the center of symmetry. All monoclinic space groups are with the 'b' axis unique. Space groups which fall into the above category require only a space group number.

In addition to the standard space groups, alternate orientations of some space groups are provided for. These are indicated by giving the standard number followed by a space group letter. Below is a list of special space groups.

Designation	Space Group
5A	A2
7A	Pa
7N	Pn
8A	Am
9A	Aa
12A	A 2/m
13A	P 2/a
13N	P 2/n
14A	P 2_1/a
14N	P 2_1/n
15A	A 2/a

System Control Cards   Group A

System Control Cards - Group A

This type of control card does not cause the loading of and transfer to a subprogram.

```
$ID
===============IIIXXXX,WWXXXXX,X,X,D============================================
```

Required Constants:

Initials in columns 16-18 are required. These must be followed immediately by a job identification number (up to four characters). This must be followed by a comma and a two-(alphabetic)-character department identification code, followed immediately by a five-digit charge number.

Optional Constants:

Maximum execution time, pages of output, and dump request fields may follow the charge number. The field after the second comma on the card is considered the maximum execution time field (minutes). That after the third comma is the pages-of-output field. If a D follows the fourth comma, a dump will be given if an error is made during the job. If no maximum execution time is specified, 2 minutes is assumed. In order to skip over a field to get to the next, put two successive commas--do not put blank spaces between commas.

Action:

1. The job is signed on and the timer is started.
2. Memory is cleared, except for system and buffers.

Note:

If a job exceeds the maximum execution time (times a certain extra factor), the job will be stopped and signed off with an error indication.

$EXECUTE CRYRM

Action:

This card causes the CRYRM system programs to be loaded into memory from the library. CRYRM is then given control. If CRYRM is in control when this card is processed, the card is ignored.

$ENDJOB

Action:

This card signals the end of input for the current job.

ERROR DUMP REQUESTED

Action:

This card sets an indicator which causes a memory dump if the job ends with an error.

PROGRAM CHANGES FOLLOW

Action:

This card prepares CRYRM to accept overlays to subprograms. All overlays in the form of absolute mode column binary cards and/or OCTAL cards must follow this card.

Note:

This card will not permit changes to the CRYRM system. If such overlays are needed, use a SYSTEM CHANGES FOLLOW card. If a complete binary deck for a subprogram is to be read in, it will contain overlays to CRYRM and therefore a SYSTEM CHANGES FOLLOW card is required.

overlays are needed, use a SYSTEM CHANGES FOLLOW card. If a complete binary deck for a subprogram is to be read in, it will contain overlays to CRYRM and therefore a SYSTEM CHANGES FOLLOW card is required.

OCTAL

```
======XXXXX=XXXXXXXXXXXX========================================================
```

Required Constants:

1. Five-digit octal number giving location to be overlaid in col. 7-11.
2. Twelve-digit octal number in col. 13-18. All zero digits must be punched as such.

Action:

The 12-digit octal number replaces the contents of the location specified by the 5-digit octal number.

Note:

This card must be preceded by a PROGRAM CHANGES FOLLOW, SYSTEM CHANGES FOLLOW, another OCTAL, or a binary card.

NAME

```
======ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ========================
```

Optional Field:

Any alphabetic or numeric identifier (for example--a compound name) may be placed in col. 7-56.

Action:

1. The fifty characters specified above are placed in the page heading.
2. If a DATA TAPE has been loaded, these characters are placed where they will be put on the DATA TAPE, if the DATA TAPE is copied when first used during the job (as, for example, the least squares routines do).

Note:

If a DATA TAPE is loaded after this card has been processed, the fifty characters from the tape will overlay those in the page heading.

```
COMMENT
=======ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
```

Action:
Card is listed with the output. Anything in col. 8-80 can be placed in the listing by this card.

```
DATA TAPE
==================XXXXXXXXXXXX======DONT=TEST=NAME.=============================
```

Required Constants:
A tape name, not to exceed 12 characters, beginning in col. 19.

Optional Field:
If DONT TEST NAME. appears in col. 37-51, the name testing feature of tape loading is suppressed.

Action:

1. A message will be printed to the operator, asking him to load the tape whose name is given on a certain tape drive on Channel 'A'. The system then waits until this is completed.
2. Unless name testing has been suppressed, the tape name recorded as the first 12 characters at the beginning of the tape will be compared to that punched on the card. "Not equal" is assumed to be an error in loading the tape. If the tape is not loaded correctly after two tries, the job is ended with an error.
3. The fifty-character page heading identifier which has been placed on the tape by a NAME card is placed in the page heading unless name testing has been suppressed.

Note:

1. If this card is used the operator should be warned that he will be asked to load the tape whose name is given above on Channel 'A'. This will allow him to have this tape ready.
2. The tape name should be identical, column for column, to that which appeared on the PREPARE DATA TAPE card which was used to make it.
3. A DATA TAPE card is not required before a PREPARE DATA TAPE, EDIT

This will allow him to have this tape ready.

2. The tape name should be identical, column for column, to that which appeared on the PREPARE DATA TAPE card which was used to make it.

3. A DATA TAPE card is not required before a PREPARE DATA TAPE. EDIT

DATA TAPE EXTRA

```
==================XXXXXXXXXXXX======DONT=TEST=NAME.=============================
```

Required Constants:

A tape name, not to exceed 12 characters, beginning in col. 19.

Optinnal Field:

If DONT TEST NAME. appears in col. 37-51, the name testing feature of tape loading will be suppressed.

Action:

1. A message will be printed to the operator, asking him to load the tape whose name is given on a certain tape drive on Channel 'B'. The system then waits until this is completed.

2. Unless name testing has been suppressed, the tape name recorded on the tape is compared to that on the card. "Not equal" is assumed to be a tape loading error. If the tape is not loaded correctly after two tries, the job is ended with an error.

Note:

1. The operator should be warned that he will be asked to load the specified tape on Channel 'B'. This will allow him to have the correct tape ready.
2. Name testing should always be suppressed on new tapes.
3. Note 2 spaces between TAPE and EXTRA.

System Control Cards - Group B

This type of control card will cause the loading of and transfer to a subprogram. It should be followed by overlays if these are required, and then by any subprogram control cards which may be required.

INITIAL DATA PROCESSING

==

Action:

The Initial Data Processing subprogram is loaded into memory, and given control after any waiting PROGRAM CHANGES are processed. See Chapter 4 for a discussion of required subprogram control cards.

PREPARE DATA TAPE

==================XXXXXXXXXXXX==

Required Constants:

A tape name, not to exceed 12 characters beginning in col. 19.

Action:

1. A message will be printed to the operator, asking him to load the tape whose name is given on a certain tape drive on Channel 'A'. The system then waits until this is completed.
2. The Data Tape Preparation and Editing subprogram is loaded into memory from the library, and given control after any waiting PROGRAM CHANGES have been processed.

See Chapter 5 for required subprogram control cards.

Note:

The tape name given here will be stored on the tape for name testing during future loading.

required subprogram control cards.

Note:

The tape name given here will be stored on the tape for name testing during future loading.

EDIT DATA ON TAPE
================XXXXXXXXXXXX==

Required Constants:
A tape name, not to exceed 12 characters, beginning in col. 19.

Action:
1. A message will be printed to the operator, asking him to load the tape whose name is given on a certain tape drive on Channel 'A'. The system then waits until this is completed.
2. The tape name recorded on the tape is compared to that on the card. "Not equal" is assumed to be a tape loading error. If the tape is not loaded correctly after two tries, the job is ended with an error.
3. The Data Tape Preparation and Editing subprogram is loaded into memory from the library, and given control after any waiting PROGRAM CHANGES have been processed.

See Chapter 5 for required subprogram control cards.

Note:
1. The operator should be warned that he will be asked to load the specified tape on Channel'A'.
2. The tape name should be identical column for column to that which appeared on the corresponding PREPARE DATA TAPE card.

LIST DATA ON TAPE
================XXXXXXXXXXXX==

Required Constants:
A tape name, not to exceed 12 characters, beginning in col. 19.

Action:
1. If a data tape has not been loaded, actions 1 and 2 above are done.
2. The Data Tape Listing subprogram is loaded into memory and given control after any waiting PROGRAM CHANGES have been processed.

No subprogram control cards are required. The DATA TAPE will be listed.

```
STRUCTURE FACTOR CALCULATION     SPACE GROUP NO.
===================================================XXXY=========================
```

Required Constants:
Space group number in col. 52-54, right justified, letter (if required) in 55.

Action:
1. Space group number is converted and stored.
2. Appropriate version of the structure-factor least squares subprogram is loaded into memory from the library.
3. It is set for the STRUCTURE FACTOR mode, then is given control after any following PROGRAM CHANGES are processed.

Note:
Error if DATA TAPE has not been loaded prior to this card.

```
LEAST  SQUARES  CYCLE            SPACE GROUP NO.
=======================ZYYY=======================XXXY=========================
```

Required Constants:
Space group number in col. 52-54, right justified, letter (if required) in 55.

Optional Constants:
Cycle letter in col. 24 and cycle number in col. 25-27. If these are left blank, the previous cycle letter and incremented previous cycle number are used. The cycle number is required if this is the first cycle with the present DATA TAPE.

Action:
1. and 2. as for STRUCTURE FACTOR CALCULATION.
3. Subprogram is set for the LEAST SQUARES mode, then is given control after any following PROGRAM CHANGES are processed.

Note:
Error if DATA TAPE has not been loaded prior to this card.

See Chapter 6 for required subprogram control cards for the structure-factor least squares subprogram.

processed.

Error if DATA TAPE has not been loaded prior to this card.

See Chapter 6 for required subprogram control cards for the structure-factor least squares subprogram.

FOURIER SPACE GROUP NO.

===XXXY=========================

Required Constants:
Space group number in col. 52-54, right justified, letter (if required) in 55.

Action:

1. Space group number is converted and stored.
2. The Fourier subprogram is loaded into memory from the library.
3. It is set for the FOURIER mode and then is given control after any following PROGRAM CHANGES are processed.

Note:
It is assumed that,prior to this, calculated structure factors have been put on the DATA TAPE if the DATA TAPE is to be used for input to the Fourier.

PATTERSON SPACE GROUP NO.

===XXXY=========================

Required Constants:
Space group number in col. 52-54, right justified, letter (if required) in 55.

Action:

1. Space group number is converted and stored.
2. The Fourier subprogram is loaded into memory from the library.
3. It is set for the PATTERSON mode and then is given control after any following PROGRAM CHANGES are processed.

See Chapter 7 for required subprogram control cards for the Fourier subprogram.

```
                                   SPACE GROUP NO.
DIFFERENCE FOURIER
=================, NO S.F. ======================XXXY=========================
                 , AND  L.S.
```

Required Constants:
Space group number in col. 52-54, right justified, letter (if required) in 55.

Optional Fields:

1. If , NO S.F. appears in col. 19-27, the partial calculated structure factors are assumed to have been put on the DATA TAPE by a previous calculation, and the entry to obtain them is suppressed.
2. If AND L.S. appears in col. 20-28, the entry to obtain the partial calculated structure factors is made in the LEAST SQUARES mode, and a cycle of least squares is done at that time.

Action:

1. Space group number is converted and stored.
2. Unless it has been suppressed, the correct version of the structure-factor least-squares subprogram is loaded into memory and entered in the STRUCTURE FACTOR or LEAST SQUARES mode for the calculation of required partial and complete structure factors.
3. After the return from the structure-factor least-squares subprogram, the Fourier subprogram is loaded into memory from the library and is entered to obtain the required difference fouriers.

Note:

1. Input of structure factors from DATA TAPE is required.
2. If an entry is to be made for the calculation of structure factors, a WEIGHTING FUNCTION card, an OMIT ATOMS card, and an atomic information deck (see Chapter 6) <u>must</u> follow this card. The subprogram control cards for the Fourier subprogram (see Chapter 7) should then follow these.
3. Any PROGRAM CHANGES for the Fourier subprogram should be placed immediately after the atomic information deck.

immediately after the atomic information deck.

COMPLETE DIFFERENCE FOURIER SPACE GROUP NO.

=======================================XXXY=======================

Required Constants:
Space group number in col. 52-54, right justified, letter (if required) in 55.

Action:

1. Space group number is converted and stored.
2. The Fourier subprogram is loaded into memory from the library.
3. It is set for the COMPLETE DIFFERENCE FOURIER mode and then is given control after any following PROGRAM CHANGES have been processed.

Note:

1. Input of structure factors from DATA TAPE is required.
2. It is assumed that,prior to this, calculated structure factors have been put on the DATA TAPE.

See Chapter 7 for required subprogram control cards.

DISTANCES, ANGLES, AND PLANES

Action:
The Distances, Angles, and Planes subprogram is loaded into memory from the library, and given control after any waiting PROGRAM CHANGES have been processed.

See Chapter 8 for required subprogram control cards.

AUTOMATIC DIFFRACTOMETER

Action:

The Automatic Diffractometer Calculations subprogram is loaded into memory from the library, and given control after any waiting PROGRAM CHANGES have been processed.

See Chapter 9 for required subprogram control cards.

UNIT CELL LEAST SQUARES

Action:

The Unit Cell Least Squares subprogram is loaded into memory from the library, and is given control after any waiting PROGRAM CHANGES have been processed. (Note two blanks after UNIT.)

See Chapter 10 for required subprogram control cards.

PUBLICATION LIST

Action:

The Publication Listing subprogram is loaded into memory from the library, and is given control after any waiting PROGRAM CHANGES have been processed.

Note:

A FORM card must follow this card. See Chapter 11 for the format of this subprogram card.

TEMPERATURE ELLIPSOIDS

Action:

The Temperature Factor Ellipsoid Inversion subprogram is loaded into memory from the library, and is given control after any waiting PROGRAM CHANGES have been processed.

See Chapter 12 for required subprogram control cards.

STATISTICAL PHASING SPACE GROUP NO. XXXY

Required Constants:

Space group number in col. 52-54, right justified, letter (if required) in 55

Action:

1. Space group number is converted and stored.
2. The Statistical Phasing subprogram is loaded into memory from the library, and is given control after any waiting PROGRAM CHANGES have been processed.

See Chapter 13 for required subprogram control cards.

PROTEIN CALCULATIONS

Action:

The Protein Calculation subprogram is loaded into memory from the library, and is given control after any waiting PROGRAM CHANGES have been processed.

Note:

For information on this subprogram, one should contact R. H. Stanford.

Chapter 4

Initial Data Processing

Programming Contributions:

D. Duchamp, film factor scaling, internal weighting, L. P., etc.
C. M. Gramaccioli, scaling between groups.
Mrs. B. Stroll, precession Lorentz-polarization correction.

Discussion:

This subprogram is a completely intermeshed set of routines for use in data reduction. It provides for entry of data directly as read from films (or at any later stage of processing), calculation of film factors, assigning of standard deviations to individual film readings, averaging between films, Lorentz-polarization corrections, absorption corrections, and scaling together of several groups of correlated intensity measurements. Any of these calculations may be done separately, or they may all be done at once.

This subprogram is called by the INITIAL DATA PROCESSING System Control card. The output may be placed on a DATA TAPE EXTRA for input to the Data Tape Preparation and Editing Subprogram or for re-input to a later phase of this subprogram. Initial input of reflection data is by cards (two types--long or short format). Thereafter data entry may be by cards or by a DATA TAPE--this DATA TAPE having been prepared as a DATA TAPE EXTRA by a previous computer run of this subprogram or the Automatic Diffractometer Subprogram.

Program Description:

The calculations of this subprogram are accomplished in three passes of the data. The particular problem of the user may require any one, any two sequential, or all three passes. If more than one pass is required, magnetic tape links between passes are provided automatically. The scheme is outlined in the table below.

Organization of the Initial Data Processing Subprogram

	Type[1] of Reflection Input	Calculations	Printed Output	Magnetic Tape Output
PASS 1	1. Long Format Data Cards 2. Long Format Mag. Tape	1. Setting of "less thans" 2. Calculation of film and pack factors. 3. Calculation of constants for internal weighting 4. Assigning internal weight	Results of all calculations are printed. If card input, cards are listed.	A Long Format Mag. Tape
PASS 2	If film factor scaling: 1. Long Format Data Cards 2. Long Format Mag. Tape if not: 1. Short Format Data Cards 2. Short Format Mag. Tape	1. Film factor scaling 2. Assigning internal weight (if not done in PASS 1) 3. Setting of "less thans" (if not done in PASS 1) 4. L. P. Correction 5. Absorption Correction 6. "Extra" correction	A reflection by reflection list of all results and correction factors	A Short Format Mag. Tape
PASS 3	1. Short Format Data Cards 2. Short Format Mag. Tape	1. Sorting of reflections according to indices. 2. Calculation of factors between correlated groups of data 3. Scaling and combining of groups	A reflection by reflection list of results, certain statistical results	A Short Format Mag. Tape

[1] Input is either by cards or magnetic tape, not both.

of groups

[1] Input is either by cards or magnetic tape, not both.

The Subprogram Deck:

The subprogram deck is composed of two parts: (1) a series of cards feeding in constants and designating options desired, and (2) the data. The cards in the first part of the deck (1) may be in any order. A CELL card and at least one calculation-requesting card must be entered.

If the reflection data is to be entered by magnetic tape, the second part of the deck (2) is a single DATA ON DATA TAPE card. When using this option, a magnetic tape with data in the proper format must be loaded as a DATA TAPE prior to the entry of the INITIAL DATA PROCESSING card.

If the reflection data is to be entered by cards, the second part of the deck (2) consists of either a long-format or a short-format data deck. The last card of either deck must be END OF DATA. The long-format data deck is designed for use in entering readings directly from films--specifically where several different exposures of the same photograph are involved, as for example, in the multiple-film Weissenberg method. The short format data deck is used for group by group (layer by layer in the case of films) entry of single intensity measurements, as, for example, for input for a Lorentz-polarization correction, or for inter-group scaling.

In the control card description the Type 1 cards are for entering constants and designating options (the first part of the subprogram deck). The Type 2 cards are for use in data entry.

Short-Format Card Input:

The short-format data deck consists of groups of short-format data cards. Each group should be preceeded by a LAYER card or a COUNTER DATA FOLLOWS card giving the necessary information about the group and separating it from other groups. For example, when entering Weissenberg data taken about the "a" axis, the zero layer forms one group, the first layer forms another group, the second layer another and so on--each group being preceeded by an appropriate LAYER card identifying that group. As many groups as are needed may be entered, one following the other. The last card of the data deck must be an END OF DATA card. This card indicates the end of the subprogram deck, and therefore only one is allowed. An example of entry of two groups (the zero and first layer about the "a" axis) follows.

Example 1:

```
    LAYER   0   AXIS   A
             0     2     3  25.2
             0     2     4  150.1
             :     :     :   :
             0    13    22  5.3
    LAYER   1   AXIS   A
             1     3     4  220.
             1     3     5  2.3
             1     3     8  15.32
             :     :     :   :
             1    13    20  4.2
    END OF DATA
```

The order of the data cards within the groups is not important to the subprogram, and is up to the user.

Long-Format Card Input:

Long-format card input is designed for use in entering film intensity data directly as read from films. It is designed for the multiple-film technique. For this option the reflections must be divided into groups as described under Short-Format Data Input.

The long-format data card provides for up to ten entries--each entry corresponding to a different film. The films should be grouped into "packs"--all films within a pack having been taken by the same exposure, as, for example, three films which were in the camera at the same time during a multiple-film Weissenberg exposure. A pack may contain from one to ten films. Within a pack, the attenuation between films is due only to the absorption of x-rays by the film.

multiple-film Weissenberg exposure. A pack may contain from one to ten films. Within a pack, the attenuation between films is due only to the absorption of x-rays by the film.

The packs are arranged on the long-format data card in order of decreasing exposure--most intense pack first. Within a pack, films must be arranged in order of decreasing intensity--the most intense film first. For example, if a layer was taken by the Weissenberg method in two exposures, a long exposure of three films and a short exposure of four films, the entries on the long-format data card would be assigned as follows

Entry 1	Entry 2	Entry 3	Entry 4	Entry 5	Entry 6	Entry 7	Entry 8	Entry 9	Entry 10
Darkest Film, Long Exposure	Middle Film, Long Exposure	Weakest Film, Long Exposure	Darkest Film, Short Exposure	Second Film, Short Exposure	Third Film, Short Exposure	Weakest Film, Short Exposure	Not Used	Not Used	Not Used

All entries should be left-justified and there should be no "not used" entries between film entries. For each entry that has been assigned to a film, there must be, within the group of reflections, at least one reflection with a reading from the film corresponding to that entry. If in the above example, a reading from the weakest film of the long exposure pack does not occur on any reflection card within the group, no entry should be reserved for that film--the four entries for the short exposure should be moved to the left one entry, Entry 7 being placed in the "not used" section.

The entry format for a given group is entered into the subprogram by use of a PACK FORM card, which tells which packs are where. The entry format must be constant within a group, but may change between groups. The designations punched on the PACK FORM card are used as headings for the columns of figures in the printed output.

In punching a long-format data card, the indices are punched, then intensity readings are punched in the entries corresponding to the films from which they came. If a reflection was not read on a given film--either because it was too dark or too weak--nothing should be punched in the entry for that film. The handling of reflections which are too weak to be observed on any film is described later under "Unobservable Reflections."

The long-format card input deck must be arranged in the following manner: First must come a PACK FORM card, specifying the entry format for the first group of data. The LAYER card for the first group must follow immediately. Then come all the long-format data cards for the first group. If the entry format of the second group is different from the first, a new PACK FORM card must immediately precede the LAYER card for the second group. (If no PACK FORM card occurs, the entry format specified on the last entered PACK FORM card is assumed.) Then follow the LAYER card and the long-format data cards for the second group. All other groups are entered in the same fashion--PACK FORM (if required), LAYER, long-format data cards. A LAYER card must precede the long-format data cards of each group, and a PACK FORM card may precede the LAYER card. At the end of all data input, and END OF DATA card must follow. Remembering that the designations punched on the PACK FORM cards are used as column headings in the printed output, some users like to enter a new PACK FORM card before every LAYER card; this is permitted. An example of the entry of long-format data is given later in this chapter.

Tape Output:

If a magnetic tape has been loaded on DATA TAPE EXTRA before the INITIAL DATA PROCESSING card enters, the final tape output will be placed on this tape in either the long or short format depending on what calculations were last completed. The normal option in this case is to write a new tape from the beginning; however, the present output may be added onto previous output (provided the format is the same) by use of an ADD TAPE OUTPUT TO DATA TAPE EXTRA card.

The short format output tape from this subprogram is suitable for use as input to this subprogram or to the Data Tape Preparation and Editing Subprogram. It does not contain sufficient information for Structure-Factor, Least Squares or Fourier calculations.

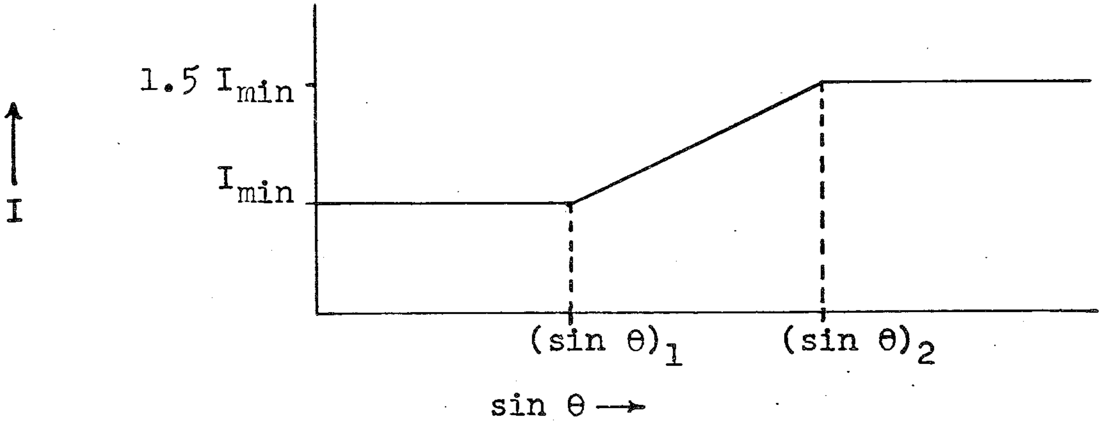
Unobservable Reflections:

Unobservable reflections are handled by setting a minimum observable intensity and saying that the reflection is "less than" this minimum observable value. "Less than" reflections are designated by making their weight negative.

Unobservable reflections may be entered in one of two ways. First, a value may be punched for each "less than" reflection, the weight being

Unobservable reflections may be entered in one of two ways. First, a value may be punched for each "less than" reflection, the weight being

made negative to indicate a "less than". Or, the weight may be made negative and no value punched--the minimum observable intensity being assigned internally by the function plotted below:



The rise at high $\sin \theta$ is designed to take into account α_1, α_2 splitting at high angle. Of course, if this option is used, unobservable reflections whose minimum observable intensities are not approximated by the above function (for example, low angle "smeared out" reflections on upper-level Weissenberg photographs) must still be entered individually. The SET LESS THANS card is used to effect automatic setting of minimum observable values.

Internal Weights:

Provisions have been made for assigning standard deviations to individual film intensity readings by the following function

$$\sigma(I) = \frac{1}{w_e}\left[a + bI + \frac{c\,I^2}{(I_{max} - I)^2}\right]\left[1 + d\ e^{-g(.5 - \sin^2\theta)^2}\right] \qquad \text{for } I < I_{max}$$

$$\sigma(I) = \text{a very large number} \qquad \text{for } I \geq I_{max}$$

where $\sigma(I)$ is the standard deviation in intensity I; w_e is an external weight punched on the data card; a, b, c, d, g and I_{max} are constants; and sin θ is the Bragg sin θ.

The function is designed primarily for use with readings obtained by visual estimation using a standard intensity scale (strip). It will be discussed more throughly elsewhere, but in brief: the "a" term is designed to raise the error assigned to a reading low on the intensity scale; the "bI" term usually dominates for a reading in the sensitive intermediate region of the intensity scale; the " $cI^2 /(I_{max} - I)^2$ " term is designed to raise the error of a reading high on the intensity scale; the second factor (with the exponential) is designed to raise the error of readings which are obscured by α_1, α_2 splitting. In the preceeding statements "raising the error" means raising the error relative to reflections which do not have the particular characteristic described. The w_e is provided so that the standard deviation of certain readings with special undesirable characteristics may be increased.

Use of the internal weighting option is effected by entering a USE INTERNAL WEIGHTS card. The above function may be used in two modes. First, all the constants--a, b, c, d, g, and I_{max}--may be entered, and weights assigned by direct use of the function.

Alternatively, an I_{min} and an I_{max} may be entered, and the program will assign constants as follows:

$a = I_{min}/3$,
$c = 0.1$,
$d = 0.25$,
$g = 50.0$,

and b will be determined by a statistical analysis of the consistency between reflections which are multiply observed in the sensitive region of the intensity scale. The "b's" so determined are calculated on an axis-by-axis basis and give an indication of how good the data about a given axis is. In practice, calculated values of "b" usually range from 0.05 to 0.15. Also in practice the determination of "b" is very sensitive to punching errors on the long-format data cards. An accurate value for "b" is dependent upon the multiple readings of each given reflection actually being <u>independent</u> visual observations, and of course on a large "b" is dependent upon the multiple readings of each given reflection actually being <u>independent</u> visual observations, and of course on a large

number of multiply-observed reflections upon which to apply the statistical criteria.

In the assigning of internal weights to readings of high-angle reflections, the I which is used in the function is the reading multiplied by a factor of 2/3 to take into account α_1, α_2 splitting since the reading is actually a combination of two lesser readings. The transition from low to high angle is made by the same method as used in assigning minimum observable intensities to less than reflections.

Once the standard deviations have been assigned to film readings, they may be carried through the remainder of the Initial Data Processing Subprogram (by propagation-of-error methods) to give a standard deviation for each scaled corrected reflection. The quantity designated a "weight" in this subprogram is the reciprocal of the standard deviation in the square of the structure factor amplitude (the intensity).

Calculation of Film and Pack Factors:

A film factor is defined as the ratio (always >1) between intensities of consecutive films within a pack. A pack factor relates two adjacent packs and is defined as the ratio of the intensity of the least intense film entered for the longer exposure pack, to the most intense film entered for the shorter exposure pack. The two films related by a pack factor are adjacent in the entry format. The factor relating two adjacent films in the entry format is the number used to multiply the readings from the later film to bring them on the same scale as the readings from the preceeding film. Thus by applying consecutive factors, each reading can be brought on the same scale as Entry 1.

The calculation of these factors can be effected by use of the CALCULATE FILM FACTORS card. Three methods of calculation are available, designated as follows:

LOGARITHMIC $$\log(K_{ab}) = \frac{\Sigma w_i \left[\log\left(\frac{I_a}{I_b}\right)_i\right]}{\Sigma w_i}$$

RATIO AVERAGE $$K_{ab} = \frac{\Sigma w_i \left(\frac{I_a}{I_b}\right)_i}{\Sigma w_i}$$

DISCRETE SUMS $$K_{ab} = \frac{\Sigma (I_a)_i}{\Sigma (I_b)_i}$$

where K_{ab} is the factor between film "a" and film "b", I_a is the reading from film "a" of a reflection (i) observed on both film "a" and film "b", I_b is the corresponding reading from film "b", w_i is the weight (calculated from the standard deviations of the readings) assigned to each multiple observation (i). The summations are over all reflections (i) which are observed on both films. The first two methods require assigned standard deviations for weighting. In practice, the LOGARITHMIC method has proven most satisfactory, and is therefore recommended.

Since the film factor is due only to attenuation from absorption by the films (in the usual case), all film factors within a given group should be the same. Accordingly, an average film factor for each layer is calculated. By taking into account the angle of incidence between the x-rays and the film, the film factors of all the groups are averaged to give an overall film factor for all the data entered. These factors once calculated are available for film-factor correction and averaging. The results of the calculation of film and pack factors are punched out on cards for possible future use.

Film Factor Correction and Averaging:

Film factor correction, as used here, means the multiplying of readings under all entries by the proper factor to put them on the same scale as the first film (Entry 1). After the readings are scaled up, they are averaged to give a value for the observed intensity for each reflection. After this is done, all reflections within a given group should be on the same scale. The scale may differ, however, between groups.

The DO FILM FACTOR CORRECTION card is used to specify this option. If film factor correction and averaging is to be done, either the film and pack factors must be calculated in the save run, or they must be read in using FACTOR_AXIS cards (and an OVERALL FILM FACTOR card if required).

The DO FILM FACTOR CORRECTION card is used to specify this option. If film factor correction and averaging is to be done, either the film and pack factors must be calculated in the save run, or they must be read in using FACTOR-AXIS cards (and an OVERALL FILM FACTOR card if required).

When doing film factor scaling, the pack factors have to be used individually and, in general, have no relation to each other. The user, however, may select how the film factors (those factors due only to attenuation from absorption by the films) are handled. One may select (1) a GENERAL film factor, meaning that one overall film factor will be used for all the data; or (2) a LAYER film factor, meaning that a separate average film factor for each group (layer) is used for all film factors within the group; or (3) an INDIVIDUAL film factor, meaning that each film factor is entered individually--the factors between films within a pack are handled as though they were unrelated to one another. The overall film factor is entered assuming normal incidence of the x-rays to the film, and is automatically scaled up for groups where non-normal incidence occurs. The GENERAL film factor is recommended except in special cases where one would expect the film factors not to be the same.

By the appropriate entries on the DO FILM FACTOR CORRECTION card, one may use weighted or unweighted arithmetic or log(geometric) averages. The weighted option is standard. It assumes that weights, derived from the internally assigned standard deviations, are available. Weighted averages are not possible when the internal weighting option is not being used. When the weighted option is being used, the standard deviations of the readings are correctly scaled and combined to give a standard deviation of the average. This is done by standard propagation-of-error methods with the assumption that the errors in film and pack factors are negligible. The weighted LOG AVERAGE is recommended.

Lorentz-Polarization Corrections:

For film data this subprogram has provisions for making Lorentz and polarization corrections for films taken by precession, rotation-oscillation, equi-inclination Weissenberg, anti-equi-inclination Weissenberg, general inclination Weissenberg, and powder techniques. The photographic method used in collecting a group of data, along with any special constants for that group, is entered on the LAYER card which preceeds the group. Groups taken by different methods may be mixed at will within a given run.

The Lorentz and polarization correction applied to data which follow a COUNTER DATA FOLLOWS card is the same as the zero-level rotation-oscillation correction. Of course, any of the film Lorentz-polarization corrections may be also used for cc nter data by just calling the data "film data."

When this correction (and also the absorption correction, etc.) is applied to an intensity value, the weight assigned to this intensity is divided by the same factor in order to keep the standard deviation of this intensity on an absolute scale.

Absorption Correction:

Provisions are made for the application of absorption correction factors by this subprogram; however, the subprogram contains no routine to calculate these factors. If a user desires to do absorption corrections he should enter a DC ABSORPTION CORRECTION card, and using the PROGRAM CHANGES FOLLOW option, he should enter a short octal overlay program at location "ABSORP" in the Initial Data Processing Subprogram. This overlay program will be entered for each reflection; it should return to location "SRUN1" with the accumulator containing the floating-point binary absorption correction factor to be multiplied by the intensity value of the reflection being processed. The main subprogram will handle all printing of the absorption factor and its actual application.

The "EXTRA" Correction:

Provisions have been made for the user to enter an overlay program by means of the PROGRAM CHANGES FOLLOW option, and to have this overlay program executed. The octal overlay program should be entered starting at location "XTRART". After it completes its calculation, it should return to location "SRUN2" with the accumulator containing a floating-point binary factor to be applied to the intensity of the reflection being processed. (Use a floating-point 1.0 if no factor is desired). This routine may be used to do special manipulation of the indices, intensity, or weights of reflections. It is entered once for each reflection. The correction factor which it generates is applied to the intensity at the same time as the Lorentz-polarization correction factor is applied.

generates is applied to the intensity at the same time as the Lorentz-polarization correction factor is applied.

Inter-Group Scaling:

Note: The following description is abstracted from a more complete account written by C. M. Gramaccioli, who is the programmer of the Pass 3 part of the Initial Data Processing Subprogram.

(a) Introduction:

The inter-group scaling part of the Initial Data Processing Subprogram provides for the scaling together of correlated groups of reflections in order to produce a set of intensity data with all reflections on the same scale. The method used in calculating scale factors will be described more completely elsewhere, but in general it is an iterative least-squares determination of scale factors--one for each group--which minimizes the function

$$\sum_k w_k \left(c_i I_{ik} - c_j I_{jk}\right)^2$$

subject to the constraint $\sum_i c_i^2 = \text{constant},$

where w_k is the weight given the pair of observations, I_{ik} and I_{jk}, of the same reflection from groups "i" and "j" respectively, and c_i and c_j are the scale factors to be applied to groups "i" and "j" respectively. Up to 15 values per reflection may be scaled together. The presence of good correlation between groups is important. Needless to say, poor correlation of certain groups will produce inferior scaling factors for these groups.

A common use of this option is to scale together several layers of photographic intensities taken about two or more axes. Another case, although less common, is the scaling together of several correlated groups of counter data.

Use of the inter-group scaling routines is effected by entering a SCALE AXES TOGETHER card. If, instead, a SORT REFLECTIONS card is entered, the routines are entered in the sorting mode. In this mode the calculation of scale factors is bypassed and the subprogram can be used as a reflection sorting routine. If it does find more than one value for a given reflection, however, it will combine the values. This mode may be useful for putting a group of randomly collected counter intensities--already on a common scale--into an order which is more convenient for future calculations.

(b) Sorting:

Before any calculations are done, all reflection data are sorted according to indices. The sorting usually is the most time consuming part of inter-group scaling, but it usually can be speeded up considerably by proper entry of systematic limitations on the ranges of the indices. Without any limitations, the sorting routine will scan the data for all possible reflections within the sphere of reflection. This can be a considerable waste of time, especially where large cells are involved.

If one or more indices have only positive values, this fact should be entered on the SCALE AXES TOGETHER card. If further restrictions may be placed on certain indices, a RANGE card should be entered. When a RANGE card is used, reflections are ignored which have at least one index outside of the range indicated. Because of this, this option may be used to select certain reflections from a larger collection of reflections; but, if scaling is being done, care must be taken not to destroy correlation.

(c) Arranging the Final Output:

As a result of a procedure to make the sorting the fastest possible, the output is arranged with the slowest moving index corresponding to the one which has the smallest range of values. The fastest moving index is the one immediately preceeding the slowest, i.e. if the slowest moving index is "ℓ", the fastest is "k"; and if the slowest is "h", the fastest is "ℓ". If another arrangement is desired, the slowest moving index can be specified on the SCALE AXES TOGETHER card. In the case of very large ranges of indices, this may be necessary, because the above mentioned criterion of the fastest sorting can lead to the impossibilitv of performing the sorting at all. In this case, the best way to proceed is to make the index with the largest range of values the slowest moving

(d) Weights:

largest range of values the slowest moving

(d) Weights:

For the best scaling together of the data groups, standard deviations of the intensity measurements are necessary. These can either be entered with the data, or, more frequently, they are assigned by the previous parts of the subprogram or by some other subprogram.

When standard deviations of the individual measurements are available, they are correctly modified by the scale factors when putting measurements on the same scale, and are used in obtaining the weighted averages of multiple measurements of the same reflection, being properly propagated to give a standard deviation in the average intensity. The reciprocal of this standard deviation is printed under "WEIGHT" in the output, and is suitable for use (unmodified) in final stages of parameter least squares. With tape input, standard deviations stored previously are automatically entered. For short-format card input, the reciprocals of the standard deviation in each intensity measurement should be punched in the WWWW position on the reflection data card.

For groups of data in which there are no standard deviations available--as, for example, with data scaled by a procedure extraneous to the CRYRM system--the NO SIGMAS option should be requested on the SCALE AXES TOGETHER card. This option sets

$$w_k = \sqrt{w_{ik}}\sqrt{w_{jk}}\left(\frac{1}{c_i I_{ik}} \cdot \frac{1}{c_j I_{jk}}\right)$$

where w_{ik} and w_{jk} are the "WEIGHTS", entered externally, for the observations I_{ik} and I_{jk} respectively, and w_k, I_{ik}, I_{jk}, c_i, and c_j are as before. In the averaging of the final results when the NO SIGMAS option is used, the weighted average uses only the external "w_{ik}"; and the "WEIGHT" printed for the average intensity reflects only the multiplicity of observation and the assigned external weights. These should not be considered as estimates of the reciprocal of the standard deviation in $F_{obs}{}^2$ and therefore should not be used in an unmodified form in parameter least-squares weighting.

(e) Axial Weights:

By use of the WEIGHT card, (up to 10 permitted), weights may be assigned to observations according to axes. Such weights multiply the weights of all observations about the axes concerned. This option is designed principally for the case where standard deviation input weights are not available, but it can be used as well if one is not satisfied with the assigned standard deviations of observations about a certain axis.

(f) Input Scaling Factors:

By use of the SCALING FACTOR card, input scale factors (up to 100 permitted) may be entered. This could reduce the number of iterations needed in calculating scale factors, especially if certain groups are very different in scale from the others. For such cases input scale factors are recommended for the groups far different in scale from the first group--but not required.

(g) Absolute Scaling of Data:

A rather different use of input scaling factors is possible. It is usually convenient to have the output structure factors as close to the absolute scale as possible. All data are scaled with respect to the first group entered. If the absolute scale factor is known for one group, this group can be entered first, and an input scale factor entered for it.

Note: All scale factors are printed with the scale factor for the first group fixed at unity. So, if a scale factor was entered for the first group, all scale factors will be printed divided by this factor.

(h) Best Fit Polynomials:

After the scaling is done, three polynomials are printed, giving the best approximations for expressing $\sigma(I)$ or $\sigma(F)$ as a function of the type

After the scaling is done, three polynomials are printed, giving the best approximations for expressing σ(I) or σ(F) as a function of the type

indicated. These σ's refer to the scaled single observations and not to their averages. In the normal procedure, these expressions are calculated from the given standard deviations of all the observations. Iber's method (Acta Cryst., 9, 652 (1956)) is used when these standard deviations are not given (see Weights).

(i) Discrepancy Factor:

In order to give an idea about the agreement of the scaled observations belonging to different sets of data, the "discrepancy factor" is printed. This factor is

$$D = \frac{\sum_i \left| I_{ave} - I_i \right|}{\sum_i I_i}$$

in which all I's are scaled.

(j) Limited Iteration Cycles:

In some cases, during the calculation of scaling factors, convergence may not be easily attained--usually because of errors in punching the data--and an enormous waste of computer time could result. To prevent this, if convergence is not attained after 5 cycles (usually 3 cycles required), the routine prints the values of the scale factors at this time and scales all the data according to these. Whereas such scale factors are usually inferior, the printed output may help in finding the trouble. The iteration cycle limit may be changed by the appropriate entry on a RANGE card.

An Example of Long-Format Input:

In the following example, only the subprogram deck is shown. Several system control cards must precede these cards; the system control card immediately before must, of course, be INITIAL DATA PROCESSING.

Example 2.

```
DC FILM FACTOR CORRECTION, GENERAL     FILM FACTOR,          LOG AVERAGE.
CALCULATE FILM FACTORS, LOGARITHMIC
OVERALL FILM FACTOR =    3.80
SCALE AXES TOGETHER       HKL  ARE ONLY POSITIVE,            SLOW  L
SCALING FACTOR  = 4.56     FOR LAYER   0    AXIS  C
DO LP CORRECTION
USE INTERNAL WEIGHTS, I(MIN) = 4.00      I(MAX) = 65.00
SET LESS THANS, MINIMUM READING =     2.00
CELL  11.189  10.463  7.220
PACK FORM            AZ6    AZ7   AZ8   BZ2   BZ3   BZ4   BZ5
LAYER   0   AXIS   C
    2    0    0                                           21.
    4    0    0                               48.   11.75 3.3
    6    0    0             36.   9.    14.   5.
    :    :    :
    :    :    :
    6   10    0 -
    3   13    0     12.5
PACK FORM            AY6    AY7   AY8   BY2   BY3   BY4
LAYER   1   AXIS   C
    2    0    1         5 40.   11.7  17.5  4.7
    :    :    :
        Many more groups
    :    :    :
   10    4    6   5 110.  29.       3 19.3
END OF DATA
```

Type 1 Initial Data Processing Control Card

Type 1 Initial Data Processing Control Cards:

These cards are used for the input of constants and the specification of options. They may be arranged in any order, but all must come before any Type 2 cards are entered.

ADD TAPE OUTPUT TO DATA TAPE EXTRA

Action:

1. The format of the output presently on the DATA TAPE EXTRA is checked against the format of the output to be placed there. These must agree, or an error message will be printed and the additions will be suppressed.
2. The information presently on the DATA TAPE EXTRA is skipped over and the present tape output is written immediately following.

Note:

A DATA TAPE EXTRA must have been loaded prior to the call to the Initial Data Processing Subprogram.

APPLY EXTRA CORRECTION

Action:

The subprogram is set to enter an Extra Correction Routine during the second pass of the data, at the same time as the Lorentz- and polarization factors are being calculated.

Note:

The user must overlay the subprogram with a routine to do the EXTRA CORRECTION if he wants to use this option. See previous discussion.

```
CALCULATE FILM FACTORS,
========================LOGARITHMIC                    ===================
                        RATIO AVERAGE
                        DISCRETE SUMS,MAX XXXXXX  MIN XXXXXX
```

Required Field:

The method for calculating film factors (see previous discussion) must be specified beginning in col. 25 as shown above. For the DISCRETE SUMS option, values for the maximum and minimum intensity readings must be punched in col. 43-48 and col. 55-60 respectively.

Action:

The subprogram is set to calculate film and pack factors by the method specified on the card.

```
CELL
======AAAAAAA=BBBBBBB=CCCCCCC=AAAAAABBBBBBCCCCCC========================
```

Required Constants:

Parameters giving the crystallographic unit cell dimensions must be entered as follows: "a" in col. 7-13, "b" in col. 15-21 (if no "b" is punched, it is assumed to be equal to "a"), "c" in col. 23-29 (if no "c" is punched, it is assumed to be equal to "b"), the angle "α" in col. 31-36, the angle "β" in col. 37-42, the angle "γ" in col. 43-48 (if any of the angles are not punched, they are assumed to be equal to 90 degrees). The axial lengths are punched in Angstrom units; the interaxial angles are punched in degrees~~units~~. and decimals.

Action:

The unit cell parameters are entered into the subprogram.

Note:

A CELL card must be entered in every Initial Data Processing Subprogram Deck.

Note:
A CELL card must be entered in every Initial Data Processing Subprogram Deck.

CELDEV

```
======XXXXXXX=XXXXXXX=XXXXXXX=XXXXXXYYYYYYXXXXXX================================
```

Required Constants:

Standard deviations in unit cell parameters must be entered as follows: σ(a) in col. 7-13, σ(b) in col. 15-21, σ(c) in col. 23-29, σ(α) in col. 31-36, σ(β) in col. 37-42, σ(γ) in col. 43-48. If any standard deviations are not entered, they are assumed to be zero. The standard deviations in the axial lengths must be in Angstrom units; the standard deviations in the interaxial angles must be in degrees ~~units~~. and decimals.

Action:

The standard deviations are entered into the subprogram, but are not used in any calculations. They will be placed on the DATA TAPE EXTRA if this option is being used.

DO ABSORPTION CORRECTION

```
================================================================================
```

Action:

The subprogram is set to apply an absorption correction factor in the second pass. The branch to the subroutine to calculate the absorption correction factor is set.

Note:

To use this option, the user must overlay the subprogram with an octal subroutine for calculation of the absorption correction factor. See the previous discussion.

```
DO FILM FACTOR CORRECTION,            FILM FACTOR
===========================GENERAL   ============,NO SIGMAS,LOG AVERAGE.========
                           LAYER
                           INDIVIDUAL
```

Required Field:

The method for obtaining film factors must be designated beginning in col. 28, as shown above. "GENERAL" means that one overall film factor, properly increased for non-normal incidence of the x-rays on the film, will be used. "LAYER" means that each group will have a separate film factor, used whenever a film factor is required within the group. "INDIVIDUAL" means that each pair of films is considered separately.

Optional Fields:

1. If ",NO SIGMAS" is punched in col. 50-59, the subprogram is set not to expect weights which are equal to 1/σ(reading). The normal mode is to expect these weights. If weights are not furnished, this option should be used.
2. If ",LOG AVERAGE." is punched in col. 60-72, when the averaging is done, the subprogram will use a weighted average of the log of the readings, instead of a weighted average of the readings themselves. The weighted average of the readings themselves is the standard option.

Action:

The subprogram will apply film factor corrections to the intensity measurements, and average the results by the method specified.

Note:

1. This card is the one to use even if there are no "film" factors, but only "pack" factors. See the discussion.
2. If this option is to be used either a CALCULATE FILM FACTORS card must be in the subprogram deck, or the film and pack factors must be entered using a FACTOR-AXIS card for each group.

DO LP CORRECTION

```
DO LP CORRECTION
================================================================================
```

Action:

The subprogram is set to calculate and apply Lorentz and polarization correction factors to all reflections.

Note:

The subprogram decides what type of correction--precession, equi-inclination Weissenberg, etc.--separately for each group from the information read in before the group.

```
FACTOR-AXIS         LAYER
============±I±I±I=======NNY==AAAAAABBBBBBCCCCCCDDDDDDEEEEEEFFFFFFXXXXXXYYYYYY==
```

Required Constants:

1. An axis designation in col. 13-18. The axis designation given here must be exactly--column for column within the six columns allowed--the same as the axis designation punched on the LAYER card of the group being referred to.
2. A layer designation in col. 26-28. The layer designation given here must be exactly--column for column--the same as the layer designation on the corresponding LAYER card.
3. A list, as required, of not more than 6 individual film factors, each entry allowed six columns, first entry in col. 31-36, second in col. 37-42, etc. If the user wants to enter a film factor for the whole group instead of individual film factors, this film factor may be punched in col. 31-36, with col. 37-66 being left blank.
4. Not more than two pack factors, as required, the first in col. 67-72, the second, if there is one, in col. 73-78.

Action:

The film and pack factors which are punched, together with the axis and layer identifiers of the group to which they refer are entered into the proper lists in the subprogram.

Note:

If film factors are being calculated, the calculated film factors will be used in preference to those entered on FACTOR-AXIS cards.

OVERALL FILM FACTOR =

```
=======================XXXXXX=========================================
```

Required Constant:

An overall film factor in col. 25-30. The factor for normal incidence of the x-rays on the film should be punched here.

Action:

An overall film factor is entered into the subprogram.

Note:

If film factors are being calculated, the overall film factor entered here will be used in preference to the calculated overall film factor.

PRINT FREQUENCY DISTRIBUTIONS OF FILM FACTORS

Action:

The subprogram is set to accumulate and print a table of frequency of occurrence of various values of the film factor within each group and for all groups. Various statistical parameters, calculated for the cumulative distribution, are also printed.

Note:

1. All film factors are corrected to normal incidence before being placed into the distribution.
2. This option is set up for a film factor of the order of 3.75 and should not be used if this is not the case.
3. The calculation and printing of this distribution is done in the first pass of the data--at the same time as film factors etc. are being calculated.

RANGE

```
======H=hmin=hmax=K=kmin=kmax=L=lmin=lmax===ITERATE UP TO xx CYCLES============
```

Optional Constants:

1. Values of maximum and minimum indices for limiting the range of the sorting in the third pass of the data. Any one or all three indices may be limited as needed. If an index is to be limited the alphabetic character ("H", "K", or "L") must be punched in the proper column (7, 19, or 31 respectively), then the minimum value for the index must be punched as a signed integer, right justified in the proper columns (9-12, 21-24, or 33-36 respectively) and the maximum value, also as a signed integer right justified, must be punched in the proper columns (14-17, 26-29, or 38-41 respectively). If an index is to have only a single value, this value is punched in the minimum position, and nothing is punched in the maximum position.
2. If "ITERATE UP TO" is punched in col. 45-57, an integer, right justified, is punched in col. 59-60, and "CYCLES" is punched in col. 62-67, the subprogram will be set to iterate up to the specified number of cycles before assuming that convergence cannot be attained.

Action:

The limiting values for the indices and/or the limiting number of iterations is read into the subprogram which is set accordingly.

Note:

1. RANGE is designed for use with the SCALE AXES TOGETHER card or the SORT REFLECTIONS card. Using it without either of these cards makes no sense.
2. For large unit cells, limiting the range of the sorting may save considerable time, since the subprogram will look for all reflections within the sphere of reflection for the wavelength being used, unless it is told not to.

SCALE AXES TOGETHER

```
=========================iiiiiARE ONLY POSITIVE, NO SIGMAS. SLOW  i=============
```

Optional Fields:

1. If certain of the reflection indices take on only positive values in the data, the corresponding alphabetic character--"H", "K" or "L" should be punched anywhere within col. 25-30, and "ARE ONLY POSITIVE" should be punched in col. 31-47.
2. If weights equal or proportional to 1/σ(Intensity value) are not being supplied with the reflections, "NO SIGMAS." should be punched in col. 50-59.
3. If the user wants to specify which index shall be slowest moving after the reflections are sorted, "SLOW" should be punched in col. 61-64, and the alphabetic character--"H", "K", or "L"--designating the index to be slowest moving should be punched in col. 67.

Action:

The subprogram is set to, in the third pass of the data, sort the reflections according to indices, calculate scaling factors for the various data groups to put them on the same scale as the first data group entered, and then scale the reflections together, producing one average value for each reflection.

SORT REFLECTIONS

```
=========================iiiiiARE ONLY POSITIVE, NO SIGMAS. SLOW  i=============
```

Optional Fields:

These are the same as on the SCALE AXES TOGETHER card.

Action:

Same as with the SCALE AXES TOGHTHER card, except the calculation of scaling factors is bypassed. In this mode the subprogram functions as a reflection sorter.

```
SCALING FACTOR =          FOR LAYER       AXIS
==================XXXXXX=============NNY========±I±I±I==========================
```

Required Constants:

1. A scaling factor, to be applied to intensity values, in col. 19-24.
2. An axis designation in col. 49-54. The axis designation given here must be exactly--column for column within the six columns allowed--the same as the axis designation punched on the LAYER card of the group being referred to.
3. A layer designation in col. 38-40. The layer designation given here must be exactly--column for column--the same as the layer designation on the corresponding LAYER card.

Action:

An input scaling factor for the data group specified is entered into the subprogram.

```
SET LESS THANS, MINIMUM READING =
====================================XXXXXX======ZZZZZZ======ZZZZZZ==============
```

Required Constants:

A value for the minimum observable intensity in col. 37-42.

Optional Constants:

Values for $(\sin \theta)_1$ and $(\sin \theta)_2$ in col. 49-54 and col. 61-66 respectively. For the significance of these values see page 4-7.

Action:

The subprogram is set to assign a minimum observable intensity to all reflections which have a negative weight, but no intensity values. The method discussed on page 4-7 is used. If no values are entered for $(\sin \theta)_1$ and $(\sin \theta)_2$ values of 0.78 and .90 respectively are assumed.

```
USE INTERNAL WEIGHTS, I(MIN) =          I(MAX) =
===============================XXXXX=============XXXXX==========================

USE INTERNAL WEIGHTS,   AXIS
==============================±I±I±I======AAAAAABBBBBBCCCCCCDDDDDDGGGGGGMAXMAX==
```

This card may take two forms as shown above. For more details see page 4-7.

Form 1.

Required Constants:

Values for the minimum and maximum intensity readings in col. 32-36 and 50-54 respectively.

Action:

The subprogram is set to determine by statistical analysis a value for the constant "b" in the standard deviation equation, assign the other constants, and assign to each intensity reading which is entered a standard deviation and a weight equal to the reciprocal of this standard deviation.

Form 2.

Required Constants:

1. An axis designation in col. 31-36. This designation must be exactly --column for column within the six columns allowed-- the same as the axis designations on the LAYER cards of the groups being referred to. If no axis designation is punched, the parameters which follow are taken to apply to all axes.
2. Values for the parameters "a", "b", "c", "d", "g", and "I_{max}" of the standard deviation equation in col. 43-48, 49-54, 55-60, 61-66, 67-72, and 73-78 respectively.

Action:

The parameters punched on this card are entered into the standard deviation equation for the axis (axes) specified. The subprogram is set to use this equation to assign to each intensity reading which is entered, a standard deviation and a weight equal to the reciprocal of this standard deviation.

```
WAVE LENGTH
===========XXXXXX==========================================================
```

Required Constant:

A value in Angstrom units for the wave length of x-rays used in col. 13-18. If instead of an Angstrom value, the characters "CU" or "MO" are punched in col. 17-18, a value of 1.54178 A. or 0.71069 A. respectively is assumed.

Action:

The wavelength is entered into the subprogram.

Note:

If no WAVE LENGTH card is entered the subprogram will assume copper radiation.

```
WEIGHT  =          FOR  AXIS
===========XXXXXX===========±I±I±I=========================================
```

Required Constants:

1. A weight in col. 13-18.
2. A axis designation in col. 31-36. This designation must be exactly--column for column within the six columns allowed--the same as the axis designations on the LAYER cards of the data groups being referred to.

Action:

Weights on an axis by axis basis are entered into the subprogram for use in the inter-group scaling section.

Note:

1. The weight given on this card multiplies the weight which comes in for each given reflection belonging to a group whose axis designation matches that on the WEIGHT card.
2. Up to ten of these cards may be entered in a given run.

Type 2 Initial Data Processing Cards:

These cards are used for the entry of data. All the Type 1 cards must come before the first Type 2 card.

The Short Format Data Card:

```
====I=HHHH==KKKK==LLLL==FFFFFF==WWWW============================================
```

Required Constants:

1. The indices h, k, and l must be punched in col. 7-10, 13-16, and 19-22 respectively, with the integer right justified. All minus signs should immediately precede the first significant digit of the integer.
2. A value for F_{obs} in col. 25-30.
3. A value for the weight of the reflection in col. 33-36, with the following special options possible: (1) If nothing is punched, the weight is set equal to 1.0; (2) if a minus sign is punched in col. 33, the weight is set equal to -1.0; (3) if a zero is punched in col. 34, the weight is set equal to 0.0. Of course, in addition to these special options, the weight may be punched with a punched decimal point or with the decimal point assumed between col. 34 and 35.

Optional Field:

If the character "I" is punched in col. 5, the value which is punched in col. 25-30 is treated as an intensity value, that is $(F_{obs})^2$.

Note:

The format described above is exactly the same as is used for the entry of data into the Data Tape Preparation and Editing Subprogram.

The Long Format Data Card:

```
=HHHH=KKKK=LLLL=WWV=IIIIV=IIIIV=IIIIV=IIIIV=IIIIV=IIIIV=IIIIV=IIIIV=IIIIV=IIII==
```

Required Constants:

1. The indices h, k, and l must be punched in col. 2-5, 7-10, and 12-15 respectively, as integers right justified. All minus signs should immediately precede the first significant digit of the integer.
2. A weight for all readings on the card in col. 17-18, with the special options: (1) if nothing is punched, a weight of 1.0 is assumed; (2) if a minus sign is punched in col. 17, a weight of -1.0 is used; (3) if a zero is punched in either col. 17 or col. 18, a weight of 0.0 results. The weight designated here multiplies the weights assigned to individual readings, and also any internally assigned weights.
3. A series of up to ten entries in col. 19-24, 25-30, 31-36, 37-42, 43-48, 49-54, 55-60, 61-66, 67-72, and 73-78. These entries are used, as required, for punching intensity readings from various films--each reading is punched in the entry assigned to the film upon which it was observed. Each entry has six columns. The first column of each entry is reserved for an individual weight for the reading being punched--if a 3 is punched for an individual weight, a weight of 0.3 results, with 1.0 being the weight assigned if nothing is punched. The last four columns of each entry are used for the entry of the intensity reading.

Note:

1. Only as many entries should be used as are needed. If no reading of the particular reflection was made on the film corresponding to a given entry, nothing should be punched within that entry--regardless of whether there was no reading because the reflection was too weak or because it was too strong.
2. A "less than" reading is indicated by a minus sign in col. 17.

PACK FORM

```
==================|AAAXX|=AAAXX|=AAAXX|=AAAXX|=AAAXX|=AAAXX|=AAAXX|=AAAXX|=AAAXX|=AAAXX|==
```

Required Fields:

Designations for assigning the entry format for the long format data cards which follow. As many as ten entries are possible, in col. 19-24, 25-30, 31-36, 37-42, 43-48, 49-54, 55-60, 61-66, 67-72, and 73-78 respectively with each entry occupying 6 columns. Anything the user desires may be punched in the last 2 columns of an entry. The second, third and fourth columns of an entry are used to designate the pack from which the film comes--if the first four columns of two successive entries are exactly, column for column, equal, then the two films are taken to be related by a "film factor"; if these columns are not exactly equal, the two films are taken to be related by a "pack factor". If "n" (less than 10) entries are to be made, only the first "n" entries should be used, the columns remaining must be left blank. There should be no blank entries between punched entries. No entry should be reserved for a given film if none of the reflections within the group which follows was observed on that film.

Note:

1. The columns for the entries on the PACK FORM card correspond to the columns for the corresponding entries on the long format data card.
2. The characters punched on the pack form card are used in page headings in the data listing which results.
3. A PACK FORM card must precede the first group of long format data cards, but thereafter need only be entered before a group if its entry format is different from that of the preceding group. Entry of a new PACK FORM card with each data group is allowed, even if the entry fromat remains the same.

LAYER AXIS

```
LAYER          AXIS
=======NNY========±I±I±IROTATION-OSCILLATION                             ==
                        GENERAL INCLINATION,MU = XXXXX
                        ANTI EQUI-INCLINATION
                        POWDER
                        PRECESSION          MU = XXXXXT1T1T1HHKKLLT2T2T2HHKKLL
```

Required Constants:

1. A layer designation in col. 8-10. The layer number must be punched as an integer, right justified, in col. 8-9. A layer letter, an alphabetic character, may be punched in col. 10. The layer letter is used if two separate data groups with the same layer number and axis number occur, the letter being different for each group. This is necessary because if the layer and axis designations are both equal, column for column, for two different data groups, the subprogram will not be able to distinguish between the groups.
2. An axis designation in col. 19-24. If the axis is one of the major axes, the character--either "A", "B", or "C"--may be punched in col. 21. Otherwise the axis must be punched by its crystallographic indices. The three indices, are punched, right justified, in col. 19-20, 21-22, and 23-24 respectively. Negative indices are allowed. No index should exceed nine.
3. If the method of obtaining the photograph was other than equi-inclination Weissenberg, the method must be punched, as shown above, beginning in col. 25, together with such other information as required. Both PRECESSION and GENERAL INCLINATION require a mu angle punched in col. 50-54, in degrees. In addition, precession photographs require that that the angle "tau" made by a line from the center of the film to the reflection spot, and the horizontal direction of the film be entered for two reflections in the same quadrant of the film. The "tau" angles, in degrees, are punched for the first and second reflections in col. 55-60, and 67-72 respectively. Following these angles, the indices of the reflections are punched as integers, right justified, as shown above.

COUNTER DATA FOLLOWS

Action:

This card is used in place of a LAYER card if counter data is to be processed.

DATA ON DATA TAPE

Action:

The subprogram is set to get reflection input from a DATA TAPE.

Note:

1. For this to be used, a DATA TAPE with the correct format data must have been loaded prior to the call to the Initial Data Processing Subprogram.
2. No LAYER, END OF DATA, PACK FORM, COUNTER DATA FOLLOWS,or data card should be used together with this card.

END OF DATA

Action:

This card is used at the end of a subprogram deck (except when DATA FROM DATA TAPE is being used) to notify the subprogram that all the data cards have been read in.

Note:

The last card in any Initial Data Processing Subprogram Deck must be either an END OF DATA card or a DATA ON DATA TAPE card.

Chapter 5

DATA TAPE Preparation and Editing

Programming Contributions: D. Duchamp, tape preparation and editing.
T. A. Beineke, statistical sections.

Discussion:

This subprogram provides for the preparation and editing of a DATA TAPE which is suitable for input to other programs. It is called into memory by a PREPARE DATA TAPE or an EDIT DATA ON TAPE card. The DATA TAPE carries on it the cell parameters and their deviations, the coefficients for calculating $\sin^2\theta/\lambda^2$, a tape name, a list of the types of atomic form factors, and for each reflection: h,k,l,F(obs), a weight for F(obs), $\sin^2\theta/\lambda^2$, the values for the different types of atomic form factors, and some other constants useable by the Structure-Factor Least Squares Subprogram. The Structure-Factor Least Squares subprogram adds the calculated structure factors and an Atomic Parameter Deck to the DATA TAPE. Provisions have been made for doing Wilson statistics to determine an approximate scale factor and an overall temperature factor, and for calculating points for a plot to approximate whether a structure is centric or acentric by the method of Howells, Phillips, and Rogers. These statistical calculations may only be made at the PREPARE DATA TAPE time.

The Subprogram Deck for the PREPARE Mode:

If a form factor curve is to be temporarily added to the table of form factor curves, the SIN TH and F cards for doing this should be first cards in the deck. These should be followed by CELL and CELDEV cards (not required for tape input from DATA TAPE EXTRA).

Any of the following control cards which may be required must come next: DISPERSION CORRECTION, ~~WAVELENGTH,~~ DO WILSON STATISTICS, APPLY WILSON SCALING FACTOR, NUMBERS OF ATOMS, DO HOWELLS. The next card following these must be a KIND F card, which specifies which atomic form factors are to be put on the DATA TAPE. Every subprogram deck for the PREPARE mode must have a KIND F card.

Input of reflection data may come either on Short Format Data Cards or from a DATA TAPE EXTRA prepared by the Initial Data Processing or Automatic Diffractometer subprograms. If reflections are to come in on cards, a deck of Short Format Data Cards, followed by an END OF DATA card must follow the KIND F card. If reflection input is by DATA TAPE EXTRA, a DATA ON DATA TAPE EXTRA card should follow the KIND F card. This is all that is required and permitted in the PREPARE mode subprogram deck.

The Subprogram Deck for the EDIT Mode:

The exact same SIN TH, F, CELL, DISPERSION CORRECTION, and KIND F, cards as were used in the preparation of the DATA TAPE being edited should appear here. If a CELL card was not used in the PREPARE mode deck, one which has the same cell parameters as those on the DATA TAPE should be used here. No WAVELENGTH, DO WILSON STATISTICS, APPLY WILSON SCALING FACTOR, NUMBERS OF ATOMS, DO HOWELLS, or CELDEV cards may be in the EDIT mode deck.

The input of reflection data for the edit must come in on Short Format Data Cards. Three types of reflection editing are allowed--SUBSTITUTIONS, DELETIONS, and ADDITIONS. To delete certain reflections, put in a DELETIONS card followed by Short Format Data Cards with h, k, l, and F(obs) punched on them exactly as they appear on the DATA TAPE, one data card for each reflection which is to be deleted. To add certain reflections to the end of the DATA TAPE, put in an ADDITIONS card, then follow this card by a Short Format Data Card for each reflection which is to be added. To do substitutions, put in a SUBSTITUTIONS card, then follow this card by two Short Format Data Cards for each substitution which is to be made--the first data card should have h, k, l, and F(obs) punched on it exactly as it appears on the DATA TAPE at present, the second should contain h, k, l, F(obs), and the weight as they are to appear after the substitution has been performed.

The data deck described above should be followed by an END OF DATA card. If this card has DO EDIT NOW punched on it, the editor will immediately do the requested edits. If DO EDIT NOW is not on the card, the editor will look at the amount of reflections to be edited. If the amount is less than some preset value in the editor, the editor will not do the editing, but will assume that a call to the Structure-Factor Least Squares subprogram is going to follow and will further assume that this subprogram will do the editing from tables which the editor leaves in memory. If the size of the edit is too large, the editor will do the edit. The END OF DATA card should be the last card in the EDIT mode subprogram deck.

No provisions have been made for changes in the cell parameters which are on the DATA TAPE, but this is no limitation since the cell parameters from the DATA TAPE will not be used in a subprogram if the user simply inserts a CELL card with the updated cell parameters in the input to the subprogram which is to use the information.

Atomic Form Factors:

Stored in the library with the DATA TAPE Preparation and Editing subprogram are form factor curves for a large number of atoms and ions. If none of these are suitable for a certain type element, the user may feed in his own form factor curve. The curve is entered in the form of a table of sin θ/λ versus atomic form factor. The sin θ part of the table is entered first by using SIN TH cards. If one card is filled before the table is exhausted, continue on another SIN TH card--as many as are needed to give all the points in the table. After the sin θ part of the table has been entered, the atomic form factor part is entered using F cards, the form factors being placed in the same position on the F cards as their corresponding sin θ/λ on the SIN TH cards. If several atomic form factor tables with the same sin θ/λ parts are to be entered, the SIN TH sequence need not be repeated. It is important that all the SIN TH cards for a given table come before any of the F cards for that table.

Atomic form factor curves are identified by a three character alphabet code, some of which may be blank. It should be noted that blank, C, blank is not the same curve as C, blank, blank or blank, blank, C. A table of the form factors which are stored on the library follows.

Identification Code	Element	Source of Curve
123		
H	H	Inter. Tables, Vol. 3, pp 202-203.
LI1	Li^{+1}	Ibid.
BE	Be	Ibid.
B	B	Ibid.
C	C(valence)	Ibid, with minor modifications.
N	N	Ibid.
N1	N^{+1}	Interpolation between other Nitrogen curves in the above reference.
O	O	Inter. Tables, Vol. 3, pp 202-203.
OHF	$O^{-1/2}$	Interpolation between other oxygen curves in the above reference.
F	F	Inter. Tables, Vol. 3, pp 202-203.
F-	F^{-1}	Ibid.
NA	Na	Ibid.
NA+	Na^{+1}	Ibid.

F-	F	Ibid.
NA	Na	Ibid.
NA+	Na^{+1}	Ibid.

Identification Code 123	Element	Source of Curve
MG	Mg	Inter. Tables, Vol. 3, pp 202-203.
MG2	Mg^{+2}	Ibid.
AL	Al	Ibid.
AL3	Al^{+3}	Ibid.
SI	Si	Ibid.
P	P	Ibid.
S	S	Ibid.
CL	Cl	Ibid.
CL-	Cl^{-1}	Ibid.
K	K	Ibid, pp 204-205.
K+	K^{+1}	Ibid.
CA	Ca	Ibid.
CA2	Ca^{+2}	Ibid.
TI	Ti	Ibid.
V	V	Ibid.
VT	V	Ibid, p 210.
CR	Cr	Ibid, pp 204-205.
CRT	Cr	Ibid, p 210.
MN	Mn	Ibid, pp 204-205.
FE	Fe	Ibid.
FE3	Fe^{+3}	Ibid.
FET	Fe	Ibid, p 210.
CO	Co	Ibid, pp 204-205.
CO2	Co^{+2}	Ibid.
COT	Co	Ibid, p 210.
NI	Ni	Ibid, pp 204-205.
NI2	Ni^{+2}	Ibid.
NIT	Ni	Ibid, p 210.
CU	Cu	Ibid, pp 204-205.
CU1	Cu^{+1}	Ibid.
CU2	Cu^{+2}	Ibid.
CUT	Cu	Ibid, p 210.

Identification Code	Element	Source of Curve
123		
ZN	Zn	Inter. Tables, Vol. 3, pp 204-205.
ZN2	Zn^{+2}	Ibid.
ZNT	Zn	Ibid, p 210.
AS	As	Ibid, pp 206-207.
SE	Se	Ibid.
BR	Br	Ibid.
BR-	Br^{-1}	Ibid.
ZR4	Zr^{+4}	Ibid.
ZR	Zr	Ibid, p 211.
NB	Nb	Ibid.
MO	Mo	Ibid.
RU	Ru	Ibid.
PD	Pd	Ibid.
AG	Ag	Ibid, pp 206-207.
AG1	Ag^{+1}	Ibid.
AGT	Ag	Ibid, p 211.
CD	Cd	Ibid.
SN	Sn	Ibid.
SB	Sb	Ibid.
I	I	Ibid.
BA	Ba	Ibid.
LA	La	Ibid.
GD	Gd	Ibid.
TA	Ta	Ibid, p 212.
W	W	Ibid.
PT	Pt	Ibid.
AU	Au	Ibid.
HG	Hg	Ibid, pp 206-207.
HG2	Hg^{+2}	Ibid.
HGT	Hg	Ibid, p 212.
TL	Tl	Ibid.
PB	Pb	Ibid.
BI	Bi	Ibid.
U	U	Ibid.

U U Ibid.

Output of the Statistical Parts:

Under the heading "RESULTS OF WILSON STATISTICS", are presented the scaling factor and overall isotropic temperature factor. These values are obtained by a least squares fit of a straight line to the usual Wilson plot. On the following page are listed the values which were used as input to the least squares routine. In order to learn how closely these points approximate a straight line, it is necessary only to plot a graph of the values in the column under "LN(AVE I/SUM F**2)" as ordinate, against the values in the column under "SIN TH**2/LAMBDA**2" as absissa. The first two columns contain the shell number and the number of observations within each shell respectively. The low angle data of shell 1 are always neglected.

A Howells, Phillips, and Rogers plot is obtained by plotting the fractions N(Z) of reflections with intensities equal to or less than a fraction Z of the local average against the values of Z from 0 to 1.[1] The first row of numbers on the output page is a list of Z in increments of .05. In the next row directly under each value of Z is found the corresponding value of N(Z). The next two rows list the theoretical values of N(Z) for centric and acentric structures respectively.

[1] E.R.Howells, D.C.Phillips, and D.Rogers, Acta Cryst, 3,210 (1950).

Statistical Calculations in Centered Space Groups:

In order to get the proper Wilson scale factor and the proper Howells, Phillips, and Rogers information, the numbers of atoms of different kinds which is entered on the NUMBERS OF ATOMS card should be the number in the unit cell multiplied by the multiplicity of the centering. For example, for a C-centered cell, the user should enter twice the number of atoms in a unit cell. For primitive space groups, one should always punch the number in one unit cell.

DATA TAPE Preparation and Editing Subprogram Cards

The following is an alphabetic listing of cards acceptable to this subprogram. The format for the Short Format Data Card, which is not listed here, may be found in Chapter 4, page 4-30.

ADDITIONS

```
==================================================================
```

Action:

The subprogram is set to regard the data cards which follow as additions to the DATA TAPE.

Note:

Allowed in the EDIT mode deck only. Short Format Data Cards must follow this card.

APPLY WILSON SCALING FACTOR

```
==================================================================
```

Action:

This card causes the F(obs) and weights of all reflections on the DATA TAPE to be scaled by the scale factor which results from doing Wilson statistics.

Note:

Allowed in the PREPARE mode deck only. This card must be used together with a DO WILSON STATISTICS card.

CELDEV

```
======XXXXXXX=XXXXXXX=XXXXXXX=XXXXXXYYYYYXXXXXX================================
```

For a format description see Chapter 4, page 4-21.

For a format description see Chapter 4, page 4-21.

CELL
```
======AAAAAAA=BBBBBBB=CCCCCCC=AAAAAABBBBBBCCCCCC================================
```

For a format description see Chapter 4, page 4-20.

Note:
This card is required in all EDIT mode decks. It is also required in PREPARE decks which have reflection input by cards. If reflection input is by DATA TAPE EXTRA, the CELL card is not required. If it is present, the cell parameters on the card will be used in preference to those on the DATA TAPE EXTRA.

DELETIONS
```
================================================================================
```

Action:
The subprogram is set to regard the data cards which follow as identifiers for reflections which are to be removed from the DATA TAPE.

Note:
Allowed in the EDIT mode only. Short Format Data Cards must follow this card.

DISPERSION CORRECTION
```
===========================fff======ΔΔΔΔΔΔ======ΔΔΔΔΔΔ==========================
```

Required Constants:
1. A three-character atomic form factor curve identifier in col. 28-30.
2. A value for the real part of the dispersion correction to the specified form factor curve in col. 37-42.

Optional Constant:
A value for the imaginary component of the dispersion correction may be placed in col. 49-54.

Action:

1. The real component given on the card is added to the designated form factor curve for all values of $\sin\theta/\lambda$.
2. If an imaginary component appears, it is converted and stored for possible use by an overlay program.

```
DATA ON DATA TAPE EXTRA
===============================================================================
```

Action:

This card tells the subprogram to get the reflection data from the DATA TAPE EXTRA which has been previously loaded.

Note:

This card is allowed in a PREPARE mode deck only. It should be the last card in the subprogram deck in which it appears.

```
DO HOWELLS,PHILLIPS,AND ROGERS CALC.
====================================NEGLECT DATA FOR WHICH  H = 0 K = 0 L = 0==
                                                           K = 0 L = 0 H = 0
                                                           L = 0 H = 0 K = 0
```

Optional Constants:

If "NEGLECT DATA FOR WHICH" is punched in col. 38-59, and either "H = 0", "K = 0", and / or "L = 0" is punched in either col. 62-66, col. 68-72, and / or col. 74-78; the designated data will be neglected in both the Howells, Phillips, and Rogers calculation and in the Wilson calculation.

Action:

The subprogram will be set to calculate points for a plot for approximating whether a structure is centric or acentric by the method of Howells, Phillips, and Rogers. This will be done in a second pass of the data.

Note:

This option is allowed only in a PREPARE mode subprogram deck, and only when Wilson statistics are also being done.

Note:

This option is allowed only in a PREPARE mode subprogram deck, and only when Wilson statistics are also being done.

```
DO WILSON STATISTICS,USE      GROUPS.              LAMBDA =
===========================NN===============================XXXXXX==============
```

Optional Constants:

1. The number of groups into which the data is to be divided for the Wilson summations may be specified in col. 28-29 as an integer, right justified. If nothing is specified, 10 groups is assumed.
2. A wavelength for use in defining the above mentioned groups may be punched in col. 61-66. If none appears, Cu is used.

Action:

The subprogram is caused to do Wilson statistics to determine an approximate scale factor and an approximate overall temperature factor. These calculations are done in the same pass of the data as the preparation of the DATA TAPE.

Note:

1. Allowed in the PREPARE mode deck only.
2. A NUMBERS OF ATOMS card must accompany this card.

```
END OF DATA
===============DO EDIT NOW======================================================
```

Optional Field (in the EDIT mode only):

If DO EDIT NOW is punched in col. 16-26, the edit will be done immediately.

Action:

The subprogram is notified that all card input of reflection data is finished. This card must be the last card in the subprogram deck in which it appears.

```
 F
==fff=aaaaaabbbbbbccccccddddddeeeeeeffffffgggggghhhhhhiiiiiijjjjjjkkkkkk========
```

Required Constants:

1. A three-character atomic form factor curve identifier in col. 3-5. This identifier must be different from any used for curves on the library.
2. Values for atomic form factors. Each value is allowed 6 columns beginning in col. 7. The values <u>must</u> be punched with the decimal point assumed between the second and third of the 6 allowed columns. Only 11 values are allowed on one card; if points are left when a card is full, the sequence may be continued on another F card with the same identifier. These continuation cards should immediately follow the first card.

Action:

This card is used in conjunction with a SIN TH card to read in atomic form factor curves which are not on the library.

```
KIND F
======aaabbbcccdddeeefffggghhhiiijjjkkklllmmmnnn================================
```

Required Constants:

Three-character atomic form factor curve identifiers for all the form factors required for the given structure must be punched one after the other, with three columns per identifier, beginning in col. 7. A maximum of 14 is allowed; a maximum of 12 is recommended.

Action:

1. The subprogram is set so that the form factors which have been designated will be placed on the data tape for each reflection, or if in the EDIT mode, for each additional or substituted reflection.
2. The subprogram is notified that all input except reflection data has entered.

2. The subprogram is notified that all input except reflection data has entered.

NUMBERS OF ATOMS

```
==================NN-fff,N-ff,NNN-ff,...                                ==
```

Required constants:

The numbers of each kind of atom must be given in the following manner:integer dash form factor identifier comma integer dash form factor identifier etc. There should be no comma after the last form factor identifier. If the form factor identifier ends in blanks, the trailing blanks need not be punched. The first integer should begin in col. 19. The numbers given should pertain to a unit cell.

Action:

A table of numbers of atoms versus types of atom is stored in memory for use by the subprogram.

Example:

```
NUMBERS OF ATOMS  43-C,22-ZN2,124-N,5-H
```

SIN TH

```
======aaaaaabbbbbbccccccddddddeeeeeeffffffgggggghhhhhhiiiiiijjjjjjkkkkkk========
```

Required Constants:

Values of sin θ/λ from a table of sin θ/λ versus atomic form factor. Each value is allowed 6 columns beginning in col. 7. The values must be punched with the decimal point assumed between the first and second of the 6 allowed columns. Only 11 values are allowed on one card; if points are left when a card is full, the sequence may be continued on another SIN TH card. These continuation cards should be placed immediately after the first SIN TH card.

Action:

This card is used in conjunction with F cards to read in atomic form factor curves which are not on the library.

```
SUBSTITUTIONS
==========================================================================
```

Action:

The subprogram is set to regard the pairs of data cards which follow as defining substitutions to be made on the DATA TAPE. The first data card of each pair should carry h, k, l, and F(obs) as they appear on the DATA TAPE. The second designates h, k, l, F(obs), and a weight as they are to appear in the substituted reflection.

Note:

Allowed in the EDIT mode only. Short Format Data Cards must follow this card.

```
GLIDE PLANES,      PA     PB     PC     110    101    011
====================X=====X=====X=====X=====X=====X=======================
```

Optional Constants:

Alphabetic identifiers, either "A", "B", "C", or "N", in col. 23, 29, 35, 41, 47, and/or 53 for defining the type of glide planes which are perpendicular to the A, B, C, 110, 101, and 011 axes respectively. If there is no glide plane perpendicular to a particular axis, nothing should be punched in the column for that axis.

Action:

Reflections in the zone of each axis which has a glide plane perpendicular to it will be treated in the special manner required. The information from this card is entered into both the Wilson and the Howells, Phillips, and Rogers calculations.

Note:

1. This card should be entered whenever glide planes are present.
2. Without this card, all reflections are treated equally.

2. Without this card, all reflections are treated equally.

Chapter 6

Structure-Factor Least Squares Calculations

Programming Contributions:

N. C. Webb, Basic program and cubic version.

A. P. Kendig, Debugging of cubic version; monoclinic-triclinic modifications.

C. M. Gramaccioli, Orthorhombic modifications.

T. A. Bieneke, Hexagonal-trigonal modifications.

D. Duchamp,
I. Ambats, } Debugging and testing.
B. D. Sharma,

R. E. Marsh, Debugging and supervision of all the above work.

Discussion:

This chapter describes the CRYRM Structure-Factor Least Squares Subprogram. It occurs in five versions: cubic, monoclinic-triclinic, orthorhombic, tetragonal, and hexagonal-trigonal. All calculations are space-group specific. All versions except cubic, provide for both isotropic and anisotropic atoms.

The proper version of the subprogram is called into memory in either the Least Squares or Structure-Factor mode by either a LEAST SQUARES or a STRUCTURE FACTOR CALCULATION system control card, or in certain cases by a DIFFERENCE FOURIER control card (see Chapter 3).

The function minimized in all cases is

$$\Sigma\, w\left(k^2|F_{obs}|^2 - |F_{calc}|^2\right)^2$$

This implies that in the final refinement $\sqrt{w}$ should be equal to $\frac{1}{\sigma(|F_{obs}|^2)}$ where $\sigma(|F_{obs}|^2)$ is the standard deviation in the square of the observed structure factor amplitude. (Note: The weight used here is the square of the term referred to a "weight" in the Initial Data Processing and Data Tape Preparation and Editing subprograms.)

The reflection data are entered by means of a DATA TAPE prepared by the Data Tape Preparation and Editing subprogram (see Chapter 5). Whenever structure factors are calculated (in a structure-factor calculation or in least squares) the latest values of F_{calc} (or A_{calc} and B_{calc}) are placed on the DATA TAPE for use by other subprograms, for example, Fourier. The structure-factor contributions of certain selected atoms may also be stored on the DATA TAPE.

The Subprogram Deck:

The subprogram deck for a Structure-Factor calculation consists of (1) first a WEIGHTING FUNCTION card, (2) special option cards as required, and (3) an Atomic Information Deck (or a substitute using the Atomic Information Deck stored on the DATA TAPE). Several cards, such as RECYCLE, and FULL MATRIX, are not allowed in the Structure-Factor mode (see individual card descriptions).

The subprogram deck for a Least-Squares calculation consists of (1) first a WEIGHTING FUNCTION card, (2) special option cards as required, (3) an Atomic Information Deck (or a substitute), and (4) RECYCLE and special option cards for more cycles if these are desired. Only certain special option cards are enterable at Recycle time; see individual card descriptions. The set up of the Least Squares subprogram deck is outlined in Figure 3.

The Atomic Information Deck:

The Atomic Information deck is composed of: COORD, ISO-AN, ISO, ANISO, and POP cards used for input of coordinates, isotropic temperature factors, anisotropic temperature factors, and population factors; a SCALE FACTOR card; if Least Squares, one or more matrix cards; and an END ATOMIC INFORMATION card. A COORD card and either an ISO or an ANISO (or an ISO-AN) card must be entered for each atom. The POP card is optional.

an ISO-AN) card must be entered for each atom. The POP card is optional.

The Least Squares Subprogram Deck:

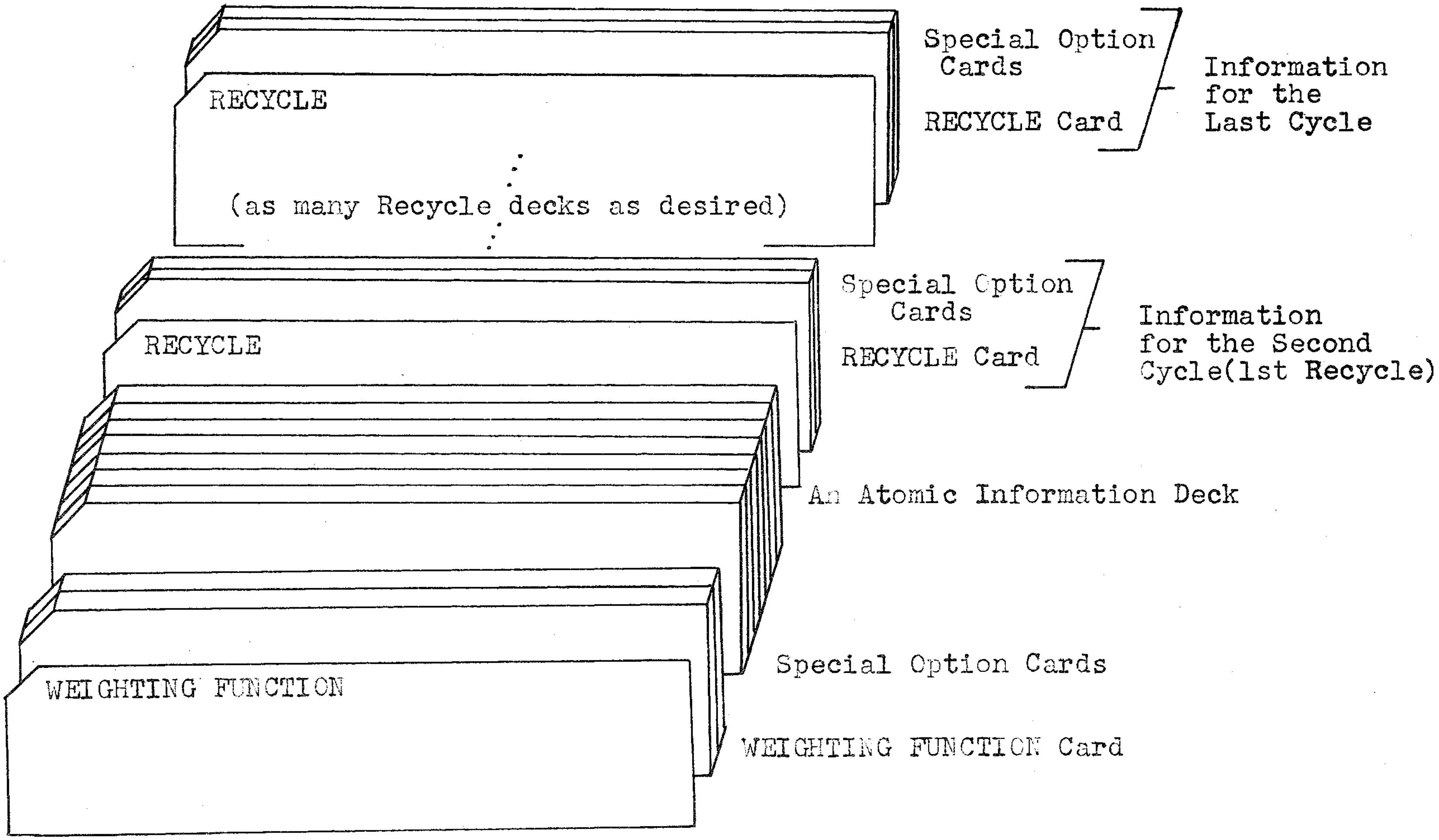


Figure 3.

No cards other than those specified above may ever be in an Atomic Information deck. The last card must always be END ATOMIC INFORMATION. For Least Squares the first card must be a matrix card. An example of an Atomic Information deck for Least Squares is given below.

Example 1.

```
FULL MATRIX     SHIFT FACTOR= 1.0
SCALE FACTOR=  1.0
ANISO     1   .0060183  .0121941  .1089064 -.0004748 -.0010085 -.0067519
  :       :
ANISO     9   .0556236  .0096826  .0873284 -.0017957 -.0014676 -.0091156
BLOCK MATRIX    SHIFT FACTOR= .5
COORD     1  N    .003025  .195415  .043551
  :       :
COORD    13  H    .241900  .078522 -.344734
NO REFINEMENT
ISO      10   3.5
  :       :
ISO      13   3.5
END ATOMIC INFORMATION
```

For the Structure-Factor mode, no matrix cards are allowed. The order of the other cards in the deck is immaterial, except as it effects the matrix setup in the Least Squares mode.

A new Atomic Information deck with the latest parameters is punched automatically after each Least Squares run (after the last cycle). If a new deck is desired after each cycle, this can be accomplished by using the appropriate control card.

As the Atomic Information deck is read in, each card is printed. This printing may be suppressed by the appropriate card. At this time the Atomic Information deck is also put on the DATA TAPE as card images.

the Atomic Information deck is also put on the DATA TAPE as card images.

At the end of a Least Squares run, the newest atomic parameters are placed on the DATA TAPE. This process requires an additional pass of the tape. Because of this, it may be desirable to use a STOP UPDATING card to suppress this in order to save time where long DATA TAPEs are involved.

Using and Editing Atomic Parameters from the DATA TAPE:

If it is desired to use the Atomic Information deck from the DATA TAPE for input to Structure-Factor or Least Squares calculations, the Atomic Information deck in the subprogram deck may be replaced by an ATOMIC PARAMETERS ARE ON DATA TAPE card. This causes the tape parameters to be used with no changes.

The Atomic Parameter deck from the DATA TAPE may be used with limited changes by using the Edit Atomic Information option. The number of cards and the matrix setup may not be changed; but COORD, ISO, ANISO, and POP cards may be substituted for those having the same atom number. An EDIT ATOMIC PARAMETERS card, followed by the new COORD, ISO, ANISO, and/or POP cards, followed by an END EDIT card form the Edit Atomic Information deck. The new cards will be substituted for the corresponding cards on the DATA TAPE, and the Atomic Information deck will be otherwise unchanged. An example of an Edit Atomic Information deck follows.

Example 2.

```
EDIT ATOMIC INFORMATION
ISO     10   3.0
COORD    1  N    .002800  .195415  .043551
END EDIT
```

The Least Squares Matrix Setup:

This subprogram provides for a multiple matrix least squares setup. The only restrictions are: (1) each parameter may be entered into only one matrix, (2) the parameters on a given input card may not be put into separate matrices (except straight diagonal), and (3) the matrices must fit into the storage available.

The parameters are entered by means of COORD, ISO, ISO-AN, ANISO, POP, and SCALE FACTOR cards. The matrix setup is defined by mixing these cards with FULL MATRIX, BLOCK MATRIX, DIAGONAL MATRIX, and NO REFINEMENT cards. The parameters on the cards following a given matrix card go into the matrix or matrices defined by that card.

A FULL MATRIX card causes all the parameters on the cards which follow (those after it but before the next matrix card or the END ATOMIC INFORMATION card) to be put into one matrix. As many FULL MATRIX cards may be used as desired--one for each matrix which involves parameters from more than one parameter card. If no such matrices are desired, no FULL MATRIX card should be entered.

After all the large matrices, if any, have been entered, a BLOCK MATRIX card may be entered. Each parameter card which follows the BLOCK MATRIX card becomes a separate matrix--for example, an ANISO card generates a 6x6 matrix; an ISO card generates a 1x1. Only one BLOCK MATRIX card may be entered. It may be followed by as many parameter cards as desired, however. There are special BLOCK MATRIX options available for temperature factors. If a "P" is punched in col. 73 of an ISO or an ANISO card, the card generates a 2x2 or a 7x7 matrix respectively, the extra parameter being a population parameter which is not shifted. A BLOCK MATRIX card should be entered only if block matrices are desired.

After all FULL and BLOCK matrices, if any, have been entered; a DIAGONAL MATRIX card may be entered, followed by any number of parameter cards. For each parameter entered in this manner, a 1x1 matrix--the diagonal term--is entered into the least squares (no cross terms are considered), for example, an ANISO card generates 6 1x1 matrices. Only one DIAGONAL MATRIX card may be entered; if no matrices of this type are desired, no DIAGONAL MATRIX card should be entered.

After all parameters in the refinement have been entered using one or more of the above options, the cards of parameters which are not to be refined, if there are any, should be placed after a NO REFINEMENT card. Only one NO REFINEMENT card may be entered; if all parameters are refined, the NO REFINEMENT card should not be entered.

have been entered using one or more of the above options, the cards of parameters which are not to be refined, if there are any, should be placed after a NO REFINEMENT card. Only one NO REFINEMENT card may be entered; if all parameters are refined, the NO REFINEMENT card should not be entered.

The order of the matrix cards is important: FULL MATRIX before BLOCK MATRIX before DIAGONAL MATRIX before NO REFINEMENT. If this order is not followed, an error results and the job will be terminated.

The storage required for a given matrix may be calculated by the formula

$$\frac{n(n + 1)}{2}$$

where n is the order of the matrix. One storage location is required for each element, only the upper triangular part of the symmetric matrix is stored. The total required storage for matrices is the sum of that required for each matrix.

Atoms in Special Positions:

An atom in a special position will not have three independently refinable coordinates and will have a multiplicity different from the maximum multiplicity of the space group. For these atoms the multiplicity (as given in the International Tables, Vol. 1) of the special position should be entered on the COORD card. The proper number of refinable parameters to enter into the Least Squares calculation and the proper shifting is handled by means of the NS and NR options on the COORD card. If a coordinate is shifted but not refined, the shift of the preceeding refined coordinate is used. Examples: (1) for an X, X, Z atom, one would shift all coordinates, but not refine Y; (2) for an 1/2, 1/2, Z atom, one would not refine and not shift both X and Y.

The Scale Factor:

This subprogram provides for the refinement of one scale factor (to be applied to all reflections). The scale factor "k" to be applied to F_{obs} is entered on the SCALE FACTOR card. The actual parameter which is refined is 1/k, or the scale factor for F_{calc}. The shift which is printed in the shift list is the shift for 1/k, however, the new and old parameters in the list are k itself.

All F_{obs} in the structure-factor list are the scaled F_{obs} (k F_{obs}). The weights, however, are based on the unscaled F_{obs} (the F_{obs} which is on the DATA TAPE).

Shift Factors:

Shift factors are enterable at three different levels. A general shift factor may be entered on the GARBAGE card. Shift factors may be entered on FULL MATRIX, BLOCK MATRIX, and DIAGONAL MATRIX cards. And finally shift factors may be entered on individual parameter cards.

If a shift factor is entered on a parameter card, this shift factor is used for all parameters on that card; all other shift factors are ignored. If no shift factor is punched on a parameter card, the shift factor on the corresponding matrix card is used. If there is no matrix card shift factor, the general shift factor is used. If no shift factor appears in any of these places, a shift factor of unity is assumed.

Anisotropic Temperature Factors:

Anisotropic temperature factors are calculated by the expression

$$e^{-(h^2B_{11} + k^2B_{22} + l^2B_{33} + hkB_{12} + hlB_{13} + klB_{23})} .$$

An isotropic atom may be converted to an anisotropic atom by use of the ISO-AN card. When this card is read in, the proper B_{ij} terms for a spherical anisotropic atom with the same size temperature factor as the isotropic are generated; and in all following cycles the six anisotropic parameters are varied separately. When a new parameter deck is punched, an ANISO card will be punched.

Unobservable and Zero Weight Reflections:

If the external weight of a given reflection is exactly zero, that reflection does not enter the Least Squares calculation or the R factor and weighted sums calculations. This condition is indicated by an asterisk after the F_{obs} in the printed output.

and weighted sums calculations. This condition is indicated by an asterisk after the F_{obs} in the printed output.

Unobservable reflections are handled by entering a minimum observable structure-factor amplitude--saying F_{obs} is "less than" the value entered. This is indicated by making the external weight of the reflection negative. If the external weight of a certain reflection is negative (indicating a "less than" reflection), the reflection enters the least squares calculation and the R factor and weighted sums calculations only if the calculated structure factor amplitude is greater than the minimum observable amplitude. It does not enter if the calculated structure-factor amplitude is within the limit. In the printed output, a "less than" reflection is indicated by making F_{obs} negative; entry into the least squares of a "less than" reflection is indicated by a ΔF with an asterisk following.

The Printed Output:

All subprogram control cards are listed in the printed output. This is followed by a listing of h, k, l, F_{obs}, F_{calc}, ΔF, and for acentric structures A_{calc} and B_{calc}. After every second page the cumulative sums, weighted sums, R-factor, and weighted R factor are printed out. The format is

$\Sigma\, kF_{obs}$	$\Sigma\lvert F_{calc}\rvert$	$\Sigma\,\lVert kF_{obs}\rvert - \lvert F_{calc}\rVert$	R	Number of Refs. entered
$\Sigma w(F_{obs})^4$	$\Sigma w(F_{calc})^4$	$\Sigma w(k^2\lvert F_{obs}\rvert^2 - \lvert F_{calc}\rvert^2)^2$	Weighted R $\left(\frac{\text{third}}{\text{first}}\right)$	Number of reflections in L. S.

where k is the scale factor for F_{obs}.

At the end of the listing the final sums are printed in the above format together with the R factor in G (Gramaccioli) units--one G = 3.95%. Also a Goodness of Fit is printed out, calculated by

$$\frac{\Sigma\, w(k^2\lvert F_{obs}\rvert^2 - \lvert F_{calc}\rvert^2)^2}{m - s}$$

where m is the number of observations, and s is the number of parameters being refined.

The actual goodness of fit may be calculated by dividing the above quantity by the fourth power of the scale factor (for F_{obs}), then taking the square root. The quantity thus obtained should approach unity if the weights are on an absolute scale. Division by a power of the scale factor is necessary because all weights are based on the unscaled structure-factors.

The total number of reflections, the number in the least squares, the number of "less than" reflections, and the number of "less than" reflections which are in the least squares are printed after the final sums.

This is all the output in the Structure-Factor mode. For the Least Squares mode, this is followed by a matrix by matrix table of shifts and standard deviations. Standard deviations are calculated by the equation

$$\sigma_i = \left[(A^{-1})_{ii} \left[\frac{\Sigma\, w(k^2 F_{obs}{}^2 - F_{calc}{}^2)^2}{m - s} \right] \right]^{1/2}$$

where σ_i is the standard deviation in parameter i, $(A^{-1})_{ii}$ is the diagonal element of the inverse matrix corresponding to parameter i, and all other parameters are as defined previously.

If more cycles of least squares are done, the output is repeated for each cycle. The printing of the structure-factor list may be suppressed by the appropriate special option card. If this is done, the final sums are still printed. Once printing is suppressed, it will remain suppressed until the option card asking for print is entered. The normal mode is to print the list. Printing of the shifts may not be suppressed.

The printing of the "normalized real" matrix (the least squares matrix A transformed so that all diagonal terms are normalized to unity) and/or the inverse matrix A^{-1} (not normalized) may be effected by the appropriate designations on the corresponding matrix cards. The normal mode is not to print any matrices.

diagonal terms are normalized to unity) and/or the inverse matrix A^{-1} (not normalized) may be effected by the appropriate designations on the corresponding matrix cards. The normal mode is not to print any matrices.

Recycle:

If more than one cycle of least squares is desired in a given run, a RECYCLE card must be entered for each extra cycle. A new WEIGHTING FUNCTION card, new $\sin^2\theta/\lambda^2$ limits (see GARBAGE card) may be entered, and certain option changing cards may be entered between cycles. No changes in the shift factors, matrix setup, or atomic parameters may be made. Cards pertaining to a given cycle should be placed immediately after the RECYCLE card which generates that cycle. (Cards for the first cycle must be placed before the Atomic Information deck.

When doing an additional cycle, the inverse matrices from the previous cycle may be used instead of accumulating new matrices. If this is done, a new ΔF^2 vector is accumulated. This is then multiplied by the old inverse matrix to obtain the shift vector. This option saves considerable time where large matrices are involved.

The Partial Structure-Factor (OMIT ATOMS) Option:

The contribution to the structure-factor of certain selected atoms may be stored on the DATA TAPE by use of an OMIT ATOMS card. This "partial structure-factor" is then available for use in difference Fourier maps, etc.

The Structure-Factor Accumulation Option:

If one desires to calculate structure-factors by entering the atoms in several batches (several passes of the DATA TAPE) the structure-factor accumulation option may be used. The steps are as follows: (1) A structure-factor calculation is run in the regular way using the first batch of atoms--the results are stored on the DATA TAPE. (2) Another structure-factor calculation is run using the second batch of atoms with the special ACCUMULATE ACALC AND BCALC card--this causes the A_{calc} and B_{calc} which are on the DATA TAPE (the contributions of the first batch of atoms) to be added respectively to those calculated in the present run, the sums being stored on the DATA TAPE. (3) As many more batches--with the special option card--may be run as desired. The last batch of atoms may be entered by either a Structure-Factor calculation or a Least Squares calculation; since this batch will produce the full A_{calc} and B_{calc}, the parameters of the atoms in the last batch may be refined.

If this option is used, the number of atoms which may be entered into a Structure-Factor calculation is unlimited. It is hoped that this option may be useful in the study of large molecules, where the size of the atom list could become a problem.

Weighting Function Control Cards:

A weighting function control card is required in the input to all Least-Squares calculations. In Structure Factor calculations, a weighting function must be used to calculate the weighted R-factor. If no weighting function control card is entered, trouble may occur.

Note:

The term "weight" in the following discussion of WEIGHTING function cards refers to the quantity $\sqrt{w}$ as w is defined on page 6-1.

WEIGHTING FUNCTION UNITY, W=1.0
==EXTERNAL WEIGHT==================

Optional Field:

58-72

If EXTERNAL WEIGHT is punched in col. 58-72, the External Weight which appears on the DATA TAPE will be multiplied by the weight generated internally, and the result used as a weight.

Action:

The internal weighting routine is set to generate a weight of unity for all reflections.

Note:

A weighting function of unity does not make much sense when used without the EXTERNAL WEIGHT option. It does NOT mean that all reflections are weighted equally. This option is included so that, for each reflection, an externally generated weight -equal or proportional to 1 / $\sigma(F_{obs}^2)$--may be used for weighting directly.

```
WEIGHTING FUNCTION HUGHES 1/FOBS         4FMIN=
================================================FFFFFF===EXTERNAL WEIGHT========
```

Required Constant:
A value for $4(F_{obs})_{minimum}$ must be punched in col. 49-54.

Optional Field:
If EXTERNAL WEIGHT is punched in col. 58-72, the External Weight which appears on the DATA TAPE will be multiplied by the weight generated internally, and the result used as a weight.

Action:
The internal weighting routine is set to generate a weight of $1/F_{obs}$ for all reflections with F_{obs} greater than $4(F_{obs})_{minimum}$. Reflections less than this are given a weight equal to $1/\sqrt{(F_{obs})4(F_{obs})_{minimum}}$. If EXTERNAL WEIGHT has been punched, this internal weight is multiplied by the External Weight which occurs on the DATA TAPE, and the result used in weighting the reflection.

```
WEIGHTING FUNCTION HUGHES 1/FOBS SQUARED 4FMIN=
================================================FFFFFF===EXTERNAL WEIGHT========
```

This card is the same as the one above except the weight is set equal to $1/F_{obs}^2$ when F_{obs} is more than $4(F_{obs})_{minimum}$, and equal to $1/(F_{obs})4(F_{obs})_{minimum}$ when it is less than this quantity.

Note:
In both the above, the F_{obs} is the unscaled observed structure factor amplitude, i.e. the F as it appears on the DATA TAPE.

WEIGHTING FUNCTION

```
==================PPPPPPQQQQQQRRRRRRSSSSSSTTTTTTUUUUUUVVVVVV=FFF==EXT WT========
```

Required Constants:

1. A value for P in col. 19-24, for Q in col. 25-30, for R in col. 31-36, for S in col. 37-42, for T in col. 43-48, for U in col. 49-54, for V in col. 55-60 may be punched. Any which are not punched are assumed to be zero.

2. If V is non-zero, a three-character form factor curve identifier must be punched in col. 62-64.

Optional Field:

If EXT WT is punched in col. 67-72, tne External Weight which appears on the DATA TAPE will be multiplied by the weight generated internally.

Action:

The internal weighting routine is set to generate weights for each reflection according to the following function:

$$\sqrt{W_i} = \frac{P + Q \sin^2\theta/\mathrm{lambda}^2}{R + S F_{obs} + T F_{obs}^2 + U F_{obs}^3 + V f_i}$$

where P, Q, R, S, T, U, and V are constants entered on the card, and f_i is the value for the specified form factor at the $\sin\theta$ of the reflection whose weight is being calculated. If EXT WT has been punched, the External Weight will be multiplied by that calculated by the above function, and the product will be used to weight the reflection.

Note:

The F_{obs} above is unscaled, i.e. as it appears on the DATA TAPE.

Matrix Control Cards:

At least one matrix control card is required in each Least Squares atomic information deck, unless the atomic parameters are being taken from the DATA TAPE, in which case the required matrix control cards are already on the DATA TAPE.

```
FULL MATRIX
================SHIFT FACTOR= SSS====PRINT REAL   ==============================
                                           INVERSE
                                           BOTH
```

Optional Constant:

If "SHIFT FACTOR=" is punched in col. 17-29, and a shift factor is punched in col. 31-33, all the atomic information parameters which go into the matrix specified by this card will be assigned this shift factor, provided no shift factor is punched on the individual atomic information card.

Optional Field:

If "PRINT" is punched in col. 38-42, and either "REAL","INVERSE" or "BOTH"is punched in col. 44-50 as shown above, the specified matrices will be printed for each cycle of Least Squares.

Action:

The subprogram is set to regard as one matrix all refinable parameters entered on atomic information cards which follow this card but precede the next matrix card (or END ATOMIC INFORMATION card).

Note:

As many FULL MATRIX cards as needed may be included in the atomic information deck. Each FULL MATRIX card generates a separate matrix.

BLOCK MATRIX

```
===============SHIFT FACTOR= S.SS====PRINT REAL    ============================
                                          INVERSE
                                          BOTH
```

Optional entries are the same as for the FULL MATRIX card.

Action:
The subprogram is set to regard as a separate matrix the refinable parameters from each of the atomic information cards which follow. This means one atomic information card = one matrix.

DIAGONAL MATRIX

```
===============SHIFT FACTOR= S.SS====PRINT REAL    ============================
                                          INVERSE
                                          BOTH
```

Optional entries are the same as for the FULL MATRIX card.

Action:
The subprogram is set to regard each refinable parameter entered after this card as one matrix. This means each parameter becomes a one by one matrix, all cross correlation with other parameters is neglected.

NO REFINEMENT

```
================================================================================
```

Action:
The subprogram is set not to refine any of the atomic parameters which are entered after this card.

Note:
All full matrices must precede all block matrices which must precede all diagonal matrices which must precede the no refinement section.

Atomic Information Cards:

```
COORD
======NNNN==FFF=XXXXXXXX=YYYYYYYY=ZZZZZZZZ===MMM===SSSNSNSNSNRNRNR==============
```

Required Constants:

1. An atom number, an integer, in col. 7-10 right justified.
2. A three character form factor identifier in col. 13-15. See Chapter 5, page 5-4 for a description of the form factor identifier codes.
3. The fractional coordinates--x, y, and z--with respect to the crystallographic axes, punched in col. 17-24, col. 26-33, and col. 35-42 respectively.

Optional Constants:

1. The multiplicity of an atom in the unit cell may be punched as an integer in col. 46-48, right justified. The maximum multiplicity of the space group is assumed if nothing is punched.
2. A shift factor in col. 52-54. If a shift factor is punched, it has precedence over shift factors on matrix cards or on the GARBAGE card.
3. If "NS" appears in col. 55-56 for x, and/or col. 57-58 for y, and/or col. 59-60 for z, the corresponding parameter(s) will not be shifted if Least Squares is done.
4. If "NR" appears in col. 61-62 for x, and/or col. 63-64 for y, and/or col. 65-66 for z, the corresponding parameter(s) will not be refined if Least Squares is done.

Action:

The information given on the card is added to the proper lists in the subprogram.

Note:

1. Atom numbers need not be sequential, but should be kept low.
2. If a parameter is shifted, but not refined, the shift of the preceding refined parameter is used. For example, if x is refined and y is shifted but not refined, the shift calculated for x is also applied to y.

2. If a parameter is shifted, but not refined, the shift of the preceding refined parameter is used. For example, if x is refined and y is shifted but not refined, the shift calculated for x is also applied to y.

```
ISO
======NNNN==BBBBBBBBBBBB================================================P==SSS==
```

Required Constants:

1. An atom number, an integer, in col. 7-10 right justified. This number should match the one on the COORD card for this atom.
2. An isotropic temperature factor in col. 13-24.

Optional Constants:

1. If a "P" is punched in col. 73, a population factor for this atom will be entered into the refinement. It will be given a value of 1.0 and not shifted.
2. A shift factor in col. 76-78. If a shift factor is punched, it has precedence over shift factors on matrix cards or on the GARBAGE card.

Action:

The information is added to the proper lists in the subprogram.

```
ISO-AN
======NNNN==BBBBBBBBBBBB================================================P==SSS==
```

This card has the same constants as the ISO card, and may be made from a previously punched ISO card by adding "-AN" in col. 4-6.

Action:

The isotropic temperature which is punched is converted to an anisotropic temperature factor of six parameters. Each of the six parameters is varied separately. When a new Atomic Parameter Deck is punched after Least Squares, the temperature factor for this atom will be on an ANISO card.

ANISO

```
======NNNN==SB11B11B11SB22B22B22SB33B33B33SB12B12B12SB13B13B13SB23B23B23P==SSS==
```

Required Constants:

1. An atom number, an integer, in col. 7-10 right justified. This number should match the one on the COORD card for this atom.
2. The six parameters for an anisotropic temperature factor defined according to the expression $e^{-(h^2B_{11}+k^2B_{22}+l^2B_{33}+hkB_{12}+hlB_{13}+klB_{23})}$, with B_{11} in col. 13-22, B_{22} in col. 23-32, B_{33} in col. 33-42, B_{12} in col. 43-52, B_{13} in col. 53-62, and B_{23} in col. 63-72.

Optional Constants:

1. If a "P" is punched in col. 73, a population factor for this atom will be intered into the refinement. It will be given a value of 1.0 and not shifted.
2. A shift factor in col. 76-78. If a shift factor is punched, it has precedence over shift factors on matrix cards or on the GARBAGE card.

Action:

The information is added to the proper lists in the subprogram.

Comment:

Each atom which is entered into the calculation must have one COORD card and one and only one temperature factor card. The POP card is optional. If this is not followed, calculations will be ended with an error.

The order of the above cards within the Atomic Information Deck is not important--COORD and temperature factor cards for the various atoms may be mixed in any order. In the case of Least Squares calculations, the order is determined by the matrix setup.

```
POP
======NNNN==PPPPP=SSS===FFF=====================================================
```

Required Constants:

1. An atom number, an integer, in col. 7-10 right justified.
2. A population factor in col. 13-17.

Optional Constants:

1. If a shift factor is punched in col. 19-21, this shift factor will be used in preference to factors entered on matrix cards or on the GARBAGE card.
2. If a three-character form factor identifier is punched in col. 25-27, the atom position specified on the COORD card is considered to be multiply occupied. Its form factor is taken as $P(f_1)+(1.0-P)(f_2)$, where P is the population factor punched on this card, f_1 is the form factor specified on the COORD card, and f_2 is the form factor identifier specified here.

Action:

The information is added to the proper lists in the subprogram.

Note:

1. The multiple occupancy option is intended for special types of disorder, as is present in certain alloys, etc. The expression given is a good approximation for such cases, but,of course, neglects "domain" considerations.
2. The POP card is optional. If no POP card is entered for a given atom, a population of unity is assumed.

```
SCALE FACTOR=
==============SSSSSSSSSS,SHIFT FACTOR= SSS======================================
```

Required Constant:

A scale factor in col. 15-24. The observed structure factor amplitude which occurs on the DATA TAPE is multiplied by this factor to put F_{obs} on the absolute scale.

Optional Constant:

If ",SHIFT FACTOR=" is punched in col. 25-38, and a shift factor is punched in col. 40-42, this shift factor will be used for the scale factor if it is refined.

Action:

A scale factor for F_{obs} is entered into the subprogram.

Note:

In a Least Squares mode subprogram deck, the SCALE FACTOR card may be placed before the Atomic Information Deck, or if it is desired to refine the scale factor, it may be placed within the range of the appropriate matrix card within the Atomic Information Deck.

```
END ATOMIC INFORMATION
================================================================================
```

Action:

This card signals the subprogram that the end of the Atomic Information Deck has been reached.

Note:

Once this card has been read in, the subprogram will not accept any more COORD, ISO, ISO-AN,POP,SCALE FACTOR, or ANISO cards. Calculations begin immediately.

Special Tape Atomic Information Cards:

The following cards are used instead of an Atomic Information Deck if the user wants to make use of the Atomic Information Deck which is on the DATA TAPE. No END ATOMIC INFORMATION card should be in the card subprogram deck if this option is being used.

ATOMIC PARAMETERS ARE ON DATA TAPE
==

Action:
The subprogram will take as input, the Atomic Information Deck which is on the DATA TAPE with no changes.

EDIT ATOMIC PARAMETERS
==

Action:
The subprogram will take as input, the Atomic Information Deck which is on the DATA TAPE, substituting each COORD, ISO, ANISO, and/or POP card which follows this card for the corresponding one on the DATA TAPE.

Note:

1. At least one substitution must be done if this card is used.
2. Cards are identified for substitution by title and atom number, neither of which may be changed.
3. No changes may be made in the matrix setup for Least Squares, since the matrix cards are on the DATA TAPE and may not be substituted. To change the matrix setup, use a card Atomic Information Deck. The same applies to adding new atoms.

END EDIT
==

Action:
This card signals the subprogram that all substitution cards for the EDIT ATOMIC PARAMETERS option have been read in.

Special Option Cards:

The following cards control special options of the subprogram. Unless otherwise indicated in the description, the user may assume that the card is enterable either before the Atomic Information Deck or at Recycle time.

GARBAGE.

```
=========GENERAL SHIFT =SSSSSS=MAXIMUM SINT/LSQ=SSSSSS,MIN.=SSSSSSMAXSET========
```

Optional Constants:

1. If "GENERAL SHIFT =" is punched in col. 10-24, and a shift factor is punched in col. 25-30, the general shift factor in the subprogram is set to the value punched.
2. Maximum and/or minimum limits on sin$^2\theta/\lambda^2$ for reflections going into the calculations. A maximum is entered by punching "MAXIMUM SINT/LSQ=" in col. 32-48, and the value in col. 49-54; A minimum value is entered by punching ",MIN.=" in col. 55-60, and the value in col. 61-66. Any reflection not falling within the specified limits will be skipped over.
3. If "MAXSET" is punched in col. 67-72, the subprogram will enter the "only isotropic" mode. This mode will allow the subprogram to accept more atoms, since an isotropic atom requires less storage than an anisotropic one. This option is for exceedingly large structures only.

Action:

The appropriate option as outlined above is set.

Note:

The GARBAGE card may be read in either before the Atomic Information Deck or at Recycle time; however, the general shift factor and MAXSET options may not be requested at Recycle time.

RECYCLE
=========SAME INVERSE===

Optional Field:
If "SAME INVERSE" is punched in col. 10-21, the subprogram is set to save the inverse matrices from the cycle just completed for use in getting the shifts for the cycle requested by this card.

Action:
The subprogram is set to do another cycle of least squares, using for input, the atomic coordinates from the preceeding cycle.

Note:
1. This card may not appear in a Structure-Factor mode deck.
2. This card may not be entered until after the Atomic Information Deck has been read in.
3. The Special Option Cards, if any, which pertain to the cycle requested here, must follow this RECYCLE card.

DO NOT PRINT ATOMIC-PARAMETER INPUT CARDS.
==

Action:
The cards in the Atomic Information Deck are not printed as they are read. Normal option is to print them.

Note:
This card, of course, must come before the Atomic Information Deck.

DO NOT PRINT H,K,L,FOBS ETC. FOR ANY REFLECTION.
==

Action:
The subprogram switch which controls printing of the structure-factor list is set not to print this list. The normal option is to print it.

PRINT H,K,L,FOBS,FCALC,ETC. FOR EACH REFLECTION

Action:

The subprogram switch which controls printing of the structure-factor list is set to print this list.

Note:

This card is designed to switch the printing on again after it has been switched off by the DO NOT PRINT card.

PUNCH NEW PARAMETER DECK AFTER EACH CYCLE

Action:

The subprogram is set to punch a new Atomic Information Deck after each cycle of Least Squares. The normal option is to punch a new deck after the last cycle only.

STOP UPDATING ATOMIC PARAMETERS ON DATA TAPE

Action:

The atomic parameters resulting from the last cycle of the run will not be placed on the DATA TAPE. The normal option is to update the DATA TAPE.

ACCUMULATE ACALC AND BCALC.

Action:

The subprogram is set for the "accumulate" mode(see page 6-11).

Note:

This card must come before the Atomic Information Deck.

OMIT ATOMS--
============N,NN,N,NNN,··· ===

Required Constants:

A list of atom numbers, entered as integers separated by commas with no blank columns in between, beginning in col. 7 and not exceeding col. 77. The atom numbers punched here refer to the atoms which have the same number punched on their COORD cards.

Action:

The contributions to A_{calc} and B_{calc} of the atoms specified in the list is stored on the DATA TAPE for future use. The total contribution of all the specified atoms is stored on a reflection by reflection basis.

Note:

The above storing of a "partial" structure-factor is in addition to the storing of the complete calculated structure factor which is automatically put on the DATA TAPE for each reflection considered.

Chapter 7

Fourier and Patterson Calculations

Programming Contribution: D. Duchamp

Introduction:

This subprogram provides for the calculation of Patterson maps, Fourier maps, and difference Fourier maps for all space groups except the hexagonal. (In practice, the hexagonal maps may be done under monoclinic or orthorhombic space groups.) Three-dimensional maps, projections down the crystallographic axes, sections with either x, y, or z set at a certain value, and sections through any specified plane in the structure may be calculated. The resultant maps may be on an inches per Angstrom scale or in fractional increments. (The general plane section must be on an inches per Angstrom scale.) Provisions have been made for saving the maps on a DATA TAPE EXTRA for future calculations.

It is assumed that anyone attempting to use this subprogram will be familiar with crystallographic Fourier techniques and expressions--in particular with the Fourier expressions for his particular space group.

This subprogram is called by either the PATTERSON, FOURIER, COMPLETE DIFFERENCE FOURIER, or DIFFERENCE FOURIER system control cards, depending upon which mode of calculation is to be done. If more than one mode are to be done, the subprogram must be called once for each mode. As many different maps as desired may be calculated within a given mode; for example, with one call to the subprogram in the Patterson mode, one may do a three-dimensional map along with a projection map.

Difference Fouriers:

The usual way to do difference Fouriers under this subprogram is to use the COMPLETE DIFFERENCE FOURIER mode. In this mode, the differencing is done between the observed structure-factor and the complete calculated structure-factor. The user should have the complete calculated structure-factor on the DATA TAPE from a previous call to the Structure-Factor Least Squares subprogram.

The DIFFERENCE FOURIER mode is designed for special cases where the user desires to subtract out only a few selected atoms from his Fourier. It is used in conjunction with the Omit Atoms option in the Structure-Factor Least Squares subprogram. The differencing is done between the observed structure-factor and the partial calculated structure-factor; the phasing may be done either by the complete calculated structure-factor or by the partial one. (See Chapters 3 and 6, pages 3-12 and 6-27.)

The calculation of either type of difference Fourier requires that reflection input be from the DATA TAPE. Of course, card input may be used if the user does the differencing and phasing elsewhere, then uses the results for card input in the FOURIER mode.

The Subprogram Deck:

The subprogram deck is arranged as follows: (1) Data processing control cards as desired--these cards contain information which the subprogram must have before it reads in the data. (If card input of reflections is used, a CELL card is required here.) (2) Map control cards--if necessary--for the first map. (3) If card input of reflections, a deck of Fourier subprogram short-format data cards, followed by an END OF DATA card. (If no card input deck is found by the subprogram, it will automatically assume DATA TAPE input of reflections; and no cards should be entered here.) (4) A calculation starting control card specifying which type of map is desired. (5) If more than one map is desired, map control cards--if needed--followed by a calculation starting control card, should followed by a calculation starting control card, should

be entered for each extra map desired. None of the options which are set by data processing control cards may be changed between successive maps; to do so requires a separate call to the subprogram. Examples of subprogram decks follow.

Example 1.

```
    THREE-DIMENSIONAL
```

The above subprogram deck (one card) will cause a three-dimensional map to be calculated using the reflections on a ΓATA TAPE which has been loaded prior to the call to the Fourier subprogram.

Note: The mode of calculation--Patterson, Fourier, etc--is set by the system control card which calls in the Fourier subprogram.

Example 2.

```
    SCALING FACTOR FOR F(OBS)  =   4.358
    USE ONLY DATA WITH SINE(SQ)THETA/LAMBDA(SQ) LESS THAN 0.3
    SCALE 3.0    INCHES/ANGSTROM
    PROJECTION  A    AXIS, Y CONSTANT HORIZONTAL
    CENTER POINT .4451 .3422 .5609       4.5    A. HORIZ.  8.0   ..A. VERTICAL
    GENERAL PLANE, DIRECTION COSINES .04562    .43557    .90023   ,Y CONSTANT HORIZ.
```

The above subprogram deck will cause a projection followed by a general plane map. Both will be on the scale of 3.0 inches/Angstrom; both will use limited data. The reflection data, as well as the size of the unit cell will be taken from the DATA TAPE.

Further examples of the use of this subprogram are contained in the examples given in Chapter 2.

General Planes:

A section through any general plane may be calculated. The user specifies the plane by giving a center point for the map and the direction cosines of the plane normal. The subprogram will orient the plane so that one fractional coordinate remains fixed in the horizontal direction on the map (the user may specify which). The user specifies the height and width of the map in Angstrom units. The maps are printed on an inches/Angstrom scale. These parameters are entered on a CENTER POINT and a GENERAL PLANE subprogram card. The CENTER POINT card must precede the GENERAL PLANE card which starts the calculation.

Phasing:

If card input of reflections is used for this subprogram, the phasing must have been done previously.

For reflection input from the DATA TAPE, the standard way of phasing the Fourier is to give F_{obs} the sign of F_{calc} in the centric case, and for the acentric case

$$A_{obs} = F_{obs} \frac{A_{calc}}{|F_{calc}|}$$

$$B_{obs} = F_{obs} \frac{B_{calc}}{|F_{calc}|}$$

For difference Fourier calculations for the acentric case, F_{obs} is replaced by $|F_{obs}| - |F_{calc}|$, and for the centric case, F_{obs} is given the sign of F_{calc} then the two phased quantities are subtracted.

Several special options are available. By using a PHASE CUTOFF FUNCTION card, all reflections with F_{calc} less than $(F_{calc})_{min}$ may be skipped, where $(F_{calc})_{min}$ is given by the function

$$(F_{calc})_{min} = F_{min}(1 + B \frac{\sin^2\theta}{\lambda^2})$$

$$(F_{calc})_{min} = F_{min}(1 + B \frac{\sin^2\theta}{\lambda^2})$$

where F_{min} and B are enterable constants.

If phases have been previously assigned by some other subprogram-- A and B for each reflection phased for direct use in a Fourier have been stored on the DATA TAPE--the use of these reflections may be effected by the PHASES PREVIOUSLY ASSIGNED control card.

When doing a map initiated by a DIFFERENCE FOURIER system control card, the differencing and phasing are both done with the partial structure-factor (stored by the Omit Atoms Option in the Structure-Factor Least Squares subprogram). If one wants to do the differencing using the partial structure-factor, but do the phasing using the complete calculated structure-factor, this may be effected by use of the DO DIFFERENCE PHASING subprogram control card.

Patterson Sharpening:

Data for use in a Patterson may be "sharpened". This option is effected by use of the SHARPEN subprogram control card. An estimated overall average temperature factor must be entered. A NUMBERS OF ATOMS card is required also. If it is desired to at the same time remove the origin peak, the card REMOVE ORIGIN should be entered. If the origin is removed, it is important that the proper scale factor for putting F_{obs} on the absolute scale is entered.

The sharpened $|F_{obs}{}^2|$ is calculated by the following expression,

$$|F_{obs}|^2_{sharpened} = \left[\frac{k^2\,(F_{obs})^2}{e^{-2B\left(\frac{\sin^2\theta}{\lambda^2}\right)}\;\Sigma n_i f_i{}^2} - 1\right] 2^{16}\left(\frac{\sin\theta}{\lambda}\right)^4 e^{-N\left(\frac{\sin\theta}{\lambda}\right)^2}$$

where n_i is the number of atoms of type i having atomic scattering factor f_i, B is the overall average temperature factor, N is an enterable parameter. If sharpening is done, reflection input must be from the DATA TAPE.

Scale Factor:

For difference Fourier maps and for sharpened Patterson maps, the scale factor necessary to put F_{obs} on the absolute scale must be entered by means of the SCALING FACTOR FOR F(OBS) card. It is advisable to enter a scale factor for all Fourier type calculations. If one is not entered, a scale factor of 1.0 is assumed. The scale factor from least squares is not saved for the Fourier; the Fourier scale factor must be entered separately.

Unobservable Reflections:

Reflections which were too weak to observe should be entered as "less than" reflections, (see Chapters 4 and 6), for input from the DATA TAPE. If this is done, the following action is taken:

a. Pattersons--$(F_{obs})_{max}$ is halved and treated as a regular reflection.

b. Difference Fouriers--If F_{calc} is greater than $(F_{obs})_{max}$, the reflection is treated as a regular reflection and enters the calculation. If F_{calc} is less than $(F_{obs})_{max}$, the reflection is skipped.

c. Fouriers (no differencing)--The reflection is skipped.

Printed Output:

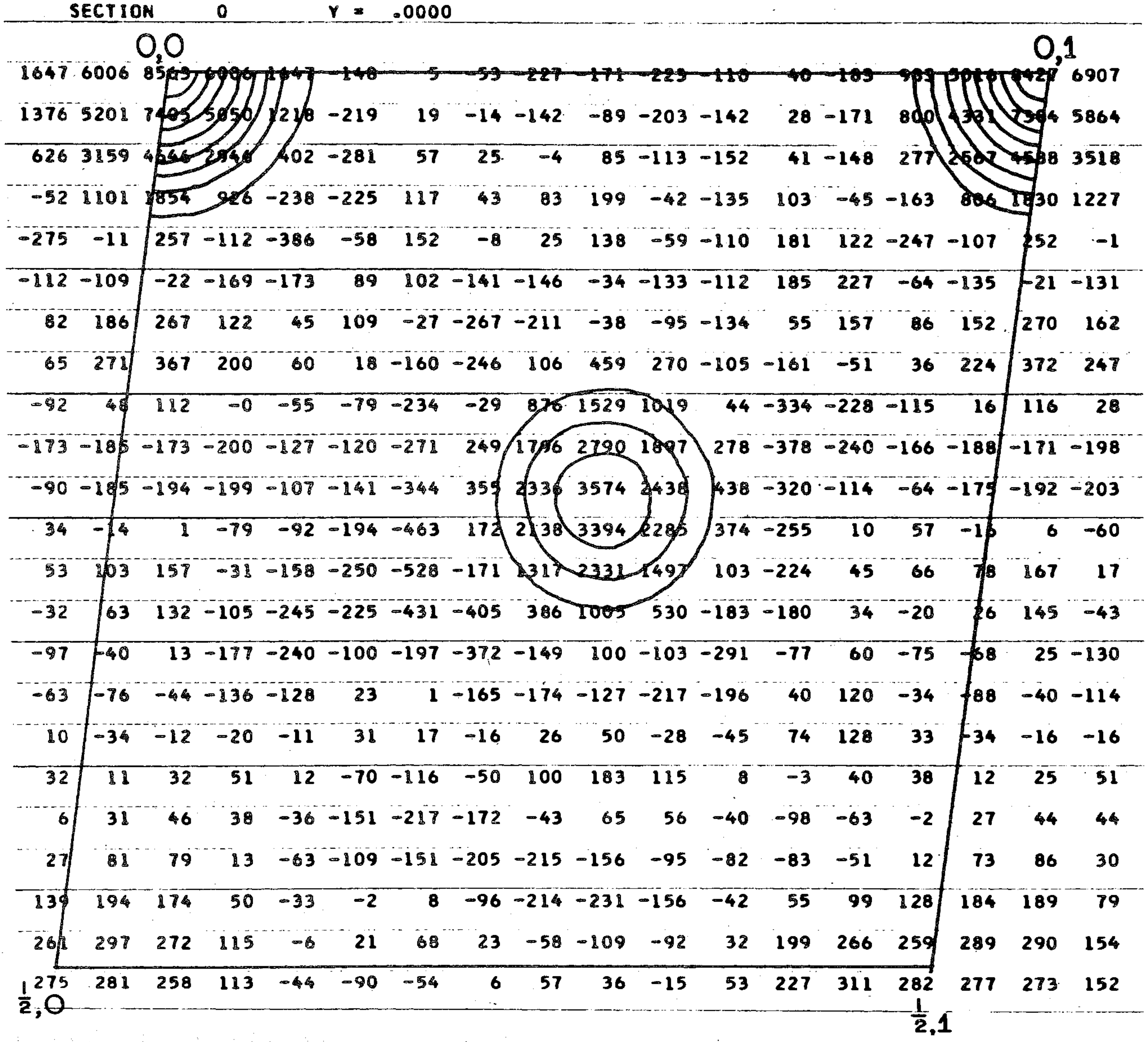
Output is arranged for a standard 131-character line with 2 map-points per inch in the horizontal direction and 3 map points per inch in the vertical direction of the page. If the map is too large, either horizontally or vertically, to fit on one page, it is continued on other pages. The largest map which can be printed is over 6 feet wide; the length is not limited.

The points are printed as signed integers, up to 4 digits long. If a point exceeds 4 digits, 9999 is printed. The normal option is to print hundredths of an electron for Fourier calculations, and tenths of an electron squared for Patterson maps. This can be changed by a POINT SCALE card.

printed. The normal option is to print hundredths of an electron for Fourier calculations, and tenths of an electron squared for Patterson maps. This can be changed by a POINT SCALE card.

The points may be either at such positions that the map is on an inches per Angstrom scale, or at fractional cell increments. Before each map, information is printed which allows the user to easily figure the fractional coordinates of any point on the map. Two inches per Angstrom is standard, but this may be changed by a SCALE card.

An example of output on an inches per Angstrom scale is shown in Figure 4. The unit cell has been drawn in afterwards. Note that the first point is not the origin, but the origin is shifted to the right on the first row, so that the entire asymmetric unit may be on the map. If the user makes a fractional coordinate grid to the proper inches per Angstrom scale, he can read fractional coordinates directly by using the grid to "measure" the map.



```
     SECTION      0          Y =  .0000

1647 6006 8563 6006 1647 -148    5  -53 -227 -171 -229 -110   40 -183  983 5016 8427 6907
1376 5201 7405 5050 1218 -219   19  -14 -142  -89 -203 -142   28 -171  800 4331 7364 5864
 626 3159 4646 2946  402 -281   57   25   -4   85 -113 -152   41 -148  277 2567 4588 3518
 -52 1101 1854  926 -238 -225  117   43   83  199  -42 -135  103  -45 -163  806 1830 1227
-275  -11  257 -112 -386  -58  152   -8   25  138  -59 -110  181  122 -247 -107  252   -1
-112 -109  -22 -169 -173   89  102 -141 -146  -34 -133 -112  185  227  -64 -135  -21 -131
  82  186  267  122   45  109  -27 -267 -211  -38  -95 -134   55  157   86  152  270  162
  65  271  367  200   60   18 -160 -246  106  459  270 -105 -161  -51   36  224  372  247
 -92   48  112   -0  -55  -79 -234  -29  876 1529 1019   44 -334 -228 -115   16  116   28
-173 -185 -173 -200 -127 -120 -271  249 1796 2790 1897  278 -378 -240 -166 -188 -171 -198
 -90 -185 -194 -199 -107 -141 -344  355 2336 3574 2438  438 -320 -114  -64 -175 -192 -203
  34  -14    1  -79  -92 -194 -463  172 2138 3394 2285  374 -255   10   57  -16    6  -60
  53  103  157  -31 -158 -250 -528 -171 1317 2331 1497  103 -224   45   66   78  167   17
 -32   63  132 -105 -245 -225 -431 -405  386 1005  530 -183 -180   34  -20   26  145  -43
 -97  -40   13 -177 -240 -100 -197 -372 -149  100 -103 -291  -77   60  -75  -68   25 -130
 -63  -76  -44 -136 -128   23    1 -165 -174 -127 -217 -196   40  120  -34  -88  -40 -114
  10  -34  -12  -20  -11   31   17  -16   26   50  -28  -45   74  128   33  -34  -16  -16
  32   11   32   51   12  -70 -116  -50  100  183  115    8   -3   40   38   12   25   51
   6   31   46   38  -36 -151 -217 -172  -43   65   56  -40  -98  -63   -2   27   44   44
  27   81   79   13  -63 -109 -151 -205 -215 -156  -95  -82  -83  -51   12   73   86   30
 139  194  174   50  -33   -2    8  -96 -214 -231 -156  -42   55   99  128  184  189   79
 261  297  272  115   -6   21   68   23  -58 -109  -92   32  199  266  259  289  290  154
 275  281  258  113  -44  -90  -54    6   57   36  -15   53  227  311  282  277  273  152
```

Figure 4

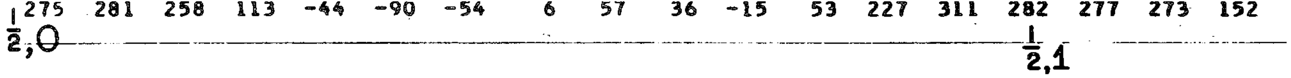


Figure 4

Data Processing Control Cards:

CELL

```
======AAAAAAA=BBBBBBB=CCCCCCC=AAAAAABBBBBBCCCCCC==============================
```

For a format description see Chapter 4, page 4-20.

Note:

This card is required when reflection input is by cards. If the reflections come in from a DATA TAPE, the cell on the DATA TAPE will be used if no CELL card is entered.

DO DIFFERENCE PHASING WITH COMPLETE CALCULATED STRUCTURE FACTORS

```
================================================================================
```

Action:

This card causes the Fourier subprogram to phase on the basis of the complete calculated structure factor when it does a difference Fourier. The differencing is still done the partial calculated structure factor.

Note:

This card is allowed in the DIFFERENCE FOURIER MODE only. Phasing on the basis of the partial calculated structure factor is standard unless this card is entered.

NUMBERS OF ATOMS

```
=================NN-fff,N-ff,NNN-ff,...                                       ==
```

For a format description see Chapter 5, page 5-13.

Note:

This card is used in conjunction with a SHARPEN card to provide the constants for sharpening data. For centered cells, multiply the number of atoms in the unit cell by the centering multiplicity.

PHASE CUTOFF FUNCTION, F(MIN)= , B =

```
==============================XXXXX======XXXXXX==============================
```

Required Constants:

1. A value for a minimum calculated structure factor in col. 32-36.
2. A parameter for the phase cutoff function in col. 43-48.

Action:

The Fourier subprogram is set to calculate a value $F(calc)_{min}$ for each reflection prior to phasing it. The following equation is used:

$$F(calc)_{min} = F(MIN)(1 + B \sin^2\theta/\lambda^2)$$

If $F(calc) < F(calc)_{min}$, the reflection is not included in the Fourier.

Note:

This option applies to tape input of reflections only.

PRINT INPUT CARDS

```
==============================================================================
```

Action:

This card causes reflection data cards to be listed as they come into the Fourier subprogram.

Note:

Allowed only when card input of reflections is used.

REMOVE ORIGIN

Action:

The Fourier subprogram is caused to subtract the appropriate term to remove the origin peak for a sharpened Patterson map.

Note:

1. This card is allowed only in the PATTERSON mode, and there only in the company of a SHARPEN and a NUMBERS OF ATOMS cards.
2. If the origin removal is to be any good, the data should be put on the absolute scale by use of a SCALING FACTOR card.

SCALING FACTOR FOR F(OBS) =

=============================XXXXXX==

Required Constant:

A scale factor to be applied to F(obs) in col. 31-36.

Action:

The observed structure factors are multiplied by the given factor as they are read in.

Note:

1. Allowed for tape input of reflections only.
2. This card is especially important for sharpened Pattersons and difference Fouriers if the F(obs) which are on the DATA TAPE differ significantly from the absolute scale.

```
SHARPEN, OVERALL AVE. TEMP FACTOR =       ,FUNCTION PARAMETER =
====================================XXXXXX=========================XXXXXX=======
```

Optional Constants:

1. An overall average temperature factor in col. 37-42. If no value is punched, 4.0 is assumed.
2. A parameter for the sharpening function in col. 67-72. If no parameter is punched, 19.36 is assumed. This value was chosen to show up peaks in standard-type organic structures.

Action:

The subprogram is set to "sharpen" the data as it comes into memory.

Note:

1. Allowed when reflection input is by DATA TAPE only; not allowed for card input.
2. A NUMBERS OF ATOMS card must accompany this card. A REMOVE ORIGIN card may accompany this card, but is optional.

```
USE ONLY DATA WITH SINE(SQ)THETA/LAMBDA(SQ) LESS THAN
==========================================================XXXXX=====================
```

Required Constant:

A maximum $\sin^2\theta/\lambda^2$ in col. 55-59.

Action:

Reflections whose $\sin^2\theta/\lambda^2$ is greater than the maximum entered value are not used in the calculations.

Note:

If this card is not used, the Fourier subprogram provides storage for all reflections up to $\sin\theta = 1.0$ for the wavelength entered. If this card is used, storage is only provided for reflections out to the maximum specified here. By using this card, the subprogram may be speeded up considerably.

out to the maximum specified here. By using this card, the subprogram may be speeded up considerably.

POINT SCALE ELECTRONS(OR ETC.)

===========XXXXXX==

Required Constant:

A constant for fixing the scale of points in the map(s) to be calculated. Points are printed out as integers; if the printed points are multiplied by the number specified here, they become Electrons/A^3(or whatever the correct units are).

Action:

The data is modified at reflection-read-in time so that the points will have the specified scale.

Note:

If this card is not used, the point scale is taken as 0.01 for Fourier calculations and as 0.1 for Patterson calculations.

WAVE LENGTH

===========XXXXXX==

For a format description see Chapter 4, page 4-29.

Note:

The Fourier will provide storage for all reflections in the specified asymmetric unit of data out to sin θ = 1.0 for the wavelength specified. If this card is not entered Cu wavelength is assumed. The Fourier subprogram may be significantly slowed down by forcing it to provide storage for reflections which never appear. If a USE ONLY card is entered, the value on that card is used in defining the necessary storage in preference to the wavelength.

Map Control Cards:

```
BOUNDS  X =          TO          , Y =          TO          , Z =          TO
============XXXXXX======XXXXXX======YYYYYY======YYYYYY======ZZZZZZ======ZZZZZZ==
```

Required Constants:

1. A lower bound for x in col. 13-18, an upper bound for x in col. 25-30.
2. A lower bound for y in col. 37-42, an upper bound for y in col. 49-54.
3. A lower bound for z in col. 61-66, an upper bound for z in col. 73-78.

Lower bounds must be negative or zero, upper bounds must be positive or zero.

Action:

The asymmetric unit within the unit cell is defined for the subprogram. This is used by the subprogram to decide the size of its projection, section and three-dimensional maps.

Note:

If this card is not entered, the asymmetric unit which is stored in the subprogram for the particular space group is used.

```
CENTER POINT                                    A. HORIZ.         A. VERTICAL
=============XXXXX=YYYYY=ZZZZZ======tttttt============ssssss==================
```

Required Constants:

1. The x,y,z coordinates of a center point (in fractional coordinates relative to the real cell axes) in col. 14-18, 20-24, 26-30 respectively.
2. Constants in Angstroms giving the horizontal and vertical extent of the general plane whose center point is defined here, in col. 37-42 and 55-60 respectively.

Action:

This card is used together with a GENERAL PLANE card to specify the parameters for defining that plane. It must be followed by the GENERAL PLANE card to which it applies.

parameters for defining that plane. It must be followed by the GENERAL PLANE card to which it applies.

```
SAVE MAP ON DATA TAPE EXTRA
===============================================================================
```

Action:
The subprogram is caused to save the maps which are calculated after this card enters on the DATA TAPE EXTRA.

Note:
A DATA TAPE EXTRA must have been loaded prior to the call to the Fourier subprogram if this card is to be used.

```
SCALE        INCHES/ANGSTROM
======XXXXXX===================================================================
SCALE        IN X,       IN Y,       IN Z
======XXXXXX=======YYYYY=======ZZZZZ===========================================
```

Two types of SCALE card are allowed. The first type specifies the scale for a to-scale map.

Required Constant:
A scale for the map(s) which follow in inches/Angstrom in col. 7-12.

The second type of SCALE card causes the intervals between points in the x, y and z directions to be as specified on this SCALE card. The maps calculated after this card is processed will therefore not be on an inches/Angstrom scale.

Required Constants:
1. A value for the increment in x in col. 7-12.
2. A value for the increment in y in col. 20-24.
3. A value for the increment in z in col. 32-36.
These values must be in fractional coordinates relative to the real cell axes.

Action:
The subprogram is set to conform with the option given on the scale card. In the absence of a SCALE card, a to-scale map on the scale 2.0 inches/Angstrom is assumed.

Card Input of Data:

```
     HHHH  KKKK  LLLL  AAAAAA  BBBBBB
==============================================================
```

Above is the format for a Short Format Data Card for input to the Fourier subprogram. All A(obs) and B(obs) which are punched must have the phase punched. If data for a Patterson is being entered, the A(obs) position must contain F(obs)^2. No phasing or scaling of card input is provided for.

Required Constants:

1. The indices h,k and l must be punched in col. 7-10, 13-16 and 19-22 respectively, with the integer right justified. All minus signs should immediately precede the first significant digit of the integer.
2. Phased values for A(obs) and B(obs) in col. 25-30 and 33-38 respectively. B(obs) of course is not required for centric structures or for Pattersons.

```
END OF DATA
==============================================================
```

Action:

When this card is encountered, the subprogram assumes that all the Short Format Data Cards for the card input of reflections have entered.

Note:

The deck for card input of data must consist of a group of Short Format Data Cards, like the one above, followed by an END OF DATA card. These cards must be placed before the first Calculation Starting Control Card in the Fourier subprogram deck.

Calculation Starting Control Cards:

These control cards cause the specified kind of map to be calculated. If a card input of reflections has not come in when the first Calculation Starting Control Card is encountered, the subprogram assumes that reflections will come in from the DATA TAPE.

```
GENERAL PLANE, DIRECTION COSINES                           , CONSTANT HORIZ.
=================================XXXXXXX==YYYYYYY==ZZZZZZZ==?==================
```

Required Constants:

1. Direction cosines, with respect to the real cell axes, for the general plane to be calculated. These numbers must be the cosines of the angles between the plane normal and the real cell axes. Cosines with respect to the a, b, and c axes are punched in col. 34-40, 43-49, and 52-58 respectively.
2. Either X, Y, or Z in col. 61, depending on which fractional coordinate is to be held constant along rows of the general plane section to be calculated.

Action:

The subprogram is caused to calculate a section through the plane specified. The map will be on an inches/Angstrom scale.

Note:

This card must have been preceded by a CENTER POINT card containing the rest of the information for defining the plane to be calculated.

```
PROJECTION        AXIS
============?========, ? CONSTANT HORIZONTAL=====================================
```

Required Constant:

Either A, B, or C in col. 14, depending upon which major axis is to be projected down.

Optional Field:

If ", CONSTANT HORIZONTAL" is punched on the card in col. 23-45, with either X, Y, or Z punched in col. 25, the map will be oriented such that the specified fractional coordinate will be held constant along rows in the map.

Action:

A projection map down the specified axis is calculated.

```
SECTION    =
========?===XXXXXX====, ? CONSTANT HORIZONTAL==================================
```

Required Constants:

1. Either X, Y, or Z in col. 9, depending on which fractional coordinate is to be held constant in the section.
2. The value at which the specified fractional coordinate is to be held, in col. 13-18.

Optional Field:

If ", CONSTANT HORIZONTAL" is punched on the card in col. 23-45, with either X, Y, or Z punched in col. 25, the map will be oriented such that the specified fractional coordinate will be held constant along rows in the map.

Action:

A section through the cell as specified on the card is calculated.

through the cell as specified on the card is calculated.

```
THREE-DIMENSIONAL
================, SECTIONS OF CONSTANT ? WITH ? CON.HORIZ.,AT INCREMENT XXXXX==
```

Optional Fields:

1. If ", SECTIONS OF CONSTANT " is punched in col. 18-41, with either X, Y, or Z in col. 41, depending upon which fractional coordinate is to be held constant in each section; the subprogram will do the three-dimensional map with sections as specified.
2. If "WITH CON.HORIZ." is punched in col. 43-59, with either X, Y, or Z in col.48, depending upon which fractional coordinate is to be held constant in each row of each section, the sections will be oriented as specified.
3. If ",AT INCREMENT" is punched in col. 61-72, with an increment in fractional coordinates for the real cell axis punched in col. 73-78, the increments between the sections of the three-dimensional map will be taken as the value specified. If no value is specified, the program will calculate a value such that the Angstrom distance between sections will be comparable to the Angstrom distance between points within given sections. If the map is not to be done to-scale, the increment on the SCALE card would be taken.

Action:

The subprogram is caused to calculate a three-dimensional map as specified.

Chapter 8

Distances, Angles, and Planes

Programming Contribution: J. K. Lo

Introduction:

This subprogram calculates distances, angles, and least squares planes from cell parameters and atomic coordinates. Individual distances or angles may be specified for calculation. Also within the group of atoms, all interatomic distances less than a certain minimum may be calculated, alone or together with all interatomic angles for which the distances from the apical atom to the other two are both within the specified minimum.

In addition, a symmetry relation may be introduced for generating another group of atoms from the original group; and between the atoms of the new group and the atoms of the old group, all distances less than a certain minimum may be calculated.

This subprogram is called in by a DISTANCES, ANGLES, AND PLANES system control card (see Chapter 3). It will handle a maximum of 200 atoms. Atom numbers need not be sequential, but calculations are faster if the largest atom number is as small as possible (in no case greater than 200).

The Subprogram Deck:

The parameter deck consists of first a CELL card (unless a cell from tape is being used, in which case, no CELL card is entered). Next a group of COORD cards for reading in atomic coordinates, followed by an END ATOMIC INFORMATION card (again if atomic coordinates are to come from the DATA TAPE, the COORD cards and the END ATOMIC INFORMATION card are omitted).

Next come the option cards, as many as required in any order.

Note: The cell and/or atomic coordinates may be entered either by input cards or from the DATA TAPE. If either are not entered by cards, the subprogram looks for a DATA TAPE, and if one has been loaded, gets the missing input from tape. If both cell parameters and atomic coordinates are entered by cards, no tape is required.

Example 1.

```
CELL   10.35   13.17   5.364
COORD    1  C    .3542      .2244     .1233
COORD    3  N    .1775      .2344     .4555
  :      :  :      :          :         :
COORD   11  H    .4667      .3442     .6221
END ATOMIC INFORMATION
COMPLETE DISTANCES AND ANGLES  3.5
PLANE 1,3,4,5,6,8,10
WEIGHT1.0,1.0,0.0,1.0,1.0,0.0,1.0
ADDED ATOMS -X          -Y          -Z            3.0
ADDED ATOMS X           1+Y         Z             3.0
```

In the above example subprogram deck, the cell and atomic coordinates are read in from cards; all distances and angles are calculated within the minimum of 3.5 Angstroms; a least squares plane involving a selected group of atoms, appropriately weighted, is calculated; then a group of symmetry related atoms is generated, and the distances which are less than 3.0 A. between atoms in the generated group and atoms in the original group are calculated; and finally another generated group and other distances are calculated.

Example 2.

```
COMPLETE DISTANCES           1.8
```

COMPLETE DISTANCES 1.8

In the above example subprogram deck (one card), the cell parameters and atomic coordinates are taken from a DATA TAPE which must have been previously loaded. Then all interatomic distances less than 1.8 A. are calculated and printed. This type of deck may be profitably used immediately after parameter least squares in order to obtain interatomic distances using the newest atomic parameters which the Least Squares Subprogram has placed on the DATA TAPE.

Orthogonalization of Coordinates:

Before the calculations of this subprogram are made, all coordinates are orthogonalized and put on an Angstrom (as opposed to fractional) basis. The equations for this operation are

$$x' = xa \sin\gamma + zc\left(\frac{\cos\beta - \cos\alpha \cos\gamma}{\sin\gamma}\right)$$

$$y' = yb + ax \cos\gamma + cz \cos\alpha$$

$$z' = zc \left(\frac{1 - \cos^2\alpha - \cos^2\beta - \cos^2\gamma + 2\cos\alpha\cos\beta\cos\gamma}{\sin^2\gamma}\right)^{1/2}$$

where x', y', and z' are the orthogonalized coordinates; x, y, and z are the fractional coordinates in the crystallographic cell; and a, b, c, α, β, and γ are the cell parameters of the crystallographic cell. The orthogonalized axes are defined as follows: the direction of b remains fixed; the orthogonalized a is chosen to be in the a,b plane perpendicular to b; the orthogonal c is chosen to complete the orthogonal set.

The orthogonalized coordinates are printed for the original set of atoms and for any symmetry related set which may be generated.

Least Squares Planes:

Least squares planes are calculated by the method given in V. Schomaker, J. Waser, R.Marsh, and G.Bergman, Acta Cryst, 12, 600(1959).

The direction cosines with respect to orthogonalized and actual cell (deorthogonalized) axial systems are printed together with the origin to plane distance. Next the deviation from the plane of all the atoms which appeared on the PLANE card are printed, together with their weights. In order to see the deviation of a certain atom from the least squares plane, but have it not influence the calculation of the plane; one should enter the atom on the PLANE card, and give it zero weight on the WEIGHT card.

If a WEIGHT card does not immediately follow the PLANE card, all weights are assumed to be unity. If a WEIGHT card does follow the PLANE card, there must be as many entries on the WEIGHT card as on the PLANE card--the nth entry on the WEIGHT card being the weight of the nth atom on the PLANE card.

If the list of atoms on the PLANE card (or the list of weights on the WEIGHT card) is longer than can be punched on one card, the list may be continued on another PLANE (or WEIGHT) card.

Ignored Cards:

In order to make this subprogram as compatible as possible with the Least Squares Subprogram, the following cards if they occur before the END ATOMIC INFORMATION card are ignored and no error results: FULL MATRIX, BLOCK MATRIX, DIAGONAL MATRIX, NO REFINEMENT, ISO, ANISO, ISO-AN, POP, and SCALE FACTOR.

Distances, Angles, and Planes Subprogram Control Cards

The following is an alphabetic listing of cards for input to this subprogram.

```
ADDED ATOMS
============XXXXXXXXXXXXYYYYYYYYYYYYZZZZZZZZZZZZLLLLLL==========================
```

Required Constants:

1. Symmetry relations for generating a new set of atoms from the original one--the equations for generating the new "x", "y", and "z" coordinates beginning in columns 13, 25, and 37 respectively. The format is as follows: 1. The letters X,Y, Z, -X,-Z,-Y (if required) referring to the coordinates of the original set; 2. A sign (+or-) followed by an integer or a common fraction. Examples of symmetry relations are:

X+1/2	-Y	Z
X	Y+5/3	-Z-1/3
1/2	X-1	-Z+2

2. A maximum distance in Angstrom units in col. 49-54.

Action:

1. A new set of atoms is generated from the original set, the fractional coordinates of each new atom being obtained by taking the fractional coordinates of the corresponding old atom and substituting them in the symmetry equations which were punched on the card. Each old atom generates one new atom. The orthogonal coordinates of the new atoms are printed.
2. All distances between atoms of the old set and atoms of the new set which are less than the maximum distance (entered on the card) are printed.

Note:

The symmetry relations punched on the card refer to fractional coordinates with respect to the real crystallographic axes, not to any orthogonalized cell. They may be punched exactly as taken from the International Tables, Vol. 1, for example.

ANGLE

```
===============NNN===NNN===NNN==================================================
```

Required Constants:
Three atom numbers, right justified, in col. 16-18, col. 22-24, and col. 28-30 respectively.

Action:
The angle formed at the second atom(by the line connecting the first and second atoms, and the line connecting the second and third atoms) is calculated and printed.

Note:
The atom numbers referred to here are the same as those punched on the COORD cards.

CELL

```
======AAAAAAA=BBBBBBB=CCCCCCC=AAAAAABBBBBBCCCCCC================================
```

For a format description see Chapter 4, page 4-20.

Note:
This card is required unless a DATA TAPE has been loaded; in which case, the cell parameters will be taken from the DATA TAPE <u>if</u> no CELL card appears in the subprogram deck. If both are possible, the parameters from the CELL card are used.

COORD

```
======NNNN==FFF=XXXXXXXX=YYYYYYYY=ZZZZZZZZ======================================
```

For a format description see Chapter 6, page 6-18.

Note:
Any constants or fields appearing after col. 43 on the COORD card have no effect on the Distances, Angles, and Planes subprogram.

COMPLETE DISTANCES

```
==============================LLLLLL============================================
```

COMPLETE DISTANCES AND ANGLES

```
==============================LLLLLL============================================
```

Required Constant:
A maximum distance in Angstrom units in col. 31-36.

Action:

1. All interatomic distances which are less than the maximum distance (entered on the card) are calculated and printed. In this calculation, all possible pairs of atoms are checked.
2. If complete angles have also been requested, next follows the calculation and printing of all interatomic angles which satisfy the following condition: The angle "Atom 1-Atom2-Atom 3" is calculated if both the distance from Atom 1 to Atom 2 and the distance from Atom 2 to Atom 3 are less than the maximum distance which has been entered.

Note:
Complete angles may not be calculated without complete distances.

DISTANCE

```
===============NNN===NNN========================================================
```

Required Constants:
Two atom numbers, right justified, in col. 16-18 and col. 22-24, respectively.

Action:
The distance in Angstroms between the two specified atoms is calculated and printed.

Note:
The atom numbers referred to here are the same as those punched on the COORD cards.

END ATOMIC INFORMATION
==

Action:
Reading this card signals the subprogram that the entry of COORD cards has been completed.

Note:
This card should not appear in the subprogram deck if atomic coordinates are being taken from the DATA TAPE. It must appear if coordinates are entered on cards.

PLANE
======N,NN,N,NN,··· ==

Required Constants:
A list of at least three atom numbers, entered as integers separated by commas with no blank columns in between, beginning in col. 7 and not exceeding col. 78. If one PLANE card is not sufficient for entry of a long list, a comma should be forced in col. 78 (with no blank columns within col. 7-78). If this occurs, the subprogram will expect the list to be continued on another PLANE card which must immediately follow this one. No comma should be punched after the last atom number in the list. The atom numbers need not be arranged in numerical order.

Action:
1. The direction cosines and origin to plane distance of the best least squares plane through the atoms of the list will be calculated and printed.
2. The deviations (in Ångstrom units) of each atom in the list from the best plane will be calculated and printed.

Note:
1. Weights for each atom for the plane calculation may be entered on a WEIGHT card.
2. Preceding zeros have no effect on atom numbers.

1. Weights for each atom for the plane calculation may be entered on a WEIGHT card.
2. Preceding zeros have no effect on atom numbers.

```
WEIGHT
======X.X,X.X,X.X,XX.X,···                                                 ==
```

Required Constants:

A list of weights, entered with punched decimal points, each entry separated by a comma from the next, the first entry beginning in col. 7 and nothing being punched beyond col. 78. If one WEIGHT card is not sufficient for entry of a long list of weights, a comma should be forced in col. 78 (with no blank columns within col. 7-78). If this occurs, the subprogram will expect the list to be continued on another WEIGHT card which must immediately follow this one. No comma should be punched after the last weight in the list. The list of weights must have the same number of entries as the list of atom numbers to which it refers. The "n th" entry in the weight list corresponds to the atom in the "n th" position of the atom number list.

Action:

The atoms entered on the PLANE card(or cards) are given the weights entered here when they are used in the least squares plane calculation.

Note:

1. If a WEIGHT card does not follow the PLANE card (or cards) which read in the atom number list, every atom in the atom number list will be assigned unit weight.
2. More than one extra WEIGHT card may be used if the weight list will not fit on two WEIGHT cards.
3. When multiple card lists are being used, all the PLANE cards (there may be more than two) which enter one atom number list must precede the first WEIGHT card of the weight list.

Chapter 9

Automatic Diffractometer Calculations

Programming Contribution: D. Duchamp.

Discussion:

This subprogram prepares input tapes for and processes output tapes from an automatic diffractometer. It is designed specifically for a G. E. Single Crystal Orienter with Datex automation using paper tape input and output. It can also be used to calculate a list of settings for any Eulerian cradle diffractometer setup. With certain modifications, it could be expanded easily to include other types of diffractometers, or other than paper tape diffractometer input and output.

Control tapes for the automatic diffractometer are generated by this subprogram, through a remote console. These control tapes are fed into the automatic diffractometer, which punches an output tape as intensities are measured. The output tape is then fed back to the computer through the remote console and into this subprogram where it is processed.

This subprogram is called by the AUTOMATIC DIFFRACTOMETER system control card. Once called, it may operate in either (but not both) the Control Tape Preparation Mode or the Output Processing Mode.

If the user so desires, the output from the subprogram's Output Processing Mode may be put on a (or added to a previously prepared) DATA TAPE EXTRA--with the same format as the short format tape generated by the Initial Data Processing subprogram. This DATA TAPE EXTRA may be used as input to the Data Tape Preparation and Editing subprogram. If the reflections are in several groups which, for some reason, are on different scales--because of, for example, different crystals or different filter arrangements--the groups may be scaled together using the Initial Data Processing subprogram--provided, of course, that enough multiply-observed reflections occur to provide correlation. In the same way, groups of reflections obtained by film methods and groups obtained by counter methods may be interscaled by gathering them together on the same DATA TAPE EXTRA and using this as DATA TAPE input to the Initial Data Processing subprogram. (See Chapter 4.)

The Control Tape Preparation Mode:

In this mode the subprogram calculates angle settings and prepares a control tape for an automatic diffractometer. The crystal position, unit cell, and Laue group must be specified. The crystal is assumed to be oriented with some unit cell axis (specified by the user) perpendicular to the phi plane (i.e. vertical when chi = 0.0). Reflection indices may either be specified by the user or generated by the subprogram. If indices are generated by the subprogram, restrictions may be placed on $\sin^2\theta/\lambda^2$ and on the range of the indices. The subprogram then generates indices for all reflections in the asymmetric unit of the Laue group (see page 1-5), provided such reflections satisfy the entered restrictions and are within the range of the diffractometer. Periodic recounting of the same reflection to check constancy of the apparatus is provided as an option (see CHECK card).

The standard option is to prepare a control tape for 2θ scans, but omega scans are also allowed. The sequence of commands on the control tape for a 2θ scan first identifies the reflection with an I.D. Transfer word. Then chi, phi, and omega, as required, are positioned by first slew positioning then slow positioning.(The slow positioning may be suppressed.) Then 2θ is moved to a position half the scan range (see SCAN RANGE card) below the 2θ calculated for the wavelength being used, by first slew positioning to a slightly lower value, then slow positioning to the precise value. Background is counted here. Next the reflection first slew positioning to a slightly lower value, then slow positioning to the precise value. Background is counted here. Next the reflection

Next the reflection is scanned the number of degrees specified in the scan range, and background is counted at the upper limit of the scan. Either of the two backgrounds may be suppressed. The program is written so that the sequence of command specified above may be easily changed, and so that other sequences may be easily added as options.

The Control Tape Preparation Mode Subprogram Deck:

This deck consists of cards giving information and specifying options for the preparation of a control tape. The CELL and CRYSTAL POSITION cards are required as is the MAKE CONTROL TAPE card. The MAKE CONTROL TAPE card must be last except in the case of specified reflections; in which case, it is followed by a group of index cards specifying the reflections--the deck in this case being ended by an END OF DATA card.

Two examples of subprogram decks for control tape preparation are given below. The first also shows the system control cards necessary.

Example 1.

```
$ID             DJDTST1,CH25000,1
$EXECUTE        CRYRM
ERROR DUMP REQUESTED
AUTOMATIC DIFFRACTOMETER
CELL   265201  164197  265514        91528
LAUE GROUP        2, C CENTERED
CRYSTAL POSITION A VERTICAL    DELTA PHI = 4.54  FOR B*.
SINE(SQ)THETA/LAMBDA(SQ) MAX= 0.05  ,MIN = 0.0
SCAN 2THETA                     NO SLOW POSITIONING     SPEED 2
SCAN RANGE  3.0   DEG. AT 2TH= 6.0    AND 4.0   DEG. AT 2TH= 100.0
CHECK   0  2  0 EVERY 15 REF.
MAKE CONTROL TAPE                                    ,SLOW H,FAST L
COMMENT
```

The above subprogram deck will generate a control tape for all reflections within the limits specified and, on the main printer, a listing of all these reflections together with their angle settings.

Example 2.

```
CELL   265201  164197  265514       91528
LAUE GROUP        2, C CENTERED
CRYSTAL POSITION A VERTICAL   DELTA PHI = 0.0    FOR B*.
MAKE CONTROL TAPE  SPECIFIED REFS.
    0    0   10
    0    0   12
    0    0   14
    0    6   15
    0    8    7
    0    2    0
    0   20    6
    0    8   19
    0    4    0
    8    0    0
   28    0    0
   24    0    0
   14    0    0
END OF DATA
```

The above subprogram deck will generate a control tape for the indicated reflections. A deck of this type might be used just to calculate preliminary settings for lining up the crystal on the diffractometer. A listing of all reflections with their settings is automatically printed on the main printer when a control tape is prepared at the remote console.

The Cutput Processing Mode:

In this mode the subprogram processes the output tape from the diffractometer and converts the information thereon to a set of observed structure factor amplitudes. The equation used in converting from counts to average intensity is:

structure factor amplitudes. The equation used in converting from counts to average intensity is:

$$I_{ave} = g\left[S - \left(\frac{B_1 + B_2}{2t}\right)\left(\frac{n}{m} + k\right)\right]$$

where B_1, B_2 = background counts (fixed time)
t = time for background count
S = scan count
m = scanning speed in °/min.
n = number of degrees scanned over
k = time lag constant (corrects time calculated from the number of degrees and the speed)
g = (I/counts) a factor.

On the DATA FROM CRYSTAL card the user must enter t (in seconds), g, and k. The B_1, B_2, S, m, and n are obtained from the diffractometer output tape. If I comes out negative, it is set equal to zero. The standard deviation is calculated by the following:

$$\sigma^2\left(\frac{I}{g}\right) = S + (B_1 + B_2)\alpha^2 + (dS)^2$$

where $\alpha = \dfrac{n}{2mt} + \dfrac{k}{2t}$

d = a weight constant and must be entered (see WEIGHT CONSTANT card)

and all the other quantities are as above. The weight constant is assumed 0.0 if none is entered.

The reflection intensity is then corrected for the Lorentz and polarization effects by multiplying the average intensity, I_{ave}, by a factor calculated by the following equation:

$$\frac{1}{LP} = \frac{\sin^2\theta - \sin^4\theta}{1 - 2(\sin^2\theta - \sin^4\theta)} .$$

This factor may easily be altered if the user prefers a different function, or if the output is from a different type of diffractometer.

Provisions are made for the application of absorption correction factors by this subprogram, however, as in the Initial Data Processing subprogram, this subprogram contains no routine to calculate these factors. If the user desires to do an absorption correction, he should enter a DO ABSORPTION CORRECTION card, and using the PROGRAM CHANGES FOLLOW option, he should enter a short octal overlay program into memory after this subprogram, and set location RUN5 to transfer to this overlay program. This overlay program will be entered for each reflection; it should return to location SRUN1 with the accumulator containing the floating-point binary absorption factor to be multiplied by the intensity value of the reflection being processed. The main subprogram will handle all printing of the absorption factor and its actual application.

Whenever factors are applied, the standard deviation is also scaled, so that the final standard deviation S I(CR) is on the same scale as the final intensity I(CORR). A weight equal to one over the standard deviation in the final intensity--$1/\sigma|F_{obs}|^2$--is printed, and stored for the magnetic tape output.

If a DATA TAPE EXTRA was loaded before the AUTOMATIC DIFFRACTOMETER card, the subprogram will automatically place the final results on this tape as outlined previously. The normal option is to start placing the results at the beginning of the tape; but if an ADD TAPE OUTPUT TO DATA TAPE EXTRA card is entered in the subprogram deck, the output previously placed on the DATA TAPE EXTRA will be skipped over, and the present output will be placed after it.

If a CHECK reflection was used in collecting the data, a CHECK card, identical to that used in generating the tape, should be entered. The CHECK reflection when it occurs in the output will be marked. The first five CHECK reflections will be averaged--each quantity I, B_1, B_2, and S--and standard deviations, based on the precision within the five observations, will be calculated. These average values will be compared with the corresponding values for each following CHECK reflection. If the disagreement is greater than three standard deviations (as calculated for the will be calculated. These average values will be compared with the corresponding values for each following CHECK reflection. If the disagreement is greater than three standard deviations (as calculated for the

first five observations), the reflection is marked as being in bad agreement. The CHECK reflection is only placed on the output DATA TAPE EXTRA once--when the average of the first five is calculated.

The Output Processing Mode Subprogram Deck:

This subprogram deck contains the diffractometer output and the information for processing it. A CELL card must be entered, along with option cards, as desired, followed by a DATA FROM CRYSTAL card. After the DATA FROM CRYSTAL card, diffractometer output must be entered with two diffractometer words per line, the words beginning in col. 1 and col. 16 respectively. This can be effected by placing the actuator board in the typewriter of the remote console after the DATA FROM CRYSTAL card has been entered, then reading the diffractometer output tape--starting with the first character of the first word. If there is an odd number of diffractometer words, the position of the last word in the last line should be left blank. After one group of diffractometer data has been entered, other groups, each preceded by a DATA FROM CRYSTAL card may be entered. After all groups have been entered, an END OF DATA card must be entered. This is the last card of the subprogram deck.

Reading in of diffractometer data from the remote console is a tricky process, so the user is advised to practice at least once before trying a real run.

An example of an Output Processing Mode subprogram deck is given; the necessary system control cards are also shown. In this example, the output is not saved on magnetic tape. If this is wanted, a DATA TAPE EXTRA system control card would have to be placed between the $EXECUTE and the AUTOMATIC DIFFRACTOMETER cards. Since a SEND PRINT TO CONSOLE card appears, a listing of the final results for each reflection would be sent to the remote console as part of the output. This listing is not as complete as the main listing of results which is printed on the computing center high speed printer. The normal mode is to print only the computing center listing.

Example 3.

```
$ID            DJDTST2,CH25000,1         ┐
$EXECUTE       CRYRM                     │
ERROR DUMP REQUESTED                     ├─  System Cards
AUTOMATIC DIFFRACTOMETER [illegible]     ┘
CELL    265201  164197  265514        91528
CHECK    0   2   0 EVERY 15 REF
DO LP CORRECTION
SEND PRINT TO CONSOLE
WEIGHT CONSTANT = 0.0001
DATA FROM CRYSTAL     A1  FIXED TIME=      60 SEC., 1/COUNTS = .100   ,DELT= 0.0
0γγγγγ2γγγ      Q1γγ6γγγγγ             ┐
J1γγ6γγ454      M1γγ6γγγγγ             │
     .               .                 ├─  Diffractometer Output
     .               .                 │
     .               .                 │
K4γγ2γ3929      N4γγγγγ37γ             │
K2γγ3γ3929      N2γγγγγγ3γ             ┘
END OF DATA
```

Cards Allowed only in the Control Tape Preparation Mode:

MAKE CONTROL TAPE

```
===================SPECIFIED REFS.                         ,SLOW I,FAST I===========
                   H=NNN - NNN K=NNN - NNN L=NNN - NNN
```

Optional Constants:

1. If "SPECIFIED REFS." is punched in columns 20-34, the subprogram is set to expect, immediately following this card, a group of index cards designating which reflections are to be scanned.
2. Values of maximum and minimum indices for limiting the range of certain or all indices. If an index is to be limited, the alphabetic characters,"H=", "K=", or "L=" is punched in col. 20-21, 32-33, or 44-45 respectively; then the minimum value , an integer right justified, is punched in col. 22-24, 34-36, or 46-48 respectively, and the maximum value, also an integer right justified, is punched in col. 28-30, 40-42, or 52-54 respectively. If no limits are entered, the subprogram will generate all indices within the allowed sphere.
3. The fastest and/or slowest moving indices in the generated set may be specified by punching ",SLOW" and/or ",FAST" in col. 55-59 or 62-66 respectively and the appropriate character --"H", "K", or"L" in col. 61 for the slowest moving and/or in col. 68 for the fastest moving index.

Action:

This card causes the subprogram to begin calculating the settings and preparing the control tape.

Note:

1. Except for the case of specified reflections, this card must be the last card in the subprogram deck.
2. If "SPECIFIED REFS" is not punched, the subprogram will automatically go into the reflection generating mode.
3. This card must be present in every Control Tape Preparation Mode subprogram deck.

The Index Card:

```
=HHHH=KKKK=LLLL=================================================================
```

Required Constants:

The indices h, k, and l must be punched in col. 2-5, 7-10, and 12-15 respectively, as integers right justified. All minus signs should immediately precede the first significant digit of the integer.

Action:

This card is used to enter reflections when generating a control tape in the specified reflections mode.

LAUE GROUP

```
================NN, X CENTERED
```

Required Constants:

1. The Laue Group Number defined as shown below must be punched as an integer, right justified, in col. 17-18.

Number	Laue Group
1	$\bar{P}$
2	2/m
3	mmm
4	4/m
5	4/m mm
6	m3
7	m3m
8	$\bar{3}$
9	$\bar{3}$m
10	6/m
11	6/m mm

2. If the unit cell is centered, "CENTERED" should be punched in col. 23-30, and the appropriate chracter--"A", "B", "C", "F", or "I"--to describe the centering should be punched in col. 21.

Note:

Required, except when using specified reflections.

to describe the centering should be punched in col. 11.

Note:
Required, except when using specified reflections.

```
CRYSTAL POSITION   VERTICAL   DELTA PHI =        FOR  *.
================X=======================XXX.XXX=====X======LEFT-HANDED=========
```

Required Constants:

1. The character "A", "B", or "C" denoting which of the crystallographic axes is vertical must be punched in col. 18. This is a "real" axis.
2. A value for Δø in col. 43-48, and the character either "A", "B", or "C" telling which reciprocal axis the Δø pertains to in col. 54, in degrees. This angle is just the ø setting for bringing reflections which lie along the particular reciprocal axis named into the reflecting position. This is determined in lining up the crystal.
3. If "LEFT-HANDED" is punched in col. 61-71, the subprogram will pick the third axis (the one not specified) by assuming a left-handed axial system. The normal mode is a right-handed system.

Note:

1. The real axis which is vertical is taken to be positive in the up direction, and the specified reciprocal axis is taken to be positive in the direction specified by the Δø.
2. This card is required in all Control Tape Preparation Mode decks.

```
SINE(SQ)THETA/LAMBDA(SQ) MAX=      ,MIN =
==============================.XXXXXX======.XXXXXX===============================
```

Required Constants:

Values for maximum and minimum values of $\sin^2\theta/\lambda^2$ in col. 31-36 and col. 43-48 respectively.

Action:

When reflections are generated, only reflections with $\sin^2\theta/\lambda^2$ within the above limits will be generated.

SCAN 2THETA

```
============XXXXXX=DEGREES===NO SLOW POSITIONING=====SPEED X==NO BACKGROUND X===
```

Optional Constants:

1. If a value of the number of degrees to be seanned over is punched in col. 13-18, and the characters "DEGREES" is punched in col. 20-26, the subprogram will assume a constant scan range of the value specified. This option should not be used if a SCAN RANGE card is entered.
2. If "NO SLOW POSITIONING" is punched in col. 30-48, the subprogram will delete the slow positioning control words which it normally generates when changing an angle. The normal mode is to "slew" to the angle, and then to "slow position" to it.
3. If "SPEED" is punched in col. 54-58, and an integer for denoting the speed is punched in col. 60, the scanning speed is changed to the value designated. If nothing is punched, a speed of will be assumed. The code for designating speed is:

1	1/4	degree	per	minute
2	1/2	"	"	"
3	1	"	"	"
4	2	"	"	"
5	4	"	"	"

4. If "NO BACKGROUND" is punched in col. 63-75, and either "1" or "2" is punched in col. 77, the subprogram will not generate the code words to count either the first or the second background. The normal mode is to count the first background, scan over the reflection, then count the second background.

Action:

The subprogram is set to do a 2θ scan.

SCAN OMEGA

```
===========XXXXXX DEGREES===NO SLOW POSITIONING=====SPEED X==NO BACKGROUND X===
```

This card is the same as the SCAN 2THETA card except the subprogram is set to do an omega scan.

```
SCAN RANGE        DEG. AT 2TH=        AND       DEG. AT 2TH=
============XXXXXX============2θ2θ2θ======XXXXXX============2θ2θ2θ==============
```

Required Constants:

Values of the number of degrees to be scanned over at two different values of 2θ, all angles in degrees. The first scan width is punched in col. 13-18, with its 2θ value in col. 31-36; the second scan width is punched in col. 43-48, with its 2θ value in col. 61-66.

Action:

The scan range will be calculated as a function of 2θ by a first degree polynomial which passes through the two points specified on the SCAN RANGE card.

Note:

1. If the scan range is to be a constant for all reflections, this constant may be specified on the SCAN 2THETA or SCAN OMEGA card.
2. If a SCAN RANGE card is used, the constant range may not be specified on a SCAN 2THETA or SCAN OMEGA card if either of these is also entered.

Cards Allowed in Either Mode:

```
CELL
======AAAAAAA=BBBBBBB=CCCCCCC=AAAAAABBBBBBCCCCCC================================
```

For a format description see Chapter 4, page 4-20.

Note:
This card is required in all Automatic Diffractometer Calculations Subprogram Decks.

```
CELDEV
======XXXXXXX=XXXXXXX=XXXXXXX=XXXXXXYYYYYYXXXXXX================================
```

For a format description see Chapter 4, page 4-21.

Note:
The only use for this card is to put the standard deviations of the cell parameters on the output magnetic tape.

```
WAVE LENGTH
===========XXXXXX===============================================================
```

For a format description see Chapter 4, page 4-29.

Note:
If no WAVE LENGTH card is entered, the subprogram will assume copper radiation.

```
END OF DATA
================================================================================
```

Use:
This card is used in the Control Tape Preparation Mode to notify the subprogram that all the index cards for specified reflections have entered. It is entered in the Output Processing Mode after all the diffractometer output has been read into the console.

the subprogram that all the index cards for specified reflections have entered. It is entered in the Output Processing Mode after all the diffractometer output has been read into the console.

```
CHECK           EVERY    REF
======HHHKKKLLL=======NN========================================================
```

Required Constants:

1. The indices h, k, and l of a check reflection must be punched in col. 7-9, 10-12, and 13-15 respectively, as integers right justified. All minus signs should immediately precede the first significant digit of the integer.
2. An integer, right justified in col. 23-24 for designating the number of reflections to be allowed between check reflections.

Action:

Control Tape Preparation Mode:

The check reflection will be scanned first, and then every 3rd reflection for 4 check reflections, and thereafter the number of reflections between check reflections will be as specified on the CHECK card.

Output Processing Mode:

This card notifies the subprogram that there is a check reflection in the diffractometer output it is to process. The first 5 values for the check reflection will be averaged and printed together with an approximate standard deviation. All check reflections which follow will be compared with this average value and if it deviates from it by more than 3 standard deviations, it will be marked in the printed output.

Note:

1. The check reflection will be put on the output DATA TAPE EXTRA from the Output Processing Mode only once.
2. Care should be taken in selecting a check reflection, with due attention being given to the size of the reflection, and to the motion of the diffractometer arcs which will be necessary for counting this reflection.

Cards Allowed only in the Output Processing Mode:

ADD TAPE OUTPUT TO DATA TAPE EXTRA
==

Action:

The output presently on the DATA TAPE EXTRA is skipped over and the present tape output is written immediately afterwards.

Note:

1. A DATA TAPE EXTRA must have been loaded prior to the call to the Automatic Diffractometer subprogram.
2. This option is designed to allow the user to read in his automatic diffractometer output in several batches, and have the magnetic tape output accumulated on a magnetic tape for later use. If this is being done, this card should not be included with the first batch of data, but should be included with the second and subsequent batches.

DO ABSORPTION CORRECTION
==

Action:

The subprogram is set to apply an absorption correction factor. The branch to the subroutine to calculate the absorption factor is set.

Note:

To use this option, the user must overlay the subprogram with an octal subroutine for calculation of the absorption correction factor. See the previous discussion.

```
DO LP CORRECTION
================================================================================
```

Action:

The subprogram is set to calculate and apply Lorentz and polarization correction factors to all reflections.

```
DATA FROM CRYSTAL        FIXED TIME=       SEC., I/COUNTS =       ,DELT=
==================NNNNNN============ssssss==================xxxxxx======kkkkkk==
```

Required Constants:

1. A data group identifier in col. 19-24, any number or letter.
2. A time in seconds for the counting time for fixed time counts in col. 37-42.
3. A factor relating net counts to intensity of reflection, in col. 61-66 (see previous discussion).
4. A lag constant for use in computing the intensity of reflection from the counter data in col. 73-78.

Action:

This card signals the subprogram that a group of diffractometer output data is about to enter, and gives it the necessary constants for processing this data.

Note:

This card must immediately be followed by diffractometer output in the format discussed previously; no other control cards are permitted in between. This card is also used to separate different groups of diffractometer output.

```
SEND PRINT TO CONSOLE
================================================================================
```

Action:
This card directs the subprogram to send a reflection by reflection listing of the results of the output processing to the remote console from which the job was submitted. The normal mode is not to send such a listing.

```
WEIGHT CONSTANT =
==================xxxxxx========================================================
```

Required Constant:
A weight constant in col. 19-24.

Action:
A weight constant for use in computing the standard deviation of an average intensity is entered into the subprogram. This constant is assumed to be 0.0 if this card does not appear in the input deck.

Chapter 10

Unit Cell Least Squares

Programming Contribution: G. N. Reeke

Note: The following is taken from a more complete write-up by G. N. Reeke.

Introduction:

This subprogram calculates unit cell parameters and their standard deviations. It is called by the UNIT CELL LEAST SQUARES system control card. It calculates that cell which gives the best least squares fit to a number of observed Bragg angles. The function minimized is

$$\sum_{n=1}^{N} w_n(\sin^2\theta_{obs} - \sin^2\theta_{calc})^2(\frac{4}{\lambda_n^2})^2 \quad .$$

If desired, systematic corrections for eccentricity and/or absorption may be included, and coefficients for these corrections included in the least squares calculation. Running time should not exceed a few seconds.

The program includes provision for operation from the Chemistry Department remote console. In this case, input is exactly as described later for punched card operation, each card being represented by one line of type at the console.

Any number of reflections may be processed up to a maximum which depends on how much memory space has been reserved by other CRYRM subprograms. This maximum is printed at the beginning of each run; it will generally exceed 3200 reflections.

Wave Lengths:

A large number of wavelengths are stored permanently in the program, and may be referenced by name. If necessary, up to ten additional wavelengths may be supplied by the user. The wavelengths stored are the alpha$_1$, alpha$_2$, and beta radiation ^{1} and the average radiation ^{2} for each of the following target elements: Chromium, iron, cobalt, copper, molybdenum, nickel, silver, zinc.

Weights:
The weighting function referred to here is $\sqrt{w_n}$. Reflections may be weighted according to any one of four functions:

a.	1	(All reflections weighted equally)
b.	E	(External weight supplied by user)
c.	$1/\sin 2\theta$	(Recommended)
d.	$E/\sin 2\theta$	(Combination of b. and c.)

Input:

Input to the program consists of a set of control cards to define the calculations wanted, and a set of data cards. It is assumed that most users will have used the CIT Precision Weissenberg camera, so provision is made to perform the preliminary calculations to convert the measured linear positions of the spots to Bragg angles by the Straumanis film calibration method. If desired, the Bragg angles may be used directly. In either case, Miller indices and wavelengths must be specified for each reflection; a weight may be given.

The Subprogram Deck:

Cards recognized by the program are of three types: Control Cards, Data Cards, and Special Cards. All Control Cards for a given calculation must come before any Data Cards. Once a single Data Card has been read, no further Control Cards may be entered until the Special Card "END OF DATA" has caused a calculation to be performed. Following this, Control Cards

Further Control Cards may be entered until the Special Card "END OF DATA" has caused a calculation to be performed. Following this, Control Cards

for other calculations using the same data may be entered. A set of such Control Cards is terminated, and a calculation is initiated, by the Special Card "RECYCLE". Following any END OF DATA or RECYCLE card, use of the current data may be terminated, and new data entered, by use of a set of Control Cards and Data Cards exactly as when the program was first called.

As an example, the deck for a single calculation would be as follows:

```
$ID             GAR,CH98092                          ┐
$EXECUTE        CRYRM                                │
NAME            5-ETHYL-6-METHYLURACIL       REEKE   ├─   System Control Cards
UNIT  CELL LEAST SQUARES                             ┘
Control Cards                         ┐
  :                                   │
Data Cards                            ├─   The Subprogram Deck
  :                                   │
END OF DATA                           ┘
$ENDJOB
```

Output:

Output from the subprogram is in five sections. First, all input cards are listed, with error messages if input errors are detected. Many of these errors prevent the program from proceeding. If input is from the remote console, only cards causing errors will be listed. Second, the matrices generated during the calculation may be printed if desired. Third, a list of Bragg angles, weight, wavelength, and certain comments. This listing is omitted at the console. Fourth, the unit cell parameters and standard deviations are printed, together with a note of thanks to the programmer. Finally, CELL and CELDEV cards are punched in the correct format for use by other CRYRM subprograms.

Error Detection:

The subprogram examines all reflections in an effort to detect bad data. This examination occurs during the processing of comparator data, and again at final output time. Two levels of discrepancy are reported, to aid the user in locating input errors.

For each reflection, the value of either film calibration point determined by that reflection alone is compared with the average value of that calibration point. A difference greater than 0.25 mm. causes an asterisk to be printed in the output for that reflection. A difference greater than some larger value which may be set by the user causes the reflection to be omitted from the least squares calculation, and the message "NO LSQ COMP" to be printed. This difference is normally taken as 0.45 mm., so that vernier-reading errors of one whole millimeter can be detected by the 0.5 mm. error they cause in the calibration point.

When the list of reflections is printed, θ(obs) and θ(calc) are compared. A difference greater than 0.2 degree causes an asterisk to be printed. A difference greater than some larger value which may be set by the user causes the message "REJECT" to be printed, and the entire calculation is then automatically repeated with all such reflections omitted. The message then becomes "NO LSQ CALC". Such automatic recycles occur until no further reflections are omitted. If no constant for comparison is specified, it is taken as 0.5 degree.

Indexing:

The program provides for changing the indexing of reflections which have been entered. Several such changes may be made, each followed by a unit cell calculation. This may be of particular benefit to persons with triclinic structures, in that several cells may be tried, and the most convenient one chosen. The change is specified by a 3 x 3 matrix of coefficients.

Once entered, an index change remains in effect until superceded by another index change or by a RESET STANDARD OPTIONS card. Therefore, the same change will be repeated if recycles are called for. Internal recycles (see Error Detection), however, do not repeat the transformation. Successive transformations operate on the current indices, not the original ones.

Partial Data:

It is not essential that the data be sufficient to completely determine the cell. The program will automatically determine as many parameters as possible from the given data. For example, if h01 data are given for a monoclinic crystal, a, c, and beta will be calculated.

[1] International Tables for X-ray Crystallography

[2] Lipson and Cochran, "The Determination of Crystal Structures," G. Bell & Sons, Ltd., London, 1957, p. 314.

Unit Cell Least Squares Control Cards:

These cards, used to establish various program options, may be used in any order whatever. The "Standard Option" assumed in the absence of the card is given in the description of each card. Control Cards cause switches to be set which then remain in effect for all succeeding calculations until superceded by the RESET STANDARD OPTIONS card. It is generally not possible to supercede one control card by another of the same type without first resetting the Standard Options.

ORTHORHOMBIC===

CUBIC

TETRAGONAL

HEXAGONAL

RHOMBOHEDRAL

MONOCLINIC

TRICLINIC

Required Field:
Crystal system name beginning in col. 1.

Action:
The crystal system is entered into the subprogram. All systems are handled separately, except rhombohedral, which is reindexed to a hexagonal basis according to the formula:

$$H = h-k$$
$$K = k-l$$
$$L = h+k+l$$

The parameters for both hexagonal and rhombohedral cells are given.

Standard Option:
Triclinic.

WEIGHTING FUNCTION 1
WEIGHTING FUNCTION E
WEIGHTING FUNCTION 1/SIN2THETA
WEIGHTING FUNCTION E/SIN2THETA

Action:
The weighting function indicated is used.

Standard Option:
E/SIN2THETA.

Note:
The four weighting functions may be arranged in a hierarchy such that any one may be replaced by one of higher (but not of equal) rank without resetting Standard Options. The hierarchy is:

$1 > 1/\sin 2\theta = E > E/\sin 2\theta$.

OMIT ABSORPTION CORRECTION

OMIT ECCENTRICITY CORRECTION

Action:
The designated correction is omitted from the calculation.

Standard Option:
Both corrections are included.

DO SYSTEMATIC COMPARATOR CORRECTION

Action:
If comparator data are used, they are corrected according to the calibration chart posted over the large comparator in Room ~~70A, Grellin~~ 044 Noyes Laboratory of Chemistry, Caltech.

Standard Option:
No correction is applied.

NORMAL MATRIX PRINT OUT

```
================================================================================
```

INVERSE MATRIX PRINT OUT

```
================================================================================
```

Action:
The indicated matrix(es) is(are) printed.

Standard Option:
Neither matrix is printedl.

WAVE LENGTH

```
============NNNNNN======LLLLLLLLLLLL============================================
```

Required Constants:

1. A wavelength name in col. 13-18, any combination of characters and blanks not already used in the permanent list.
2. A wavelength in Angstroms in col. 25-36.

Action:
The wavelength is entered into the table of wavelengths stored in the program, and remains there for the duration of the run.

Note:
This WAVE LENGTH card is different from the standard CRYRM one.

TRANSFORM INDICES

```
==================AAAAAABBBBBBCCCCCCDDDDDDEEEEEEFFFFFFGGGGGGHHHHHHIIIIII========
```

Required Constants:
The coefficients A,···,I in col. 19-24,25-30,31-36,37-42,43-48,49-54, 55-60,61-66, and 67-72. If decimal points are not punched, the coefficients are assumed to be right justified integers.

Action:
The coefficients are stored for use in transforming the indices

Action.

The coefficients are stored for use in transforming the indices

according to:

$$H = Ah + Bk + Cl$$
$$K = Dh + Ek + Fl$$
$$L = Gh + Hk + Il$$

The user is cautioned that fractional coefficients must be used with care to assure that the resulting indices are integers.

SET ERROR LIMIT, BAD THETAS

```
==============================LLLLLL======CCCCCC================================
```

Required Constants:

1. A limit in degrees in col. 31-36.
2 A limit in mm. in col. 43-48.

Action:

The limits are entered. They remain in effect for the duration of the run unless changed by another SET ERROR LIMIT card.

Standard Option:

Limits of 0.5 degrees and 0.9mm.

RESET STANDARD OPTIONS

```
================================================================================
```

Action:

The effect of all previous control cards is negated, with certain exceptions noted above.

COMMENT

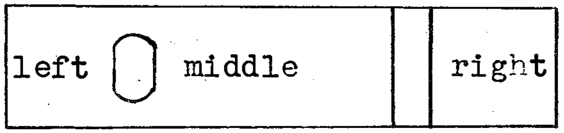
```
=======ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
```

Action:

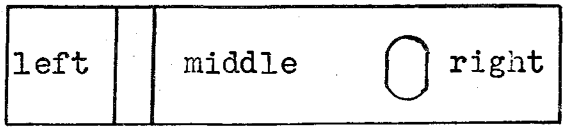
The card is listed only. Anything in col. 8-80 may be placed in the listing by this card.

Unit Cell Least Squares Data Cards:

Data cards are of two types, depending upon whether Bragg angle or comparator data are to be entered. The subprogram distinguishes the two types by their format. Comparator data are entered as one or more of three readings: left, middle, or right, as illustrated below. It is physically impossible for a given spot to occur in both the left and right regions. Cards containing such data cause the job to be terminated. When read, the film may be in one of two possible orientations. The "left handed" film is assumed, and the RIGHT HANDED COMPARATOR DATA card must be used if the right handed orientation is used.



Right Handed Film



Left Handed Film

Data from several films may be entered into a single calculation. If comparator data are being used, data from each film other than the last are followed by the Special Card "END FILM" which causes the program to process that block of data, calculating and storing the Bragg angles. If direct angular data are used, there is no need to distinguish films. It is also possible to use mixed data. Films of the two types may be mixed in any sequence. Left and Right handed films may be mixed. Within one film, data may be mixed provided that all angular data must precede any comparator data.

```
==HHH==KKK==LLL===LEFTLEFTLEFT======MIDDLEMIDDLE======RIGHTRIGHTRIEEEEEENNNNNN==
==HHH==KKK==LLL=========THETATHETATH==============================EEEEEENNNNNN==
```

Required Constants:

1. The indices, h, k, and l, as right justified integers in col. 3-5, col. 8-10, and col. 13-15 respectively.
2. Comparator Data: Values of left, middle, and/or right readings in col. 19-30, col. 37-48, and col. 55-66 respectively.
   Theta Data: A value of the Bragg angle in col. 25-36.
3. An external weight for this reflection in col. 67-72. If this is left blank, a weight of 1.0 is assumed.
4. A wavelength identifier in col. 73-78. For radiations stored permanently in the program, only col. 73-75 are used; and the identifier consists of two letters--the chemical symbol of the target element (CR, FE, CO, CU, MO, NI, AG, ZN)--followed by one of the symbols 1, 2, A, or B, where

   1 indicates alpha(1) radiation
   2 indicates alpha(2) radiation
   A indicates alpha average radiation
   B indicates beta radiation.

   For example, molybdenum beta radiation is named MOB. All six positions are available for wavelengths entered by the user. A blank wavelength field is interpreted as copper average radiation.

Action:

The data on the card is stored in tables until a calculation is requested by one of the Special Cards.

Unit Cell Least Squares Special Cards:

END FILM

Action:

Preceding data are lumped together in a Straumanis calculation of Bragg angles. If angular data are entered, the card has no effect.

END OF DATA

Action:

This card produces the same action as the END FILM card, followed by an actual unit cell calculation.

Warning:

After the last film, do not use both END FILM and END OF DATA cards.

RECYCLE

Action:

A new unit cell calculation is performed using data already entered for a previous calculation. It may be used, for example, to calculate cell parameters with and without the absorption correction.

RIGHT HANDED COMPARATOR DATA

LEFT HANDED COMPARATOR DATA

Action:

The subprogram is informed that the <u>following</u> data are of the type specified. With no entry, left handed is assumed.

The subprogram is informed that the following data are of the type specified. With no entry, left handed is assumed.

Chapter 11

Publication List

Programming Contribution: D. Duchamp

Description:

This is a short subprogram which will print a structure-factor list in a form which is suitable for publication. The subprogram requires a DATA TAPE with the values of the calculated structure-factors which are to be printed. It will print out the structure-factors in the same order as they appear on the DATA TAPE. The listing which is produced may be cut up and pasted together to produce a large table, which can then be reduced photographically. In this manner as many as 2,000 reflections or more may be made to fit on an 8 1/2 x 11 page.

The user must enter into the subprogram a FORM card giving the information for setting up the list. This is the only control card for the Publication List subprogram. The subprogram is called in by the PUBLICATION LIST system control card. If a list is requested from the remote console, a list is sent both to the main printer and to the remote console.

In order to make sure that the output from the printer is suitable for publication, the user should confer with the computer operator, asking him to reverse the paper in the printer, and to put in a dark ribbon.

The Output Format:

The output is arranged in groups--each with a heading followed by a list. A new group is started whenever an index other than the fastest moving changes. The reflections are printed in the order in which they appear on the DATA TAPE.

Example 1, Centric:

```
    2 10 L

 0   60  -56
 1   43   40
 2    3    4
```

Columns are fastest moving index (in this example, L), cF_{obs}, cF_{calc}; where c is the factor punched under "PRINT F TIMES" on the FORM card.

Example 2, Acentric:

```
    H 10 11

 1   29   31 320
 2  130  128  53
 4   58   60  20
```

Columns are fastest moving index (in this example, H), cF_{obs}, $c|F_{calc}|$, and $\propto$; where c is the same as above, and $\propto$ is the phase angle in degrees for F_{calc}.

Zero weight reflections are indicated by an asterisk after F_{obs}. Less-than reflections are indicated by making F_{obs} negative.

The position of the alphabetic character in the group heading may be shifted one column in either direction by an overlay to either location HKLET or location HKLET + 1.

Required Control Card:

```
FORM-FAST  ,        , PRINT F TIMES          SCALE FACTOR =
==========I=ACENTRIC================xxxxxx==================ssssss==============
             CENTRIC
```

Required Constants:

1. An identifier in col. 11, punched as an alphabetic character, either "H", "K", or "L", denoting which index is to be treated as the fastest moving.
2. Either "ACENTRIC" or "CENTRIC " in col. 13-20.
3. A numerical factor in col. 37-42. This factor multiplies both F_{obs} and F_{calc}, and the integral part of the result is printed.
4. A scale factor for F_{obs} in col. 61-66. This factor is used to put the F_{obs} which is on the DATA TAPE on the same scale as F_{calc}.

Action:

The reflections are listed according to the format specified.

Chapter 12

Temperature Ellipsoids

Programming Contribution: Mrs. B. Stroll

Description:

This subprogram calculates principal axes of vibration of thermal ellipsoids from anisotropic thermal parameters according to the method of Jurg Waser, Acta Cryst. 8, 731 (1955). The TEMPERATURE ELLIPSOIDS system control card is used to call this subprogram. The lengths of the principal axes in "B" units, and their direction cosines with respect to the crystallographic cell axes are printed. Direction cosines with respect to any orthogonal set of axes may be obtained by use of the SPECIAL PLANE option.

The SPECIAL PLANE Option:

A plane may be specified in either of two ways: 1. The direction cosines (with respect to crystallographic cell axes) of the plane normal, and a vector (with crystallographic cell fractional coordinate components) specifying the horizontal direction in the plane are entered, or 2. the plane is chosen with either the X, Y, or Z crystallographic cell fractional coordinate held constant in the plane, and either the A, B, or C crystallographic cell axis in the horizontal direction in the plane.

From the information mentioned above, an orthonormal system of axes is defined. In the output, the vertical direction in the plane becomes the first axis of the orthonormal system; the horizontal direction in the plane becomes the second, and the plane normal becomes the third. For each ellipsoid which has been calculated, the direction cosines with respect to these orthogonal axes are calculated and printed. Also the coordinates in the orthogonal system of the ends of the major axes are printed.

Subprogram Input:

Input of anisotropic thermal parameters to the subprogram may be either by ANISO cards or from the DATA TAPE. If no ANISO cards are entered in the subprogram deck, the subprogram will take ANISO cards from a DATA TAPE which must have been loaded before calling the subprogram. If no ANISO cards have been entered, and no DATA TAPE has been previously loaded, an error results. No special card telling the subprogram to get the parameters from tape is allowed. If ANISO cards in the subprogram deck are used to read in anisotropic thermal parameters, a DATA TAPE is not required.

The Subprogram Deck:

The subprogram deck should be arranged in the following order:

a. A CELL card. (This card may be omitted if the input of anisotropic parameters is from the DATA TAPE, and if it is desired to use the cell parameters presently on the DATA TAPE.)

b. ANISO cards...as many as desired. (This section is omitted if input of anisotropic parameters is from the DATA TAPE.)

c. SPECIAL PLANE cards...if desired. (As many SPECIAL PLANE cards may be entered as the user desires. No SPECIAL PLANE cards may be entered until all the ANISO cards have been entered. Each SPECIAL PLANE card applies to all the atoms whose temperature ellipsoids are being calculated.)

Example 1:

```
SPECIAL PLANE X CONSTANT, B HORIZONTAL
```

The above subprogram deck (one card) will cause the subprogram to look for a DATA TAPE, invert all the anisotropic thermal parameters which occur on this DATA TAPE, and print the information relative to the orthonormal set of axes defined from the information on the above SPECIAL PLANE card. Since a CELL card is not entered, the cell parameters on the DATA TAPE will be used.

Example 2:

```
CELL  10.356  8.562   3.490
ANISO    1   .008532   .013455   .045675  -.000056   .000568   .003490
  :      :      :         :         :         :         :         :
ANISO    9   .006945   .029872   .069234   .098735  --.000985   .000009
SPECIAL PLANE   .5000      .5000      .705        VECTOR - -.030 -.080 .018
SPECIAL PLANE Y CONSTANT, A HORIZONTAL
SPECIAL PLANE Z CONSTANT, A HORIZONTAL
```

The above example subprogram deck will cause the subprogram to calculate the thermal ellipsoids for all the atoms whose ANISO card is present, and print the information relative to the three SPECIAL PLANEs specified. The SPECIAL PLANE cards apply to all the ANISO cards.

Temperature Ellipsoids Subprogram Control Cards:

```
CELL
======AAAAAAA=BBBBBBB=CCCCCCC=AAAAAABBBBBBCCCCCC==============================
```

For a format description see Chapter 4, page 4-20.

```
ANISO
======NNNN==SB11B11B11SB22B22B22SB33B33B33SB12B12B12SB13B13B13SB23B23B23========
```

For a format description see Chapter 6, page 6-20. (Col. 73-80 are ignored by this subprogram.)

```
SPECIAL PLANE                                   VECTOR -
==============XXXXXXXXX===YYYYYYYYY===ZZZZZZZZZ============XXXXXXYYYYYYZZZZZZ==
```

```
SPECIAL PLANE  CONSTANT,  HORIZONTAL
==============?===========?====================================================
```

Required Constants:

FORM 1:

1. Direction cosines with respect to the real cell axes--with respect to the a, b, and c axes in col. 16-24, 28-36, and 40-48 respectively.
2. Fractional components with respect to the cell edges of a vector specifying the horizontal direction in the plane--x, y, and z in col. 61-66, 67-72, and 73-78 respectively. (This vector need not be exactly in the plane.)

FORM 2:

1. Either X, Y, or Z in col. 15, depending upon which fractional coordinate is to be held constant in the plane.
2. Either A, B, or C in col. 27, depending upon which axis is to be the horizontal direction in the plane. ?

Action:

1. From the information specified, an orthogonal coordinate system with unit basis vectors is defined such that the first basis vector is in the plane in the vertical direction, the second is in the plane in the horizontal direction, and the third is normal to the plane.
2. For each ellipsoid which has been inverted, the direction cosines of the major axes with respect to these orthogonal axes are calculated and printed, together with coordinates in the orthogonal system of the ends of the major axes, the center of the atom being considered as the origin, and the lengths being expressed in B units.

Chapter 13

Statistical Phasing

Programming Contribution: T. A. Beineke.

Note: The following description is taken from notes by T. A. Beineke.

Introduction:

The Statistical Phasing Subprogram calculates unitary structure-factors, U, normalized unitary structure-factors, E[1], and the distribution of E, and as an option, lists the normalized structure-factors in the Sayer's relationships or Σ_2[2,3].

Certain classes of reflections, depending on the space group, require that U's or E's be divided by $\sqrt{2}$, $\sqrt{3}$, or 2. This is done automatically for centric space groups in the monoclinic and triclinic systems; for others, it must be done by hand.[4] (Eventually this will be done for all space groups, but the present version treats only the monoclinic and triclinic systems.)

This subprogram is called into memory by the STATISTICAL PHASING System Control Card. It requires reflection input from a DATA TAPE prepared by the Data Tape Preparation and Editing Subprogram. A scaling

[1] M.M. Woolfson, "Direct Methods in Crystallography," Oxford, 1961, p. 13.

[2] D. Sayer, Acta Cryst, 5, 60 (1952).

[3] H. Hauptman and J. Karle, "Solution of the Phase Problem, I. The Centrosymmetric Crystal," A.C.A. Monograph No. 3, Wilmington: the Letter Shop.

[4] In cases other than monoclinic or triclinic, the distribution criteria for the E's will naturally be slightly in error because certain classes of reflections have not been divided by the appropriate number.

factor and an overall isotropic temperature factor are input parameters. The isotropic temperature factor may be on the DATA TAPE or may be entered by a card.

The Output:

First is a list of h,k,l, F_{obs}, U, E, Epsilon (i.e. the expectation value of U^2), and Weight. The purpose of including the Weight column is to indicate the reflections which are "less than" reflections. Following this list, is a comparison of the distribution of E with a theoretical two-tailed Gaussian. (E should be normally distributed with expectation value of unity and standard deviation of unity.) On the following page are more distribution criteria, the expectation value of$_2$ $|E|$, the expectation value of $|E^2 - 1|$, and the expectation value of $E^2 - 1$ compared with theoretical values for centric and acentric crystals. Because of the limited number of characters available for print out, the usual expectation brackets $\langle\,\rangle$ are represented by parentheses ( ) and the absolute value sign || is represented by / /.

If the LIST SAYER'S RELATIONSHIPS option is used (At present, this option may be used only on centric monoclinic and triclinic crystals, otherwise an error message results.), a listing of the reflections for which E is larger than E(min), an input parameter, follows. For each reflection, h,k, and l are listed followed by E. Following this is a complete list of pairs of reflections which fulfill the Sayer's relationship with the given reflection. These pairs are listed three across a page . For each pair, both sets of indices are printed within the parentheses separated by /. Immediately following the last parenthesis is a signed product of the E's followed by the probability that this particular relationship is true. The sign printed before the product of the E's indicates the relationship between the sign of that particular product and the sign of the E at the left. At the end of the row is a number (usually 1.0) which represents the maximum probability that the sign of the E listed at the left will be equal to the sign of the sum of all the products of the E's on the right. At the far right is printed the number of these pairs which interact with

At the far right is printed the number of these pairs which interact with

this reflection.

Example:

```
7 1 7   2.64  ( -1 9 6/-8 8 1) -4.57 .93   ( 0 3 1/-7 2 6) 4.42 .92    .998  2
```

In the above, the reflection 7 1 7 has an E of 2.64 and is related to 2 pairs of reflections, (-1 9 6/-8 8 1) and (0 3 1/-7 2 6). The probability that the sign of 7 1 7 is opposite to the sign of the product of the E's of -1 9 6 and -8 8 1 is .93. The probability that the sign of the product, 4.42, of the E's of 0 3 1 and -7 2 6 is the same as the sign of 7 1 7 is 0.92. The maximum probability (if all relationships hold) that the sign of 7 1 7 is the same as the sum of the signed products in this list is 0.998. We are told that 2 products interact with 7 1 7.

The Subprogram Deck:

The subprogram deck is composed of a number of option and information cards followed by a DATA ON DATA TAPE card. The DATA ON DATA TAPE card is always last in the subprogram deck. A NUMBERS OF ATOMS card is required. If the reflections on the DATA TAPE have already been put on an absolute scale by applying a Wilson scale factor when the DATA TAPE was prepared, no scale factor need be entered; otherwise, one should be entered by a SCALING FACTOR FOR F(OBS) card. The overall isotropic temperature factor may be obtained from the DATA TAPE if the Wilson scale factor was applied when the DATA TAPE was prepared; if this has not been done, or if another temperature factor is preferred, it may be entered on an OVERALL ISOTROPIC TEMPERATURE FACTOR card. In order to obtain a list of the Sayer's relationships, the LIST SAYER'S RELATIONSHIPS card is used. The first listing of E's and U's may be suppressed by the DON'T LIST UNITARY STRUCTURE FACTORS card. If a radiation other than copper was used, a WAVE LENGTH card must be entered. If one wishes to restrict the list of Sayer's relationships, a USE ONLY DATA WITH SINE(SQ)THETA/LAMBDA(SQ) LESS THAN card may be entered.

Hints and Warnings:

1. In calculating unitary structure factors for space groups other than centric monoclinic or triclinic, use space group number 2. If, however, the classes of reflections to be divided by a constant are the same as for one of the centric monoclinic space groups, use the space group number of that corresponding monoclinic space group.

2. The Statistical Phasing subprogram has a dislike for acentric space groups; so, whether or not you believe the space group is centric, tell the program that it is.

3. For listings of Sayer's relationships for triclinic space groups, each triple product is listed under each reflection; however, this is not necessarily true for monoclinic space groups.

Statistical Phasing Subprogram Cards:

```
DATA ON DATA TAPE
================================================================================
```

Action:
This card notifies the subprogram that card input has ended and calculations should be started. It must be the last card in all Statistical Phasing Subprogram Decks.

```
DON'T LIST UNITARY STRUCTURE FACTORS
================================================================================
```

Action:
The first listing of U's and E's is suppressed.

Note:
Normal option is to print U's and E's, however this may not be wanted if the user is mainly interested in Sayer's relationships.

```
LIST SAYER'S RELATIONSHIPS    E(MIN)  =
==========================================xx.xxxx================================
```

Required Constant:
A value for E(MIN) in col. 43-48. When the Sayer's relationships are listed, only reflections having E's greater than E(MIN) will be considered. Blanks (nothing punched) gives a value of 1.5 for E(MIN).

Action:
When calculations are done, Sayer's relationships involving reflections whose E's are greater than E(MIN) will be listed.

NUMBERS OF ATOMS

```
==================NN-fff,NN-ff,NNN-ff,•••                                      ==
```

For a format description see Chapter 5, page 5-13.

Note:

1. This card is required in Statistical Phasing Subprogram Decks.
2. The numbers of atoms in one unit cell should be punched for all space groups, irrespective of centering.

OVERALL ISOTROPIC TEMPERATURE FACTOR, B =

```
==========================================xxxxxx================================
```

Required Constant:
An overall temperature factor in col. 43-48.

Action:
The overall isotropic temperature is entered into the program.

SCALING FACTOR FOR F(OBS) =

```
============================xxxxxx==============================================
```

For a format description see Chapter 7, page 7-11.

USE ONLY DATA WITH SINE(SQ)THETA/LAMBDA(SQ) LESS THAN

```
==========================================================xxxxx=================
```

For format description see Chapter 7, page 7-12.

WAVE LENGTH

```
============xxxxxx==============================================================
```

For a format description see Chapter 4, page 4-29.

Appendix I

Card Input via Magnetic Tape

The CRYRM input deck as described in Chapter 2, is for input using the Caltech setup--the input cards are read into the 7040 computer which loads them on magnetic disk for input to the 7094. If it is desired to run CRYRM by loading the input cards on a magnetic tape by some other computer (for example, an IBM 1401), and using this input tape for input to the 7094; the deck described in Chapter 2 should be changed as follows:

1. The $EXECUTE card should come first.
2. The $ID card should come second.

(The order of the two above cards is reversed from Chapter 2.)

3. The $ENDJOB card is not needed, and if not deleted by the tape loading program, it will cause an error when it reaches CRYRM. Therefore, this card should not be used unless required by the specific tape loading program being used.

The jobs may be stacked. If CRYRM is in control and receives either a a $IBSYS card or a $EXECUTE some system other than CRYRM, it will sign off the present job and return control to the IBSYS monitor. If it receives a $EXECUTE itself card, it will sign off the current job and continue to read the input tape.

Appendix II

CRYBUG, The Debugging Subprogram

Programming Contribution: I. Ambats.

Discussion:

This subprogram provides the user with a very powerful debugging aid when running under CRYRM. Basicly it is designed to allow the user to obtain information about what is actually going on in his program at various times without stopping the calculations. This is usually hard to come by since the user may not sit at the console and run the 7094 computer.

CRYBUG is given control by calling it at some point in the calculations by use of the sequence outlined later. When it is called, it is given a time in seconds and an end location. CRYBUG will begin tracing the program which called it, executing it instruction by instruction. It begins with the instruction immediately following the CALL statement. If it is still tracing when the designated time expires, or if the user references a place in CRYRM or IBSYS, other than a communication region, the run will be terminated with an error message. If CRYBUG reaches the end location without detecting an error, it gives control back to the program being traced.

While CRYBUG is in control, it will print for each reflection, the contents of all registers and the contents of memory locations referenced. (This print out may be suppressed on all except non-successive Branch-type instructions.) CRYBUG may also be asked to dump certain regions of memory whenever a non-successive Branch-type instruction is encountered. Up to ten regions may be specified.

Hints and Warnings:

1. CRYBUG may not be used to trace any subprogram which has instructions at memory locations lower than the end of CRYBUG. (15246 octal in the present version.) Therefore any subprogram to be traced by CRYBUG should be origined at a higher memory location than the end of CRYBUG.

2. One must consider the large amount of printed output which can result if CRYBUG is used for a very large number of instructions (as for example, in loops). The user should carefully figure on this, switching off CRYBUG in places where it is not needed.

3. The execution time for a program under the trace will go up by about a factor of 60 to 100.

Calling Sequence:

```
        SXD     AARD,4
        CALL    CRYBUG(AARD)
```

where AARD is the first location of a sequence of the following format

```
AARD    SSS     N,,**
        PZE     END,,TIME
        PZE     S1,,L1     ┐
        PZE     S2,,L2     │
         :       :   :     ├  N
        PZE     SN,,LN     ┘
```

where

SSS = PZE gives full output
    = MZE gives output only on branch instructions

TIME gives the maximum number of seconds wanted under the trace (Overflow gives error termination of job.)

N gives the number of region dumps specified, limited to ten (specify in octal).

SX gives the Xth region's starting location
LX gives the Xth region's octal length.

END gives the octal location of the instruction at which CRYBUG is to give control back to the program being traced.

When CRYBUG is called as described above, all registers are preserved in the condition they were in before the SXD AARD,4 instruction.

Appendix III

Preparation of an Additional Subprogram

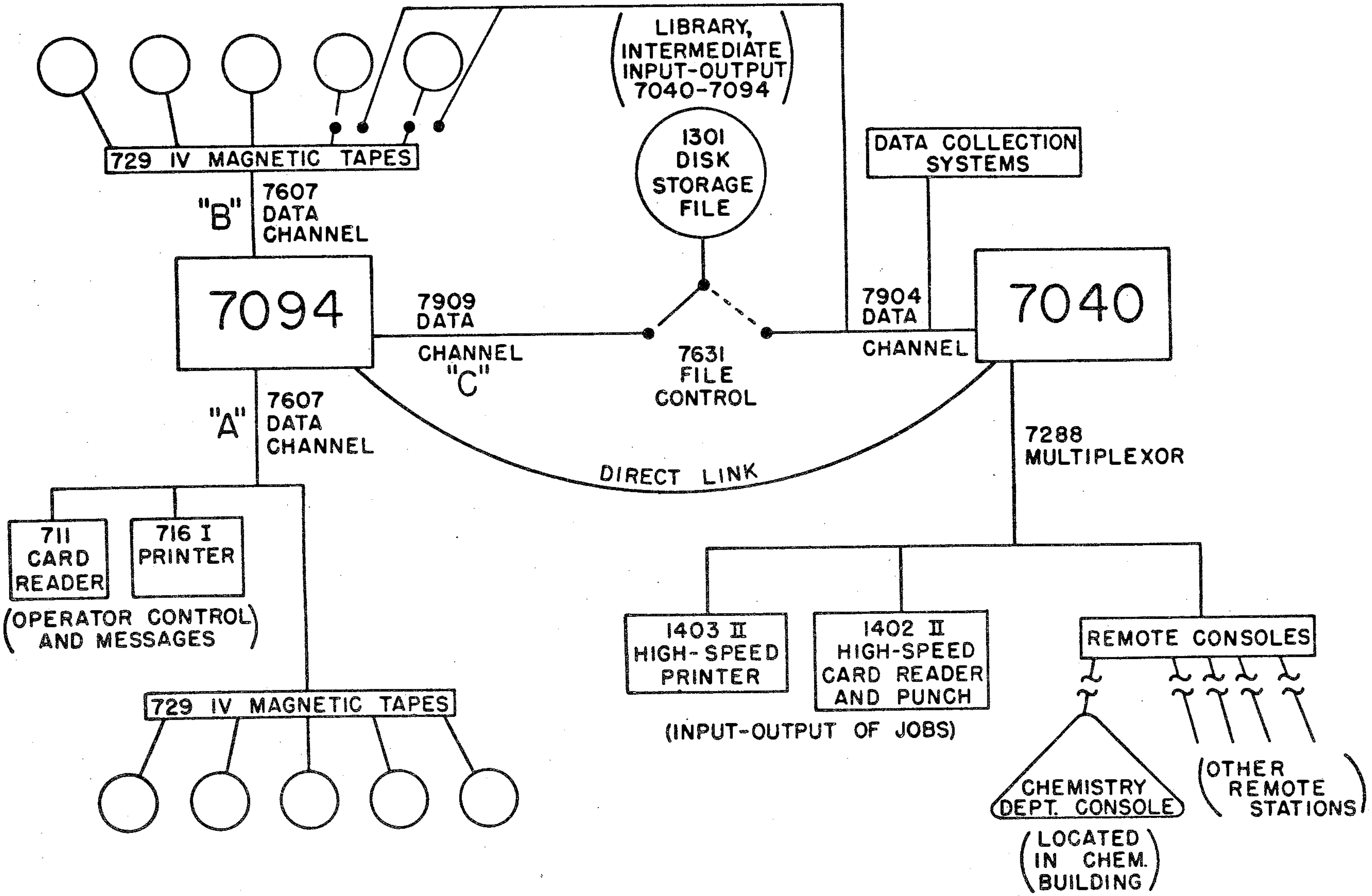
If it is desired to prepare an additional subprogram, either for temporary use or for permanent inclusion in the CRYRM system, the coding should be done in the MAP assembler language and assembled in the absolute mode. (If the routine is short enough the user may wish to translate his program into octal machine language himself; this is also permitted.) The system program should be referenced only through the communication region (see an assembly of the system program). If input and output are to be done, the input - output subroutines in the system program should be used. A number of functional subroutines are also available.

In order to run a subprogram(which is not on the CRYRM system library) under the CRYRM moniter, the user should set up CRYRM running deck as usual, loading a DATA TAPE, etc as required. When it is desired to transfer to the new subprogram, a SYSTEM CHANGES FOLLOW should be inserted, followed by the new subprogram (on absolute mode binary cards and/or OCTAL cards). In addition to entering the new subprogram, these changes should (1) make the location BRANCH in the system communication region equal to zero, (2) place into location PROGIN in the communication region, a transfer instruction to the entry point of the new subprogram, and (3) make location SY2REC in the communication region non-zero. Any cards to be read by the new subprogram should follow these changes. After the SYSTEM CHANGES are read in, the system will transfer to location PROGIN, thus handing over control to the new subprogram. When the new subprogram finishes its calculations, it should return to the system at location PROGRT in the communication region for a successful return, or location NEWJOB for an error termination. If the return is to PROGRT, the system will continue to read input and execute further calculations if more are requested.

A subprogram may be added to the CRYRM system library by the standard IBSYS Editor. If so, tables in the system Action subroutine, which reads the library, should be expanded to include the additional subprogram; and the tables in the system moniter control card scanning routine should be expanded to include a control card which will call in the new subprogram.

Provided the new subprogram does not occupy space in memory which is used by CRYBUG, the CRYRM debugging aid program, this program is available if debugging problems should arise.

Appendix IV



CALTECH COMPUTER SETUP

Appendix V

The DATA TAPE Format

The DATA TAPE has 31-word records numbered 0-30, according to the following format:

Word No.	
	1st Record
0-1	Tape identifier words (taken from col. 19-30 on PREPARE card)
2-11	Special name for page heading (taken from a NAME card)
16	Wavelength (not always present)
17	Cycle letter (in BCD for Least Squares)
18	Cycle number (fixed-point binary for Least Squares)
19-24	a,b,c,α,β,γ --cell parameters (floating-point binary)
25-30	σ_a,σ_b,σ_c,σ_α,σ_β,σ_γ --standard deviations in cell parameters (floating-point binary)
	2nd Record
0-13	Form factor identifiers (BCD--3 characters, 3 zeros--from KIND F)
14-19	S_{11}, •••,S_{23} Coefficients for calculating $\sin^2\theta/\lambda^2$ (floating-point)
20	Overall temperature factor (floating-point binary)

With Atomic Information Deck

	3rd Record and following Records
0-13	BCD card images of the cards of the Atomic Information Deck
	⋮
0-13	BCD card image of the END ATOMIC INFORMATION card.

Without Atomic Information Deck

	3rd Record
0-30	All zeros.

Next records are Data Records, one per reflection, as many as required

0-2	h, k, l (fixed-point binary, right-justified)
3	F_{obs} (floating-point binary)
4	Weight (floating-point binary)
5-7	$2\pi h$, $2\pi k$, $2\pi l$ (floating-point binary)
8	$\sin^2\theta/\lambda^2$ (floating-point binary)
9-22	$f_1,\cdots,f_{14}$ atomic scattering factors for this reflection (floating-point binary)
21-22	A_{calc}(special), B_{calc}(special) (floating-point binary) Note the multiple use of 21-22. (also used for f_{13} and f_{14} if these should ever occur)
23-28	h^2,k^2,l^2,hk,hl,kl (floating-point)
29-30	A_{calc}, B_{calc} (floating-point binary)

Last three Records

0	777777777777 in octal.

End of File

Appendix VI

Selected 7094 Running Times

Subprogram	Calculation Description	Time
Initial Data Processing	Complete 3-pass data reduction from Weissenberg film readings of all layers about 2 axes, 1120 reflections afterwards	4 min. 10 sec.
Initial Data Processing	Weiss. LP correction, 108 reflections	1 sec.
Least Squares	$P2_12_12_1$, full matrix, 13 anisotropic atoms (118 parameters), 11 isotropic atoms (not refined), 1102 reflections	5 min./cycle
Least Squares	$P2_1/c$, full matrix, (21 parameters), 5 isotropic atoms, 189 reflections	10 sec.
Least Squares	C2/c-2 cycles-90 anisotropic atoms (all refined), 38 isotropic atoms (not refined) 11562 reflections (total), Multiple matrix setup: 6-45x45 matrices; 90-6x6 matrices; one 1x1 matrix--total of 811 parameters; Recycle with same inverse.	1 hr. 45min. 18 sec.
Least Squares	$P2_1/n$, full matrix, (185x185), 16 anisotropic and 10 isotropic atoms, 2219 reflections, with tape copy.	17 min. 39sec.
Structure-Factor Calc.	Cubic, Fd3m, 21 isotropic atoms, 1215 independent reflections	2 min. 30sec.

Structure Factor Calc.	$P\bar{1}$, 11 anisotropic and 10 isotropic atoms, 1589 reflections	41 sec.
Fourier	$P\bar{1}$, general plane section, 9.0 A x 9.0 A, 2035 points, 1589 reflections	1 min. 51 sec.
Fourier	Patterson, orthorhombic, 2 projections, 200 reflections/projection	8 sec.
Fourier	$P\bar{1}$, 3 dimensional; 32 sections, 1008 points/ section, 1500 reflections	3 min. 18 sec.
Distances, Angles, and Planes	Complete distances and angles for 13 atoms	2 sec.
Publication List	1200 reflections, acentric	30 sec.

Control Card Index

Control Card Index

Index 2

Index 3

Index 3

The CRYM Crystallographic Computing System (1991)

by David J. Duchamp[1], Richard E. Marsh[2], Jean Westphal and many co-workers

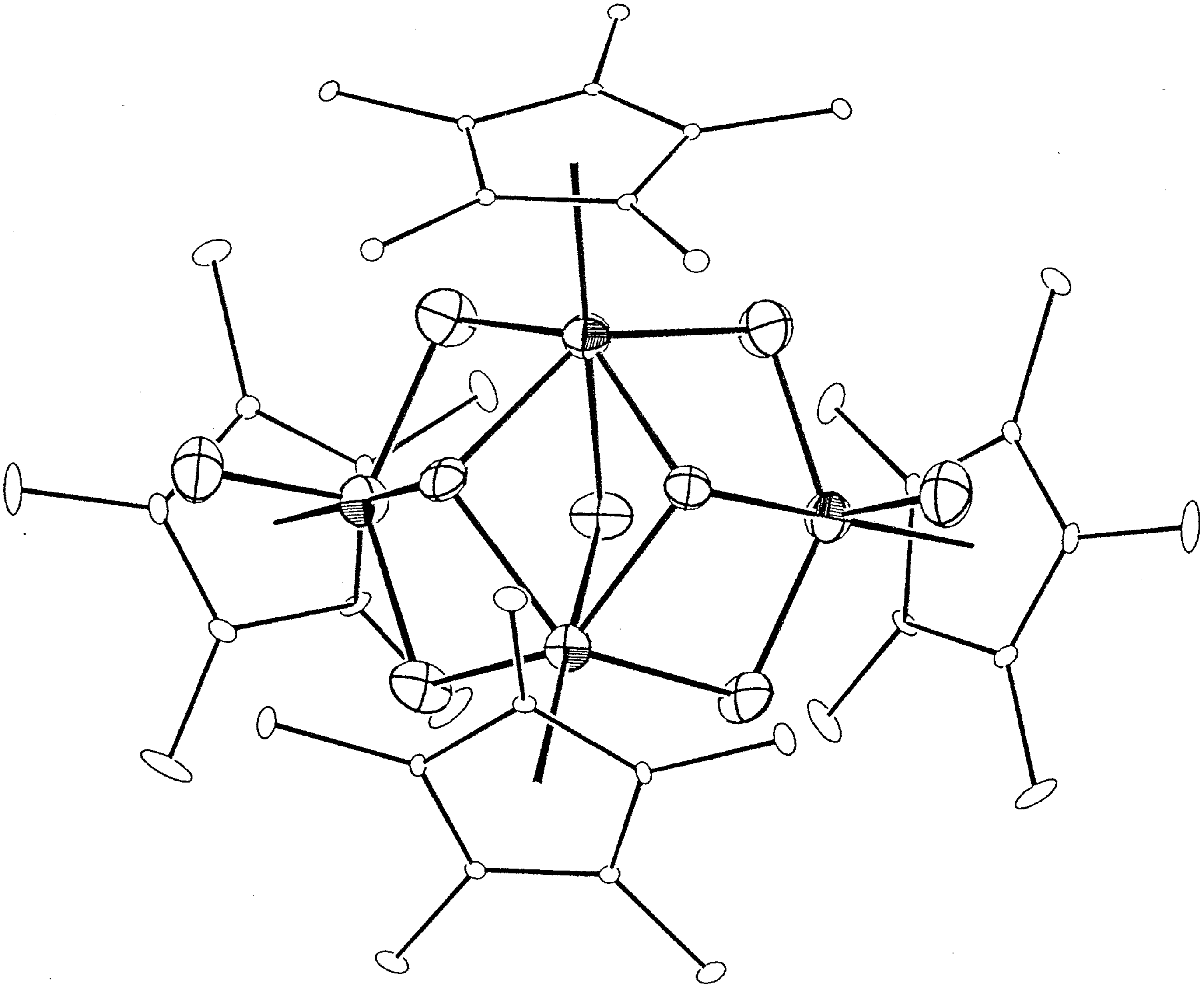
1) 6209 Litchfield Lane Kalamazoo, Michigan 49009, USA. E-mail: djduchamp@mac.com ; 2) Beckman Institute, 139-74 California Institute of Technology Pasadena, California 91125, USA. E-mail: lmh@caltech.edu & rmarsh@caltech.edu

Place holder page. Report-proper starts on following page

A Guide to

X-RAY COMPUTING

at Caltech



The

CRYM

Crystallographic Computing System

by

David J. Duchamp

Richard E. Marsh

Jean Westphal

and

many co-workers

X-ray Crystallography Facility
Beckman Institute and
Division of Chemistry and Chemical Engineering
California Institute of Technology
Pasadena, California 91125

Table of Contents

III)Completing the Structure

IV)Miscellaneous programs

Introduction

XRAY is the command which runs **CRYM** jobs on the *XRAY VAX* computer. There are several different versions of this command which may be used. [The word 'XRAY' is a symbol to the computer; the definition of symbols can be learned by typing the DCL command:
SHOW SYMBOL *name*
where *name* is the symbol to be explained.]

In order to run any **CRYM** job, it is first necessary to create an *input file*. This file will contain the various card–images as described in this **CRYM** booklet, beginning with the program title card (*Structure Factors*, *Least Squares*, *Prepare data tape*, or whatever) and ending with an *End X-ray* card. While this file may be created using a file editor, it is usually more convenient to use the prompting routine called by typing *INPUT*. This routine will prompt you for the information necessary to create the *input file* of card images. [*INPUT* runs the standard input program. If there appears to be an error in this current program, *OLDINPUT* can be used to run a prior version of the input program. *TESTINPUT* utilizes the input program under development; it should not be used in normal circumstances. It is unlikely that there will be a *SPECINPUT* for unusual applications.

The *input file* is defined by a file name, a file type, and a version number (in the form *filnam.typ;j*). The file name may be up to nine characters long, the file type up to three characters, and the version number from 1 to 999. It is convenient to make the file name a compound identifier that remains the same during the entire crystal–structure analysis. The file type should then be related to the first two or three letters of the program you are running—*name.SF*, *name.LSQ*, *name.PRE*, or whatever. This helps you and others more clearly identify an input file in a directory by noting the file type.

Special note. Some file types have been defined internally; it is therefore mandatory to use only these file types when running those particular programs. Here are the important ones: *Prepare* and *Initia* require the file types *PRE* and *INI* respectively; *Taperead* must have the file type *CAD* if data are from the *Enraf-Nonius CAD4* diffractometer, or *TAP* for the $P\bar{1}$ diffractometer. (Suggestions for the specification of file types are given in Table I–1.)

The XRAY command

When you have created your input file and are ready to run it, type *XRAY*. You will then be asked for your compound ID (the file name) and the program code (the file type). [Note: the latest version of *name.typ* will be used, but any other version can be specified by responding to the program code with file type and version number, i.e., *typ;j* instead of just *typ*.] You will then be asked whether you wish to run the job interactively or on one of the batch queues—"fast" (*BATCH*) or "slow" (*BACK*). Jobs which take more than a minute or two (e.g., *Structure Factors*, *Least Squares*, *Wilson*, *Absorption*, *Fourier* and the like) should **not** be run interactively. *BATCH* is the normal priority queue for batch jobs. The other possible priority is *BACK* which may be slower.

You may ask that your batch job be on hold or suspended in the specified queue. If you do so, the program will not run until a *NEXTJOB* command is issued. This command is automatically issued at the end of any **CRYM** job, so if you want one job to start after another has successfully completed, you should ask to *hold* the second job. Both jobs must be in the same batch queue.

The command file will also ask if you would like to add a comment to your *REC* file. The *name.REC* file is a formatted file which includes the input file, the date and time submitted, and a comment, if desired, for each **CRYM** program you run for a particular structure (identified by *name*). Since it is formatted, it may be read, edited, typed, or printed.

Printed output files from any job are available as *name.OUT*. They may be examined on a terminal (*TYPE name.OUT* or *EDIT name.OUT*), or listed on one of the printers (*PRINT name.OUT* or *HPP name.OUT*). The log of any batch job, which includes file assignments and directory information, is available as the file *name.LOG*, and can be read, typed, edited, or printed. If the program does not terminate normally (i.e., a *fatal error*), the log is available as *XRAYSET.LOG* and could provide an explanation of the error. The log of an interactive job is displayed on the screen during execution of the program but is not saved.

A backup file, *name.BKP*, is created by several programs (*Structure Factors*, *Least Squares*, *Wilson Statistics* and *Edit data tape*). This is a binary (unformatted) data file of the type *name.HKL* and contains the latest version of the reflection data. If a program aborts in such a way as to destroy or damage the original data file, *name.HKL* (this can be determined by comparing the *size—DIR/SIZE*—of the *name.HKL* and *name.BKP* files), the backup file can be substituted by typing:

RENAME name.BKP name.HKL

The earlier versions of *name.HKL* should presumably be deleted. On the other hand, if a program has run successfully, the backup file, *name.BKP*, should be deleted. (The *name.HKL* and *name.BKP* files will be the same size, within a block or two.)

When issuing the *XRAY* command, prompts can be avoided by giving all required information in the command line. Note that in this case no default values are supplied:

XRAY P1 P2 P3 P4 P5 (P6)

P1 is the compound id
P2 is the program code extension
P3 is S, F, or I for the desired queue
P4 is N or Y for hold status
P5 is a one word comment (enclose in " " for multiple words or to preserve lowercase)
P6 is needed only for *.CAD Taperead* program; Y or N if data is to be copied from diffractometer computer

For example type:

XRAY JMP3 DIS F N N

to run a *Distan* job in the *BATCH* queue immediately.]

[Just as with *INPUT*, *XRAY* runs the standard program. If there appears to be an error in this current program, *OLDXRAY* can be used to run a prior version. *TESTXRAY* utilizes the program under development; it should not be used in normal circumstances. *SPECXRAY* will use the special program written for unusual applications.]

Organizational notes.

Typing *INPUT* executes the cry:xrayin.com command file which in turn runs the cry:input.exe file. The source (*name*.for) and object (*name*.obj) modules for cry:input.exe are in cry_input:. (The *name*s of the input subroutines are related to the names of the **CRYM** programs for which the command file is created. The cry:input.exe file is created by executing cry_input:linkinput.com.

OLDINPUT executes the cry:oldxrayin.com command file which in turn runs the cry:oldinput.exe file. This is merely a previous incarnation of cry:input.exe.

The testcry: input programs are set up in similar fashion. However, here older versions of files are merely that; none are renamed with an '*OLD*' prefix.

Typing *XRAY* executes the cry:xrayrun.com command file which then executes the cry:xrayset.com command file which runs the cry:cryvax.exe file. (Thus the origin of the above mentioned [*username*]xrayset.log upon error termination before the .log file is renamed.) The source (*name*.for) and object (*name*.obj) modules for cry:cryvax.exe are in cry_source. The cry:cryvax.exe file is created by executing cry_source:linkcry.com. Graphics routines from cry_graphics: and input/output routines from cry_lib: are also incorporated. [**N.B.** Merely creating or copying a higher version cryvax.exe does not mean that it will be used. Cryvax.exe is an installed image and the last lines of linkcry.com ensure that the latest version is used. Cry:cryvax.exe is the only **CRYM** program installed.]

TESTXRAY does the same things *XRAY* does, except in testcry:, a different directory. (Testcry:cryvax.exe is not installed.) Again, the analogous '*OLD*' files are merely previous versions.

SPECXRAY is set up somewhat differently. Modified subroutines are written in speccry_source; these are given a different name to distinguish them from the unaltered source, such as 'lsmon_atwin.for' for 'lsmon.for'. Speccry_source:linkcry_master.com then is edited (and there named as well) to include this changed program; all other subroutines are taken from the appropriate cry: directories.

There are several other **CRYM** subdirectories. For example:

- cry_mtcopy: obsolete magnetic tape programs
- cry_misc:, testcry_misc: miscellaneous material not incorporated into **CRYM**
- cry_manual:, the .TeX files of this manual

The INF file

The purpose of the *name.INF* file is to keep a permanent record of important diffractometer information and crystal data such as diffractometer scan speed and 2θ range and the unit cell dimensions and volume. This file may be created at the same time your tape is processed (called by the input program as part of *Taperead*) or at any time through the input program by requesting program *Crystal*. This file may be updated as necessary.

Many of the data are required as input to **CRYM** programs (for instance, the *Cell* card is used by several programs). The input program will take the data from your *name.INF* file to save the trouble of re-entry.

Furthermore, some programs (*Structure Factors*, *Least Squares* and *Absorption*) will add some information (such as the number of reflections, R factor and the absorption coefficient) to the file, if it exists when they are run. If the programs are used several times (for example, several cycles of *Least Squares*), the parameters will be updated. This file will provide a useful record of information required by most journals.

NOTE: The *name*.INF file is purged after being written to.

How to handle problems commonly encountered when running **CRYM** programs:

(1) Stopping a program while it is running:

(a)Batch jobs (on queue *BACK* or *BATCH*):

Type *DELETE/ENTRY=n* where *n*, the job number, may be found by typing *SQB* and locating your job.

(b)Interactive jobs:

Type *ctrl–y* (i.e., *y* while the control key is held down) to immediately stop any interactive job.

(2)Stopping a current print job (on queue *PRINTER, HP4M* or *LASER*):

Type *DELETE/ENTRY=n* where *n*, the job number, may be found by typing *SQL* and locating your job. Please use this feature only when absolutely necessary on the *Laser* printer. In particular, a TEX job should only be deleted in emergencies! (Any problems with the printers should be reported to room 128 Beckman Institute.) The paper tray can be pulled out to halt the laser printer

(3)Mistakenly viewing a binary file (e.g., *name.HKL*, *name.MAG*, etc.):

If you catch your error soon enough (a binary file will appear on the terminal screen as a random bunch of nonsense characters), *ctrl–y* will exit you from the file. However, if you are not quick, *ctrl–y* will not work and you will be stuck in the file and your terminal will be locked. To solve this problem, you will have to kill your process.

First, log onto another terminal and type *SH SYS*. This will give a list of all current processes. Locate your bad process. It should have *LEF* under the column *state*, a high priority (7 or greater) and either your name or terminal as the process name and record the Pid number. Now type *KILL*. This will initiate the kill routine and the system will respond with *KILL:*. Type *K n* where *n* is the 8–digit process ID listed for the bad process. The system will give some information on the process you have named and ask if it is the correct process. If it is, type *Y* and the process will be killed. Type *E* to exit from the kill routine.

Description of Crym Programs

Taperead

Reads the data from a diffractometer, lists the contents and creates a binary* *name.MAG* file of diffractometer information, F's and weights.

Begin

Creates a *name.INF* file, records crystal and data collection parameters.

Crystal Data

Creates a *name.INF* file and calculates the density, volume and its sigma and the absorption coefficient.

Background

Calculates average backgrounds of weak reflections as a function of 2θ.

2Background

Applies average backgrounds as calculated in *Background* and stores corrected F's and weights on the *name.MAG* file.

Decay

Calculates crystal decay as a function of time by examining the check reflections on the *name.MAG* file.

Update

Lists and deletes or restores reflections from the *name.MAG* file.

Absorption Correction

Calculates the transmission coefficients on F^2 and stores them on the *name.MAG* file.

Initial Data Processing

Performs Lp, absorption and decay corrections and averages reflections according to point group symmetry; deletes classes of reflections designated as systematic absences; creates a binary *name.INT* file.

Prepare

Creates, from the *name.INT* file or from hand-entered data, a binary* *name.HKL* file of information needed by most further programs.

Re-Prepare

Updates your *name.HKL* file for changes in cell dimensions, atom form factors or anomalous dispersion or for index switching.

List and *List Phased Disk*

Lists the contents of your *name.HKL* file.

Edit

Allows you to add, delete and/or substitute reflections on your *name.HKL* file.

Make Phased Disk

Creates a *name.HKL* file from special data cards containing h, k, ℓ, weight and A_{calc} and B_{calc} or the phase angle.

Wilson Statistics

Estimates the absolute scale and average overall temperature factor and applies the scale factor to F_{obs} and weight in the *name.HKL* file; generates E's and lists intensity statistics.

Structure Factors

Calculates structure factors and phase angles; lists reflection data and agreement values (goodness-of-fit and R's).

Least Squares

Calculates structure factors and phase angles, lists reflection data and agreement values (goodness-of-fit and R's), generates H atom parameters, and performs the least squares refinement.

Fourier, *Difference Fourier* and *Patterson*

Generates a Fourier, Difference Fourier or Patterson map.

Peak List

Searches a saved map, generated by Fourier, Difference Fourier or Patterson, for peaks.

Contour Map

Produces a contoured map after running a Fourier, Difference Fourier or Patterson map calculation.

Distances, Angles, and Planes

Performs distance, angle, plane and symmetry calculations; calculates principal components of Gaussian ellipsoids.

H Pos

Generates coordinates for hydrogen and other atoms having known geometry.

H Optimize

Converts General Plane Fourier map coordinates to atomic x, y and z coordinates and optimizes the positions of 3 to 9 atoms so as to form a regular polygon.

Cosine

Calculates direction cosines and center point for a plane normal to the line between two atoms, for use by Fourier, Difference Fourier or Patterson.

Rcheck

Calculates various sums, including R, for specific groups of data by reading the *name.-HKL* file.

Parameter Switch

Transforms coordinates, Gaussian ellipsoids and cell dimensions with a unit cell transformation matrix.

Cell

Calculates unit cell dimensions given data from a diffractometer or measurements from comparator film.

Publication List

Creates a structure factor listing suitable for publication from the contents of the *name.HKL* file.

Clist

Prepares a table of coordinates and/or temperature factors suitable for publication from your *name.TAB* file.

Dlist

Prepares a table of distances and/or angles suitable for publication from your *Distan* output file.

Tlist

Prepares tables of crystal data from your *name.INF* file.

Convert

Converts CRYM format *Coord* and *Temp* cards to ORTEP format.

* A binary file is unformatted and cannot be read, editted, typed or printed.

The Binary Files

name.MAG—Created by *Taperead*; used by *Absorption*, *Decay*, *Background*, *2Background*, *Update*, and *Initia*; modified by *Update*, *2Background*, and *Absorption*:

ID number, $h, k, \ell, 2\theta, \chi, \phi$, scan range, scan speed, background1 counts, scan counts, background2 counts, x-ray time, tb/ts and 3 blank slots for each reflection. The first two slots are for the intensity and sigma of the intensity corrected for background and remain empty until *2Background* is run. [Thee slots may be used temporarily by *Taperead.*] The third slot is for the transmission coefficient from *Absorption* and remains empty until *Absorption* is run requesting an updated *name.MAG* file.

name.BGD—Created by *Background*; used by *2Background*:

The number of 2θ intervals, the width of these intervals, and the starting value, the mean background in c.p.s. and the weight for each of the intervals for both distributions. The reflections to be deleted from the *name.MAG* file when *2Background* is run are also included.

name.INT—Created by *Initia*; used by *Prepare*:

h, k, ℓ, F_{obs}, and weight for each reflection.

name.HKL—Created by *Prepare* or *Absorption*; used by *Wilson*, the *Fourier* programs, *Structure Factors*, *Least Squares*, *Re-Prepare*, *Edit*, *List*, *Rcheck*, *Publication List*, and *Absorption*; modified by *Re-Prepare*, *Edit*, *Wilson*, *Absorption*, and *Least Squares*:

h, k, ℓ, F_{obs}, weight $(= 1/(\sigma F_o^2))$, $2\pi h, 2\pi k, 2\pi\ell$, $\frac{\sin\theta^2}{\lambda^2}$, 12 form factors,$A_{spec}$, B_{spec}, sp1, sp2, sp3, sp4, e, sharpened F, A_{calc}, and B_{calc} for each reflection. (Sp1-sp4 are special slots.)

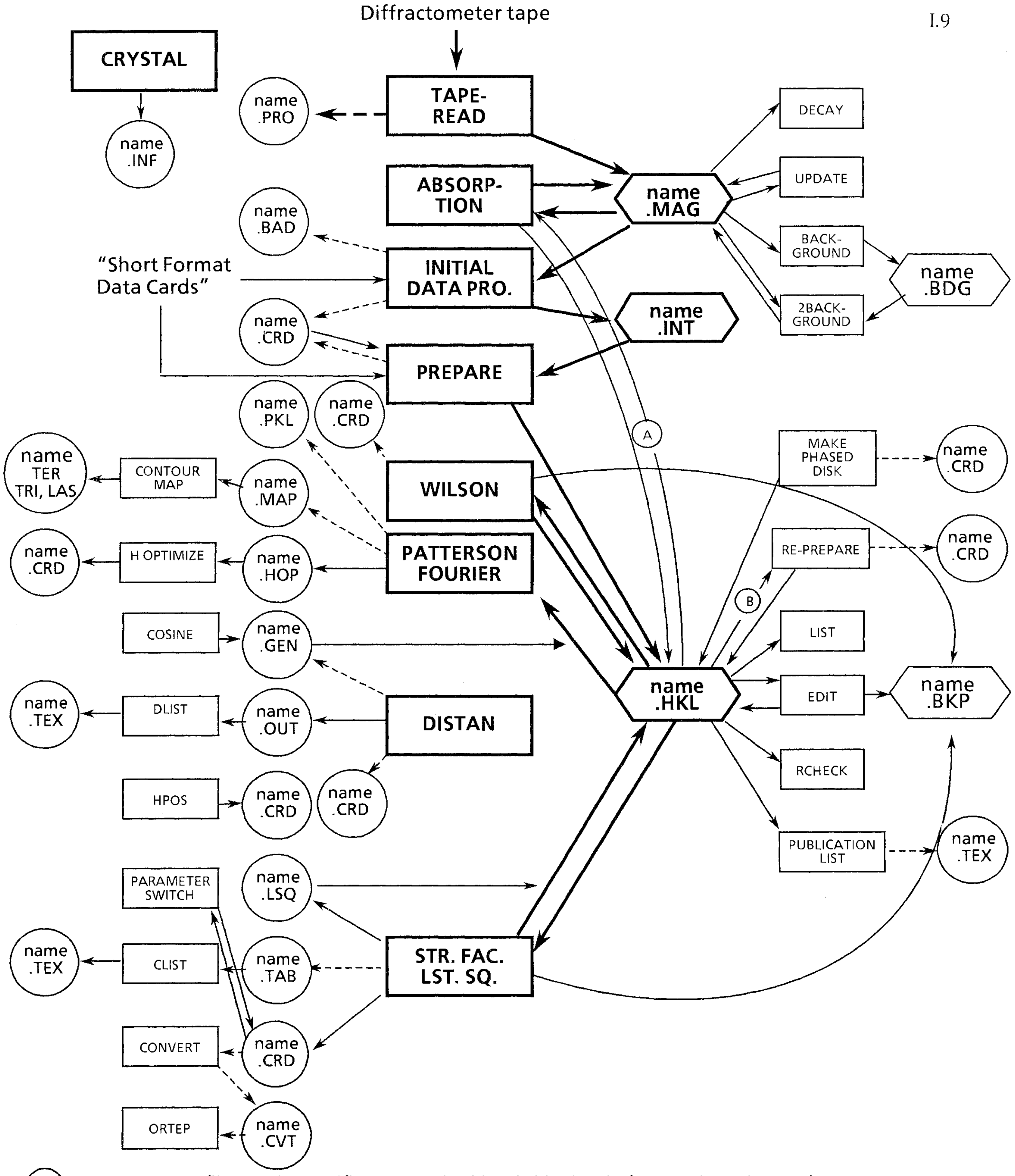
The first three records on the *name.HKL* file are special. They contain:

1)scale factor, the first 10 anomalous dispersion factors,$a, b, c, \alpha, \beta, \gamma, \cos\alpha, \cos\beta, \cos\gamma$, the last 2 anomalous dispersion factors and 9 blanks;

2)12 *Kind F* values, shh $(= \frac{1-\cos a^2}{det\times a^2})$, skk, sll, skl $(= \frac{\cos b\times\cos c-\cos a}{b\times c\times det})$, shk, shl, and 13 blanks;

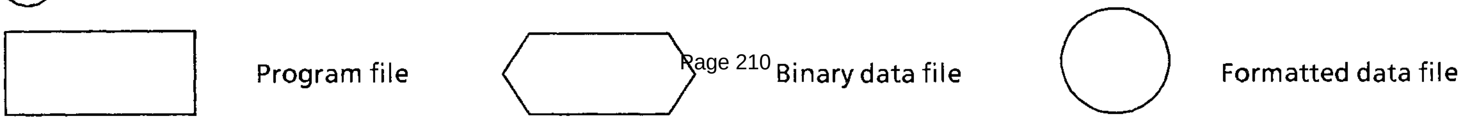
3)31 blanks.

Note: The *name.BKP* file, created by *Wilson*, *Least Squares* and *Edit* is of the same type as *name.HKL*. The *name.BKP* file will contain your original data while the *name.HKL* file will contain the updated data after any of these programs has completed. If the program is succesful, delete *name.BKP*. If you do not want the updated data or if the program fails while running, rename *name.BKP* to *name.HKL*.



Ⓐ Your name.HKL file must be modified to contain Phi and Chi values before ABSORPTION may be run.

Ⓑ Your name.HKL file must be renamed OLDHKL.HKL before RE-PREPARE may be run.



Taperead

This program reads a disk file on either VaxStation 3100, KIRBY or XRAY, from the *CAD4* or *P$\overline{1}$* diffractometers and creates a binary file *name.MAG* and a listing of its contents (optional). *Taperead* will process only the latest .DAT file on disk. It need not be run interactively.

Note: Make sure your file type corresponds to the diffractometer used: *.CAD* for the CAD4, or *.TAP* for the P$\overline{1}$. For files from the *CAD4* diffractometer, the file name, *name.DAT*, must be the same name as your input file name (*name.CAD*).

Note: If you collected data in the non-bisecting mode on the *CAD4* diffractometer, you must run *Absorption* before running *2Background* because *Taperead* (Nonius only) places the value of omega in the corrected intensity slot (11) of the *name.MAG* file. This omega angle is needed for Gaussian absorption corrections when non-bisecting mode was used to collect data and the value of omega will be overwritten when *2Background* is run.

Input cards

1)Title card—Format(A4,5X,A3)
1–8 *TAPEREAD*
10–12 ⟨*CAD*⟩/⟨*TAP*⟩ Data from ⟨*CAD4*⟩/⟨*P$\overline{1}$*⟩ diffractometer. *CAD4* diffractometer.

2)File card—Format(I5,I2,1X,A4,1X,A7,1X,A6)
1–5 ID number of last reflection to be read.
6–7 File number of file to be read; this is ignored for *CAD4* diffractometer data.
9–12 ⟨blank⟩/⟨non-blank⟩ ⟨listing⟩/⟨no listing⟩ of data.
14–20 ⟨blank⟩/⟨non-blank⟩ ⟨do not⟩/⟨do⟩ create a *name.PRO* file of profiles (only if data **is** from the *P$\overline{1}$* diffractometer).
22–27 ⟨non-blank⟩ Use scan counts rather than intensity in generating *name.PSI* file of $\psi-$scan data for empirical absorption correction (only if data **is** from the *CAD4* diffractometer).

3)Scan card—Format(2A4)—*optional*—Not needed if data are from the *CAD4* diffractometer.
1–8 ⟨*SCAN 2-THETA*⟩/⟨*SCAN OMEGA*⟩ ⟨2θ⟩/⟨ω⟩ scan. (The default is *SCAN 2-THETA*)

4)Delete card(s)—Format(A6,1X,11(I5,A1),I5)—*Optional*
1–6 *DELETE*
8–12, 14–18, etc. ID numbers of reflections to be deleted.
13, 19, etc. ⟨ ⟩/⟨–⟩ ⟨single ID⟩/⟨range of ID's⟩ entered.

5)End card—Format(A3)
1–3 *END*

Description of individual cards

1)Title card

Columns 10–12 contain a diffractometer identifier: *CAD* for the CAD4 or *TAP* for the P$\bar{1}$.

2)File card

The file card contains some necessary information concerning the file to be read—the ID number of the last reflection to be read, the file number of the file to be read and whether or not the contents should be printed.

The default value of *0* for the last ID number causes all reflections to be read.

The default for the file number is *1*; only one file is read for *CAD4* diffractometer data. For *P*$\bar{1}$ diffractometer data, several files may be read. There must be one *File* card (followed by any corresponding *Scan* or *Delete* cards that are needed) per file. Every file need not be read (i.e., you may have file 1 read followed by file 3, or you may request that only file 2 be read). The only restriction is that the files be read in the order in which they appear (i.e., file 1 followed by file 3, rather than file 3 followed by file 1).

You may also request that a separate file of profiles be created if your data are from the $P\bar{1}$ diffractometer. If requested, file *name.PRO* will be created containing ID, h, k, ℓ, 2θ, and the base-2 representation of the 96 steps (i.e. 0–1 counts will be listed as *0*, 2–3 counts as *1*, 4–7 counts as *2*, etc. through 2^{20}—after *9*, $A, B, \ldots, K$ are used) for each reflection.

For empirical *psi*−scan absorption correction on the *CAD4* diffractometer, the default is to use intensities to create the *name.PSI* file. If requested, scan counts (ignoring background) will be used.

3)Scan card

No scan card need be entered if your data are from the *CAD4* diffractometer. Otherwise, if no scan card is found, *Scan 2θ* is assumed. If ω-scan is wanted, a *Scan Omega* card must be used.

4)Delete card

This option may be used to enter single reflections or ranges of reflections to be deleted. A single reflection is deleted by entering the ID number of the reflection. A range is deleted by entering the lower inclusive ID number, followed by "–", followed by the upper inclusive ID number.

Note: Reflections deleted in *Taperead* **are not** saved on the *name.MAG* file and may therefore not be used in any calculations. For this reason, deletions should only be made at this time if reflections are known to be unusable for some reason (i.e. diffractometer malfunctions.)

Input Files Needed

name.CAD or *name.TAP* Data from a diffractometer.

Output

name.MAG

A *name.MAG* file of the contents. This file is binary and may not be looked at or edited.

name.OUT

If a listing of the data has been requested, you will have ID, h, k, ℓ, B1, scan, B2, I, $\sigma(I)$, 2θ, ϕ, χ, x-ray time, speed and tb/ts for each reflections. The x-ray time is listed in minutes—a negative value means that the attenuator was used.

If you have used the delete option, a list of deleted reflections will be included.

name.OUT ;+1

For the *CAD4* diffractometer only, a second output file will list selected reflections and sums as an aid in determining space groups.

name.PRO (if requested)

For P$\bar{1}$ diffractometer only, a listing of ID, H, K, L, 2θ and the 96 profile counts for each reflection.

name.PSI

For the *CAD4* diffractometer only, a sorted listing of ϕ and relative intensities or scan counts.

name.INF (if requested)

Additional references

Dunitz, pp. 276–281.

Crystal Data

This program calculates the density, volume (and its sigma) and the absorption coefficient, given the unit cell dimensions and the atomic composition of the crystal. Its purpose is to provide data that are required by most journals.

Many of the cards in the following data deck may be contained in your *name.INF* file, if you created one. The input program will then take this information directly from your file to obviate the necessity of reentering these data. (A *name.INF* file of crystal data may be created at any time through the input program by requesting program *Begin*.)

Input cards

1)Title card—Format(A12)
1–12 *CRYSTAL DATA*

2)Cell card—Format(A4,2X,3F8.5,3F8.4)
1–4 *CELL*
7–14 a (Å)
15–22 b (Å)
23–30 c (Å)
31–38 α (°)
39–46 β (°)
47–54 γ (°)

3)Celsig card—Format(A6,3F8.5,3F8.4)
1–6 *CELSIG*
7–14 $\sigma(a)$
15–22 $\sigma(b)$
23–30 $\sigma(c)$
31–38 $\sigma(\alpha)$
39–46 $\sigma(\beta)$
47–54 $\sigma(\gamma)$

4)Kind F card—Format(A4,2X,12A3,6X,I2)
1–6 *KIND F*
7–9, 10–12, etc. Atom form factor labels.
49–50 Number of additional form factors to be entered. (*Optional.*)

4a)Trailer cards for additional atom form factors—Format(10F8.4/10F8.4/10F8.4/A3) *Optional*
1–10, 11–20, etc. 30 f-values at intervals of .05 in $\frac{sin\theta}{\lambda}$.
/
1–3 Atom form factor label.

5)Numbers of Atoms card—Format(A4,16X,12F4.0)
1–16 *NUMBERS OF ATOMS*
21–24, 25–28, etc. Numbers of atoms of each type in the unit cell (in the same order as on the *Kind F* card).

Description of Individual Cards

The information on cards 2, 4 and 5 is contained in the *name.INF* file. If you have created the file, the input program will take the information from it.

See *Prepare* for a detailed description of the *Cell* and *Kind F* cards and *Wilson* for the *Number of Atoms* card.

3)Celsig card

This card contains the sigmas of the unit cell parameters and is necessary for determining the sigma of the volume of the cell.

Input Files Needed

name.CRY

Output

name.OUT

Values of the cell density, volume, sigma of the volume and the absorption coefficient.

Background

This program uses the *name.MAG* file(s). It examines, at intervals of 2θ, the backgrounds of the weak reflections. For each 2θ interval, the average background, the esd of this average and the goodness-of-fit among the individual contributors are printed. It is anticipated that the results of this program will be used in *2Background* to apply average backgrounds to **all** reflections.

It is the purpose of these programs to establish (*Background*) and apply (*2Background*) background counts that, because of averaging over many reflections, are more precise—and, presumably, more accurate—than the individual background counts that are measured separately for each reflection. They are more accurate because weak reflections are presumably not greatly affected by non-Bragg diffraction (thermal-diffuse scattering, etc.), multiple diffraction effects (Renninger overlap, etc.) or truncation effects. For these programs to be applicable, it must be presumed that the inherent background scattering for any reflection depends only on the Bragg angle (that is, 2θ) for that reflection. Anisotropy in the background scattering is **not** taken into account. Since a highly absorbing crystal of irregular shape or (even worse) a crystal mounted within a capillary will probably show highly anisotropic backgrounds because of absorption by either the crystal or the capillary, such crystals are not good candidates for these programs. (Provision is made to delete from consideration reflections with χ values greater than a prescribed minimum value, and this may be a satisfactory remedy.)

A second purpose of these programs is to partially correct for the effect of scan truncation. Wavelength dispersion in the incident beam, mosaicity in the crystal, and incomplete collimation may cause the effective Bragg scattering for a reflection to extend into the region normally presumed to be background alone causing the measured background for a strong reflection to be too large. In establishing the average background function (*Background*), only the backgrounds from the weaker reflections are used. The average function is then applied to **all** reflections in *2Background*.

Two separate calculations will be made, based on different cut-off values in terms of $\frac{I}{\sigma_I}$; the default values are $a_1 = 8.0$ and $a_2 = 3.0$. With these values, the first calculation will include only those reflections whose net intensity (scan−background) is less than eight times the esd of that intensity (based on counting statistics: $\sigma_I = \sqrt{\text{scan} + \text{background}}$); the second calculation would be based on those reflections with net intensities less than three times the esd. Either distribution can be used by *2Background*. The higher the cut-off value the greater the number of reflections whose backgrounds will contribute to the average; but if the cut-off intensity is too large, truncation effects may become important.

The 2θ interval over which reflections will be averaged can be adjusted. The default is *1*°, and this value should be increased if there are not enough weak reflections within that interval to create useful averages.

Reflections will not be included in the averaging if their two background measurements are seriously disparate; the criterion for deletion is that $B_1 - B_2 > 6\sigma$ ($\sigma = \sqrt{B_1 + B_2}$, the Poisson esd of the background counts) or that $(B_2 - B_1) > 3\sigma$. (B_1 is the low 2θ background and B_2 is the high 2θ background.)

Reflections can also be excluded from the calculations on the basis of their χ value

(Eulerian diffractometer geometry). For example, if the crystal was mounted in a capillary, reflections at high χ may show anomalous backgrounds because of scattering from or absorption by the glass, and can be omitted. Presumably, these same reflections should be omitted from *2Background* calculations.

Reflections can also be deleted from the calculations upon user instruction. These reflections (but not those deleted automatically on the basis of background inequality or those deleted on the basis of χ) will be deleted from the main *name.MAG* file(s) when *2Background* is run unless the user instructs otherwise.

The printed output from this program should be examined carefully before its results are used in *2Background*. The most important consideration is whether the backgrounds are indeed isotropic, that is, whether the generated function can satisfy backgrounds in all regions of reciprocal space; the observed distribution of background counts vs. 2θ should be monotonic. The goodness-of-fit values are of some help here; in addition, the list of reflections, if any, whose backgrounds deviate by more than 3σ from the generated function should be examined for systematic trends. The decision must also be made as to whether the a_1 or a_2 function is more appropriate (or, perhaps, another cutoff should be used in a rerun of the program). Finally, a decision must be made as to the minimum value of 2θ at which a satisfactory function can begin. (See further instructions under *2Background*.)
Note: If you collected data in the non-bisecting mode on the *CAD4* diffractometer, you must run *Absorption* before running *2Background* because *Taperead* (Nonius only) places the value of omega in the corrected intensity slot (11) of the *name.MAG* file.

Input cards

1)Title card—Format(A10,1X,10I2)
1–10 *BACKGROUND*
12–13, 14–15, etc. Version number(s) of *name.MAG* file(s). (The default is use of highest version only.)

2)Limit card—Format(A5,1X,2F6.2,I6,2F7.2) Optional
1–5 *LIMIT*
7–12 1^{st} $\frac{I}{\sigma_I}$ cut-off value. (The default is $a_1 = 8.0$.)
13–18 2^{nd} $\frac{I}{\sigma_I}$ cut-off value. (The default is $a_2 = 3.0$.)
19–24 Width of 2θ intervals in degrees. (The default is *1*°.)
25–31 Lower inclusive χ limit. ($-180 \leq \chi_{min} \leq 180$) (*Optional.*)
32–38 Upper inclusive χ limit. (*Optional.*)

3)Delete card—Format(A6,1X,11(I5,A1),I5) *Optional*
1–6 *DELETE*
8–12, 14–18, etc. ID numbers of reflections to be deleted.
13, 19, etc. $\langle\ \rangle/\langle - \rangle$ $\langle$single ID$\rangle/\langle$range of IDs$\rangle$ entered.

4)End card—Format(A3)
1–3 *END*

Description of individual cards

1)Title card

This card contains the version number(s) of the *name.MAG* file(s) that are to be used. The default, if no version numbers are found, is use of the highest version *name.MAG* file in your directory.

2)Limit card—Optional

This card includes the cut-off values a_1 and a_2 in terms of $\frac{I}{\sigma_I}$. The first calculation will include only those reflections whose net intensity is less than a_1 times the esd of that intensity and the second, those reflections whose intensity is less than a_2 times the esd where a_1 and a_2 are entered on the card with $a_1 \geq a_2$.

The card is also used to enter the width of the 2θ interval. The default is *1*°, but this may be increased if there are not enough weak reflections to create useful averages.

Finally, the card contains the valid χ range for reflections to be accepted. If no range is found, there will be no rejection on the basis of χ values. The lower χ limit, χ_{min}, must be between $-180°$ and $+180°$.

Note: This card is optional. If not entered, all default values will be used.

3)Delete card

This card may be used to enter single reflections or ranges of reflections to be deleted. Single reflections are made by entering the ID number of the reflection. A range is deleted by entering the lower inclusive ID number, followed by a "–", followed by the upper inclusive ID number.

Note: Reflections deleted in *Background* will be stored on the *name.BGD* file(s). **They will only be deleted from the *name.MAG* file when *2Background* is run.** The *Restore* command may be used to override this option.

Input files needed

name.BAC
name.MAG

Output

name.BGD

A *name.BGD* file containing the two background distribution curves. This file is unformatted and may not be read, typed, edited or printed. It is used only by *2Background* and may therefore be deleted after *2Background* has run successfully.

name.OUT

A printed list of 2θ intervals including, for each interval, the average background, the esd of this average and the goodness-of-fit among the individual contributors.

If the delete option is used, a list of deleted reflections will be included. These reflections will be permanently deleted from the *name.MAG* file when *2Background* is run unless otherwise directed (*i.e.* via the *Restore* command).

2Background

This program uses the *name.MAG* file(s) and the *name.BGD* file created by *Background*. It uses one of the two background distributions, obtained in *Background* from the weak reflections and stored on a *name.BGD* file, to calculate improved backgrounds for all reflections. In favorable cases, this process increases the accuracy in the measurements of the weaker reflections and also takes partial account of the effects of truncation on the stronger reflections. The revised backgrounds and their esd's are used to calculate revised I and σ_I values which are stored in special slots on the original *name.MAG* file(s) for use by *Initia*. (*Initia* contains an option to disregard these revised values if desired.)

The revised backgrounds are calculated as follows:

(1)If a reflection has a 2θ value greater than a minimum specified by the user, the value of the background counts is obtained by linear least-squares interpolation of the three surrounding values stored on *name.BGD*;

(2)If the 2θ value is less than that minimum, the background is obtained from a quadratic least-squares fit to all *BGD* values below this minimum.

Note the following requirements:

(1)Above the minimum 2θ, there must be no empty intervals out to the limit of the data. It is advisable that there be at least a moderate number of entries (10, say) in all intervals above the minimum 2θ. The minimum 2θ value that may be used is included in *Background* output;

(2)Below the minimum 2θ, there must be at least three non-zero entries in the intervals created by *Background* in order for a quadratic fit to be successful. (If there are fewer than three non-zero entries, no correction will be applied below the minimum 2θ value.)

Reflections may be excluded from this process if their χ values fall outside specified limits; in this case, the special slots will be filled with values of I and σ_I as calculated from the original background.

Reflections deleted in *Background*, and hence stored on the *name.BGD* file, will be marked as deleted on the *name.MAG* file(s) unless otherwise instructed (*i.e.* via the *Restore* command). Additional reflections can be deleted or restored in *2Background*.

The interpolation table will automatically be taken from the highest version number of *name.BGD*. If a lower version is wanted, it must be *RENAME*d.

Warning: If you collected data in the non-bisecting mode, you must run *Absorption* before running *2Background* because *Taperead* (Nonius only) places the value of omega in the corrected intensity slot (11) of the *name.MAG* file. This omega angle is needed for Gaussian absorption corrections when non-bisecting mode was used to collect data and the value of omega will be overwritten when *2Background* is run.

Input cards

1)Title card—Format(A4,8X,10I2)
1–11 *2BACKGROUND*
13–14, 15–16, etc. Version number(s) of *name.MAG* file(s). (The default is use of highest version only.)

2)Reflection card—Format(F5.0,1X,I1,1X,2F7.2,1X,A4)
1–5 2θ cut-off.
7 ⟨Blank, *0* or *1*⟩/⟨*2*⟩ ⟨First⟩/⟨Second⟩ distribution from *Background* used.
9–15 Lower inclusive χ limit (χ_{min}). ($-180 \leq \chi_{min} \leq 180$) *Optional.*
16–22 Upper inclusive χ limit. (χ_{max}) *Optional.*
24–27 ⟨Blank⟩/⟨Non-blank⟩ ⟨Do⟩/⟨Do not⟩ list reflections with updated I and σ_I values.

3)Restore card—use one of the following—*Optional*
a)Format(A7,11(I5,A1),I5)
1–7 *RESTORE*
8–12, 14–18, etc. ID numbers of reflections to be restored.
13, 19, etc. ⟨ ⟩/⟨–⟩ ⟨Single ID⟩/⟨Range of IDs⟩ entered.

b)Format(A11)
1–11 *RESTORE ALL*

4)Delete card—Format(A6,1X,11(I5,A1),I5)
1–6 *DELETE*
8–12, 14–18, etc. ID numbers of reflections to be deleted.
13, 19, etc. ⟨ ⟩/⟨–⟩ ⟨Single ID⟩/⟨Range of IDs⟩ entered.

5)End card—Format(A3)
1–3 *END*

Description of individual cards

1)Title card

The *Title* card is used to enter the version number(s) of the *name.MAG* file(s) to be used. The default, if no version numbers are specified, is use of the highest version *name.MAG* file in your directory only and may be used if your directory contains only one *name.MAG* file.

2)Reflection card

This card is used to give information concerning the calculation of the new backgrounds. A 2θ value is entered such that above this value, an average background is determined by a linear least-squares fit through the three surrounding values. Therefore, no intervals above this value may be empty. Below this value, there will be one quadratic least-squares fit made through all of the intervals. If there are fewer than three such intervals, reflections falling in this category will have their special I and σ_I slots filled with the I and σ_I values based on the original background.

Also wanted is the number of the distribution from *Background* to be used (*1* or *2*).

A valid χ range may be entered. Reflections falling outside this range will have their special slots on the *name.MAG* file(s) filled by the I and σ_I as calculated from the original background. If no range is entered, there will be no rejection on the basis of χ values.

The lower χ limit must be between -180 and $+180$. **Note:** If you have specified a valid χ range in *Background*, you will probably want to use the same one here.

3)Restore card

This option is used to request that reflections deleted previously be included in *2Background* calculations and **not** marked as deleted when the *name.MAG* file(s) is rewritten. *Restore All* restores all deleted reflections. *Restore* is used to restore single reflections or ranges of reflections. Single restorations are made by entering the ID number of the reflection. A range is restored by entering the lower inclusive ID number, followed by a "–", followed by the higher inclusive ID number.

4)Delete option

This option is used to specify single reflections or ranges of reflections to be deleted. This is done exactly as on the *Restore* card.

Note: All reflections deleted when *2Background* is run (including reflections deleted in *Background*) will be marked as deleted on the *name.MAG* file(s) and ignored in all future calculations unless directed otherwise (i.e., restored in *Initia* or in *Update*). A list of all reflections marked as deleted when *2Background* is run is included in output.

Input Files Needed

name.2BA
name.MAG
name.BGD

Output

name.MAG

Updated *name.MAG* file(s). This file(s) will have the same version number as the original file. The new I and σ_I based on the new background will be included in this updated version.

name.OUT

If the list option is used, all reflections will be printed with their updated I and σ_I values.

All reflections currently marked as deleted on the *name.MAG* file(s) will be printed.

Decay

This program reads the check reflections (marked by a negative ID number) from the *name.MAG* file(s) and calculates the coefficients for three types of curves—constant (no decay), linear and quadratic in time—for each check reflection by means of a least-squares fit through the F_{obs} values. A preliminary calculation of the curves is made using all reflections. Then, reflections having a large discrepancy between their F_{obs} and F_{calc} values will be rejected before calculating the final coefficients. The criterion for rejection is that $|\frac{F_{obs}-F_{calc}}{\sigma F_{obs}}|$ be greater than a given maximum, δ_{max}, where F_{calc} comes from the preliminary calculation of the decay curves and σ_F is read from the *name.MAG* file and is normally based on counting statistics. Coefficients, derived from these results, may be used as input to *Initia* on the *Scale on Time* card.

Input cards

1)Title card—Format(A5,1X,10A2)
1–5 *DECAY*
7–8, 9–10, etc. Version number(s) of *name.MAG* file(s) to be used. (The default is use of highest version only.)

2)Starting time and δ_{max} card—Format(A4,1X,F8.2,F6.2) *Optional*
1–4 *TIME*
6–13 Starting x-ray time in minutes. (The default is *0.0*.)
14–19 δ_{max}. (The default is *10.0*.)

3)Delete card—Format(A6,1X,11(I5,A1),I5) *Optional*
1–6 *DELETE*
8–12, 14–18, etc. ID numbers of reflections to be deleted.
13, 19, etc. ⟨ ⟩/⟨–⟩ ⟨Single ID⟩/⟨Range of IDs⟩ entered.

4)End card—Format(A3)
1–3 *END*

Description of individual cards

1)Title card

The *Title* card is used to enter the version number(s) of the *name.MAG* file(s) to be used. The default, if no version numbers are specified, is use of the highest version *name.MAG* file in your directory only and may be used if your directory contains only one *name.MAG* file.

2)Starting time and δ_{max} card

This card is used to specify the starting exposure time in minutes (default is *0.0*) and δ_{max}. If $|\frac{F_{obs}-F_{calc}}{\sigma F_{obs}}| > \delta_{max}$ then that check reflection will be discarded when calculating the final coefficients of the decay curves (default is *10.0*). F_{calc} is the value obtained from a preliminary calculation of the decay curves with all check reflections included.

If this card is not included, the default values will be used.

3)Delete card

This card may be used to enter single reflections or ranges of reflections to be deleted. Single reflections are deleted by entering the ID number of the reflection. A range is deleted by entering the lower inclusive ID number, followed by a –, followed by the upper inclusive ID number.

Note: Reflections deleted at this time are only deleted from the *Decay* calculations. They are **not** permanently deleted from the *name.MAG* file(s).

Input Files Needed

name.DEC
name.MAG

Output

name.OUT

A list of the check reflections, their intensities, and the weighted residuals of these values $(= \frac{F_{obs}-F_{calc}}{\sigma F_{obs}})$ for the three decay curves before and after reflections are deleted based on δ_{max};

The coefficients, F_{calc} values, and the GOF for the three decay curves for each check reflection. The coefficients are in the forms requested by the *Scale on Time* card in *Initia*. The percent decay is calculated for linear and quadratic curves, as required for *Acta Cryst.* manuscripts;

A list of all check reflections deleted from *Decay* calculations;

A graph of I vs. exposure time for up to five check reflections.

2Decay

This program reads the check reflections (marked by a negative ID number) from the *name.MAG* file(s) and applies a decay correction to them, as chosen from the results of DECAY. It then calculates the coefficients for three types of curves—constant (no decay), linear and quadratic in time—for each check reflection by means of a least-squares fit through the F_{obs} values. A preliminary calculation of the curves is made using all reflections. Then, reflections having a large discrepancy between their F_{obs} and F_{calc} values will be rejected before calculating the final coefficients. The criterion for rejection is that $|\frac{F_{obs}-F_{calc}}{\sigma F_{obs}}|$ be greater than a given maximum, δ_{max}, where F_{calc} comes from the preliminary calculation of the decay curves and σ_F is read from the *name.MAG* file and is normally based on counting statistics. If the chosen decay correction constants are appropriate, the goodness of fit of the "no decay" curve should be approximately the same as the other goodness of fit values.

Input cards

1)Title card—Format(A5,1X,10A2)
1–6 *2DECAY*
7–8, 9–10, etc. Version number(s) of *name.MAG* file(s) to be used. (The default is use of highest version only.)

2)Starting time and δ_{max} card—Format(A4,1X,F8.2,F6.2) *Optional*
1–4 *TIME*
6–13 Starting x-ray time in minutes. (The default is *0.0*.)
14–19 δ_{max}. (The default is *10.0*.)

3)Delete card—Format(A6,1X,11(I5,A1),I5) *Optional*
1–6 *DELETE*
8–12, 14–18, etc. ID numbers of reflections to be deleted.
13, 19, etc. ⟨ ⟩/⟨–⟩ ⟨Single ID⟩/⟨Range of IDs⟩ entered.

4)Scale card—Format(A4,5X,A1,12X,F8.2,2F10.6,F8.2)
1–13 SCALE ON TIME
23–30 k in $F_{new} = F_{obs} \times \frac{k}{1+A(T-T_o)+B(T-T_o)^2}$
31–40 A in equation
41–50 B in equation
51–58 T_o in equation

5)End card—Format(A3)
1–3 *END*

Description of individual cards

1)Title card

The *Title* card is used to enter the version number(s) of the *name.MAG* file(s) to be used. The default, if no version numbers are specified, is use of the highest version *name.MAG* file in your directory only and may be used if your directory contains only one *name.MAG* file.

2)Starting time and δ_{max} card

This card is used to specify the starting exposure time in minutes (default is *0.0*) and δ_{max}. If $|\frac{F_{obs}-F_{calc}}{\sigma F_{obs}}| > \delta_{max}$ then that check reflection will be discarded when calculating the final coefficients of the decay curves (default is *10.0*). F_{calc} is the value obtained from a preliminary calculation of the decay curves with all check reflections included.

If this card is not included, the default values will be used.

3)Delete card

This card may be used to enter single reflections or ranges of reflections to be deleted. Single reflections are deleted by entering the ID number of the reflection. A range is deleted by entering the lower inclusive ID number, followed by a –, followed by the upper inclusive ID number.

Note: Reflections deleted at this time are only deleted from the *Decay* calculations. They are **not** permanently deleted from the *name.MAG* file(s).

4)Scale card

This is identical to the SCALE ON TIME card for Initial Data Processing and has the overall scale factor k (usually equal to 1.0), $a * 10^3$ and $b * 10^6$, plus a T_o value for non-zero starting times (equal to starting X-ray time on card 2).

Input Files Needed

name.2DE
name.MAG

Output

name.OUT

A list of the check reflections, their intensities, and the weighted residuals of these values ($= \frac{F_{obs}-F_{calc}}{\sigma F_{obs}}$) for the three decay curves before and after reflections are deleted based on δ_{max};

The coefficients, F_{calc} values, and the GOF for the three decay curves for each check reflection. The coefficients are in the forms requested by the *Scale on Time* card in *Initia*. The percent decay is calculated for linear and quadratic curves, as required for *Acta Cryst.* manuscripts;

A list of all check reflections deleted from *Decay* calculations;

A graph of I vs. exposure time for up to five check reflections.

Update

This program uses the *name.MAG* file(s). It can be used for any combination of the following:

1)List the contents of the *name.MAG* file(s);

2)Delete, according to reflection number, any reflections or groups of reflections;

3)Restore any reflections or groups of reflections that have been previously deleted.

The output will include a list of all reflections marked as deleted at the conclusion of the program.

Input cards

1)Title card—Format(A6,1X,10I2)
1–6 *UPDATE*
8–9, 10–11, etc. Version number(s) of *name.MAG* file(s). (The default is use of highest version only.)

The following three cards may be entered in any order. At least one must be entered for anything meaningful to happen.

2)List card—Format(A4)
1–4 *LIST*

3)Restore card—use one of the following
a)Format(A7,11(I5,A1),I5)
1–7 *RESTORE*
8–12, 14–18, etc. ID numbers of reflections to be restored.
13, 19, etc. ⟨ ⟩/⟨–⟩ ⟨Single ID⟩/⟨Range of IDs⟩ entered.

b)Format(A11)
1–11 *RESTORE ALL*

4)Delete card—Format(A6,1X,11(I5,A1),I5)
1–6 *DELETE*
8–12, 14–18, etc. ID numbers of reflections to be deleted.
13, 19, etc. ⟨ ⟩/⟨–⟩ ⟨Single ID⟩/⟨Range of IDs⟩ entered.

5)End card—Format(A3)
1–3 *END*

Description of individual cards

1)Title card

The *Title* card is used to enter the version number(s) of the *name.MAG* file(s) to be used. The default, if no version numbers are specified, is use of the highest version *name.MAG* file in your directory only and may be used if your directory contains only one *name.MAG* file.

2)List card

This option requests a complete listing of the *name.MAG* file(s). The listing includes: ID number, h, k, ℓ, 2θ, ϕ, χ, scan range, scan speed, B1 counts, scan counts, B2 counts, exposure time and background-to-scan-time ratio. If the *Background* programs have been run, net intensities and their sigmas, calculated from corrected backgrounds, will be included. If *Absorption* has been run, the transmission coefficient will be included.

Reflections that have been deleted will be identified by a negative scan speed. This flag can be removed so that the reflection may be included in subsequent calculations by the *Restore* command.

3)Restore card

This option is used to request that reflections deleted previously (either by the *Background* programs or by an earlier use of this program) **not** be marked as deleted when the *name.MAG* file(s) is rewritten. *Restore All* restores all deleted reflections. *Restore* is used to restore single reflections or ranges of reflections. Single restorations are made by entering the ID number of the reflection. A range is restored by entering the lower inclusive ID number, followed by a "–", followed by the higher inclusive ID number.

4)Delete option

This option is used to specify single reflections or ranges of reflections to be deleted. This is done exactly as on the *Restore* card.

Input Files Needed

name.UPD
name.MAG

Output

name.MAG

Updated *name.MAG* file(s) having the same version number as the original file(s) with appropriate deletions and/or restorations.

name.OUT

A listing of all reflections that have been coded as "deleted," either by this program or by the *Background* programs.

If you have used the list option, for each reflection, ID, h, k, ℓ, 2θ, ϕ, χ, range, speed, B1, scan, B2, exposure time, and t_b/t_s will be printed. If the *Background* programs have been run, I and σ_I, corrected for the averaged background, will be included. If *Absorption* has been run, the transmission coefficient will be printed.

Absorption Correction

This program calculates an absorption correction. The different methods available are:

1) Gaussian quadrature (Busing and Levy, *Acta Cryst.*, **10**, 180, 1957). The integration is calculated over a grid of $8 \times 8 \times 8$ points.
2) Psi-scan correction. Relative intensities from high χ scans are used to calculate (by interpolation; cubic or quartic fit to be substituted) the absorption correction constant. A θ dependence is included as a default option, but may not be the best method.
3) Analytical function. The absorption correction is evaluated from a function fit to ψ-scan data. (In ABS.FOR, check the program if you need this.)

The calculated absorption correction is applied to F_{obs} and to the *weight* ($= \sigma_{F_{obs}}^{-2}$). Input is normally from *name.MAG* file(s), but can also come from a *name.HKL* file if it has been modified to include ϕ and χ values, or from a specially prepared *name.CRD* file. You may choose whether or not the corrections are written in your *name.OUT* file or applied to your input file (*name.MAG*, *name.HKL* or *name.CRD*).

Note: If you collected data in the non-bisecting mode on the *CAD4* diffractometer, you must run *Absorption* before running *2Background* because *Taperead* (Nonius only) places the value of omega in the corrected intensity slot (11) of the *name.MAG* file.

Input cards

1)Title card—Format(A4,18X,10I2)
1–21 *ABSORPTION CORRECTION*
23–24, 25–26, etc. Version number(s) of *name.MAG* file(s) if used.

2)Constants card—Format(A4,I4,F10.6,A3,A3,3X,A3,F10.2,2X,A7)

1–3 ⟨*CON*⟩/⟨*PSI*⟩⟨*PSI1*⟩ Conventional gaussian quadrature, semi-empirical psi-scan method or analytical function.

5–8 ⟨*Number of faces*⟩/⟨*Number of interpolation intervals*⟩.

9–18 Linear absorption coefficient in cm^{-1}.

19–21 ⟨*MAG*⟩/⟨*HKL*⟩/⟨*CRD*⟩ Input from ⟨*MAG*⟩/⟨*HKL*⟩/⟨*CRD*⟩ file.

22–24 ⟨*MAG*⟩/⟨*HKL*⟩/⟨*CRD*⟩/⟨blank⟩ Updated output to ⟨*MAG*⟩/⟨*HKL*⟩/⟨*CRD*⟩/⟨no⟩ file.

28–30 ⟨Blank⟩/⟨Non-blank⟩ ⟨Do not⟩/⟨Do⟩ list updated reflections in *name.OUT* .

31–40 Φ offset in degrees for ψ-scan correction.

43-49 ⟨Blank⟩/⟨Non-blank⟩ ⟨*NOTHETA*⟩ ⟨Use⟩/⟨Do not use⟩ θ dependence for ψ-scan correction.

3a)Face cards—Format(A4,2X,4F8.5)
1–4 *FACE*
7–14 χ (positive or negative).
15–22 ϕ
23–30 Distance from center of rotation to crystal face in *cm.*
31–37 Distance from center of rotation to crystal face in *mm.*

3b)Psi-scan curve data cards—Format(3X,6(F6.3))
1–3 Blank
4–9,10–15,etc. Values of the absorption coefficient at intervals in ϕ, starting at $\phi = 0$. The number of intervals specified on the *Constants* card determines the number of these cards needed.

3c)PSI1—(4F9.4)

4)Reflection cards—Format(4X,3I6,6F8.2) Use only if *CRD* is in columns 19–21 on your *Constants* card

5–10,11–16,17–22	h,k,ℓ
23–30	F_{obs}
31–38	CRYM weight
39–46	2θ
47–54	ϕ
55–62	χ

Description of individual cards

1)Title card

If input is from *name.MAG* file(s), this card includes the version number(s) of the *name.MAG* file(s) to be used. The default is use of the highest version *name.MAG* file only and may be used if your directory contains only one *name.MAG* file.

2)Constants card

This card determines the type of method to be used to calculate the absorption correction factor abscor to be added to the *name.MAG* file.

CON specifies the gaussian quadrature method in which case the second parameter on the card is the number of observed (and measured) crystal faces. There must be one *Face* card for each face specified here. *PSI* indicates a psi-scan curve will be used; here the second parameter is the number of intervals, typically 36 for 10 degree steps (maximum number of intervals is 72).

This card also contains the input and output file types. Input may be *name.MAG* file(s), a *name.HKL* file or reflection cards (or a *name.CRD* file of reflection cards). Output may be to an updated *name.MAG* file(s), a *name.HKL* file or a *name.CRD* file. **Beware of input-output incompatibility.** If input is from your *name.MAG* file(s), output may be to a *name.MAG* file(s) or to a *name.CRD* file. If input is a *name.HKL* file, output may be to a *name.HKL* file or a *name.CRD* file. If the input is a set of reflection cards, output may be to a *name.CRD* file only. You may also request that no file be updated.

If you request that your *name.MAG* file be updated, the transmission coefficient will be placed in a special slot on the file for use by *Initia*. If *Absorption Correction* is run again, the stored transmission coefficient is **not** used in the calculations. A new coefficient will replace the stored one if an updated file is requested.

Note: If input is from a *name.HKL* file, that file must have been modified so as to contain ϕ and χ values (by use of the program *HKL Add*).

You may also request that the updated *F* values be included in your *name.OUT* file.

3a)Face card - used only for conventional gaussian quadrature.

There must be the same number of *Face* cards as is specified on the *Constants* card. Each card contains χ (positive or negative) and ϕ, in degrees, when the face is in reflecting position, and the distance from the center of rotation to the crystal face in *cm* or *mm.*

3b)Psi-scan curve data card - used for psi-scan correction.

These cards contain the values of the relative absorption coefficient of the crystal as a function of the diffractometer angle ϕ, at intervals starting at $\phi = 0$. There are six values per card. The coefficient is obtained by dividing the actual intensity at a given ϕ value by the average intensity; the data are usually obtained for a number of reflections (usually six or more including Friedel mates if applicable) and smoothed, averaged values used.

3c)PSI1

$$\text{ABSCOR} = a_1(1 - a_2 e^{(-a_3|cos(x+a_4|))})$$

4)Reflection cards

If input is from cards, the reflection cards follow the last *Face* card. The last reflection card must be blank to indicate the end of card read-in. Each card contains h, k, ℓ, F_{obs}, CRYM weight, 2θ, ϕ and χ.

Note: If input is a *name.CRD* file of reflection cards, it must be copied into your input file after the last *Face* card.

Note: This program does not require an *End* card. It may be followed by another program or terminated with an *End X-ray* card.

Input files needed

name.ABS

name.MAG or *name.HKL* (as specified on the *Constants* card)

Output

name.MAG, *name.HKL* or *name.CRD*

An updated *name.MAG* file(s), *name.HKL* file or *name.CRD* file including new F values with the transmission coefficient applied, if requested.

name.OUT

The reflections with the maximum and minimum transmission coefficients and, if requested, a listing of the updated reflections.

Initial Data Processing

Initial Data Processing (*Initia*) can perform any or all of the following operations:

1)Lp corrections;

2)Delete unwanted reflections, such as systematic absences or diffractometer malfunctions, from the data set;

3)Correct F_{obs} values for decay, using results of the *Decay* program;

4)Transform indices so as to conform with standard space-group designation;

5)Average F^2 values according to any desired point-group symmetry;

6)Assign weights to the averaged F^2 values, including *lack-of-confidence* factors.

Input is normally from a *name.MAG* file(s), but also can come from a hand-created file. If input is from a hand-created file, operations *1* and *3* do not apply. There are two types of input instructions—*program* instructions and *data* instructions. *Program* instructions are entered only once, and define conditions for the entire operation. There are two kinds of *data* instructions. The first type is specific to each particular input data block (*name.MAG* file as specified on the *Disk* card or a set of *Short Format Data* cards). The second type affects all following data blocks unless a change is specified.

Note: The input file made up of the following cards **must** be given the name *name.INI*.

Input cards

Program Instructions—entered only once to define conditions for the entire operation

1)Title card—Format(A4,20X,I2)
1–23 *INITIAL DATA PROCESSING*
25–26 Point group number (see chart, page 7.9).

2)Weight card—Format(A6,1X,2F6.3,1X,2(A3,1X)) *Optional*
1–6 *WEIGHT*
8–13 Weight constant before averaging. (The default is *.014*.)
14–19 Weight constant after averaging. (The default is *.014*.)
21–23 ⟨Blank⟩/⟨*AVG*⟩ Final F^2 on basis of ⟨weighted⟩/⟨simple⟩ average of equivalent reflections.
25–27 ⟨Blank⟩/⟨*AVG*⟩ Final weight on basis of ⟨sum⟩/⟨average⟩ of the weights of the individual contributors.

3)Data card—Format(I2,1X,A4,2X,A4,I3,1X,A3)
1–2 Order for sorting (see chart, page 7.4).
4–7 ⟨Blank⟩/⟨*LIST*⟩ ⟨Do not⟩/⟨Do⟩ print input data. (Not to be used if input is from *Short Format Data* cards.)
10–13 ⟨Blank⟩/⟨*CARD*⟩/⟨*BOTH*⟩ Output to ⟨binary *name.INT* file⟩/⟨*name.CRD* file⟩/⟨*name.INT* and *name.CRD* files⟩.
14–16 Number of types of systematic absences. (The default is *0*.)
17–19 ⟨Blank⟩/⟨Non-blank⟩ ⟨Do not⟩/⟨Do⟩ create a *name.BAD* file of reflections with high GOF values.

3a)Systematic Absences card—Format(10I5) (used only if specified on *Data* card)
1–5, 6–10, etc. Symmetry code constants (see chart, page 7.8).

Type 1 Data Instructions

1)Switch cards—Format(A4,3X,I5,2X,I5)/(9F5.0) *Optional*—consists of two cards
1–6 *SWITCH*
8–12 Lower inclusive ID number (> 0) for reflections to be switched. (*Optional.*)
15–19 Upper inclusive ID number (> 0) for reflections to be switched. (*Optional.*)
/
1–5, 6–10, etc. The nine elements A_1–A_9 (see p. 7.5) of the transformation matrix.

2)Restore card—use one of the following—*Optional* (Not to be used if input is from *Short Format Data* cards.)
a)Format(A7,11(I5,A1),I5)
1–7 *RESTORE*
8–12, 14–18, etc. ID numbers of reflections to be restored.
13, 19, etc. ⟨ ⟩/⟨–⟩ ⟨Single ID⟩/⟨Range of IDs⟩ entered.

b)Format(A11)
1–11 *RESTORE ALL*

3)Delete card—Format(A6,1X,11(I5,A1),I5) *Optional* (Not to be used if input is from *Short Format Data* cards.)
1–6 *DELETE*
8–12, 14–18, etc. ID numbers of reflections to be deleted.
13, 19, etc. ⟨ ⟩/⟨–⟩ ⟨Single ID⟩/⟨Range of IDs⟩ entered.

Type 2 Data Instructions—affect all following data blocks unless a change is indicated

1)Monochromator card—Use one of the following if input **is** from *name.MAG* file(s)
a)Format(A4,12X,2F8.2,F5.3)
1–13 *MONOCHROMATOR*
17–24 θ (The default is *6.08*°.)
25–32 ϵ (The default is *90.0*°.)
33–37 η, the degree of imperfection constant. (The default is *0.5*.)

2)Scale card—Use one of the following—*Optional* (if none is used, the scale factor is assumed to be *1.0*)
a)Format(A4,18X,F8.2)
1–5 *SCALE*
23–30 Scale factor on F_{obs}.

b)Format(A4,5X,A1,12X,F8.2,2F10.6,F8.2)
1–13 *SCALE ON TIME*
23–30 k in $F_{new} = F_{obs} \times \frac{k}{1+A(T-T_0)+B(T-T_0)^2}$.
31–40 A in equation.
41–50 B in equation.
51–58 T_0 in equation.

c)Format(A4,5X,A1,12X,F8.2,6X,I5,4X,I5)
1–11 *SCALE ON ID*
23–30 Scale factor (applied to specified range of IDs).
37–41 Lower inclusive ID number (> 0) for reflections to be scaled. (*Optional.*)
46–50 Upper inclusive ID number (> 0) for reflections to be scaled. (*Optional.*)

3)Background card—Format(A4) *Optional* (Not to be used if input is from *Short Format Data* cards)
1–13 $\langle$*BACKGROUND*$\rangle$/$\langle$*NO BACKGROUND*$\rangle$ $\langle$Do$\rangle$/$\langle$Do not$\rangle$ use I and σ_I values corrected for background in *2Background* program.

4)Absorption card—Format(A4) *Optional* (Not to be used if input is from *Short Format Data* cards.)
1–13 $\langle$*ABSORPTION*$\rangle$/$\langle$*NO ABSORPTION*$\rangle$ $\langle$Do$\rangle$/$\langle$Do not$\rangle$ apply transmission coefficients in calculations.

5)F Min card—Format(A5,4X,I1,12X,F8.2) *Optional*
1–5 *F MIN*
10 $\langle$Blank or *0*$\rangle$/$\langle$*1*$\rangle$ Assigns 0.0 to weight and F_{obs} if $\langle F_{obs} < R\rangle / \langle F < R \times \sigma F_{obs} \rangle$.
23–30 R from appropriate equation.

6)You must use at least one of the following two cards:

a)Disk card—Format(A4,4X,I2)
1–4 *DISK*
9–10 Version number of *name.MAG* file. (The default is highest version only.)

b)Short Format Data cards—Format(4X,3I6,F8.3,F8.5)
5–10 h
11–16 k
17–22 ℓ
23–30 F_{obs}
31–37 CRYM weight ($= \sqrt{w}$)

7)End card—Format(A3)
1–3 *END*

Description of individual cards

1)Title card

This card includes the point group number according to the chart on page 7.9.

2)Weight card

This optional card specifies how to treat weights, particularly when multiple observations are to be averaged. Two weight constants (or *lack-of-confidence factors*) may be entered; they both decrease the weight of an observation by adding a term $(\alpha F^2)^2$ to the variance of F^2. [See Stout & Jenson, pp. 435–440; McCandlish, Stout, & Andrews, *Acta Cryst.*, **245** (1975).] The first value adds the term before that observation is averaged; the second adds the term to the final averaged value. Thus, the first is designed to reflect unknown errors in the individual measurements while the second is designed to compensate for systematic errors such as absorption or errors in the model. The default value for each constant is *0.014*.

The *Weight* card may be used to request simple averaging, rather than a weighted average, of equivalent reflections, though the latter is advised.

Finally, you may request that the program establish a final weight on the basis of the average (rather than the sum) of the weights of the individual contributors, but this is not recommended.

If this card is not encountered, all default values will be used.

3)Data card

This card establishes the order of indexes for the final, averaged data file and defines some output options. The order for sorting is as follows:

Order	Fast	Slow
1	h	ℓ
2	h	k
3	k	h
4	k	ℓ
5	ℓ	h
6	ℓ	k

You may also request that your data appear in a *name.CRD* file rather than a *name.INT* file. The *name.INT* file, which can be read by *Prepare*, is unformatted and cannot be read, typed, edited or printed. The *name.CRD* file, which can be used as input to *Prepare CRD*, may be edited, but requires more disk space. You may also request that both a *name.CRD* and *name.INT* file be created.

You may enter the number of types of systematic absences which you want deleted from the data set. If you choose this option, you must enter a trailer card containing the specified number of codes describing the conditions for keeping a class of reflections according to the chart on page 7.8. The index conditions defining the systematic absences refer to the final indexing scheme, **after *switching* has been done**.

Finally, you may request that a *name.BAD* file be created. This file will contain the ID numbers and goodness-of-fit for reflections with high (starred in output) GOF's.

Type 1 Data Instructions (applies only to the following data block)

1)Switch card

This option calls for a trailer card containing the nine elements A_1–A_9 of a transformation matrix for the reflection indices. All reflections in the following data block with ID numbers within the specified range will have their reflection indices transformed as follows:

$$\begin{aligned} h_{new} &= h * A_1 + k * A_2 + l * A_3 \\ k_{new} &= h * A_4 + k * A_5 + l * A_6 \\ l_{new} &= h * A_7 + k * A_8 + l * A_9 \end{aligned} \quad \text{or} \quad \begin{pmatrix} h \\ k \\ l \end{pmatrix}_{new} = \begin{pmatrix} A_1 & A_2 & A_3 \\ A_4 & A_4 & A_6 \\ A_7 & A_8 & A_9 \end{pmatrix} \begin{pmatrix} h \\ k \\ l \end{pmatrix}$$

If no range is specified, the transformation will be applied to all reflections within the next data block. There can be no more than 100 sets of these two cards preceding any data block.

2)Restore card

This option is used to request that reflections deleted previously be included in *Initia* calculations and **not** be deleted from the *name.INT* and/or *name.CRD* output file. *Restore All* restores all deleted reflections. *Restore* is used to restore single reflections or ranges of reflections. Single restorations are made by entering the ID number of the reflection. A range of reflections to be restored is specified by entering the lower inclusive ID number, followed by a "–", followed by the higher inclusive ID number.

3)Delete card

This option is used to specify single reflections or ranges of reflections to be deleted. This is done exactly as on the *Restore* card.

Type 2 Data Instructions (applies to all following data blocks unless changed)

1)Monochromator card

This card is used for Lorentz and polarization correction. It includes θ_m for the monochromator at a particular wavelength. Examples for a graphite monochromator are:

Mo radiation	$\theta_m = 6.08°$	*Cu* radiation	$\theta_m = 13.28°$
Ag radiation	$\theta_m = 4.80°$	*Co* radiation	$\theta_m = 15.48°$
Fe radiation	$\theta_m = 16.78°$	*Cr* radiation	$\theta_m = 19.97°$

The default is *6.08°*. Also included on this card is ϵ, the angle between the diffraction plane of the diffractometer (horizontal) and the diffraction plane of the monochromator. It is 90° for the CAD4, $P2_1$ and $P\bar{1}$ with tube mount horizontal, but 0° for the $P2_1$ or $P\bar{1}$ with tube mount vertical. The default is *90°*. (See Arndt & Willis, *Single Crystal Diffractometry*, p. 287.) Typically, $L = \frac{1}{\sin 2\theta}$ and $p = \frac{1+\cos^2 2\theta \cos^2 2\theta_m}{1+\cos^2 2\theta_m}$ for $\epsilon = 0°$ or $p = \frac{\cos^2 2\theta + \cos^2 2\theta_m}{1+\cos^2 2\theta_m}$ for $\epsilon = 90°$.

The card also contains η, the *degree of imperfection* constant in Lp correction. Monochromators are typically more perfect than the usual crystalline samples grown, but less perfect than the "ideally perfect" crystals of diamond or germanium grown by zone melting (see Stout & Jenson, pp. 390–394; Dunitz, pp. 291–296). $p = \eta p_{perfect} + (1-\eta)p_{imperfect}$. Either this card or a *No Monochromator* card must be used before the first data block if input is from *name.MAG* file(s). The values on the entered card then become the default values for all further data blocks until and unless another *Monochromator* or *No Monochromator* card is encountered.

2)Scale card

The *Scale* card is used to apply a scale factor to F_{obs}. There are 3 types of scale cards. The first type applies the given constant to all reflections. The second type, *Scale on Time*, corrects all reflections for decay according to

$$F_{new} = F_{obs} \times \frac{k}{1 + A(t-t_0) + B(t-t_0)^2}$$

where k is the scale factor, A $(*10^3)$ and B $(*10^6)$ are coefficients that may be obtained from *Decay* output and t_0 is the starting x-ray time in hours. The third type of *Scale* card, *Scale on ID*, applies the same scale factor to all reflections having ID numbers within the given range. There can be no more than 100 of these *Scale on ID* cards preceding any data block.

The starting default value for the scale factor is *1.0*. If a scale card is encountered, it becomes the default for all following data blocks until and unless another *Scale* card is encountered.

3)Background cards

The *Background* and *No Background* cards are used to specify whether or not the I and σ_I values, as obtained from *2Background*, are to be used in *Initia* calculations. These cards should be disregarded if you have not run the *Background* programs. The starting default is to use the corrected values if they appear on the *name.MAG* files. Thus, a card need be entered only if you wish to change this default and request that the original intensities (and their sigmas) be used instead. To do this, enter a *No Background* card. This then becomes the default for all further data blocks until and unless a *Background* card (requesting use of the corrected values) is encountered.

4)Absorption cards

The *Absorption* and *No Absorption* cards are used to specify whether or not the transmission coefficients be used in *Initia* calculations. These cards should be disregarded if you have not run the *Absorption* program. The starting default is to use the coefficients. Thus, a card is needed only if you wish to change this default and request that the coefficients be disregarded. To do this, enter a *No Absorption* card. This then becomes the default for all further data blocks until and unless an *Absorption* card (requesting use of the coefficients) is encountered.

5)F min card

This option is used to assign weights and F's of 0.0 to reflections either with F_{obs} less than a given constant or with F less than $\sigma_{F_{obs}}$ times a given constant. The normal procedure is to retain all reflections, unaltered; if the averaged value of F^2 is negative, the negative sign is transferred to F (as a signal only).

6)Disk card and Short Format Data cards

This card calls for input of data from the specified version of a *name.MAG* file. There **must** be a separate *Disk* card for each *name.MAG* file to be read. If no disk card is found, the program will presume that data will be entered as *Short Format Data* cards, which must be either entered by hand or copied into the instruction file after the pertinent data instructions.

Note: Since *Short Format Data* cards do not contain ID numbers or x-ray time, the *Delete*, *Restore*, *Scale on ID* and *Scale on Time* options do not apply.

Short Format Data cards and the *Disk* card may be interspersed, but remember to precede each data block with the appropriate data instructions.

Input files needed

name.INI

name.MAG (or *Short Format Data* cards)

Output

name.INT and/or *name.CRD*

A binary *name.INT* file, for use by *Prepare*, and/or a card-image file, *name.CRD*, that can be edited and used by *Prepare CRD*.

name.OUT

A listing of the averaging results, including statistical comparisons among averaged reflections;

A listing of all reflections deleted from the *name.INT* and/or *name.CRD* file.

name.BAD (Optional)

A listing of all reflections with high GOF values.

name.INF

GOF, R-merge values and the transmission coeffiecient will be added or updated.

Symmetry Code			Symmetry Code		
Constant	For	Observed	Constant	For	Observed
1	$h\ k\ 0$	$h = 2\mathrm{N}$	23	$h\ 0\ 0$	$h = 4\mathrm{N}$
2	$h\ 0\ l$	$h = 2\mathrm{N}$	24	$0\ k\ 0$	$k = 4\mathrm{N}$
3	$h\ k\ 0$	$k = 2\mathrm{N}$	25	$0\ 0\ l$	$l = 4\mathrm{N}$
4	$0\ k\ l$	$k = 2\mathrm{N}$	26	$h\ k\ 0$	$h + k = 4\mathrm{N}$
5	$h\ 0\ l$	$l = 2\mathrm{N}$	27	$h\ 0\ l$	$h + l = 4\mathrm{N}$
6	$0\ k\ l$	$l = 2\mathrm{N}$	28	$0\ k\ l$	$k + l = 4\mathrm{N}$
7	$h\ k\ 0$	$h + k = 2\mathrm{N}$	29	$h\ k\ l$	$h < k < l$ (reorder)
8	$h\ 0\ l$	$h + l = 2\mathrm{N}$	30	$h\ k\ l$	h=0 (projection)
9	$0\ k\ l$	$k + l = 2\mathrm{N}$	31	$h\ k\ l$	k=0
10	$h\ 0\ 0$	$h = 2\mathrm{N}$	32	$h\ k\ l$	l=0
11	$0\ k\ 0$	$k = 2\mathrm{N}$	33	$h\ h\ l$	$l = 2\mathrm{N}$
12	$0\ 0\ l$	$l = 2\mathrm{N}$	34	$h\ k\ h$	$k = 2\mathrm{N}$
13	$h\ k\ l$	$h + l = 2\mathrm{N}$	35	$h\ k\ k$	$h = 2\mathrm{N}$
14	$h\ k\ l$	$h + k = 2\mathrm{N}$	36	$h\ h\ l$	$2h + l = 4\mathrm{N}$
15	$h\ k\ l$	$k + l = 2\mathrm{N}$	37	$h\ k\ h$	$2h + k = 4\mathrm{N}$
16	$h\ k\ l$	$h = 2\mathrm{N}$	38	$h\ k\ k$	$2k + h = 4\mathrm{N}$
17	$h\ k\ l$	$k = 2\mathrm{N}$	39	$0\ 0\ l$	$l = 3\mathrm{N}$
18	$h\ k\ l$	$l = 2\mathrm{N}$	40	$0\ 0\ l$	$l = 6\mathrm{N}$
19	$h\ k\ l$	$h + k + l = 2\mathrm{N}$	41	$h\ k\ 0$	$h + k = 4\mathrm{N}$
20	$h\ k\ 0$	$h = 2\mathrm{N},\ k = 2\mathrm{N}$	42	$h\ 0\ l$	$h + l = 4\mathrm{N}$
21	$h\ 0\ l$	$h = 2\mathrm{N},\ l = 2\mathrm{N}$	43	$0\ k\ l$	$k + l = 4\mathrm{N}$
22	$0\ k\ l$	$k = 2\mathrm{N},\ l = 2\mathrm{N}$	44	$h\ \overline{h}\ l$	$l = 2\mathrm{n}$

No.	Point Group	Space Group	Equivalent Forms	No. of Equiv. Forms	Conditions on Parent Form
1	1	1	hkl	1	none
2*	$\bar{1}$	2	$hkl, \bar{h}\bar{k}\bar{l}$	2	2 indexes ≥ 0
3	2	3–5	$hkl, \bar{h}k\bar{l}$	2	$l \geq 0$
4	m	6–9	$hkl, h\bar{k}l$	2	$k \geq 0$
5*	$2/m$	10–15	$hkl, h\bar{k}l, \bar{h}\bar{k}\bar{l}, \bar{h}k\bar{l}$	4	$k, l \geq 0$
6	222	16–24	$kl, h\bar{k}\bar{l}, \bar{h}k\bar{l}, \bar{h}\bar{k}l$	4	$h, k \geq 0$
7	$mm2$	25–46	$hkl, h\bar{k}l, \bar{h}kl, \bar{h}\bar{k}l$	4	$h, k \geq 0$
8*	mmm	47–74	$7 + hk\bar{l}$, etc.	8	$h, k, l \geq 0$
9	4	75–80	$hkl, k\bar{h}l, \bar{h}\bar{k}l, \bar{k}hl$	4	$h, k \geq 0$
10	$\bar{4}$	81–82	$hkl, k\bar{h}\bar{l}, \bar{h}\bar{k}l, \bar{k}h\bar{l}$	4	$h, k \geq 0$
11*	$4/m$	83–88	$9 + hk\bar{l}$, etc.	8	$h, k, l \geq 0$
12	422	89–98	$9 + kh\bar{l}$, etc.	8	$h, k, l \geq 0$
13	$4mm$	99–110	$9 + khl$, etc.	8	$h \geq k \geq 0$
14	$\bar{4}2m$	111–114;121–122	$10 + khl$, etc.	8	$h \geq k \geq 0$
15	$\bar{4}m2$	115–120	$10 + kh\bar{l}$, etc.	8	$h, k, l \geq 0$
16*	$4/mmm$	123–142	$11 + khl$, etc.	16	$h, k, l \geq 0; h \geq k$
17	3	143–146	$hkil, kihl, ihkl$	3	$h \geq k, i$
18*	$\bar{3}$	147–148	$17 + \bar{h}\bar{k}\bar{i}\bar{l}$, etc.	6	$h \geq k, i; l \geq 0$
19	321	150;152;154–155	$17 + khi\bar{l}$, etc.	6	$h \geq k, i; l \geq 0$
20	312	149;151;153	$17 + \bar{k}\bar{h}\bar{i}\bar{l}$, etc.	6	$h \geq l, i; l \geq 0$
21	$3m1$	156;158;160–161	$17 + \bar{k}\bar{h}\bar{i}l$, etc.	6	$h, k \geq 0$
22	$31m$	157;159	$17 + khil$, etc.	6	$h \geq k \geq i$
23*	$\bar{3}m1$	164–167	$18 + \bar{k}\bar{h}\bar{i}l$, etc.	12	$h, k, l \geq 0$
24*	$\bar{3}1m$	162–163	$18 + khil$, etc.	12	$h \geq k \geq i; l \geq 0$
25	6	168–173	$hkil, kihl, ihkl, \bar{h}\bar{k}\bar{i}l, \bar{k}\bar{i}\bar{h}l, \bar{i}\bar{h}\bar{k}l$	6	$h, k \geq 0$
26	$\bar{6}$	174	$hkil, kihl, ihkl, hki\bar{l}, kih\bar{l}, ihk\bar{l}$	6	$h \geq k, i; l \geq 0$
27*	$6/m$	175–176	$25 + hki\bar{l}$, etc.	12	$h, k, l \geq 0$
28	622	177–182	$25 + khi\bar{l}$, etc.	12	$h, k, l \geq 0$
29	$6mm$	183–186	$25 + khil$, etc.	12	$h \geq k \geq 0$
30	$\bar{6}m2$	187–188	$26 + \bar{k}\bar{h}\bar{i}l$, etc.	12	$h, k, l \geq 0$
31	$\bar{6}2m$	189–190	$26 + khi\bar{l}$, etc.	12	$h \geq k \geq i; l \geq 0$
32*	$6/mmm$	191–194	$29 + hki\bar{l}$, etc.	24	$h \geq k \geq 0; l \geq 0$
33	23	195–199	$hkl, klh, lhk, h\bar{k}\bar{l}, k\bar{l}\bar{h}, l\bar{h}\bar{k},$ $\bar{h}k\bar{l}, \bar{k}l\bar{h}, \bar{l}h\bar{k}, \bar{h}\bar{k}l, \bar{k}\bar{l}h, \bar{l}\bar{h}k$	12	$h \geq k, l; h, k \geq 0$
34*	$m3$	200–206	$33 + \bar{h}\bar{k}\bar{l}$, etc.	24	$h \geq k, l; h, k, l \geq 0$
35	432	207–214	$33 + \bar{k}hl$, etc.	24	$h \geq k, l; h, k, l \geq 0$
36	$\bar{4}3m$	215–220	$33 + khl$, etc.	24	$h \geq k \geq l; h, k \geq 0$
37*	$m3m$	221–230	$36 + \bar{h}\bar{k}\bar{l}$, etc.	48	$h \geq k \geq l \geq 0$

The centrosymmetric point groups are indicated by an asterisk; these should be used unless anomalous dispersion is to be taken into account.

Data Disk Programs

Prepare INT
Prepare CRD

These programs create a binary *name.HKL* file of data which is required by most further programs. *Prepare INT* (or just *Prepare*) requires as input the binary *name.INT* file created by *Initia*. *Prepare CRD* requires *Short Format Data* cards as input.

Note: This program requires the file type *PRE*, *i.e.*, *name.PRE*.

Input Cards

1)Title card—Format(A4,4X,A3,1X,A5,3X,A4)
1–7 *PREPARE*
9–11 ⟨Blank or *INT*⟩/⟨*CRD*⟩ Input from ⟨*name.INT* file⟩/⟨*Short Format Data* cards⟩.
13–17 ⟨Blank⟩/⟨*PRINT*⟩ ⟨Limited⟩/⟨Full⟩ listing of reflections.
21–24 ⟨Blank⟩/⟨*CARD*⟩ Output to ⟨*name.HKL*⟩/⟨*name.CRD*⟩ file.

2)Cell card—Format(A4,2X,6F8.4)
1–4 *CELL*
7–14 a
15–22 b
23–30 c
31–38 α
39–46 β
47–54 γ

3)Kind F card—Format(A4,2X,12A3,6X,I2)
1–6 *KIND F*
7–9, 10–12, etc. Atom form factor labels.
49–50 Number of additional form factors to be entered. (*Optional.*)

3a)Trailer cards for additional atom form factors—Format(10F8.4/10F8.4/10F8.4/A3) *Optional*
1–10, 11–20, etc. 30 f-values at intervals of .05 in $\frac{sin\theta}{\lambda}$.
/
1–3 Atom form factor label.

4)Anomalous Dispersion card—Format(A4,6X,7(A3,F7.3))
1–9 *ANOMALOUS*
11–13, 21–23, etc. Atom form factor labels.
14–20, 24–30, etc. Real part, $f_j^{'}$, of the anomalous dispersion factors for atom j.

5)Short Format Data cards—Format(4X,3I6,F8.2,F8.5) *Prepare CRD* option only

5–10	h
11–16	k
17–22	ℓ
23–30	F_{obs}
31–38	CRYM weight ($= \sqrt{w}$)

6)End card—Format(A3) *Prepare CRD* option only

1–3	*END*

Description of individual cards

1)Title card

If input is from *Short Format Data* cards, use *Prepare CRD*. If input is from a *name.INT* file created by *Initia*, use *Prepare* (or *Prepare INT*).

For *Prepare INT*, the default is to list only the first and last five reflections in your output file. Entering "PRINT" in columns 13–17 results in the listing of all reflections.

Finally, if you want to create a *name.CRD* file rather than a *name.HKL* file, columns 21–24 must be nonblank.

2)Cell card

The *Cell* card requires unit cell dimensions a, b, c, α, β and γ. There are no defaults.

3)Kind F card

This card specifies the atom form factors that are to be used. The values of these form factors are written on the *name.HKL* file along with other data for each reflection. Most form factors are stored internally as the nine coefficients tabulated by Cromer (*Int. Tables*, **Vol. IV**, pp. 99–101). The stored atom form factor labels corresponding to these coefficients are given on the table on pages 8.11 and 8.12.

If additional atom types are wanted, they must be entered as interpolation tables on trailer cards. The procedure is as follows:

(1)Identify the atom types on the Kind F card; make sure that these additional labels do not match any of those internally stored (see table);

(2)Specify the number of additional atom types you are entering;

(3)For **each** additional atom type, enter 4 trailer cards (immediately after the *Kind F* card). The first 3 cards contain the values of f_j^0 at intervals of .05 in $\frac{sin\theta}{\lambda}$, beginning at 0. The fourth card contains the form factor label as specified on the *Kind F* card. The f_j^0 values for the individual reflections are obtained by cubic interpolation.

4)Anomalous Dispersion card

This card accommodates the real part, f'_j, of the anomalous dispersion factors for up to seven atom types. (The imaginary part, F''_j, is entered on the *Coord* card in *Structure Factors* or *Least Squares*.) The real part of the anomalous dispersion factor is stored internally for elements H through Cf for Mo $K\alpha$ radiation (values taken from *International Tables*, Vol. IV, pp. 149–151) and only the form factor label as on the *Kind F* card need be entered to recall these values. For other atoms or other types of radiation, the atom form factor labels are entered followed by the corresponding anomalous terms.

5)Short Format Data cards

These cards are **not** needed if input is from a *name.INT* file (*Prepare* or *Prepare INT*).

They contain the reflection indexes h, k, ℓ, the structure factor F_{obs} ($\sim \sqrt{intensity}$) and the CRYM weight. (The quantity called *weight* in this write-up and in the program is actually $\sqrt{w} = \sigma_{F^2}^{-1}$ rather than $\sigma_{F^2}^{-2}$.

Input files needed

name.PRE
name.INT (for *Prepare INT*) or *Short Format Data* cards (for *Prepare CRD*)

Output

name.HKL (or *name.CRD*)

A binary *name.HKL* (or formatted *name.CRD* file if requested) containing the reflections and data for each reflection.

name.OUT

A list of the reflections including h, k, ℓ, F_{obs}, $1/\sigma(F_{obs})^2$, $(sin^2\theta/\lambda^2)$ and the values of the form factors.

Re-Prepare

This program reads the *name.HKL* file and creates an updated *name.HKL* file. It is convenient for changing cell parameters, atom form factors or anomalous dispersion factors and for switching indexes.

Note: Your ***name.HKL*** **file must be renamed for use by** *Re-Prepare*. Type:

RENAME name.HKL OLDHKL.HKL

This gives your *name.HKL* file the **necessary** name *OLDHKL.HKL*.

Input cards

1)Title card—Format(A4,8X,A5,3X,A4)
1–10 *RE-PREPARE*
13–17 ⟨Blank⟩/⟨*PRINT*⟩ ⟨LIMITED⟩/⟨FULL⟩ listing of reflections.
21–24 ⟨Blank⟩/⟨*CARD*⟩ Output to ⟨*name.HKL*⟩/⟨*name.CRD*⟩ file.

2)Cell card (*Optional*)
See *Prepare* for description.

3)Switch card—Format(A4/9F5.0) *Optional*
1–6 *SWITCH*
/
1–5, 6–10, etc. The 9 elements A_1–A_9 of the transformation matrix.

4) and 4a)Kind F card and trailers (*Optional*)
See *Prepare* for description.

5)Anomalous card (*Optional*)
See *Prepare* for description.

6)End card—Format(A3)
1–3 *END*

Explanation of individual cards

1)Title card

The default of *Re-Prepare INT* is to list only the first and last five reflections in your output file. Entering “PRINT” in columns 13–17 results in the listing of all reflections.

If you want to create a *name.CRD* file rather than a *name.HKL* file, columns 21–24 must be nonblank.

2)Cell card, 4) and 4a)Kind F card and trailers and 5)Anomalous Dispersion card

These cards need only be entered in this program if you wish to change the corresponding information previously entered in *Prepare* and stored on the *name.HKL* file. To make a change, enter the appropriate card with the corrected information. This will then replace the information currently stored on the *name.HKL* file. If no changes are necessary, the card need not be entered.

Note: If you are entering any form factors by hand, i.e., if columns 49–50 on the *Kind F* card are nonblank, you **must** use the *Kind F* card and trailers.

3)Switch card

This option requires a trailer card containing the nine elements A_1–A_9 of a transformation matrix for the reflection indexes. All reflections will have their Miller indexes transformed as follows:

$$\begin{aligned} h_{new} &= h*A_1 + k*A_2 + l*A_3 \\ k_{new} &= h*A_4 + k*A_5 + l*A_6 \\ l_{new} &= h*A_7 + k*A_8 + l*A_9 \end{aligned} \quad \text{or} \quad \begin{pmatrix} h \\ k \\ l \end{pmatrix}_{new} = \begin{pmatrix} A_1 & A_2 & A_3 \\ A_4 & A_4 & A_6 \\ A_7 & A_8 & A_9 \end{pmatrix} \begin{pmatrix} h \\ k \\ l \end{pmatrix}$$

The coefficients, A_i, should be entered as real numbers. (*.333* is sufficient for $\frac{1}{3}$ and -*.333* is sufficient for $-\frac{1}{3}$.) If the new indexes are not integers, you will receive an error message. Five such errors results in termination of the program.

Note: No *End* card is needed. This program may be followed by another program or terminated with an *End X-ray* card.

Input files needed

name.REP
name.HKL **which has been renamed to** *OLDHKL.HKL*

Output

name.HKL or *name.CRD*

An unformatted *name.HKL* or formatted *name.CRD* file containing the reflections and data for each reflection. If the program has completed successfully, you should delete the *OLDHKL.HKL* file.

name.OUT

A list of the reflections including h, k, ℓ, F_{obs}, $1/\sigma(F_{obs})^2$, $(sin^2\theta/\lambda^2)$ and the values of the form factors.

List
List Phased Disk

These programs list the contents of the *name.HKL* file. *List Phased Disk* includes A_{calc} and B_{calc} for each reflection in the output. The *name.HKL* file is not altered.

Input cards

1)Title card—Format(A4,1X,A4)
1–4 *LIST*
6–16 ⟨Blank⟩/⟨*PHASED DISK*⟩ ⟨Do not⟩/⟨Do⟩ include A_{calc} and B_{calc} values in output.

Note: No *End* card is needed. This program may be followed by another program or terminated by an *End X-ray* card.

Explanation of individual cards

1)Title card

This card is used to request a listing of the *name.HKL* file, with or without A_{calc} and B_{calc} values. The *name.HKL* file will not be affected.

Input files needed

name.LIS
name.HKL

Output

name.OUT

A list of the reflections including h, k, ℓ, F_{obs}, $1/\sigma(F_{obs})^2$, the values of the scattering factors, and A_{calc} and B_{calc} (if requested).

Edit

This program operates on the *name.HKL* file. It is used to add, delete or change reflections. *Edit* does not list the contents of the *name.HKL* file; it merely tells whether or not the corrections were made successfully.

Input cards

1)Title card—Format(A4)
1–4 *EDIT*

2) and 2a)Kind F cards and Trailers—*Optional*
See *Prepare* for description.

3)Type card—Format(A3)
1–3 ⟨*ADD*⟩/⟨*DEL*⟩/⟨*SUB*⟩ Following group of *Short Format Data* cards should be ⟨added⟩/⟨deleted⟩/⟨substituted⟩ on the *name.HKL* file.

4)Short Format Data cards—Format(4X,3I6,F8.2,F8.5)
See *Prepare* for description.

5)End card—Format(A3)
1–3 *END*

Description of individual cards

2) and 2a)Kind F cards and Trailers

These cards are needed only if you are entering any form factors by hand, i.e., if columns 49–50 on the *Kind F* card are nonblank.

3) and 4)Type and Short Format Data cards

The *Type* cards are followed by the corresponding data on *Short Format Data* cards.

Data following *ADD* is one *Short Format Data* card per added reflection containing h, k, ℓ, F_{obs} and CRYM weight.

Data following *DEL* is one *Short Format Data* card per deleted reflection containing h, k, ℓ and, if wanted, F_{obs}. If the F_{obs} field is blank (**not 0.0**), the F_{obs} comparison will be neglected and the first encountered reflection having the given h, k and ℓ will be deleted. If an F_{obs} is entered, it must be within 0.1 of the F on the *name.HKL* file (i.e., the **scaled** F). Negative signs **must** be entered for negative F's! The DEL card permanently removes the reflection from the *name.HKL* file. Hence, a deleted reflection may only be replaced by use of the ADD command.

Data following *SUB* is one **pair** of *Short Format Data* cards per substitution. The first of the pair is the **old** h, k, ℓ and, if wanted, F_{obs}. If the F_{obs} field is blank (**not 0.0**), the F_{obs} comparison will be neglected and the first encountered reflection having the given h, k and ℓ will be substituted. If an F_{obs} is entered, it must be within 0.1 of the F on the *name.HKL* file. Negative signs **must** be entered for negative F's! The second of the pair is the **new** h, k, ℓ, F_{obs} and CRYM weight.

The *Type* and *Short Format Data* cards may be repeated for other types.

Note: For substitutions and deletions, the input program (called by typing *INPUT*) cannot distinguish between 0.0 and a blank field. Both will result in a blank F_{obs} field. If you need 0.0, your input file must be edited.

Input files needed

name.EDT
name.HKL

Output

name.HKL
A *name.HKL* file containing all reflections including the corrections.

name.OUT
A list of edit errors, if any.

Make Phased Disk

This program creates a *name.HKL* file from input data cards of h, k, ℓ, A_{calc}, B_{calc} and CRYM weight or from h, k, ℓ, phase angle and CRYM weight. The magnitude of F will be stored in both the F_{obs} and F_{calc} slots on the *name.HKL* file.

Input cards

1)Title card—Format(A4,16X,A4)
1–16 *MAKE PHASED DISK*
21–24 ⟨Blank⟩/⟨*CARD*⟩ Output to ⟨*name.HKL*⟩/⟨*name.CRD*⟩ file.

2)Cell card—Format(A4,2X,6F8.4)
2)Cell card
See *Prepare* for description.

3) and 3a)Kind F card and trailers
See *Prepare* for description.

4)Anomalous Dispersion card
See *Prepare* for description.

5)Data cards—use one of the following types:
a)Format(4X,3I6,2F8.3,8X,F8.5)

5–10	h
11–16	k
17–22	ℓ
23–30	A_{calc}
31–38	B_{calc}
47–54	CRYM weight $(= \sqrt{w})$

b)Format(4X,3I6,F8.3,8X,2F8.5)

5–10	h
11–16	k
17–22	ℓ
23–30	F
39–46	Phase angle
47–54	CRYM weight $(= \sqrt{w})$

6)End card—Format(A3)
1–3 *END*

Input files needed

name.MAK

Output

name.HKL or *name.CRD*

A binary *name.HKL* or formatted *name.CRD* file containing the reflections and data for each reflection.

name.OUT

A list of the reflections, including h, k, ℓ, F_{obs}, $1/\sigma(F_{obs})^2$, the values of the form factors and A_{calc} and B_{calc}.

All form factors must be left-justified!

Form	Sym-	Form	Sym-	Form	Sym-
factor	bol	factor	bol	factor	bol
$\mathrm{H}_{(SDS)}$	HS	V^{5+}	V5	Nb^{5+}	NB5
$\mathrm{H}_{(HF)}$	H	Cr	CR	Mo	MO
H^{-}	H1	Cr^{++}	CR2	Mo^{3+}	MO3
Li	LI	Cr^{3+}	CR3	Mo^{5+}	MO5
Li^{+}	LI1	Mn	MN	Mo^{6+}	MO6
Be	BE	Mn^{++}	MN2	Tc	TC
Be^{++}	BE2	Mn^{3+}	MN3	Ru	RU
B	B	Mn^{4+}	MN4	Ru^{3+}	RU3
C	C	Fe	FE	Ru^{4+}	RU4
C_{val}	CV	Fe^{++}	FE2	Rh	RH
N	N	Fe^{3+}	FE3	Rh^{3+}	RH3
O	O	Co	CO	Rh^{4+}	RH4
O^{-}	O1	Co^{++}	CO2	Pd	PD
F	F	Co^{3+}	CO3	Pd^{++}	PD2
F^{-}	F1	Ni	NI	Pd^{4+}	PD4
Na	NA	Ni^{++}	NI2	Ag	AG
Na^{+}	NA1	Ni^{3+}	NI3	Ag^{+}	AG1
Mg	MG	Cu	CU	Ag^{++}	AG2
Mg^{++}	MG2	Cu^{+}	CU1	Cd	CD
Al	AL	Cu^{++}	CU2	Cd^{++}	CD2
Al^{3+}	AL3	Zn	ZN	In	IN
Si	SI	Zn^{++}	ZN2	In^{3+}	IN3
Si^{4+}	SI4	Ga	GA	Sn	SN
P	P	Ga^{3+}	GA3	Sn^{++}	SN2
S	S	Ge	GE	Sn^{4+}	SN4
Cl	CL	Ge^{4+}	GE4	Sb	SB
Cl^{-}	CL1	As	AS	Sb^{3+}	SB3
K	K	Se	SE	Sb^{5+}	SB5
K^{+}	K1	Br	BR	Te	TE
Ca	CA	Br^{-}	BR1	I	I
Ca^{++}	CA2	Rb	RB	I^{-}	I1
Sc	SC	Rb^{+}	RB1	Cs	CS
Sc^{3+}	SC3	Sr	SR	Cs^{+}	CS1
Ti	TI	Sr^{++}	SR2	Ba	BA
Ti^{++}	TI2	Y	Y	Ba^{++}	BA2
Ti^{3+}	TI3	Y^{3+}	Y3	La	LA
Ti^{4+}	TI4	Zr	ZR	La^{3+}	LA3
V	V	Zr^{4+}	ZR4	Ce	CE
V^{++}	V2	Nb	NB	Ce^{3+}	CE3
V^{3+}	V3	Nb^{3+}	NB3	Ce^{4+}	CE4

Pr	PR	Hf	HF	Bi^{5+}	BI5
Pr^{3+}	PR3	Hf^{4+}	HF4	Po	PO
Pr^{4+}	PR4	Ta	TA	At	AT
Nd	ND	Ta^{5+}	TA5	Rn	RN
Nd^{3+}	ND3	W	W	Fr	FR
Pm	PM	W^{6+}	W6	Ra	RA
Pm^{3+}	PM3	Re	RE	Ra^{++}	RA2
Sm	SM	Os	OS	Ac	AC
Sm^{3+}	SM3	Os^{4+}	OS4	Ac^{3+}	AC3
Eu	EU	Ir	IR	Th	TH
Eu^{++}	EU2	Ir^{3+}	IR3	Th^{4+}	TH4
Eu^{3+}	EU3	Ir^{4+}	IR4	Pa	PA
Gd	GD	Pt	PT	U	U
Gd^{3+}	GD3	Pt^{++}	PT2	U^{3+}	U3
Tb	TB	Pt^{4+}	PT4	U^{4+}	U4
Tb^{3+}	TB3	Au	AU	U^{6+}	U6
Dy	DY	Au^{+}	AU1	Np	NP
Dy^{3+}	DY3	Au^{3+}	AU3	Np^{3+}	NP3
Ho	HO	Hg	HG	Np^{4+}	NP4
Ho^{3+}	HO3	Hg^{+}	HG1	Np^{6+}	NP6
Er	ER	Hg^{++}	HG2	Pu	PU
Er^{3+}	ER3	Tl	TL	Pu^{3+}	PU3
Tm	TM	Tl^{+}	TL1	Pu^{4+}	PU4
Tm^{3+}	TM3	Tl^{3+}	TL3	Pu^{6+}	PU6
Yb	YB	Pb	PB	Am	AM
Yb^{++}	YB2	Pb^{++}	PB2	Cm	CM
Yb^{3+}	YB3	Pb^{4+}	PB4	Bk	BK
Lu	LU	Bi	BI	Cf	CF
Lu^{3+}	LU3	Bi^{3+}	BI3		

*Coefficients for analytical approximation to the scattering factors (Table 2.2.b, *International Tables for X-ray Crystallography*, **Vol. IV**, pp. 99–101.)

Wilson Statistics

This program operates on the *name.HKL* file.

Unless otherwise directed (see *Title* card), the absolute scale and over-all temperature factors are calculated using the method of H. Levy *et al* (*Chemistry Division Annual Progress Report*, Oak Ridge National Laboratory, Sept., 1970, p. 135ff). This is a non-linear least squares procedure capable of refining the over-all temperature factor as a sphere or as an ellipsoid compatible with the *Laue* group. Initial estimates of the scale and overall β are obtained from a preliminary *Wilson* plot of the standard variety (see Stout & Jensen, pp. 188–195; Dunitz, pp. 153–156) or may be entered by the user. After refinement has terminated and the *name.HKL* file is updated, the program puts the F's on the absolute scale (by applying the overall scale factor), and calculates sharpening factors, E's and various intensity statistics (this is necessary for a sharpened *Patterson*).

Input cards

1)Title card—Format(A4,3X,A5,1X,A4,F8.4,1X,2F10.2)
1–6 *WILSON*
8–12 ⟨Blank⟩/⟨*PRINT*⟩ ⟨Limited⟩/⟨Full⟩ listing of reflections.
14–17 ⟨Blank⟩/⟨Non-blank⟩ ⟨Do not⟩/⟨Do⟩ create a *name.CRD* file of E-values.
18–25 Minimum E-value for reflections to be included in the *name.CRD* file. (The default is *1.2*.)
27–36, 37–46 Scale and $2B$. (*Optional*—not prompted by *Input* program.)

2)Numbers of Atoms card—Format(A4,12X,I4,12F4.0,3I2)
1–16 *NUMBERS OF ATOMS*
17–20 Space group number.
21–24, 25–28, etc. Numbers of atoms of each type in the unit cell (in the same order as on the *Kind F* card).
70 ⟨Blank or *0*⟩/⟨*1*⟩ ⟨Do⟩/⟨Do not⟩ update the *name.HKL* file.
72 Centering factor (see chart, page 9.2). (The default is chosen according to the space group number.)
73–74 ⟨Blank or *0*⟩/⟨Nonzero integer⟩ ⟨Do not⟩/⟨Do⟩ read in ϵ and variance arrays regardless of space group number.

2a)Epsilon and 2b)Variance cards—Format(13F4.0/14F4.0) *Optional*
1–4, 5–8, etc. ϵ array.
/
1–4, 5–8, etc. Variance array.

3)Plot or Init card—Format(A4,1X,2F10.2,1X,A4) Use one of the following:
a)Plot card
1–4 *PLOT*
6–15 $2\theta_{max}$ in degrees. (The default is *60°*.)
16–25 Wavelength in angstroms. (The default is *.71°*.)
27–30 ⟨Blank⟩/⟨*ANIS*⟩ ⟨Do not⟩/⟨Do⟩ continue with anisotropic refinement.

b)Init card
1–4 *INIT*
6–15 Initial value of scale factor.
16–25 Initial value of temperature factor.
27–30 ⟨Blank⟩/⟨*ANIS*⟩ ⟨Do not⟩/⟨Do⟩ continue with anisotropic refinement.

Description of individual cards

1)Title card

Entering "PRINT" in columns 8–12 on this card results in a listing of the Wilson scaling applied to each reflection. The default is to print only the first and last five reflections.

This card may be used to request that a *name.CRD* file of data cards in format (3I5,F10.5) be created. The cards consist of h, k, ℓ, and E-values. If you request this option, a minimum E-value may be entered such that reflections with E-values less than this value will not be included in the *name.CRD* file. The default is *1.2*.

If the scale and overall β are already known, you may choose to bypass all *Wilson* calculations and proceed directly to the scaling and intensity statistics calculations. Should this be desired, the scale and overall β are included on the *Title* card in columns 21–30 and 31–40 repectively. There are no default values. Nonzero entries in either of these fields signals bypass of the *Wilson* calculations. If using this option, no *Plot* or *Init* card (card 3) is entered. Because of the infrequency of the use of this option, the *Input* program does not recognize it. The constants must be edited in and card 3 must be deleted.

2)Numbers of Atoms card

This card contains the number of atoms of each type in the unit cell. The order must be the same as the order in which the atom form factor labels appeared on the *Kind F* card in *Prepare* (and hence on the *name.HKL* file).

You may request that the *name.HKL* file not be updated. The default is an updated file with the *Wilson* scale factor applied to all of the F's.

The space group number, as given in the *International Tables*, must also be included.

This card contains the centering factor:

= 1 if cell is *primitive*;

= 2 if cell is A-, C- or *I-centered*;

= 3 for trigonal groups which are listed in the *International Tables* with both hexagonal and rhombohedral axes. All CRYM programs require such groups to be based on the triply-primitive hexagonal cell;

= 4 if cell is *F-centered.*

The default is chosen according to the space group number.

The program will automatically construct correct ϵ and variance arrays for space groups 1–142 (triclinic to tetragonal) subject to the proviso that monoclinic groups must have the b-axis unique. For space groups greater than 142 (trigonal to cubic), the program will read two trailer cards, the first containing the ϵ array, the second the variance array. The order of ϵ on the card is hk0, h0ℓ, 0kℓ, h00, 0k0, 00ℓ, hhℓ, hh0, and, if cubic, hkh, h0h, hkk, 0kk, hhh. The order of the variance array is first hkℓ, and then hk0, h0ℓ, etc. (as above). Any ϵ or variance read as *0.0* will be set to *1.0.*

Note: The ϵ array begins with hk0, the variance array with hkℓ. Values of ϵ and variance can be obtained from the *International Tables*, **Vol. 2**, pp. 355–6. The variance is *1.0* for acentric (A) reflections and *2.0* for centric (C) reflections. ϵ is the quantity S/Φ, the last column in tables 7.2.1A and B. ϵ is set at *1.0* for the general hkℓ reflections in all space groups.

You may request to enter ϵ and variance arrays described above regardless of space group number. This option may be used to handle space groups of non-standard settings. Note, however, that rhombohedral space groups must have hexagonal axes.

3)Plot or Init card

The *Plot* card is used to request that initial estimates of the scale and overall β be made using a *Wilson* calculation. You need to enter the maximum 2θ in degrees and the wavelength in Å; the default values are *60*° and *.71*Å. The initial estimates of k and $2B$ are then refined isotropically until convergence or divergence occurs, up to a maximum of 6 cycles. Convergence is assumed when the absolute value of each parameter shift is less than 0.3 times its e.s.d. You may also request that the program continue with anisotropic refinement under the same limiting conditions as for isotropic with the additional requirement that the resulting $2B_{ij}$ matrix be positive definite.

An *Init* card is used to enter initial values of the scale and overall β, if they are known. Isotropic least squares refinement will proceed immediately, and you may request that the program then continue with anisotropic refinement as above.

Note: This program does not require an *End* card. It may be followed by another program or terminated with an *End X-ray* card.

Input files needed

name.WIL
name.HKL

Output

name.HKL

An updated *name.HKL* file (with the same version number as input) including the new F's and weights with the *Wilson* scale factor applied (if requested).

name.CRD

h, k, ℓ, and E-values for all reflections with E-values over the specified minimum (if requested).

name.OUT

A listing of all reflections including scaled F_{obs} and weight, E, E^2 and E^3 (if requested);

Various E statistics and theoretical values for centric, hypercentric and acentric distributions.

Structure Factors
Least Squares

The *Structure Factors* program calculates, from a set of atom coordinates and Gaussian amplitudes, values of F_{calc} for each reflection stored on the *name.HKL* file. The *Least Squares* program makes the same calculation, but also carries out a least squares adjustment of any desired parameters.

F_{calc} is calculated for any reflection (hkℓ) according to the expression:

$$F_{calc} = \sum_j f_j T_j p_j e^{2\pi i(hx_j + ky_j + lz_j)}$$

where the sum is over all atoms j having coordinates x_j, y_j, z_j, form factor f_j, Gaussian amplitude T_j (either isotropic or anisotropic) and population p_j. The least squares process minimizes the quantity $\sum w\Delta^2$ where w is the weight (typically $\sigma_{F^2_{obs}}^{-2}$) and $\Delta = F^2_{obs} - (\frac{F_{calc}}{k})^2$; k is the scale factor on F_{obs}. Since F_{calc} is a non-linear function of the parameters, several cycles of least squares refinement are usually needed to arrive at convergence (see Stout & Jenson, pp. 358–363). Note that least squares calculations are extensive, with single cycles often taking 30 minutes or more to run, so the input data should be carefully checked.

In least squares refinements, the parameters can all be put into one matrix or can be blocked into several matrices; alternatively, they can be designated for *No Refinement*, in which case their contributions are included in the F_{calc} calculation but are not adjusted. The coordinates of an atom can be adjusted by the least squares process with the B's or U's held fixed (or vice versa). If the contributions of some atoms are to be held fixed during successive least squares cycles (usually hydrogen atoms or solvent molecules), all parameters of these atoms—coordinates and B's or U's—can be included under *No Refinement Save*; these contributions will be stored in the special slots A_{sp} and B_{sp} where they can be *Use*d in subsequent cycles without being recalculated. Another option allows the inverse(s) of the least squares matrix(es) calculated during the first least squares cycle to be used during subsequent cycles in the same job, rather than requiring that they be recalculated; this saves the major portion of computing time during the subsequent cycles, at the expense of slower convergence. Non-refined atoms (such as H) with calculable geometries may be repositioned between cycles. The B value can be set to a fraction of that of the attached atom.

Note: There are no programs for cubic symmetry

For *Least Squares*, you should use the file type *LSQ*. This will result in the creation of a higher version of *name.LSQ* upon completion of the program containing your input data deck with the updated parameters. (The earlier version **will not** be destroyed.)

Input cards

1)Title card—use one of the following
a)Format(A4,14X,I3,A1,3X,A4)
1–17 *STRUCTURE FACTORS* (must be < 195)
19–21 Space group number given in *International Tables*, **Vol. I** or **Vol. A**.
22 Letter to indicate alternate representation of a few space groups. (Only 7A, 7N, 9A, 13A, 13N, 14A, 14N and 15A are accepted.)
26–29 ⟨Blank⟩/⟨*SAVE*⟩/⟨ *USE*⟩ ⟨No⟩/⟨Establish⟩/⟨Apply⟩ partial contributions. (Not available in Hexagonal or Cubic systems.)

b)Format(A4,14X,I3,A1,4X,A3)
1–13 *LEAST SQUARES*
19–21 Space group number given in *International Tables*, **Vol. I** or **Vol. A**.
22 Letter to indicate alternate representation of a few space groups.
27–29 ⟨Blank⟩/⟨*USE*⟩ ⟨Do not⟩/⟨Do⟩ apply partial contributions.

2)Weighting card—use one of the following
a)Format(A4,3X,A4,3X,A4,4X,F8.3)
1–13 *WEIGHT HUGHES*
15–21 ⟨Blank⟩/⟨*SQUARED*⟩ $\sqrt{w} = \langle\frac{1}{F_{obs}}\rangle/\langle\frac{1}{F^2_{obs}}\rangle$.
23–29 Value of $4F_{min}$.

b)Format(A4,3X,A4)
1–12 *WEIGHT UNITY*

c)Format(A4,3X,A4)
1–15 *WEIGHT EXTERNAL*

d)Format(A4,3X,6F6.5)
1–6 *WEIGHT*
8–13, 14–19, etc. P, Q, R, S, T and U from equation, page 10.6.

3)Garbage card—Format(A4,4X,F8.5,2X,I1,1X,2F6.4,1X,D5.0) *Optional*
1–7 *GARBAGE*
9–16 m such that reflections with $F^2_{obs} \leq m\sigma$ are considered separately.
19 ⟨Blank or *0*⟩/⟨*1*⟩ ⟨Omit⟩/⟨Include in separate sums⟩ reflections based on m cut-off.
21–26 Maximum $\frac{\sin^2\theta}{\lambda^2}$.
27–32 Minimum $\frac{\sin^2\theta}{\lambda^2}$.
34–38 Test diagonal matrix elements and reset to zero any that fall below the tolerance.

4)Scale card—Format(A4,2X,F10.7,1X,F3.1)
1–5 *SCALE*
7–16 Scale factor, applied to F_{obs}. (The default is to take the scale factor from the *name.HKL* file.)
18–20 Partial shift factor for refinement of scale factor. (*Optional.*)

5)Secondary Extinction card—Format(A4,17X,F8.5,1X,F5.3,1X,F10.7,1X,A3) *Optional*
1–20 *SECONDARY EXTINCTION*
22–29 Isotropic secondary extinction factor ($\times 10^6$).
31–35 Partial shift factor for the refinement of this parameter. (*Optional.*)
37–46 Wavelength in Å. (The default is *0.71* Å.)
48–50 ⟨Blank⟩/⟨*NEG*⟩ ⟨Do not⟩/⟨Do⟩ allow secondary extinction to be negative.

6)Accumulate A_{calc} and B_{calc} card—Format(A4) *Optional*
1–26 *ACCUMULATE ACALC AND BCALC*

7)Print, Do Not Print options
a)Do Not Print card—Format(A4,9X,F6.4)
1–12 *DO NOT PRINT*
14–19 Value of A such that only those reflections with $|\sqrt{w}\Delta| > A$ are listed. (*Optional.*)

b)Print card—Format(A4)
1–5 *PRINT*

8)Matrix cards and No Refinement card—use one of the following per block of parameters
a)Full Matrix card—Format(A4,8X,A4,14X,F3.1)
1–11 *FULL MATRIX*
13–20 ⟨Blank⟩/⟨*OVERRIDE*⟩ Allow maximum of ⟨300 parameters per matrix⟩/⟨1500 parameters in one matrix⟩.
31–33 Partial shift factor for every parameter in the matrix. (The default is *1.0*.)

b)Block or Diagonal Matrix card—Format(A4,26X,F3.1)
1–15 ⟨*BLOCK MATRIX*⟩/⟨*DIAGONAL MATRIX*⟩ to indicate matrix blocking scheme.
31–33 Partial shift factor for every parameter. (The default is *1.0*.)

c)No Refinement card—Format(A4,12X,A4)
1–13 *NO REFINEMENT*
17–20 ⟨Blank⟩/⟨*SAVE*⟩ ⟨Do not⟩/⟨Do⟩ save A_{calc} and B_{calc} for the following atoms in special slots on the *name.HKL* file for later use.

9)Coord cards—Format(A5,1X,A4,2X,A3,3F9.6,2X,F4.0,F6.2,6A2,F6.2,F7.5)
1–5 *COORD*
7–10 Atom identifier.
13–15 Atom form factor label.
16–24 x (in fractions of a cell edge).
25–33 y (in fractions of b cell edge).
34–42 z (in fractions of c cell edge).
45–48 Site multiplicity. (*Optional.*)
49–54 Partial shift factor for refinement of coordinates. (*Optional.*)
55–56 Shift constraints on x. (*Optional.*)
57–58 Shift constraints on y. (*Optional.*)
59–60 Shift constraints on z. (*Optional.*)
61–62 $\langle$Blank$\rangle/\langle NX\rangle$ $\langle$Normal$\rangle/\langle$No$\rangle$ refinement of x. (*Optional.*)
63–64 $\langle$Blank$\rangle/\langle NY\rangle$ $\langle$Normal$\rangle/\langle$No$\rangle$ refinement of y. (*Optional.*)
65–66 $\langle$Blank$\rangle/\langle NZ\rangle$ $\langle$Normal$\rangle/\langle$No$\rangle$ refinement of z. (*Optional.*)
67–72 f'' value from *International Tables*, **Vol. IV**, pp. 149–150. (*Optional.*)
73–79 Radial standard deviation of atomic position in Å. (*Optional.*)

10)Iso, Iso-An and Aniso cards—use one of the following per atom
a)Iso and Iso-An cards—Format(2A3,A4,2X,F12.8,47X,F4.2,4I1)
1–3 *ISO*
4–6 $\langle$Blank$\rangle/\langle$*-AN*$\rangle$ $\langle$Isotropic$\rangle/\langle$Convert iso to anisotropic$\rangle$.
7–10 Atom identifier.
13–24 Isotropic Gaussian amplitude (B).
73–76 Partial shift factor for refinement of this parameter. (*Optional.*)
77–80 W, X, Y and Z—special symmetry restriction parameters (see chart, page 10.9). (*Optional.*)

b)Aniso cards—Format(A5,A1,A4,2X,6F10.9,1X,F3.2,I4)
1–5 *ANISO*
6 $\langle$Blank or $B\rangle/\langle U\rangle$ $\langle\beta_{ij}\rangle/\langle U_{ij}\rangle$ elements entered.
7–10 Atom identifier.
13–22 β_{11} or U_{11}.
23–32 β_{22} or U_{22}.
33–42 β_{33} or U_{33}.
43–52 β_{12} or U_{12}.
53–62 β_{13} or U_{13}.
63–72 β_{23} or U_{23}.
74–76 Partial shift factor for refinement of these parameters. (*Optional.*)
77–80 W, X, Y and Z—special symmetry restriction parameters (see chart, page 10.9). (*Optional.*)

11)Pop cards—*Optional*
a)Format(A3,8X,F6.3,F4.2,3X,12A4)
1–3 *POP*
12–17 Partial population, p, of reference atom.
18–21 Shift factor for refinement of this parameter. (*Optional.*)
25–28, 33–36,..., 65–68 Atom identifiers where Pop= p.
29–32, 37–40,..., 69–72 Atom identifiers where Pop= $1.0 - p$.

Continuation card for type A Pop card—Format(24X,12A4) *Optional*
25–28, 33–36,..., 65–68 Atom identifiers where Pop= p.
29–32, 37–40,..., 69–72 Atom identifiers where Pop= $1.0 - p$.

b)Format(A3,3X,A4,1X,F6.3,F4.2,3X,A3)
1–3 *POP*
7–10 Atom identifier.
12–17 Partial population of first atom type.
18–21 Shift factor for refinement of this parameter. (*Optional.*)
25–27 Atom form factor label of second atom type.

12)End Atomic Information card—Format(2A4)
1–22 *END ATOMIC INFORMATION*

13)Type card(s)—Format(3X,I,4A4,2F10.5,6X,A4,X,A3,X,F10.6)
4 Ntyp (see chart, page 13.4).
5–8 Atom identifier of Atom 1, the atom to which the generated atom(s) is attached.
9–12, 13–16, 17–20 Atom identifiers of adjacent atoms.
21–30 Distance from Atom 1 to generated atom(s) in Å. (The default is 0.95 Å.)
31–40 Angle between two generated atoms in degrees for *Ntyp*s 2 (the default is 120°), 4, 5 and 6 (the default is 109.5°).
47–50, 51–54, 55–58 Atom identifier(s) to be assigned to generated *Coord* cards. (*Optional.*)
60–62 Atom form factor label to assign to generated *Coord* cars. (The default is *HS*.)
64–73 Multiplier for B value of generated atom relative to attached atom.

13a)Iso card(s)—Format(A3,9X,F10.5)
1–3 *ISO*
13–22 B.

14)Recycle card—Format(A4,4X,A4,9X,A3) *Optional* (*Least Squares* only)
1–7 *RECYCLE*
9–20 ⟨Blank⟩/⟨*SAME INVERSE*⟩ ⟨Create new matrix and inverse⟩/⟨Use previous matrix inverse) for next least squares cycle.
22–24 ⟨Blank⟩/⟨*USE*⟩ ⟨Ignore⟩/⟨Use⟩ A and B values stored in special slots on *name.-HKL* file. (Only valid if previously *save*d on a *No Refinement* card.)

15)End Least Squares card—Format(2A4,10X,A4) (*Least Squares* only)
1–17 *END LEAST SQUARES*
19–22 ⟨Blank⟩/⟨*LIST*⟩ ⟨Do not⟩/⟨Do⟩ create a *name.TAB* file of final coordinates and Gaussian amplitudes.

Description of individual cards

1)Title card

The *Title* card is used to initiate either the *Structure Factor* or *Least Squares* calculation. You may specify whether to apply or, for *Structure Factors*, to establish partial contributions. The card also contains the space group number as given in *International Tables*, **Vol. I** or **Vol A**, and a letter to specify an alternate choice of axes—only 7A, 7N, 9A, 13A, 13N, 14A, 14N and 15A are accepted.

2)Weighting card

The *Hughes* weighting scheme [*J. Am. Chem. Soc.*, **63**, 1737 (1941)] is used to give the appropriate set of weights for an F^2 refinement corresponding to *unit* weight in an F refinement. In this scheme,

$$\sqrt{w} = 1/F_{obs} \text{ for } F_{obs} \geq 4F_{min}$$
$$\sqrt{w} = 1/\sqrt{4F_{obs}F_{min}} \text{ for } F_{obs} < 4F_{min}$$

(see Stout & Jenson, p. 435). The *Weight Unity* card assigns weights of *1.0* to all reflections. The *Weight External* card uses weights from the *name.HKL* file, i.e., weights derived from counting statistics. The fourth weight option generates a weight w by the relation:

$$\sqrt{w} = \frac{P + Q\frac{\sin^2\theta}{\lambda^2}}{R + S \times F_{obs} + T \times F_{obs}^2 + U \times F_{obs}^3}.$$

3)Garbage card

The *Garbage* card is needed if the sphere of data should be limited, the matrix test value changed, or data rejected by the σ test. The matrix test value, *TEST*, is a double precision number used in the tolerance test of diagonal matrix elements. In some cases, parameters—usually β_{ij}—may be unrefinable because of the symmetry of the atom site. The matrix elements for such parameters, which should be identically zero, may be non-zero because of computer round-off. Accordingly, a tolerance test is built in, such that diagonal matrix elements below this tolerance will be set to zero. If this tolerance—normally set at 1×10^{-1}—is too large, the matrix element of a legitimate parameter may be zeroed, and that parameter will not be shifted. (This is most likely to happen when the scale factor is far from 1.0 and, hence, the weights are poorly scaled.) In such cases, lowering the tolerance will allow these legitimate parameters to survive the test, and to be adjusted. If any matrix element is less than *TEST*, the element will be set to *1.0* with normalizing constant *0.0*. The default value for *TEST* is *1.D−1*. The leading integer may be zero, and the final integer must be right justified. Examples include: *1.D+0*, *1.D−3* or *0.D+0*.

The "σ rejection test" is used to either reject some reflections from the refinement completely, or simply to calculate sums and agreement indices based on a subset of the reflections in addition to the sums and indices for all reflections. A value m is entered to select the reflection set and a flag is set to indicate if the selected reflections are to be ignored in the refinement or if separate sums are to be kept.

m	Col. 19	Result
3.0	0	Only data with $F^2 > 3\sigma(F^2)$ are used in the refinement.
-2.0	0	Only data with $F^2 < 2\sigma(F^2)$ are used in the refinement.
2.5	1	All data used; two sets of sums printed, one for all data and one for data with $F^2 < 2.5\sigma(F^2)$.
-3.0	1	All data used; two sets of sums printed, one for all data and one for data with $F^2 > 3.0\sigma(F^2)$.

If you want to use this card for a later cycle of *Least Squares*, place it immediately before the appropriate *Recycle* card.

4)Scale card

You must have a *Scale* card! For *Structure Factors*, or for *Least Squares* when you do **not** want to refine the scale factor, the *Scale* card must follow the *Weight* card (or *Garbage* card, if used). If you want to refine the scale factor (in *Least Squares*), place the card after the appropriate *Matrix* card.

5)Secondary Extinction card

This card adds the refinable parameter g to the calculation of F:

$$\left(F^*_{calc}\right)^2 = \frac{F^2_{calc}}{1 + g\beta F^2_{calc}}.$$

[Equation 3, Larson, *Acta Cryst.*, **23**, p. 664 (1967)], where β, assuming no absorption, is $\frac{1+\cos^4 2\theta}{\sin 2\theta(1+\cos^2 2\theta)}$ and a typical value for organic crystals is $g \sim 10 \times 10^{-6}$. To allow the secondary extinction factor to be negative, write NEG in columns 48–50.

To refine this parameter, include this card after the appropriate *Matrix* card. Otherwise, it must come immediately before the first *Matrix* card.

6)Accumulate A$_{calc}$ and B$_{calc}$ card

This option causes the values of A$_{calc}$ and B$_{calc}$ from separate structure factor runs to be accumulated in the special slots A_{sp} and B_{sp} on the *name.HKL* file for each reflection. It is normally used only for very large structures, such as proteins, for which all atoms cannot be included in one calculation.

7)Print, Do Not Print cards

These cards control the amount of output to the *name.OUT* file. The starting default is to print all structure factors after each cycle. To change this, use a *Do Not Print* card before the data deck or the appropriate *Recycle* card. No cycles will be printed from that point on until and unless a *Print* card is encountered before a later *Recycle* card. (Since *Print* is the default, it is not accepted before the data deck.)

To request the printing of only those structure factors whose $|\sqrt{w}\Delta|$ values are greater than a certain cut-off, use a *Do Not Print* card with the appropriate cut-off value in columns 14–19.

These card can be in several different places in the deck.

8)Matrix and No Refinement cards

The *Matrix* cards indicate the extent of blocking. A general shift factor, applying to all parameters within the matrix, may be given; blank is interpreted as *1.0*, i.e. no damping. The *Full Matrix* card places all parameters in the next data block in the same matrix. If the *Override* option is used, the storage allotment is increased so that 1500 parameters will be allowed in one matrix rather than 300 parameters per matrix. The *Block Matrix* card puts the parameters on each card (*Coord*, *Iso*, etc.) in a separate matrix and the *Diagonal Matrix* card puts each individual parameter in a separate matrix.

The *No Refinement* card requests that the parameters in the next data block not be refined. If the *Save* option is used, A and B will be calculated during the first cycle and stored in the special slots A_{sp} and B_{sp}. They may be recalled by the *Use* option on the *Recycle* card, or on a subsequent *Structure Factors* or *Least Squares* card, so that the contributions of these atoms need not be re-calculated each time. If any parameter of an atom is in a *No Refinement Save* block, then all parameters for that atom must be in that block.

9)Coord cards

Fractional atom coordinates, shift constraints, multiplicity and incorporation of f'' (the imaginary component of the anomalous dispersion factor) into the F_{calc} are entered on these cards.

The multiplicity specifies the number of atoms in the unit cell; it will generally coincide with the *multiplicity* of the atom site as given in the *International Tables*. The default is the maximum multiplicity for the particular space group—that is, the number of general positions. **Note:** This is an F4.0 field, so that fractional occupancy of an atom site can be designated here (by entering the product of the normal multiplicity and the occupation factor); however, if the occupancy is to be refined, a *Pop* card must be used.

The shift factor overrides that given on the *Matrix* card. If the *Coord* shift factor is *0.0* or blank, the *Matrix* shift is applied.

The *shift* and *refine* fields, columns 55–66, are used to indicate constraints on any of the parameters as imposed by the site symmetry of the atom—or, in a polar space group, to specify the origin by fixing one or more coordinates of one atom. If any coordinate is to be held fixed during the refinement, enter *NX*, *NY* or *NZ* in the appropriate columns of **both** the shift and refine fields; for example, to hold the y parameter fixed, enter *NY* in columns 57–58 **and** in columns 63–64. If, on the other hand, the site symmetry requires that one coordinate of an atom be related to another coordinate of that same atom (this occurs only in space groups of tetragonal or higher symmetry), the shift and refine fields will contain different entries. For example, if the atom lies in a position such that $y = -x$, enter $-X$ in columns 57–58 and *NY* in columns 63–64, indicating that the y coordinate should not be included in the least squares matrix (the matrix term involving x is adjusted accordingly), but that its shift should be set equal to $-\Delta x$. In hexagonal space groups, atoms lying in special positions such as (x,$2x$,z) must be transposed to an equivalent positions such as (x,$-x$,z) where the shift on y can be constrained to either $+x$ or $-x$. In addition, if $y = -x$, enter a shift factor of 0.577 ($1/\sqrt{3}$); if $y = +x$ no factor is needed. (This will slow down the convergence of z, but will prevent x and y from possibly diverging.)

The f'' value (from *International Tables*, **Vol. IV**, pp. 149–150) may be entered. The radial standard deviation (in Å) is also optional. After a cycle of least squares, it is calculated as $(\sigma(x)+\sigma(y)+\sigma(z))/3$ and stored on the *name.CRD* and *name.LSQ* files. For a pair of atoms i and j with radial esd.'s of σ_i and σ_j, the esd of the distance between the atoms as calculated in *Distan* is simply $\sigma_d = \sqrt{\sigma_i^2 + \sigma_j^2}$. Note that this estimate assumes isotropic errors in the coordinates, that the atom positions are uncorrelated, and that the errors in cell dimensions are negligible. (See Dunitz, pp. 261–265.) A minus sign preceding σ indicates that at least one of the coordinates of that atom is fixed.

10)Iso, Iso-An and Aniso cards

The isotropic Gaussian amplitudes, in the form of B's, are entered on the *Iso* and *Iso-An* cards. The partial shift factor is applied during a least squares cycle if it is different from the general shift factor on the *Matrix* card. The *Iso-An* card converts a single isotropic B to six anisotropic coefficients β_{ij} before building the matrix. Constraints on the anisotropic terms are indicated in columns 77–80.

The *Aniso* card is used to enter anisotropic Gaussian parameters. Either β_{ij}:

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

or U_{ij}:

$$\exp(2\pi^2)(-U_{11}h^2a^{*^2} - U_{22}k^2b^{*^2} - U_{33}\ell^2c^{*^2} - 2U_{12}hka^*b^* - 2U_{13}h\ell a^*c^* - 2U_{23}k\ell b^*c^*)$$

elements may be entered, although refinement is based on the β's–the algebraic coefficients multiplying h^2, k^2, $h\ell$, etc. You may also request special symmetry restrictions in the *WXYZ* slots, columns 77–80. To request that the matrix element *12*, *13* or *23* not be shifted or refined, leave *W* (column 77) blank and put a *1* in the *X*, *Y* or *Z* column respectively. For tetragonal and hexagonal symmetries, *W* nonblank requests special restrictions according to the following charts:

Tetragonal

W	*X*	*Y*	*Z*	FOR
				(numbers refer to matrix elements)
2			0	$11 = 22, 23 = +13$
2			2	$11 = 22, 23 = -13$
2	1	1	1	$11 = 22, 12 = 13 = 23 = 0.0$

Hexagonal

W	*X*	*Y*	*Z*	FOR
				(numbers refer to matrix elements)
2			0	$11 = 22, 23 = +13$
2			2	$11 = 22, 23 = -13$
2		1	1	$11 = 22, 13 = 23 = 0.0$
3				$11 = 12 = 22, 13 = 23 = 0.0$
4		2		$12 = 22, 13 = 23 \times 0.5$
4			1	$12 = 22, 23 = 0.0$
5		5		$11 = 12, 23 = 13 \times 0.5$
5	1			$11 = 12, 13 = 0.0$

For help in determining the appropriate constraints, see *Acta Cryst.*, **Vol. IX**, p. 359, 1956.

11)Pop cards

There are two types of cards for specifying partial populations. The first type is used if an atom does not fully occupy a site, or is divided between two sites; the second type is used if a single site is occupied by two different atom types.

On the first type of card, up to six different atoms (each with the same population factor p) may be entered; if desired, up to six paired atoms occupying alternative sites (each with population $1.0 - p$) can also be entered. If additional atoms need to be constrained to the same population factor, up to five trailer cards with up to six atoms (or atom pairs) each are accepted. Up to 10 of these cards (each with their corresponding trailers) are allowed.

The second type of *Pop* card calls for a fractional content p of the atom type specified on the *Coord* card and content $1 - p$ of the atom type specified in columns 25–27. Up to 10 of these cards are allowed.

Both types of population parameters may be included within a least squares matrix, and all atoms designated on the cards will contribute to the derivative terms. Note, though, that an atom designated on a *Pop* card which is contained within a least squares matrix cannot be among the *No Refinement Save* atoms; if one parameter card for an atom is in the *No Refinement Save* parameter block, all parameters for that atom must be.

Population factors can also be entered by means of the *Multiplicity* field on the *Coord* cards; however, these numbers are not refined.

12)End Atomic Information card

This card must follow the last atom parameter card.

13)Type card.

These cards are used to reposition non-returnable atoms. They may be easily generated by Hydrogen position input. The isotropic displacement parameter for the atom is based on that of the attached atom; a multiplier may be input; default is 1.0.

13a)Alternatively, an *ISO* card may be included. 14)Recycle card

This card requests that another cycle of *Least Squares* be performed. One cycle will always take place. An additional cycle will occur for each *Recycle* card used.

The *Same Inverse* option requests that the matrix inverse(s) from the previous least squares run be used in the next cycle.

The *Use* option requests that the A and B values stored in the special slots on the *name.HKL* file (available only if the *Save* option was used previously) be used in the next least squares cycle.

Note: Cards 3, 6 and 7 may be used after card 12 before the appropriate *Recycle* card.

15)End Least Squares card

Besides terminating a least squares run, this card may also be used to request the creation of a *name.TAB* file containing a list of final coordinates, β's, U's and σ's. The *name.TAB* file is required by *Clist*.

Input files needed

name.STR or *name.LSQ*
name.HKL

Output

name.OUT

h, k, ℓ, kF_{obs} (k = scale factor), F_{calc}, $\Delta F(= kF_{obs} - F_{calc})^*$, $\sqrt{w}\Delta(= \sqrt{w}[(kF_{obs})^2 - F^2_{calc}])$, A_{calc} and B_{calc} for each reflection for all cycles (unless a *Do Not Print* card requests otherwise);

Various sums, weighted and not.

name.TAB (if requested, *Least Squares* only)

A list of final coordinates, Gaussian amplitudes and esd's. This file is required by *Clist.*

name.CRD (*Least Squares* only)

A new data deck (from the first *Matrix* card through the *End Atomic Information* card) of the refined parameters from the final least squares cycle, in the same order as the original input. Parameters from other cycles will also be included if the *Punch* option was used.

name.LSQ (*Least Squares* only)

An updated version of your entire input data deck containing the parameters from the last least squares cycles.

name.INF

Any of the following items which are affected by the program will added or updated:
Number of independent reflections
Number of $F^2_{obs} > 0$
R-index
Final Goodness-of-Fit for *nnn* reflections and *nnn* parameters
Number of F^2_{obs} < or > *nnn* with a sigma of *nnn*
R for F^2_{obs} < or > *nnn* with a sigma of *nnn*

[*] $\Delta F = 0$ if $F^2_{obs} \leq 0$.

Fourier
Difference Fourier
Patterson

These programs calculate Patterson, Fourier or Difference Fourier maps in each crystallographic space group, using standard settings. The maps may cover any portion of the unit cell, in fractional coordinates or in inch/Å scale. Three-dimensional maps, projections, single sections and general plane sections can be calculated. The structure factors from the *name.HKL* file are used. Provision is made to edit the input data according to various limiting criteria, to perform Patterson sharpening, to generate difference coefficients between various data sets or to make maps using calculated structure factors or phases. Occasionally, spikes of unknown origin will occur in these maps. Also, peaks may sometimes be absent in Patterson map.

The previous chapter describes the calculation of the structure factors for a given electron distribution described by a set of atom coordinates and other factors. Fourier or difference Fourier maps perform the inverse operation, namely the generation of electron density maps from the set of structure factors (Stout & Jensen, pp. 196–241, 279–300). The seminal relation is

$$\rho(x,y,z) = \frac{1}{V}\sum_h\sum_k\sum_\ell C_{hk\ell}e^{-2\pi i(hx+ky+\ell z)}$$

where $C_{hk\ell}$ is the ***phased*** coefficient of the Fourier series, i.e.,

$$C_{hk\ell} = |C_{hk\ell}|e^{i\alpha_{hk\ell}}$$

with magnitude $|C_{hk\ell}|$ and phase angle $\alpha_{hk\ell} = \tan^{-1}\left(\frac{B_{hk\ell}}{A_{hk\ell}}\right)$;

$$A_{hk\ell} \equiv \sum_j f_j \cos 2\pi(hx_j + ky_j + \ell z_j)$$

and

$$B_{hk\ell} \equiv \sum_j f_j \sin 2\pi(hx_j + ky_j + \ell z_j).$$

To incorporate anomalous dispersion, $f_j = f_j^o + f_j' + if_j''$.

In the Fourier mode, the coefficients C are kF_{obs} (k = scale factor on F). For a centrosymmetric structure, $B = 0$ and the phase angles are either 0 or π, taken from A_{calc}. For a non–centrosymmetric structure, the full calculation outlined above is carried out,

$$A = kF_{obs}\frac{A_{calc}}{|F_{calc}|};\quad B = kF_{obs}\frac{B_{calc}}{|F_{calc}|}.$$

In the difference Fourier mode, the coefficients C are $|F_{obs}| - |F_{calc}|$. Specifically, A_{calc} and B_{calc} are generated:

$$A = kF_{obs}\frac{A_{calc}}{|F_{calc}|} - A_{calc};\quad B = kF_{obs}\frac{B_{calc}}{|F_{calc}|} - B_{calc}.$$

In the *Patterson* mode, the Patterson space group corresponding to the primitive or centered Laue space group is taken from an internally stored table (see Stout & Jensen, p. 283). The coefficients are $(k \times F_{obs})^2$. The Patterson can be "sharpened" and the origin can be "removed" by use of the *Sharpen* option (see Stout & Jensen, pp. 281–283).

Scaling:

Two scale factors are used in this program. The scale factor referred to above as k is used to place observed data on an absolute (F_{calc}) scale. When k has been properly entered, the program makes use of the unit cell volume to calculate all densities in electrons per Å^3 (electrons per Å^2 for projections). The program then multiplies the output by the reciprocal of a *point scale* (normally *0.01*—see *Point Scale* option) to give values suitable for printing as integers.

Editing:

By use of the *Limits* option, the program can be set to reject all reflections whose indices, $\frac{\sin^2\theta}{\lambda^2}$ or $\sigma_{F_{obs}}$ exceed certain values, or whose F_{obs} is less than a certain value. Reflections with zero weight can also be eliminated from *Fourier* and *Patterson* maps; they are automatically eliminated from *Difference Fourier* maps.

In the *Fourier* and *Difference Fourier* modes, the program can be set to reject all reflections for which F_{calc} does **not** exceed the expression:

$$F_{calc}(min) = C_1 + C_2 F_{obs} + C_3 \frac{\sin^2\theta}{\lambda^2}$$

by use of the *Phase Cutoff Function* option.

Unobserved Reflections (*less thans*):

Any reflection entered with a negative weight is taken to be unobserved. The F_{obs} entered is taken to be the minimum observable F, so an F of -2.1 reads "less than 2.1" (a carry-over from the days of visual intensity measurements). Such reflections are treated as follows:

Patterson: Enter $\frac{F_{min}}{2}$;
Fourier: Reject reflection;
Difference Fourier: Include if $F_{calc} > F_{obs}$.

Asymmetric Unit of Data:

All input reflections may be tested for conformity to the asymmetric units as defined in *Initia* (page 7.9). This test is mainly intended to prevent the user from accidentally including different reflections of the same form, which would be added together. Since the Fourier summation will work correctly with any indices, an *Omit Asymmetric Unit Test* option is provided.

Printed Output:

Output is arranged with two map points per inch horizontally and three map points per inch vertically. The vertical interval may be changed by use of the *Square Grid* option.

There is a maximum of 26 columns per page. If the map is too large to fit on one page, continuation pages are produced. A limit of 156 columns (6 pages) is imposed to protect the user from input errors. Points are printed as four-digit integers; larger values are printed as *9999*.

Subprogram Deck:

Input to this program consists of *control cards* and a *map starting card*. *Control cards* may be in any order, but each card can be used only once. There is a default for every option which is not explicitly specified (see descriptions below). Most options are read in *free format*. The first 4 columns of the card identify the option. Up to 8 columns may be used by each variable. Two successive blank columns terminate the reading of any card.

Control cards

Title card—Format(A6,13X,I3,A1)
1–18 ⟨*PATTERSON*⟩/⟨*FOURIER*⟩/⟨*DIFFERENCE FOURIER*⟩
20–22 Space group number. (Taken from name.INF file, if available.)
23 Letter for special representation of a few space groups (only 5A, 7A, 7N, 8A, 9A, 12A, 13A, 13N, 14A, 14N, 15A, 65A are allowed).

Cell card—Format(A4,2X,6F8.4)
1–4 *CELL*
7–14 a
15–22 b
23–30 c
31–38 α
39–46 β
47–54 γ
Default: If this card is not used, cell parameters will be taken from your *name.HKL* file.

SCALE FACTOR = scale
Action: This card is used to enter the scale factor (k) by which F_{obs} is to be multiplied (to be put on the absolute scale, normally).
Default: If this option is not used, the scale factor will be taken from your *name.HKL* file, assuming one is there. (Only *Structure Factors* or *Least Squares* places a scale factor on the *name.HKL* file, so normally, *Patterson* will not be able to make use of this default.)

LIMITS [*HMAX* = h_{max}], [*KMAX* = k_{max}], [*LMAX* = l_{max}], [*RHOMAX* = ρ_{max}],
[*FMIN* = F_{min}], [*SIGMAX* = σ_{max}], [*DELETE ZERO WEIGHT*]
Format: *LIMITS* followed by any combination of the above options in any order. A continuation card may be used if needed; in this case, the last nonblank character on the preceding card must be a comma and the first five columns on the continuation card must be blank.
Action: The meanings of the various options are as follows:

HMAX = h_{max}, *KMAX* = k_{max}, *LMAX* = l_{max}: Reflections whose indices exceed the given values are omitted.
RHOMAX = ρ_{max}: Reflections whose $\frac{\sin^2\theta}{\lambda^2}$ value exceed the given value ρ_{max} are omitted.
FMIN = F_{min}: Reflections for which F_{obs} is less than the given F_{min} are omitted.
SIGMAX = σ_{max}: Reflections for which the standard deviation of F_{obs} is larger than the given value σ_{max} are omitted.
DELETE ZERO WEIGHT: Reflections with zero weight are omitted. **Note:** Zero weight reflections are always omitted from *Difference Fourier*.
Defaults: All tests are omitted.
Example:

```
LIMITS HMAX = 10, LMAX = 12, RHOMAX = 0.04, FMIN = 0.01,
     DELETE ZERO WEIGHT
```

LIST REJECTS
Action: The indices of all reflections omitted from the Fourier calculation for any reason (except due to elimination by *RHOMAX*) will be printed in the output.
Default: Any reflection omitted because of the asymmetric unit test, by a centering test, or by *HMAX*, *KMAX* or *LMAX* is not listed.

OMIT ASYMMETRIC UNIT TEST
Action: Testing for the correct asymmetric unit is eliminated if this option is used. See discussion on *Asymmetric Unit*, page 11.2.

`PHASE CUTOFF FUNCTION, FCMIN =` m + n**FOBS* + p**RHO*
Action: The program will omit any reflections for which F_{calc} does not exceed the value of *FCMIN* calculated using the F_{obs} and ρ for that reflection according to the given equation.
Note: This card is not permitted in *Patterson*.
Example:

```
PHASE CUTOFF FUNCTION, FCMIN = 0.01 + 1.0*FOBS + 0.0*RHO
```

POINT SCALE value
Action: *Value* is a real number ($e^-/Å^3$). The Fourier coefficients are multiplied by the reciprocal of this value; hence, if the output points are multiplied by the specified point scale, they become $e^-/Å^3$.
Default: *0.01* for *Difference Fourier*, *0.01* for *Fourier* and *1.0* for *Patterson*.

SHARPEN [*REMOVE*[= r]]
Action: This card may be used in *Patterson* only and the program *Wilson* **must** have been run previously. It requests sharpening of a Patterson map. The *Remove* option specifies origin removal. A numerical value can be given to specify partial removal of a fraction r of the origin peak ($0.0 \leq r \leq 1.0$).
Default: If no r value is given in the *SHARPEN* option, *r=1.0*.
Example:

```
SHARPEN REMOVE = 0.5
```

BOUNDS [*X* = *(*from,to*)*], [*Y* = *(*from,to*)*], [*Z* = *(*from,to*)*]
Format: *BOUNDS* followed by any combination of the three coordinates (x, y, z) in any order. For each coordinate, lower and upper bounds are given separated by a comma and enclosed in parentheses.
Default: For each space group, an appropriate asymetric unit is stored in the program (see pages 11.11–11.16).
Example:
BOUNDS X = (-0.5,0.5), Y = (0.0,0.32), Z = (0.0,1.0)

SAVE MAP
Action: The *name.MAP* file created after this option is encountered will be saved for subsequent use by *Peak List* (run as a separate program—see p. 11.7) or *Contour Map.*

SCALE size
SCALE X = Δx, *Y* = Δy, *Z* = Δz
Action: True scale, undistorted maps are specified by the word *SCALE* followed by a real value for the scale in in./Å. Fractional scale maps are specified by giving the intervals between point in each direction in crystallographic coordinates.
Default: If this option is not used, the scale will be set to *2.0* in./Å. For three-dimensional maps, the interval between sections will be the same as the horizontal increment.
Examples:

```
SCALE 1.5
SCALE X = .02, Y = .04, Z = .05
```

Map starting cards
Use one of the following:

GENERAL `ss.sssaa.aaaabb.bbbbcc.ccccdd.ddddee.eeeehh.hhhww.www i,`
[*MINPRINT* = density], [*ID* = 'any characters'], [*SQUARE GRID*]

PROJECTION [*DOWN* axis], [axis *CON.HORIZ.*], [*MINPRINT* = density],
[*ID* = 'any characters'], [*SQUARE GRID*]

SECTION axis = value, [axis *CON.HORIZ.*], [*MINPRINT* = density],
[*ID* = 'any characters'], *SQUARE GRID*

THREE-DIMENSIONAL [*DOWN* axis], [axis *CON.HORIZ.*], [*INCREMENT* = δ],
[*MINPRINT* = density], [*ID* = 'any chracters'], *SQUARE GRID*

Format and Action: Each of the above cards initiates the calculation of a map of the specified type. The various options on the card can be in any order. Continuation cards may be used if needed; in this case, a comma must be the last nonblank character on the preceding card and the first five columns on the continuation card must be blank.

General card—Format(A7,1X,F6.3,6F7.4,2F6.3,1X,I1)
1–7 *GENERAL*
9–14 Scale for general plane maps in in./Å. The default is *2.0*.
15–35 The 3 directional cosines relative to *a*, *b* and *c*.
36–56 The 3 fractional coordinates of the center of the map.
57–62 The map width in Å. (The default is *5.0*.)
63–68 The map height in Å. (The default is *5.0*.)
70 ⟨*1*⟩/⟨*2*⟩/⟨*3*⟩ ⟨*X*⟩/⟨*Y*⟩/⟨*Z*⟩ constant horizontal.
The first 70 columns are fixed format as described above. The *Minprint*, *ID* and *Square Grid* options may be used in free format after column 70 (put a comma in column 71 and use a continuation card with first five columns blank).

Note: The *General* card may be created by the program *Cosine* or in *Distan* by use of the *Gen* option.

Axis *CON.HORIZ.* (All map types except *General*)
Axis may be *X*, *Y* or *Z* to specify which fractional coordinate is to be held constant along rows of the map. (For orthogonal axes, this is the vertical coordinate.)
Default: *X CON.HORIZ.*

MINPRINT = density (All map types)
This option specifies the minimum numerical value to be printed (an integer). All smaller densities are printed as blanks. The density may be negative.
Default: Normally, all densities are printed.

ID = 'any characters' (All map types)
The map ID is used to identify the printed and tape outputs. Up to twelve characters are allowed, enclosed in quotes.

SQUARE GRID (All map types)
This option specifies the printing of maps on every third line. Points are then $\frac{1}{2}$ inch apart vertically and horizontally. The scale of true-scale maps is adjusted so that they remain undistorted (the sampling is less dense). The scale of fractional-scale maps is unchanged. You must use 6 lines per inch printing for this to work!
Default: Maps are normally printed on every second line.

DOWN axis (*Projection* and *Three-Dimensional* only)
This option specifies the coordinate (*X*, *Y* or *Z*) which is to be held constant within each section of a *Three-Dimensional* map, or the coordinate which is to be *Projected* down.
Note: The program automatically selects the appropriate two-dimensional data for a *Projection* from whatever data are read in.
Default: *DOWN Z*

Axis = value (*Single Sections* only)
This option specifies which coordinate (*X*, *Y* or *Z*) is to be held constant in a *Single Section* map and supplies the value of that coordinate.

INCREMENT = δ (*Three-Dimensional true-scale* maps only)
This option specifies the fractional increment, δ, between sections. For fractional-scale maps, the appropriate increment from the *Scale* card is used instead. If no increment is specified, the program will calculate a value such that the distance between sections is comparable to the distance between points within the sections.

Examples:

```
GENERAL    2.0  0.2738 0.3053 0.2500   0.00    0.00    1.00 4.5    4.5    1,
     ID = 'MIRROR'
```

(The ID is on a continuation card with the first five columns blank.)

```
PROJECTION DOWN X, Y CON.HORIZ., SQUARE GRID

SECTION Y = 0.5, MINPRINT = 2

THREE-DIMENSIONAL INCREMENT = 0.025
```

Peak List option—Format(A4,6X,F6.0,F6.3,2I2,1X,F6.0,F6.3)
1–9 *PEAK LIST*
11–16 Minimum height in printed units for peaks to be listed.
17–22 Minimum distance between peaks in Å (eliminates overlapping of peaks). Pairs of peaks separated by less than this amount will have their coordinates averaged. The default is *0.1*.
23–24 ⟨*0*⟩/⟨*1*⟩ ⟨Do not⟩/⟨Do⟩ create a *name.PKL* file of *Coord* cards for the peaks. The *Coord* cards in the *name.PKL* file are in the regular *Coord* format needed by *Distan* and *Structure Factors, Least Squares*. However, the cards include only x, y and z and an incremented integer as an atom identifier. Any other information needed on the cards (*Structure Factors* and *Least Squares* **demand** a form factor label and matching *Iso* or *Aniso* cards) must be edited in.
25–26 ⟨*0*⟩/⟨*1*⟩ ⟨Do not⟩/⟨Do⟩ print a section-by-section listing. You may request this option only with a *3-DIM* map starting card. If used, the peaks for each map section will be listed separately (in addition to the three-dimensional peaks).
If negative peaks are wanted:
28–33 Minimum height for negative peaks. The default is *150*; that is, no negative peak will be listed unless its value is less than −150 on the map scale.
34–39 Minimum distance between the negative peaks, in Angstroms. The default is *0.5* Å.

Note: If, after running your *Fourier* (*Difference Fourier*, *Patterson*) program with the *Save Map* option, you wish to do a *Peak List* as a separate job (or if you wish to do a second *Peak List* with different parameters), you may use this card alone (followed by an *End X-ray* card) to initiate the calculations using your *name.MAP* file.

End Card—Format(11X) 1–11 END FOURIER

Input files needed:

name.FOU, *name.DIF* or *name.PAT*

name.HKL

name.INF

Output

name.MAP (if requested)
A binary file of information used by *Peak List* or *Contour Map*.

name.PKL (if requested by *Peak List*)
A formatted file of *Coord* cards.

name.OUT
The Fourier, difference Fourier or Patterson map.

name.GEN (from *General Plane Fourier* only)
A file of *Gen* cards for use as input to *HOPT*.

Contour Map

This program produces a contour map of a *Fourier*, *Difference Fourier* or *Patterson* calculation. The map will be output to one of three types of files, suitable for viewing on a VT125 terminal or for printing on the TRILOG, HP4M or the LASER printer.

Before running *Contour Map*, it is first necessary to run the appropriate *Fourier* program using the *Save Map* option. This will create a *name.MAP* file which is needed by *Contour Map*.

Input cards

1)Title card—Format(A4,8X,3F5.1,13X,I1)
1–11 *CONTOUR MAP*
13–17 Beginning contour level (may be negative).
18–22 Contour increment.
23–27 Ending contour level.
Note: Contour levels and increment refer to printed units on the corresponding *Fourier* map.
41 Version number of *name.MAP* file to be used. (The default is the use of the highest version *name.MAP* file in your directory.)

2)Output specifier card—Format(A80)
1–80 ⟨*name.TER/VT125*⟩/⟨*name.TRI/TRILOG*⟩/⟨*name.LAS/LASER*⟩ Output to file ⟨*name.TER* suitable for typing on a VT125 terminal⟩/⟨*name.TRI* suitable for printing on the TRILOG printer⟩/⟨*name.LAS* suitable for printing on the LASER printer⟩. Each of these files **must** be used on the specified device since they contain control sequences peculiar to the individual devices. Hopefully, the file extensions (*TER*, *TRI* and *LAS*) will prevent any accidents. The files are **not** binary and hence may be read or even edited, but will probably not make sense to the average user.

Note: This program does not require an *End* card. It may be followed by another program or terminated with an *END X-ray* card.

Input files needed

name.CON

name.MAP

Output

name.TER, *name.TRI* or *name.LAS*

The contour map will be output to the specified file. The *name.TER* file may be typed on a VT125 terminal (*T name.TER*) while the *name.TRI* and *name.LAS* files may be printed on the TRILOG or LASER printer (*PRINT name.TRI* or *LASER name.LAS*).

name.OUT

A small output file which may be helpful if there is an error in your map.

Defaults for Space Group Bounds

Space Group	X	Y	Z
1	0.0,1.0	0.0,1.0	0.0,1.0
2	0.0,1.0	0.0,0.5	0.0,1.0
3	0.0,0.5	0.0,1.0	0.0,1.0
4	0.0,1.0	0.0,0.5	0.0,1.0
5	0.0,0.5	0.0,0.5	0.0,1.0
6	0.0,1.0	0.0,0.5	0.0,1.0
7	0.0,1.0	0.0,0.5	0.0,1.0
8	0.0,1.0	0.0,0.25	0.0,1.0
9	0.0,1.0	0.0,0.25	0.0,1.0
10	0.0,0.5	0.0,0.5	0.0,1.0
11	0.0,1.0	0.0,0.25	0.0,1.0
12	0.0,0.5	0.0,0.25	0.0,1.0
13	0.0,0.5	0.0,0.5	0.0,1.0
14	0.0,1.0	0.0,0.25	0.0,1.0
15	0.0,0.5	0.0,0.25	0.0,1.0
16	0.0,0.5	0.0,0.5	0.0,1.0
17	0.0,0.5	0.0,0.5	0.0,1.0
18	0.0,0.5	0.0,0.5	0.0,1.0
19	0.0,0.5	0.0,0.5	0.0,1.0
20	0.0,0.25	0.0,0.5	0.0,1.0
21	0.0,0.25	0.0,0.5	0.0,1.0
22	0.0,0.25	0.0,0.25	0.0,1.0
23	0.0,0.25	0.0,0.5	0.0,1.0
24	0.0,0.25	0.0,0.5	0.0,1.0
25	0.0,0.5	0.0,0.5	0.0,1.0
26	0.0,0.5	0.0,0.5	0.0,1.0
27	0.0,0.5	0.0,0.5	0.0,1.0
28	0.0,0.25	0.0,1.0	0.0,1.0
29	0.0,0.25	0.0,1.0	0.0,1.0
30	0.0,0.5	0.0,1.0	0.0,0.5
31	0.0,0.25	0.0,1.0	0.0,1.0
32	0.0,0.25	0.0,1.0	0.0,1.0
33	0.0,0.25	0.0,1.0	0.0,1.0
34	0.0,0.25	0.0,1.0	0.0,1.0
35	0.0,0.25	0.0,0.5	0.0,1.0
36	0.0,0.25	0.0,0.5	0.0,1.0
37	0.0,0.25	0.0,0.5	0.0,1.0
38	0.0,0.5	0.0,0.25	0.0,1.0
39	0.0,0.5	0.0,0.25	0.0,1.0

Space Group	X	Y	Z
40	0.0,0.25	0.0,0.5	0.0,1.0
41	0.0,0.5	0.0,0.25	0.0,1.0
42	0.0,0.25	0.0,0.5	0.0,0.5
43	0.0,0.25	0.0,0.25	0.0,1.0
44	0.0,0.25	0.0,0.5	0.0,1.0
45	0.0,0.25	0.0,0.5	0.0,1.0
46	0.0,0.25	0.0,1.0	0.0,0.5
47	0.0,0.5	0.0,0.5	0.0,0.5
48	-0.25,0.25	0.0,1.0	0.0,0.25
49	0.0,0.5	0.0,0.5	0.0,0.5
50	0.0,0.25	0.0,1.0	0.0,0.5
51	0.0,0.25	0.0,0.5	0.0,1.0
52	0.0,0.5	0.0,0.25	0.0,1.0
53	0.0,0.25	0.0,0.5,	0.0,1.0
54	0.0,0.25	0.0,0.5	0.0,1.0
55	0.0,0.25	0.0,1.0	0.0,0.5
56	0.0,0.25	0.0,1.0	0.0,0.5
57	0.0,0.5	0.0,1.0	0.0,0.25
58	0.0,0.25	0.0,1.0	0.0,0.5
59	-0.25,0.25	-0.25,0.25	0.0,0.5
60	0.0,0.25	0.0,0.5	0.0,1.0
61	0.0,0.25	0.0,1.0	0.0,0.5
62	0.0,0.5	0.0,0.25	0.0,1.0
63	0.0,0.25	0.0,1.0	0.0,0.25
64	0.0,0.25	0.0,1.0	0.0,0.25
65	0.0,0.25	0.0,0.5	0.0,0.5
66	0.0,0.25	0.0,0.5	0.0,0.5
67	0.0,0.25	0.0,0.25	0.0,1.0
68	0.0,0.25	0.0,0.25	0.0,1.0
69	0.0,0.25	0.0,0.25	0.0,0.5
70	0.0,0.125	-0.125,0.125	0.0,1.0
71	0.0,0.25	0.0,0.5	0.0,0.5
72	0.0,0.25	0.0,0.5	0.0,0.5
73	0.0,0.25	0.0,0.5	0.0,0.5
74	0.0,0.25	0.0,0.25	0.0,1.0
75	0.0,0.5	0.0,0.5	0.0,1.0
76	0.0,1.0	0.0,1.0	0.0,0.25
77	0.0,1.0	0.0,0.5	0.0,0.5
78	0.0,1.0	0.0,1.0	0.0,0.25
79	0.0,0.5	0.0,0.5	0.0,0.5
80	0.0,0.5	0.0,1.0	0.0,0.25
81	0.0,0.5	0.0,1.0	0.0,0.5

Space Group	X	Y	Z
82	0.0,0.5	0.0,0.5	0.0,0.5
83	0.0,0.5	0.0,0.5	0.0,0.5
84	0.0,0.5	0.0,0.5	0.0,0.5
85	-0.25,0.25	-0.25,0.25	0.0,0.5
86	-0.25,0.25	0.0,1.0	0.0,0.25
87	0.0,0.5	0.0,0.5	0.0,0.25
88	0.0,0.5	-0.25,0.25	0.0,0.25
89	0.0,0.5	0.0,0.5	0.0,0.5
90	0.0,0.5	0.0,0.5	0.0,0.5
91	0.0,0.5	0.0,0.5	0.0,1.0
92	0.0,0.5	0.0,0.5	0.0,0.5
93	0.0,0.5	0.0,1.0	0.0,0.25
94	0.0,0.5	0.0,1.0	0.0,0.25
95	0.0,0.5	0.0,0.5	0.0,1.0
96	0.0,0.5	0.0,0.5	0.0,0.5
97	0.0,0.5	0.0,0.5	0.0,0.25
98	0.0,0.5	0.0,1.0	0.0,0.125
99	0.0,0.5	0.0,0.5	0.0,1.0
100	0.0,0.5	0.0,0.5	0.0,1.0
101	0.0,0.5	0.0,1.0	0.0,0.5
102	0.0,0.5	0.0,1.0	0.0,0.5
103	0.0,0.5	0.0,0.5	0.0,0.5
104	0.0,0.5	0.0,0.5	0.0,0.5
105	0.0,0.5	0.0,0.5	0.0,0.5
106	0.0,0.5	0.0,0.5	0.0,0.5
107	0.0,0.5	0.0,0.5	0.0,0.5
108	0.0,0.5	0.0,0.5	0.0,0.5
109	0.0,0.5	0.0,0.5	0.0,0.25
110	0.0,0.5	0.0,0.5	0.0,0.25
111	0.0,0.5	0.0,1.0	0.0,0.5
112	0.0,0.5	0.0,0.5	0.0,0.5
113	0.0,0.5	0.0,1.0	0.0,0.5
114	0.0,0.5	0.0,0.5	0.0,0.5
115	0.0,0.5	0.0,0.5	0.0,0.5
116	0.0,0.5	0.0,1.0	0.0,0.25
117	0.0,0.5	0.0,0.5	0.0,0.5
118	0.0,0.5	0.0,1.0	0.0,0.25
119	0.0,0.5	0.0,0.5	0.0,0.25
120	0.0,0.5	0.0,0.5	0.0,0.25
121	0.0,0.5	0.0,1.0	0.0,0.25
122	0.0,0.5	0.0,0.5	0.0,0.5
123	0.0,0.5	0.0,0.5	0.0,0.5

Space Group	X	Y	Z
124	0.0,0.5	0.0,0.5	0.0,0.25
125	0.0,0.25	0.0,1.0	0.0,0.5
126	-0.25,0.25	-0.25,0.25	0.0,0.25
127	0.0,0.5	0.0,0.5	0.0,0.5
128	0.0,0.5	0.0,0.5	0.0,0.25
129	0.0,0.25	0.0,1.0	0.0,0.5
130	0.0,0.25	0.0,1.0	0.0,0.25
131	0.0,0.5	0.0,0.5	0.0,0.25
132	0.0,0.5	0.0,0.5	0.0,0.5
133	0.0,0.25	0.0,1.0	0.0,0.25
134	-0.25,0.25	0.0,1.0	0.0,0.25
135	0.0,0.5	0.0,0.5	0.0,0.25
136	0.0,0.5	0.0,1.0	0.0,0.25
137	-0.25,0.25	-0.25,0.25	0.0,0.25
138	0.0,0.25	0.0,1.0	0.0,0.25
139	0.0,0.5	0.0,0.5	0.0,0.25
140	0.0,0.5	0.0,0.5	0.0,0.25
141	0.0,0.25	0.0,0.25	0.0,1.0
142	0.0,0.25	0.0,0.25	0.0,1.0
143	0.0,1.0	0.0,1.0	0.0,1.0
144	0.0,1.0	0.0,1.0	0.0,0.333
145	0.0,1.0	0.0,1.0	0.0,0.333
146	0.0,1.0	0.0,1.0	0.0,0.333
147	0.0,1.0	0.0,1.0	0.0,0.5
148	0.0,0.5	0.0,1.0	0.0,0.333
149	0.0,1.0	0.0,1.0	0.0,0.5
150	0.0,1.0	0.0,1.0	0.0,0.5
151	0.0,1.0	0.0,1.0	0.0,0.333
152	0.0,1.0	0.0,1.0	0.0,0.333
153	0.0,1.0	0.0,1.0	0.0,0.333
154	0.0,1.0	0.0,1.0	0.0,0.333
155	0.0,1.0	0.0,1.0	0.0,0.333
156	0.0,1.0	0.0,1.0	0.0,1.0
157	0.0,0.5	0.0,1.0	0.0,1.0
158	0.0,1.0	0.0,1.0	0.0,0.5
159	0.0,1.0	0.0,1.0	0.0,0.5
160	0.0,1.0	0.0,1.0	0.0,0.333
161	0.0,1.0	0.0,1.0	0.0,0.1667
162	0.0,1.0	0.0,1.0	0.0,0.5
163	0.0,1.0	0.0,1.0	0.0,0.25
164	0.0,1.0	0.0,1.0	0.0,0.5
165	0.0,1.0	0.0,1.0	0.0,0.25

Space Group	X	Y	Z
166	0.0,1.0	0.0,1.0	0.0,0.1667
167	0.0,1.0	0.0,1.0	0.0,0.1667
168	0.0,0.5	0.0,1.0	0.0,1.0
169	0.0,1.0	0.0,1.0	0.0,0.1667
170	0.0,1.0	0.0,1.0	0.0,0.1667
171	0.0,1.0	0.0,1.0	0.0,0.1667
172	0.0,1.0	0.0,1.0	0.0,0.1667
173	0.0,1.0	0.0,1.0	0.0,0.5
174	0.0,1.0	0.0,1.0	0.0,0.5
175	0.0,0.5	0.0,1.0	0.0,0.5
176	0.0,1.0	0.0,1.0	0.0,0.25
177	0.0,0.5	0.0,1.0	0.0,0.5
178	0.0,1.0	0.0,1.0	0.0,0.1667
179	0.0,1.0	0.0,1.0	0.0,0.1667
180	0.0,1.0	0.0,1.0	0.0,0.1667
181	0.0,1.0	0.0,1.0	0.0,0.1667
182	0.0,1.0	0.0,1.0	0.0,0.25
183	0.0,0.5	0.0,1.0	0.0,1.0
184	0.0,0.5	0.0,1.0	0.0,0.5
185	0.0,1.0	0.0,1.0	0.0,0.5
186	0.0,1.0	0.0,1.0	0.0,0.5
187	0.0,1.0	0.0,1.0	0.0,0.5
188	0.0,1.0	0.0,1.0	0.0,0.25
189	0.0,1.0	0.0,1.0	0.0,0.5
190	0.0,1.0	0.0,1.0	0.0,0.25
191	0.0,0.5	0.0,1.0	0.0,0.5
192	0.0,0.5	0.0,1.0	0.0,0.25
193	0.0,0.5	0.0,1.0	0.0,0.25
194	0.0,1.0	0.0,1.0	0.0,0.25
195	0.0,0.5	0.0,0.5	0.0,1.0
196	0.0,0.25	0.0,0.5	0.0,0.5
197	0.0,0.5	0.0,0.5	0.0,0.5
198	0.0,0.5	0.0,0.5	0.0,1.0
199	0.0,0.5	0.0,0.5	0.0,0.5
200	0.0,0.5	0.0,0.5	0.0,0.5
201	-0.25,0.25	-0.25,0.25	0.0,0.5
202	0.0,0.25	0.0,0.25	0.0,0.5
203	0.0,0.125	-0.125,0.125	0.0,1.0
204	0.0,0.25	0.0,0.5	0.0,0.5
205	0.0,0.5	0.0,0.5	0.0,0.5
206	0.0,0.125	0.0,0.5	0.0,0.5
207	0.0,0.5	0.0,0.5	0.0,0.5

Space Group	X	Y	Z
208	0.0,0.25	0.0,1.0	0.0,0.5
209	0.0,0.25	0.0,0.25	0.0,0.5
210	0.0,0.125	0.0,0.25	0.0,1.0
211	0.0,0.25	0.0,0.5	0.0,0.5
212	-0.25,0.25	-0.25,0.25	0.0,1.0
213	-0.25,0.25	-0.25,0.25	0.0,1.0
214	0.0,0.125	0.0,0.5	0.0,1.0
215	0.0,0.5	0.0,0.5	0.0,1.0
216	0.0,0.25	0.0,0.25	0.0,1.0
217	0.0,0.5	0.0,0.5	0.0,0.5
218	0.0,0.5	0.0,0.5	0.0,0.5
219	0.0,0.25	0.0,0.5	0.0,0.5
220	0.0,0.25	0.0,0.5	0.0,0.5
221	0.0,0.5	0.0,0.5	0.0,0.5
222	-0.25,0.25	-0.25,0.25	0.0,0.25
223	0.0,0.25	0.0,0.5	0.0,0.5
224	-0.25,0.25	0.0,1.0	0.0,0.25
225	0.0,0.25	0.0,0.25	0.0,0.5
226	0.0,0.25	0.0,0.25	0.0,0.25
227	0.0,0.125	0.0,0.5	0.0,1.0
228	0.0,0.125	0.0,0.5	0.0,0.5
229	0.0,0.25	0.0,0.5	0.0,0.5
230	0.0,0.25	0.0,0.5	0.0,0.25
5A	0.0,0.5	0.0,0.5	0.0,1.0
7A	0.0,1.0	0.0,0.5	0.0,1.0
7N	0.0,1.0	0.0,0.5	0.0,1.0
8A	0.0,1.0	0.0,0.25	0.0,1.0
9A	0.0,1.0	0.0,0.25	0.0,1.0
12A	0.0,1.0	0.0,0.25	0.0,0.5
13A	0.0,1.0	0.0,0.5	0.0,0.5
13N	0.0,0.25	0.0,1.0	0.0,1.0
14A	0.0,1.0	0.0,0.25	0.0,1.0
15A	0.0,1.0	0.0,0.25	0.0,0.
65A	0.0,0.5	0.0,0.5	0.0,0.25

Distances, Angles, and Planes

Distances, Angles and Planes (*Distan*) can calculate the following:
1)Interatomic distances;
2)Interatomic angles;
3)Torsion angles;
4)Least squares planes;
5)Dihedral angles between pairs of least squares planes;
6)Principal axes of Gaussian amplitude ellipsoids;
7)Coordinate orthogonalization;
8)Symmetry transformations;
9)Atom connectivity transformations.

Atom coordinates may be entered as part of the input file, recalled from a *name.CRD* file or recalled from temporary storage if the *Distan* job immediately follows *Least Squares* or *Structure Factors*.

All distances between pairs of atoms closer than and/or more distant from a designated value can be calculated, as can all angles A–B–C in which the A–B and B–C distances are below another designated value. Symmetry transformations necessary to find distances or angles involving additional asymmetric units can be created according to the space group (standard settings only; *International Tables*—if the available atomic coordinates do not correspond to the standard setting of a space group, they can be transformed by the program *Parameter Switch* before any calculation takes place), or can be specified by the user.

Torsion angles A–B–C–D (the torsion angle about the bond B–C, corresponding to the right-handed screw) require entry of the four atoms A, B, C, D, all of which must be in the original asymmetic unit; no symmetry transformation will be applied.

Least squares planes can be calculated through any designated atoms in any asymmetric unit. Weights and esd.'s can be entered separately, or taken from a *name.HKL* file. Dihedral angles (and esd.'s, including allowance for atoms that are common to the two planes) can be calculated for any pair of such planes.

Principal axes are calculated from the β_{ij} or U_{ij} values as available on *Aniso* cards, either entered separately, recalled from a *name.CRD* file or recalled from temporary storage if the *Distan* job immediately follows *Structure Factors* or *Least Squares*.

Coordinate orthogonalization is automatically performed before any bond distance, angle or plane calculations. An additional listing of orthogonalized coordinates for any desired asymmetric units can be demanded.

Atoms can be transformed from one asymmetric unit to another so as to establish optimum intramolecular connectivity.

Finally, if the available atomic coordinates do not correspond to the standard setting of a space group, they can be transformed before any calculation takes place.

Symmetry transformation sequence numbers: Symmetry numbers are optional on several cards. These numbers indicate which space group symmetry transformation is to be used in the calculation. They may only be used if *Sym* cards have been entered. In all cases, if no symmetry number is entered, no transformation will be applied in the calculation. A negative symmetry number denotes the inverse of the symmetry transformation. A transformation $\mathbf{R} = \alpha\mathbf{r} + \mathbf{t}$, which moves an atom from position vector $\mathbf{r}$ to position vector $\mathbf{R}$, and in which α is the matrix of the rotational part and $\mathbf{t}$ is the translation, has the sometimes surprising inverse $\mathbf{r} = \alpha^{-1}\mathbf{R} - \alpha^{-1}\mathbf{t}$. For example, the inverse of $\mathbf{R} = -\mathbf{r} + \mathbf{t}$ is $\mathbf{r} = -\mathbf{R} + \mathbf{t}$, but of $\mathbf{R} = \mathbf{r} + \mathbf{t}$ is $\mathbf{r} = \mathbf{R} - \mathbf{t}$. Another example: Suppose DIST ALL to reveal that an A atom moved by a $\mathbf{4}_1$ axis at $(\frac{1}{4}, \frac{3}{4}, z)$ from (x_A, y_A, z_A) to $(1 - y_A, \frac{1}{2} + x_A, \frac{1}{4} + z_A)$ is bonded to atom B at (x_B, y_B, z_B). What are the coordinates of the B atom that is bonded to the original atom A? Answer: $(y_B - \frac{1}{2}, 1 - x_B, z_B - \frac{1}{4})$, in view of $\mathbf{t} = (1, \frac{1}{2}, \frac{1}{4})$ and, in terms of its rows, $\alpha = (0, -1, 0; 1, 0, 0; 0, 0, 1)$.

Input cards

The first two cards in the *Distan* data deck must be the *Title* card and the *Cell* card. The last card must be the *End Distan* card. All other cards may be entered in any order subject to the noted restrictions.

1)Title card—Format(A4,26X,I3,A1)
1–29 *DISTANCES, ANGLES, AND PLANES*
31–33 Space group number given in *International Tables*, **Vol. I**.
34 Letter to indicate alternative representation of a few space groups (only 14N for now).

2)Cell card—use one of the following
a)Format(A4,2X,6F8.4)
1–4 *CELL*
7–14 a (Å)
15–22 b (Å)
23–30 c (Å)
31–38 α (°)
39–46 β (°)
47–54 γ (°)
b)Format(3A4)
1–15 *CELL IN STORAGE*

3)Coord card—use one or both of the following
a)Format(A5,1X,A4,2X,A3,3F9.6,30X,F7.5)
1–5 *COORD*
7–10 Atom identifier.
13–15 Atomic scattering factor label. (*Optional.*)
17–24 x (in fractions of a cell edge).
25–33 y (in fractions of b cell edge).
34–42 z (in fractions of c cell edge).
73–79 Radial standard deviation of atomic position (σ) in Å. (*Optional.*)

b)Format(3A4)
1–16 *COORD IN STORAGE*

4)Torsion card—Format(A7,4(1X,A4))
1–7 *TORSION*
9–12, 14–17, 19–22, 24–27 4 atom identifiers needed to define the torsional angle about the line joining the second and third atoms.

5)Aniso card—use one or both of the following
a)Format(A5,A1,A4,2X,6F10.6)
1–5 *ANISO*
6 ⟨Blank or B⟩/⟨U⟩ ⟨β_{ij}⟩/⟨U_{ij}⟩ values entered.
7–10 Atom identifier.
13–22 β_{11} or U_{11} value.
23–32 β_{22} or U_{22} value.
33–42 β_{33} or U_{33} value.
43–52 β_{12} or U_{12} value.
53–62 β_{13} or U_{13} value.
63–72 β_{23} or U_{23} value.
b)Format(3A4)
1–16 *ANISO IN STORAGE*

6)Sym card—use one or both of the following
a)Format(A3,3X,12F6.3)
1–3 *SYM*
7–12 A (see equation, page 12.7).
13–18 B
19–24 C
25–30 D
31–36 E
37–42 F
43–48 G
49–54 H
55–60 P
61–66 Q
67–72 R
73–78 S
b)Format(A7,1X,F10.0,1X,I1,1X,I3)
1–7 *SYM ALL*
9–18 Symmetry cutoff distance.
20 ⟨Blank or *0*⟩/⟨*1*⟩ ⟨Do not⟩/⟨Do⟩ create a *name.CRD* file of *Sym* cards.
22–24 Number of transformations to be printed ($\leq$ 500). (The default is *50*.)

7)Orthog card—Format(A6)
1–6 *ORTHOG*

8)Connect card—Format(A7,1X,2F5.2)
1–7 *CONNECT*
9–13 Minimum distance (Å) for presuming a bond to exist.
14–18 Maximum distance (Å) for presuming a bond to exist.

9)Dist card—use one or both of the following
a)Format(A4,2X,2A4,3X,I4)
1–4 *DIST*
7–10, 11–14 2 atom identifiers—distance is calculated between these 2 atoms.
18–21 Symmetry transformation number, applied to first atom. (*Optional.*)

b)Format(A8,1X,2F10.2)
1–8 *DIST ALL*
10–19 Maximum distance cutoff in Å. (Default based on atom type.)
20–29 Minimum distance cutoff in Å. (The default is *0.0*.)

10)Angle card—use one or both of the following
a)Format(A5,2X,3A4,1X,2I4)
1–5 *ANGLE*
8–11, 12–15, 16–19 3 atom identifiers—angle is calculated between these 3 atoms with the second being the apex atom.
21–24 Symmetry transformation number, applied to first atom. (*Optional.*)
25–28 Symmetry transformation number, applied to third atom. (*Optional.*)

b)Format(A9,2X,F6.0)
1–9 *ANGLE ALL*
11–16 Maximum distance cutoff for angle calculations in Å. (The default is *1.6*.)

11)Plane card—Format(A5,A1,9(A4,I4),A2)
1–5 ⟨*PLANE*⟩/⟨*TWOPL*⟩ Request calculation for ⟨one plane⟩/⟨two planes including the dihedral angle between them⟩.
6 ⟨Blank⟩/⟨Non-blank⟩ ⟨Limited⟩/⟨Full⟩ printed output.
7–10, 15–18, etc. Atom identifiers for plane calculation.
11–14, 19–22, etc. Symmetry transformation numbers of preceding atoms. (*Optional.*)
79–80 Plane identifier. (*Optional.*)

Continuation cards (if needed) for card 10—Format(6X,9(A4,I4),A2)
7–10, 15–18, etc. Atom identifiers for plane calculation.
11–14, 19–22, etc. Symmetry transformation numbers of corresponding atoms. (*Optional.*)
79–80 Plane identifier. (*Optional.*) Must match identifier on card which it follows.

Number Common Atoms card (if needed) for *Twopl* card—Format(A4,16X,I2)
1–19 *NUMBER COMMON ATOMS*
21–22 Number of atoms common to both planes in two plane calculation. **Note:** Atoms in common must be first atoms listed on **both** *Twopl* cards. A *Number Common Atoms* card must follow the first as well as the second of the *Twopl* cards.

Gen card (if wanted) to accompany card 10—Format(A3,1X,3F6.3,I2)
1–3 *GEN*
5–10 Map scale in in./Å. (The default is *1.0*.)
11–16 Vertical width of map in Å. (The default is *10.0*.)
17–22 Horizontal width of map in Å. (The default is *10.0*.)
23–24 $\langle$*0* or *1*$\rangle$/$\langle$*2*$\rangle$/$\langle$*3*$\rangle$ $\langle x\rangle/\langle y\rangle/\langle z\rangle$ constant horizontal. (The default is x.)

12)Weight card—use one or more of the following
a)Format(A6,9(A4,F4.0))
1–6 *WEIGHT*
7–10, 15–28, etc. Atom identifiers.
11-14, 19–22, etc. Weights of corresponding atoms.

b)Format(3A4)
1–14 *WEIGHT=1/SIGMA*

c)Format(4A4)
1–22 *WEIGHT=1/SIGMA SQUARED*

13)Sigma cards—use one or more of the following
a)Format(A5,1X,9(A4,F4.0))
1–5 *SIGMA*
7–10, 15–18, etc. Atom identifiers.
11–14, 19–22, etc. Positional standard deviations in Å($\times 10^5$) of corresponding atoms.

b)Format(3A4)
1–16 *SIGMA IN STORAGE*

Continuation cards (if needed) for cards 10a, 11a—Format(6X,9(A4,I4))
7–10, 15–18, etc. Atom identifiers.
11-14, 19–22, etc. Weights or esd.'s of corresponding atoms.

14)End card—Format(2A4)
1–10 *END DISTAN*

Description of individual cards

2)Cell card

This card is used to provide the unit cell dimensions. *Cell in Storage* calls for the parameters which are available on the *name.HKL* file if, and only if, a *Structure Factor* or *Least Squares* calculation immediately precedes *Distan* in the same job.

Note: This card should be the first card in the *Distan* data deck since all other calculations require this information.

3)Coord card(s)

The *Coord* card is used to enter coordinate data. As for *Cell in Storage*, *Coord in Storage* recalls all *Coord* cards entered in *Structure Factors* or *Least Squares* and may therefore only be used when *Distan* follows *Structure Factors* or *Least Squares* in the same job. Both types of *Coord* cards may be used if needed. Atomic scattering factor labels are optional but should be used if you plan a *Least Squares* refinement with the *Coord* cards.

Note: This card should follow the *Cell* card as it is necessary for most other calculations.

4)Torsion card

This card requests the calculation of torsional angles. If used, it must come directly after the *Cell* and *Coord* cards. The sign of the calculated angle is assigned by the convention of Klyne and Prelog (right-hand screw), (*Experentia*, **16**, 251–253).

Note: Esd.'s will be applied to torsion calculations if they have been previously entered on the *Coord* cards or are available from the *name.HKL* file (*Coord in Storage* option).

5)Aniso card(s)

This card is used to enter the anisotropic Gaussian amplitudes for the equation

$$e^{-(\beta_{11}h^2+\beta_{22}k^2+\beta_{33}l^2+\beta_{12}hk+\beta_{13}hl+\beta_{23}kl)}$$

or, if requested, the U_{ij} terms in matrix notation:

$$e^{-(2\pi^2(U_{11}h^2a^{*^2}+U_{22}k^2b^{*^2}+U_{33}l^2c^{*^2}+2U_{12}hka^*b^*+2U_{13}hla^*c^*+2U_{23}klb^*c^*))}.$$

Aniso in Storage recalls all *Aniso* cards entered in *Structure Factors* or *Least Squares* as for the *Coord in Storage* card.

6)Sym card(s)

This card is used to enter symmetry transformations appropriate to the space group. *Sym All* generates automatically, according to the standard setting of the space group, all transformations which will place one or more of the previously entered atoms (*Coord* cards) within the specified distance from another atom in the original asymmetric unit. Transformations are numbered sequentially as they are generated. If you wish to apply such a transformation to an atom in any subsequent calculation, the number of the transformation must be taken from a previous run with the same data. The first 50 transformations generated will always be printed in output. You may request that up to 500 transformations be printed if needed.

The *Sym* card is used to request a specific transformation of the form:

$$x' = Ax + By + Cz + D, \quad y' = Ex + Fy + Gz + H, \quad z' = Px + Qy + Rz + S$$

or:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} A & B & C \\ E & F & G \\ P & Q & R \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} D \\ H \\ S \end{pmatrix}.$$

These transformations are numbered sequentially as they are encountered. The constants are entered as real numbers; for $\frac{1}{3}$ enter *.3333*.

7)Orthog card

This card requests the printing of both the fractional and the orthogonalized coordinates of symmetry transformed atoms. The *Orthog* card should follow the *Sym* card(s). Up to 50 transformations (unless this number has been changed on the *Sym All* card) will be printed on output followed by the coordinates which result from each transformation.

8)Connect card

This option attempts to perform symmetry operations on the atom coordinates, in accordance with the space group equivalences, to generate (as far as possible) discrete molecules or ions—that is, molecules or ions in which the atoms are directly connected to one another without requiring symmetry transformations. Output includes a *name.CRD* file of the atoms, appropriately transformed as needed.

Minimum and maximum bond lengths are needed. If an interatomic distance, either before or after symmetry transformation, is less than the minimum bond length, the second atom to be encountered is assumed to be a duplicate and will be omitted from the final *name.CRD* file; if an atom is not within the maximum bond length of any other atom, it will be presumed to lie in a separate fragment.

This option may be useful following *Fourier* or *Direct Methods*, where the atoms may be scattered among various molecules.

9)Dist card(s)

This card requests the calculation of interatomic distances. The *Dist All* card requests the calculation of all distances less than the specified maximum and greater than the specified minimum distance. If a symmetry card has been entered, all symmetry transformations in the list will be used.

The *Dist* card requests the distance between two specified atoms. A symmetry transformation may be applied to the first atom.

Note: Esd.'s will be applied to the calculated distances if previously entered (as for *Torsion* card).

10)Angle card

This card requests the calculation of interatomic angles. The *Angle All* card requests the calculation of all angles whose Atom1–Atom2 and Atom2–Atom3 distances are less than the specified maximum distance. This distance must be less than or equal to the distance on the *Dist All* card.

The *Angle* card requests the calculations of the angle between the three specified atoms, the second being the apex atom. Symmetry transformations may be applied to the first and/or third atoms.

Note: Esd.'s will be applied to the calculated angles if previously entered (as for the *Torsion* card).

11)Plane card

This card is used to request the calculation of the least squares plane through a set of atoms. The planes are calculated by the method given in V. Schomaker, J. Waser, R. Marsh and G. Bergman, *Acta Cryst.*, **12**, 600 (1959).

The *Plane* card calculates the parameters of the plane, including their variances and covariances, from the coordinates and weights of the entered atoms. Symmetry transformations may be applied to each atom before it is placed in the plane list. If more than 9 atoms are involved, continuation cards may be entered for up to a total of 150 atoms.

The *Twopl* (*Two Plane*) card does the above calculations for two planes (entered on separate *Twopl* cards) along with the dihedral angle θ between the planes. The *Twopl* cards **must** be entered in pairs, each one being followed immediately by any necessary continuation cards. The standard deviation of θ becomes ill-defined when θ becomes very small. For this contingency, the program provides an upper bound for the standard deviation of θ (see Waser, Marsh and Cordes, *Acta Cryst.*, **B29**, 2703 (1973)). Up to 40 atoms are allowed in either plane (continuation cards may be used). If there are atoms common to both planes, two *Number Common Atoms* cards (one following each *Twopl* card) are used and the atoms in common must be the first atoms on **both** of the *Twopl* cards (with the order the same on both cards).

If you would like to create a *General* card for use as input to *Fourier* for a particular plane calculation, follow the *Plane* (or *Twopl*) card(s) with a *Gen* card. The *Gen* card contains some *Fourier* map information. All values have defaults. Your *General* card will be found in *name.GEN* upon completion of the program.

Each atom involved in a plane calculation is assigned a default weight of *1.0*. If a different weight is wanted, a *Weight* card may be used. The *Weight* card must follow the appropriate *Plane* or *Twopl* card. For example, if it is desired to see the out-of-plane distance (deviation from the plane) of an atom, but not use the atom to calculate the best plane, it may be assigned a weight of *0.0*. Similarly, to permit calculation of variances and covariances, each atom is assigned a default positional standard deviation of *0.01*Å. If a different set of esd.'s is wanted, a *Sigma* card may be used. The *Sigma* card must follow the corresponding *Weight* card. (**Note:** The weights and sigmas may be unrelated; see Waser, Marsh and Cordes.)

12)Weight card

This card changes the weights of atoms involved in the best plane calculations from the assigned weight of *1.0*. The *Weight* card assigns weights to specific atoms. Continuation cards may be used.

Note: Only atoms requiring a weight different from *1.0* need to be included on the *Weight* card.

The *Weight=1/Sigma* and *Weight=1/Sigma Squared* cards assign those weights to all atoms involved in the plane calculations. These cards may only be used if nonzero esd.'s have been assigned to all atoms involved (via the *Coord in Storage* card or individually on the *Coord* cards).

A mixture of weights may be assigned by entering a *Weight=1/Sigma* card or a *Weight=1/Sigma Squared* card followed by a *Weight* card for specific atoms.

13)Sigma card

This card specifies the positional standard deviation of atoms for use in calculating variances and covariances of the best plane(s); the default assigned sigma is *0.01*Å. The *Sigma* card assigns esd.'s to specific atoms. Continuation cards may be used.

Note: Only atoms requiring a sigma different from *0.01*Å(or stored sigma values if a *Sigma in Storage* card precedes this card) need be included on the *Sigma* card.

Sigma in Storage assigns stored values of sigma (from *Structure Factors* or *Least Squares*) to all atoms involved in the plane calculations. This card may only be used if nonzero sigmas have been assigned to all atoms involved.

A mixture of sigmas may be assigned by entering a *Sigma in Storage* card followed by a *Sigma* card for specific atoms.

Input files needed

name.DIS

Output

name.GEN (if requested)

General cards which may be used as input to the *Fourier* programs.

name.OUT

Cell card: Unit cell, metric tensor for real space and its inverse (the metric tensor for reciprocal space).

Coord cards: List of entered coordinates and the corresponding orthogonalized coordinates. The orthogonalized axes are defined as follows: the direction of B remains fixed; the orthogonalized A is chosen to be in the AB plane perpendicular to B; the orthogonalized C is chosen to complete the orthonormal set. Hence:

$$\begin{aligned}
x' &= xa\sin\gamma + zc\left(\frac{\cos\beta - \cos\alpha\cos\gamma}{\sin\gamma}\right);\\
y' &= yb + ax\cos\gamma + cz\cos\alpha;\\
z' &= zc\sqrt{\frac{1-\cos^2\alpha-\cos^2\beta-\cos^2\gamma+2\cos\alpha\cos\beta\cos\gamma}{\sin^2\gamma}}.
\end{aligned}$$

Torsion card: Torsional angle(s).

Connect card: A "Distance log" of interatomic distances lying between D_{min} and D_{max}; a "Final atom list" in which the atoms have been transformed so as to establish optimum connectivity. (For pairs of atoms less than D_{min} apart, one will be given a Flag of *3*.)

Aniso card: Anisotropic thermal parameters; mean square amplitude of the vibration ellipsoid in the direction of each principal axis; the direction cosines of the principal axes with respect to the unit cell axes.

Sym card: Symmetry transformation coefficients; symmetry transformation sequence number.

Orthog card: Fractional and orthogonalized coordinates of symmetry-transformed atoms.

Dist card: 2 atom identifiers; interatomic distance; symmetry transformation sequence number (applied to first atom).

Angle card: 3 atom identifiers; interatomic angle (if cosine of angle is less than $-.99999$, the angle is set to *180* and marked by a $*$; if cosine is greater than $+.99999$, the angle is assumed to be *0.0* and is omitted from listing); 2 symmetry transformation sequence numbers (applied to first and third atoms).

Plane card: Origin-to-plane distance (always positive), the direction cosines of the plane normal with respect to the real crystallographic axes, and the out-of-plane deviations (in Å) for all atoms; the components of the plane normal in several coordinate systems; variances, covariances and standard deviations associated with these plane parameters.

Twopl card: Same as for *Plane* card plus the dihedral angle between the two planes and its standard deviation.

Number Common Atoms card: None.

Weight card: None.

Sigma card: None.

name.CRD (from *Connect* option only)

A list of atoms, in *Coord* card format, transformed so as to establish optimum connectivity. Atoms with a Flag of *3* are omitted.

Hydrogen Position

Hydrogen Position (*H Pos*) generates coordinates for hydrogen or other atoms having known geometry, given the positions of the adjacent atoms and the bonding configuration. Only the following input file is needed.

Input cards

1)Title card—Format(A4)
1–5 *H POS*

2)Heading card—Format(A80)
1–80 Any title.

3)Cell card—Format(A4,2X,2F8.4)
1–4 *CELL*
7–14 *a* (Å)
15–22 *b* (Å)
23–30 *c* (Å)
31–38 α (°)
39–46 β (°)
47–54 γ (°)

4)Coord cards—Format(A5,1X,A4,2X,A3,3F9.6)
1–5 *COORD*
7–10 Atom identifier.
13–15 Atomic scattering factor label. (*Optional.*)
16–24 *x* in fractions of *a* cell edge.
25–33 *y* in fractions of *b* cell edge.
34–42 *z* in fractions of *c* cell edge.

5)Type card(s)—Format(3X,I1,4A4,2F10.5,6X,3A4,1X,A3)
4 Ntyp (see chart, page 13.4).
5–8 Atom identifier of Atom 1, the atom to which the generated atom(s) is attached.
9–12, 13–16, 17–20 Atom identifiers of adjacent atoms.
21–30 Distance from Atom 1 to generated atom(s) in Å. (The default is *0.95* Å.)
31–40 Angle between two generated atoms in degrees for *Ntyp*s 2 (the default is *120*°), 4, 5 and 6 (the default is *109.5*°.)
47–50, 51–54, 55–58 Atom identifier(s) to be assigned to generated *Coord* cards. (*Optional.*)
60–62 Atom form factor label to assign to generated *Coord* cards. (The default is *HS*.)

6)Iso card(s)—Format(A3,9X,F10.5)
1–3 *ISO*
13–22 *B*.

7)End card—Format(A3)
1–3 *END*

Description of individual cards

3)Cell card

This card contains unit cell dimensions a, b, c, α, β and γ.

4)Coord cards

Coord cards are used to give the position of the atoms adjacent to the generated atoms. All that is needed on *H Pos Coord* cards is the atom identifier and x, y and z. There must be a *Coord* card for each atom used on a *Type* card and the atom identifiers must be the same on both cards.

5)Type cards and 6)Iso cards

Type cards are used to identify the relationship of the generated atoms to the other atoms in the cell. All atoms specified on a *Type* card must have a corresponding *Coord* card.

The card contains *Ntyp* according to the chart on page 13.4. It also contains the atom identifiers of the atom to which the generated atom(s) is attached (Atom 1) and the adjacent atoms as shown in the chart. (A fourth atom identifier is needed only for *Ntyp 3*.) The atom identifiers must match the atom identifiers on the *Coord* cards.

This card also contains the distance from Atom 1 to the generated atom(s) (the default is *0.95*Å) and the angle between two generated atoms necessary only for *Ntyp*s 2 (the default is *120*°), 4, 5 and 6 (the default is *109.5*°—tetrahedral).

This card may contain individual atom identifier(s) (1 for *Ntyp*s 1 and 3; 2 for *Ntyp*s 2 and 4; 3 for *Ntyp*s 5 and 6) to be assigned to the generated *Coord* cards. If this option is not used, generated *Coord* cards will be assigned numerical identifiers starting with *1*.

Finally, this card contains the atomic scattering factor label to be assigned to the generated *Coord* card (*C* or *HS*, for example). The default is *HS*.

The *Type* card may be repeated as often as necessary. If you want *Iso* cards for a particular *Type* card to be included in your *name.CRD* file, an *Iso* card must follow the appropriate *Type* card. The created *Iso* cards will have the same labels as the corresponding *Coord* cards.

Input files needed

name.HPO

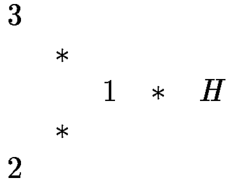
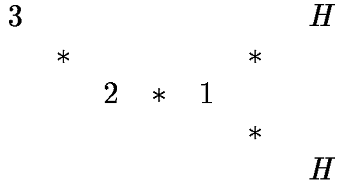
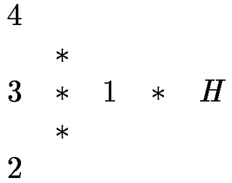
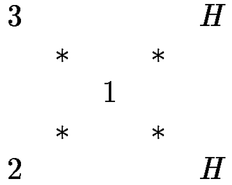
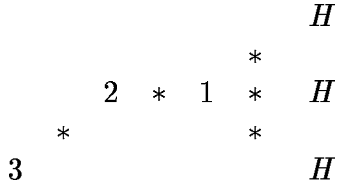
Output

name.CRD

Coord cards (and *Iso* cards, if requested) for the generated atoms. These cards may be used as input to *Distan*, *Structure Factors* or *Least Squares*.

name.OUT

A listing of the generated atoms and their calculated coordinates.

Ntyp	Type	
1	Trigonal $1H$	
2	Trigonal $2H$	
3	Tetrahedral $1H$	
4	Tetrahderal $2H$	
5 or 6	Tetrahderal $3H$ staggered Tetrahedral $3H$ Eclipsed	

Cosine

This program calculates the direction cosines and center point for a plane normal to the line between two atoms, such as the plane of hydrogens of a methyl group. The *name.GEN* file created by *Cosine* contains *General Plane* card(s) which may be used as input to the *Fourier* programs.

Input cards

1)Title card—Format(A4)
1–6 *COSINE*

2)Cell card—Format(A4,2X,6F8.4)
1–4 *CELL*
7–14 a (Å)
15–22 b (Å)
23–30 c (Å)
31–38 α (°)
39–46 β (°)
47–54 γ (°)

3)Constants card—Format(3F10.5,4X,I1,3F10.5) *Optional*
1–10 The desired Fourier map scale in in./Å. (The default is *2.0* in./Å.)
11–20 The horizontal map size in Å. (The default is *5.0* Å.)
21–30 The vertical map size in Å. (The default is *5.0* Å.)
35 $\langle$Blank, *0* or *1*$\rangle/\langle$*2*$\rangle/\langle$*3*$\rangle$ $\langle x\rangle/\langle y\rangle/\langle z\rangle$ to be constant horizontal on the map.
36–45 The distance from the target atom to the first plane in Å. (The default is *0.3* Å.)
46–55 The increment to calculate a series of parallel planes at equal spacings. (*Optional.*)
56–65 The distance from the target atom to the last plane. (*Optional.*)

4)Coord cards (in pairs)—Format(A5,1X,A4,2X,A3,3F9.6)
1–5 *COORD*
7–10 Atom identifier.
13–15 Atomic scattering factor label. (*Optional.*)
17–24 x (in fractions of a cell edge).
25–33 y (in fractions of b cell edge).
34–42 z (in fractions of c cell edge).

5)End card—Format(A3)
1–3 *END*

Description of individual cards

2)Cell card

This card needs only the unit cell dimensions a, b, c, α, β and γ.

3)Constants card

This optional card contains information necessary to create the input cards to the *Fourier* programs. It includes the desired Fourier map scale in in./Å (the default is *2.0* in./Å) and the horizontal and vertical map sizes in Å (the defaults are *5.0* Å).

It also contains the orientation code which tells which direction should be kept constant horizontal on the map (the default is x constant).

The card contains the distance from the target atom to the first plane in Å (the default is *0.3* Å). If you are calculating a series of planes at different distances, you must also enter the distance increment between the planes and the last distance (the distance from the target atom to the last plane) in Å.

If this card is not encountered, all default values will be used.

4)Coord cards

Coord cards are entered in pairs telling which two atoms determine the direction cosines. The first of the pair is the atom further from the plane, the second, the atom closer to the plane (the *target* atom, such as the methyl carbon if one wished to define the plane of its hydrogen atoms).

See *Structure Factors*, *Least Squares* for a detailed description of the *Coord* card.

Input files needed

name.COS

Output

name.OUT

The direction cosines and center point for the plane normal to each set of planes.

name.GEN

General Plane cards which may be used as input to the *Fourier* programs.

Rcheck

This program reads a *name.HKL* file after *Structure Factors* or *Least Squares* has been run and calculates various sums and R for specific groups of data. The data are broken down according to h- k- and ℓ-values, parities, $\frac{\sin^2\theta}{\lambda^2}$ and $\frac{F^2}{\sigma}$ ratios.

Input cards

1)Title card—Format(A4)
1–6 *RCHECK*

2)Scale card—Format(F10.6) *Optional*
1–10 Scale factor—if blank or *0.0*, the scale factor will be taken from the *name.HKL* file.

3)End card—Format(A3)
1–3 *END*

Description of individual cards

2)Scale card

This card is used to enter a scale factor to be applied to F_{obs} for all reflections. If the *Scale* card is not used or the scale factor is read as *0.0*, the scale factor will be taken from the *name.HKL* file.

Input files needed

name.RCK

name.HKL (after *Structure Factors* or *Least Squares* has been run)

Output

name.OUT

Various sums and R, weighted and unweighted, for specific groups of data on your *name.HKL* file.

Parameter Switch

This program transforms coordinates and anisotropic Gaussian *temperature factor* parameters to a new unit cell description, given the old unit cell parameters and the transformation matrix. It reads *Coord* and/or *Aniso* cards and creates a *name.CRD* file with the appropriate parameters updated. The input cards may be contained in a *name.CRD* file created by *Least Squares* or entered as part of the following input file. If the cards are in a *name.CRD* file, the program will read the specified file, transform all *Coord* and *Aniso* parameters and copy all other cards without change into a new *name.CRD* file.

If a *Cell* card is included, cell parameters are transformed according to the given transformation matrix. Coordinates and anisos are then transformed by use of the inverse of the transpose of the given matrix. Anisotropic β_{ij} values are assumed entered and are written in output in equation form:

$$\beta_{11}h^2 + \beta_{22}k^2 + \beta_{33}\ell^2 + \beta_{12}hk + \beta_{13}h\ell + \beta_{23}k\ell;$$

if the input anisos are coded as U_{ij}, both input and output values will be in matrix form:

$$hk\ell \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{pmatrix} \begin{pmatrix} h \\ k \\ \ell \end{pmatrix}$$

unless otherwise requested.

Input cards

1)Title card—Format(A4,3X,I2)
1–16 *PARAMETER SWITCH*
18–19 The version number of the *name.CRD* file to be used (if input is contained in a *name.CRD* file). The default is use of the highest version *name.CRD* file in your directory.

2)Cell card—Format(A4,2X,6F8.4) Required if anisos are in U_{ij} form; optional otherwise
1–4 *CELL*
7–14 a (Å)
15–22 b (Å)
23–30 c (Å)
31–38 α (°)
39–46 β (°)
47–54 γ (°)

3)Transformation card—Format(9F6.3,1X,2A1)
1–6, 7–12, etc. The 9 elements of the unit cell transformation matrix.
56 Nonblank indicates that anisotropic β_{ij} values are entered in matrix form rather than equation form or that U_{ij} values are entered in equation form rather than matrix form.
57 Nonblank requests that anisotropic β_{ij} values be printed (in output) in matrix form rather than equation form or that U_{ij} values be printed in equation form rather than matrix form.

4)Coord cards (if not contained in a *name.CRD* file)
See *Structure Factors*, *Least Squares* for description.

5)Aniso cards (if not contained in a *name.CRD* file)
See *Structure Factors*, *Least Squares* for description.

6)End card—Format(A3)
1–3 *END*

Description of individual cards

2)Cell card

The *Cell* card includes the unit cell parameters. This card is necessary if any U_{ij} anisotropic parameters are being transformed. In any case, if used, the transformed cell dimensions will be included in the printed output.

3)Transformation card

The nine elements $A_{11}, A_{12}, \ldots, A_{33}$ of the unit cell transformation matrix are entered on this card.

$$\begin{pmatrix} a' \\ b' \\ c' \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = Aa$$

The inverse of the transpose of this matrix is used to transform the coordinates and temperature parameters.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = (A^{-1})^T \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

The normal procedure is to enter and list (on output) the β_{ij}'s in equation form or the U_{ij} parameters in matrix form (matrix form has the off-diagonal values multiplied by a factor of $\frac{1}{2}$). This may be changed on the *Transformation* card.

4)Coord cards and 5)Aniso cards

If none of these cards is found, coordinate and anisotropic parameters will be taken from the appropriate *name.CRD* file (as specified on the *Title* card).

See *Structure Factors*, *Least Squares* for a description of the cards.

Input files needed

name.SWI

name.CRD (if parameters not included in above input file)

Output

name.CRD

The given *name.CRD* file with coordinates and anisotropic temperature factor parameters transformed according to the given matrix. All other cards in the original *name.CRD* file (if used) will be included unchanged in the new file. The new *name.CRD* file will be the highest version *name.CRD* file in your directory once the program has completed.

name.OUT

Transformed cell dimensions (if a *Cell* card is included), the transformation equations for coordinates and temperature parameters and the transformed coordinates and temperature parameters.

Cell

Using h, k, ℓ and θ or 2θ data from a diffractometer or measurements from a comparator film, *Cell* calculates by least squares the unit cell dimensions a, b, c, α, β and γ and their esd's. The cell dimensions can be constrained so as to accord to the crystal system. The quantity minimized is:

$$\sum w(\sin^2\theta_{obs} - \sin^2\theta_{calc})^2$$

where w is equal to $\sin^{-2} 2\theta$; this assumes that all measurements of θ are equally accurate. The number entered in the weight column of a *Data* card is multiplied by the square of this term.

Input cards

1)Title card—Format(A4)
1–4 *CELL*

2)Cell Title card—Format(A80)
1–80 Any title.

3)Crystal System card—Format(A4)
1–12 *TRICLINIC*, *MONOCLINIC*, *ORTHORHOMBIC*, *TETRAGONAL*, *HEXAGONAL* or *CUBIC*.

4)Wavelength card—Format(A4,7X,I2,F10.6) (Enter as many cards as needed)
1–10 *WAVELENGTH*
12–13 Wavelength identifier.
14–23 Wavelength.

5)Data Type card—Format(2A4)
1–10 ⟨*THETA*⟩/⟨*2THETA*⟩/⟨*COMP RIGHT*⟩/⟨*COMP LEFT*⟩ ⟨Diffractometer data with θ values⟩/⟨Diffractometer data with 2θ values⟩/⟨Comparator film with beam stop on left⟩/⟨Comparator film with beam stop on right⟩.

6)Data cards—use one of the following types of cards:
a)Following *Theta* or *2Theta* card—Format(4X,3I6,F8.2,18X,I2,F10.5)
5–10 h
11–16 k
17–22 ℓ
23–30 θ or 2θ
49–50 Wavelength identifier. (See page 18.2; the default is *0*.)
51–60 Weight. (The default is *1.0*.)

b)Following *Comp Left* or *Comp Right* card—Format(4X,3I6,3F8.2,2X,I2,F10.5)
5–10 h
11–16 k
17–22 ℓ
23–30 Left portion measurement in mm.
31–38 Middle portion measurement in mm.
39–46 Right portion measurement in mm.
49–50 Wavelength identifier. (The default is *0*.)
51–60 Weight. (The default is *1.0*.)

7)End card—Format(A3)
1–3 *END*

8)Start card—Format(A5)
1–5 *START* (to start execution).

Description of individual cards

4)Wavelength card

The wavelength card is used to add to or change the list of stored wavelengths. The following wavelengths are internally stored (Vol. IV, *Int. Tables*, pp. 6–9):

Wavelength ID	Radiation	Wavelength
0	CuKα	1.54184
1	CuKα_1	1.54056
2	CuKα_2	1.54439
3	CuKβ	1.3926
4	CoKα	1.79026
5	CoKα_1	1.78897
6	CoKα_2	1.79285
7	CoKβ	1.62079
8	FeKα	1.93735
9	FeKα_1	1.93604
10	FeKα_2	1.93998
11	FeKβ	1.75661
12	MoKα	0.71073
13	MoKα_1	0.70930
14	MoKα_2	0.71359
15	MoKβ	0.63229
16	CrKα	2.29100
17	CrKα_1	2.28970

Wavelength ID	Radiation	Wavelength
18	CrKα_2	2.29361
19	CrKβ	2.08487
20	AgKα	0.56087
21	AgKα_1	0.559408
22	AgKα_2	0.563798
23	AgKβ	0.49707

To change or add to this list, enter the ID number by which you will refer to this wavelength and the corresponding wavelength. If the ID number is 23 or less, the internally stored value will be changed to the specified value for the particular run of the program. An ID number over 23 will add the given wavelength to the list. The wavelengths are referred to by ID number on the *Data* cards. Hence, if all your data are based on MoKα radiation, you could enter a *Wavelength* card with *ID*=0 and *Wavelength*=0.71073. You would then not have to enter a wavelength ID on your *Data* cards since you have changed the *0* default value to the desired MoKα wavelength.

5)Data Type card

This card is used to specify what type of data will be entered. A *Theta* or *2Theta* card is used if your data are from a diffractometer (*Theta* for the CAD4 and *2Theta* for the others). *Comp Right* and *Comp Left* cards are used for comparator film data. Left-handed comparator film has the beam stop on the left side of the film and the hole for the beam on the right side.

6)Data cards

Data cards following a *Theta* or *2Theta* card consist of h, k, ℓ, θ or 2θ, wavelength ID and weight. *Data* cards following either of the *Comp* cards consist of h, k, ℓ, left distance, middle distance, right distance, wavelength ID and weight. If the wavelength ID is blank, wavelength *0* (CuKα radiation unless changed on a *Wavelength* card) will be assumed. The default for weight is *1.0*. A weight of 0.0 may be entered if, for example, the indexes are uncertain.

If data come from a comparator, at least one (and preferably several) *Data* cards must have entries in both the *left portion* and the *middle portion*, and one (or several) other cards must have entries in both the *middle portion* and the *right portion*. These reflections are used to obtain the radius of the camera so that the measurements can be converted into θ values.

7)End card

This card specifies the end of one set of data. Cards 5–7 may be repeated for different data types.

8)Start card

This card signifies the end of all data cards and initiates the calculation.

Input files needed

name.CEL

Output

name.OUT

Unit cell dimensions a, b, c, α, β and γ, and their esd's;
A listing of observed and calculated 2θ values for all reflections entered.

Publication List

This program reads your *name.HKL* file and creates a structure factor listing suitable for deposit or for publication in journals (after reduction). A multi-column table is prepared. No sorting is done; the reflections in the *name.HKL* file are assumed to be ordered with one index moving fastest.

Structure factor output options are as follows:
a)h, k, ℓ, F_{obs}, F_{calc};
b)h, k, ℓ, F_{obs}, F_{calc}, A_{calc}, B_{calc};
c)h, k, ℓ, F_{obs}, F_{calc}, Phase angle;
d)h, k, ℓ, F_{obs}, F_{calc}, $\sigma_{F_{obs}}$; (default)
e)h, k, ℓ, F_{obs}, F_{calc}, $\frac{\Delta F^2}{\sigma_{F^2_{obs}}}$.

The input program will ask if your output will be printed on the PostScript laser printer or the Talaris laser printer (the output can also be sent to the line printer). If so, the output will include special instructions to make it suitable for laser printing.

Input cards

1)Title card—Format(A4,13X,A5)
1–16 *PUBLICATION LIST*
18–22 ⟨it HP ⟩/⟨*LASER*⟩/⟨Blank⟩ Output to ⟨HP4M Laser⟩/⟨Talaris Laser⟩/⟨Line⟩ printer. (Default is HP4M.)

2)Column card—Format(A4,3X,I3,I2,7I3,I4,2F6.2,F8.5) *Optional*
1–6 *COLUMN*
8–10 ⟨*1*⟩/⟨*2*⟩/⟨*0* or *3*⟩ ⟨h⟩/⟨k⟩/⟨ℓ⟩ fastest-moving index. (The default is ℓ.)
11–12 Number of columns per index. (The default is *6*.)
13–15 Number of columns for F_{obs}. (The default is *5*.)
16–18 Number of columns for F_{calc} (negative for F_{calc} to be signed; output options *a*, *d* and *e* only). (The default is same number of columns as for F_{obs}.)
19–21 Number of columns for A_{calc} and B_{calc}. (Must be nonblank for output option *b*.)
22–24 Number of columns for phase angle. (Must be nonblank for output option *c*.)
25–27 Number of columns for $\sigma_{F_{obs}}$. (Must be nonblank for output option *d*.) (Default is 4.)
28–30 Number of columns for $\frac{\Delta F^2}{\sigma_{F^2_{obs}}}$. (Must be nonblank for output option *e*.)
31–33 Number of columns of reflections per page. (The default is the maximum possible on 8 inches of printout.)
34–37 Number of lines per column. (The default is *78*.)
38–43 Scale factor. (If blank or *0.0*, taken from *name.HKL* file.)
44–49 Printing scale for F_{obs}, F_{calc}, A_{calc}, B_{calc}, $\sigma_{F_{obs}}$, and $\Delta F^2/\sigma_{F^2_{obs}}$. (The default is *10.0*.)

50–57 $\frac{\sin^2\theta}{\lambda^2}$ maximum cutoff. (The default is no cutoff.)
3)Publication Title card—Format(A55,A4)
1–55 Title for table.
56–59 ⟨Blank⟩/⟨*PAGE*⟩ ⟨Do not⟩/⟨Do⟩ print page numbers.

Description of individual cards

1)Title card

If columns 18–22 are non-blank, your output file will include a sequence of printer control characters appropriate to the laser printer you chose.

2)Column card

The *Column* card is used to enter the number of columns for each output item, the scale factor, the fastest-moving index and other information. The default for the fastest-moving index is ℓ. Before using this default, be sure that ℓ is the fastest-moving index on your *name.HKL* file (see the output from your most recent *Structure Factors*, *Least Squares* calculation). The default for the number of columns per index is *6* (including spaces), for F_{obs} is *5* and for F_{calc} is the same number of columns as for F_{obs} (*5* unless you have entered a different value for F_{obs}). If you want output option *a*, *d* or *e*, you may request that F_{calc} be replaced by A_{calc} to give a signed value for centrosymmetric structures with $B_{calc} = 0$; this is done by entering a negative sign before the number of columns for F_{calc}.

Depending on the output option desired, you may enter the number of columns for A_{calc} and B_{calc} (output option *b*), the phase angle (output option *c*), $\sigma_{F_{obs}}$ (output option *d*) **or** $\frac{\Delta F^2}{\sigma_{F^2_{obs}}}$ (output option *e*). For output option *a*, leave **all** of these fields blank. For output option *d*, the value of $\sigma_{F_{obs}}$ is taken equal to $\sqrt{|F_{obs}|^2 + \sigma_{F^2_{obs}}} - |F_{obs}|$.

You may also enter the number of columns of reflections for each output page (the default is the maximum possible on 8 inches of printout) and the number of lines per column (the default is *78*). If the scale factor is read as blank or *0.0*, it will be taken from the *name.HKL* file. The printing scale operates on all printed values except h, k, ℓ and the phase angle. All numbers are printed as integers; thus a printing scale of *10.0* multiplies the values by *10.0* and rounds to the nearest integer. Finally, you may specify a $\frac{\sin^2\theta}{\lambda^2}$ maximum cutoff value (the default is no cutoff).

If no data card is found, output is option *e* with F_{calc} unsigned and the number of columns for $\frac{\Delta F^2}{\sigma_{F^2_{obs}}}$ is *5*. The fastest-moving index is ℓ and the scale factor is taken from the *name.HKL* file. All other default values are used. The print-out fills an 8.5 × 11 inch page.

Note: This program does not require an *End* card. It may be followed by another program or terminated with an *End X-ray* card.

Input files needed

name.PUB

name.HKL

Output

name.OUT
Output according to the specified option. (Type *HPP name.OUT* if HP4M output was requested, *PRINT name.OUT* otherwise.)

name.TEX
Title page in TEX format. To print, type "DOTEX name". Your output will be from the HP4M PostScript printer. **Note:** This file is created by the *Input* program for *Publication List*.

CLIST

This program creates tables of coordinates or anisotropic temperature factors suitable for publication. One of the output files, *name.TEX*, will create a TEXed table.

Input is a *name.TAB* file from *Least Squares* (created by the *LIST* option on the *END LEAST SQUARES* card). The output files are *name.OUT* and *name.TEX* of either coordinates x, y and z and U_{eq} and their estimated standard deviations or anisotropic U_{ij} values and their e.s.d.'s. Hydrogen and non-hydrogen coordinates may be listed separately.

Input cards

1)Title card—Format(A5)
1–5 *CLIST*

2)Spacing card—Format(A6,3X,I1,3X,I3)
1–6 ⟨Blank⟩/⟨*DOUBLE*⟩ ⟨Single⟩/⟨Double⟩ spacing. The default is single spacing.
10 ⟨Blank⟩/⟨*2*⟩ Heading is ⟨one⟩/⟨two⟩ lines long. The default is one line.
14–16 ⟨BLANK⟩/⟨###⟩ User supplied baselineskip is ⟨Blank⟩/⟨Number⟩. The default is blank.

3)Table title—Format(A70)
1–70 Title of table.

4)Table name—Format(A10)
1–10 Name of table (printed on each page of table).

5)Type card—Format(A5,1X,A4,3X,A4)
1–5 ⟨*COORD*⟩/⟨*ANISO*⟩ Table will include ⟨coordinates and U_{eq}⟩/⟨anisotropic U_{ij} values⟩. (The default is *COORD*.)
7–12 ⟨*NO H*⟩/⟨*H ONLY*⟩/⟨*ALL*⟩ List ⟨non-hydrogen⟩/⟨hydrogen only⟩/⟨all⟩ coordinates (*COORD* option only). (The default is *NO H*.)
14–17 ⟨*UEQ*⟩/⟨*B*⟩/⟨*BOTH*⟩ List ⟨U_{eq}⟩/⟨B⟩/⟨Both⟩ (*COORD* option only). (The default is U_{eq}.)

Description of individual cards

2)Spacing card

This card allows you to request that your table be double-spaced rather than single-spaced. You may also request that your heading consist of two lines rather than only one (to accommodate longer titles).

3)Table title and 4)Table name

These cards are used to specify the title of the table (e.g., *Coordinates * 10**4.*) and the table number (e.g., *Table I.*). The title will be printed at the beginning of the table while the table number will be printed on the top of each page.

5)Type card

This card is used to specify the type of table that will be created. If *COORD* is entered in columns 1–5, the table will consist of coordinates x, y and z and U_{eq} and their estimated standard deviations. Coordinates and U_{eq} and their e.s.d.'s are listed ($\times 10^4$). If the e.s.d. for any parameter is 0, that parameter is assumed to be constrained and no e.s.d. is listed. If .00001 $\leq$ e.s.d. $<$ .0001, the e.s.d. will be listed with a decimal point.

If *ANISO* is entered in columns 1–5, the table will include anisotropic U_{ij} values and their e.s.d.'s. Anisotropic U_{ij} values and their e.s.d.'s are listed ($\times 10^4$). If the e.s.d. for any parameter is 0, that parameter is assumed to be constrained and no e.s.d. is listed.

If the *COORD* option is specified, you may choose whether you want to include non-hydrogen, only hydrogen, or all coordinates in your table and whether you will be listing only U_{eq} values, only B values or some of both.

Note: This program does not require an *End* card. It may be followed by another program (including *Clist* to create a different table) or terminated with an *End X-ray* card.

Input Files Needed

name.CLI
name.TAB

Output

name.TEX

Typing *DOTEX name* creates a TEXed table from this file. It will be sent to the Laser printer. You may edit this file to, for example, get proper lower case letters for element symbols before typing *DOTEX*. (Note that it is usually easiest to edit your *name.TAB* file to get proper lower case letters; then all of the output will be correct.)

name.OUT

This table may be *PRINT*ed on the line printer.

DLIST

This program creates a table of distances and angles suitable for publication. Output includes a *name.TEX* file, which will produce a TEXed table. You may specify either single or double spacing (some journals, such as Inorganic Chemistry, require double spacing) and may ask for either one or two lines of heading.

Two input files are needed: the DLIST file described here and a *name.OUT* file of distances and angles as produced by *Distan*. The program will operate on all distances and angles in this *name.OUT* file; if some are not wanted or others are to be added, this file must be edited. (Other information included in the *name.OUT* file from *Distan*, such as plane calculations, will be ignored.)

Input cards

1)Title card—Format(A5,1X,A3,I2)
1–5 *DLIST*
7–9 Extension of file containing the desired distances and angles in *Distan* output form. (The default is *OUT*.)
10–11 Version number of file (as above). (The default is the highest version *name.ext* file in your directory.)

2)Spacing card—Format(A6,3X,I1)
1–6 ⟨Blank⟩/⟨*DOUBLE*⟩ ⟨Single⟩/⟨Double⟩ spacing. The default is single spacing.
10 ⟨Blank⟩/⟨*2*⟩ Heading is ⟨one⟩/⟨two⟩ lines long. The default is one line.

3)Table title—Format(A70)
1–70 Title of table.

3a)Table title continued—Format(A70) *Optional*
1–70 Title of table cont.

4)Table number—Format(A10)
1–10 Number of table (printed on each page of table).

Description of individual cards

1)Title card

This card may include the extension and version number of the file containing the desired distances and angles in *Distan* output format. *Distan* puts the information in a *name.OUT* file. Since all **CRYM** programs create a *name.OUT* file, you may want to *RENAME* the *Distan* output file to distinguish it from others: *RENAME name.OUT name.ext*. If this is done, you must enter the new extension in columns 7–9.

The version number of the file is entered in columns 10–11. The default is the highest version *name.ext* file in your directory. If you want an earlier version of a *name.OUT* or *name.ext* file, columns 7–9 may be blank, but columns 10–11 must contain the version number.

If columns 7–11 are all blank, the program will read the highest version *name.OUT* file in your directory.

2)Spacing card

This card allows you to request that your table be double-spaced rather than single-spaced. You may also request that your heading consist of two lines rather than only one (to accommodate longer titles).

3)Table title, 3a)Table title cont. and 4)Table number

These cards are used to specify the title of the table (e.g., *Distances and Angles.*) and the table number (e.g., *Table 3.*). The title will be printed only at the beginning of the table while the table number will be printed on the top of each page.

Note: This program does not require an *End* card. It may be followed by another program or terminated with an *End X-ray* card.

Input Files Needed

name.DLI
name.OUT (or *name.ext*)

Output

name.TEX

Typing *DOTEX name* creates a TEXed table from this file. It will be sent to the Laser printer. You may edit this file to, for example, get proper lower case letters for atom symbols before typing *DOTEX*. (Note that it is usually easiest to edit the input to the *Distan* program to get proper lower case letters; then all the output will be correct.)

name.OUT

This table may be *PRINT*ed on the line printer.

TLIST

This program creates Tables I and IV of crystal data (including cell dimensions, diffractometer information, final GOF and r-factor, etc.) suitable for publication in *Inorg. Chem.* from your *name.INF* file. Output includes files *nameI.TEX* and *nameIV.TEX* which will produce TeXed tables.

The *name.INF* file may be created at *Taperead* time or at any time by using the input program and requesting program *Begin*. You may also add or change information in the file by requesting the limited output option in *Begin*. Several programs (including *Least Squares*, *Initia*, and *Crystal Data*) add to and update the information in the *name.INF* if it exists at run time. The *name.INF* file is **not** binary and hence may be *READ*, *EDIT*ed or *TYPE*d. Be careful to retain the correct formats if you hand-edit the file.

Input Cards

1)Title card—Format(A5)
1–5 *TLIST*

2)Space group card—Format(12X,A20)
13–32 Space group name in TeX format (i.e. "P2_1").

Description of individual cards

2)Space group card

This card is used to enter the space group name. This should be entered in TeX math mode format, i.e.:

Enter	For
P1	$P1$
P\overline{1}	$P\overline{1}$
P2/n	$P2/n$
P2_{1}	$P2_1$

See *The TeXbook* by Knuth or your resident TeXpert for further help.

Note: This program does not require an *End* card. It may be followed by another program or terminated with an *End X-ray* card.

Input Files Needed

name.TLI
name.INF

Output

nameI.TEX, *nameIV.TEX*

Typing *DOTEX nameI* and *DOTEX nameIV* creates TEXed tables from this file. It will be sent to the Laser printer. You may edit this file to, for example, update your structure name before typing *DOTEX*. (Note that it is usually easiest to edit your *name.INF* file before running *Tlist*; then all future output will be correct.)

name.OUT

This table may be *PRINT*ed on the line printer.

Convert

This program converts CRYM format *Coord* and temperature factor (*Iso* and *Aniso*) cards to ORTEP format. The input cards may be contained in a *name.CRD* file created by *Least Squares* and/or included as part of the following input file. If the cards are in a *name.CRD* file, *Convert* will read the specified file directly. Output incudes a *name.CVT* file of the converted cards. Converted *Aniso B* cards will have a *1*, *Aniso U* cards a *4* and *Iso* cards a *6* in column 63. *Convert* accepts up to 600 *Coord* cards and requires the same number of temperature factor cards (for the same atoms) as *Coord* cards. The *name.CVT* file is complete for use by *Ortep* including the *1* in the first column of the last temperature factor card.

You may enter symmetry transformations to be applied to all converted atoms, in order to complete molecules that lie on symmetry elements. Furthermore, if you do not want all atoms to be included in your plot, you may request that certain atom types not be converted and/or that individual atoms be converted. These transformations and restrictions will be applied to all atoms on your atom list (*name.CRD* file and/or hand-entered atoms).

Input cards

1)Title card—Format(A4,4X,A3,I2)
1–7 *CONVERT*
9–11 ⟨*CRD*⟩/⟨Blank⟩ ⟨Do⟩/⟨Do not⟩ take *Coord* and temperature factor cards from your *name.CRD* file.
12–13 Version number of *name.CRD* file to be used (if input is contained in a *name.CRD* file). (The default is use of the highest version *name.CRD* file in your directory.)

2)Coord cards (if wanted)
See *Structure Factors*, *Least Squares* for description.

3)Iso and/or Aniso cards (to accompany *Coord* cards)
See *Structure Factors*, *Least Squares* for description.

4)Sym cards—Format(A3,1X,A30) *Optional*
1–3 *SYM*
5–34 Symmetry transformation in *free format* (i.e. $-X,1/2+Y,-1+Z$).

5)Omit card—Format(A4,10(1X,A3)) *Optional*
1–4 *OMIT*
6–8, 10–12, etc. Form factor labels of atoms **not** to be converted.

6)Atom cards—Format(A4,10(1X,A6)) *Optional*
1–4 *ATOM*
6–11, 13–18, etc. Atom identifiers of individual atoms to be converted.

7)End card—Format(A3)
1–3 *END*

Description of individual cards

1)Title card

This card is used to tell whether or not *Coord* and temperature factor cards should be taken from your *name.CRD* file. If not, cards 2 and 3 must be included in your input file. If your *name.CRD* file is read, you may enter additional *Coord* and temperature factor cards (cards 2 and 3) in the input file if desired.

2)Coord cards and 3)Aniso and/or Iso cards

See *Structure Factors*, *Least Squares* for a description of these cards.

If *Coord* and temperature factor cards are not going to be read from your *name.CRD* file, you must include these cards. If your *name.CRD* file is going to be used, you may also enter additional *Coord* and temperature factor cards in your input file if desired.

Note: *Convert* requires that there be both a *Coord* and a temperature factor (*Aniso* or *Iso*) card for each atom on the atom list.

4)Sym cards

The *Sym* cards are used to enter symmetry transformations. The coordinates and anisotropic temperature factors—β_{ij} or U_{ij}—of all converted atoms will be transformed accordingly and included in your *name.CVT* file along with the corresponding original atom. If a transformation results in the exact duplication of any original atom, the transformed atom will be rejected. Up to 10 transformations may be entered.

Symmetry transformations can be made **only for triclinic, monoclinic, or orthorhombic classes**. Symmetry transformations can be done in *Ortep* itself for any space group.

Transformations are entered in *free format* with the following restrictions:
1)Translation components must precede the letter (*−1/2+X* not *X−1/2*);
2)Fractions must be indicated by a "/" sign (*1/2+X* not *.5+X*);
3)Commas must separate *X*, *Y* and *Z* designators (*−X,Y,−1/3+Z*).

Blanks are permissible except on either side of a "/" sign. No transformation may exceed the 30-character field.

5)Omit card

This card is used to specify atom types **not** to be converted. For instance, if you do not want hydrogen atoms included in your plot, enter the form factor label *HS* on the *Omit* card. All atoms of a type *not* specified on the *Omit* card will be converted. Up to 10 form factor labels may be entered.

Individual atoms specified on an *Atom* card override atom types entered on an *Omit* card. Therefore, if you want only a few atoms of a particular type to be included, you may omit all atoms of that type with an *Omit* card and then specify the individual atoms to be included on an *Atom* card.

6)Atom card

This card is used to specify individual atoms on your atom list to be converted. If no *Omit* card and no *Atom* cards are used, all atoms will be converted. If no *Omit* card is used and *Atom* cards **are** included, only the individual atoms specified on the *Atom* cards will be conveted. If an *Omit* card is included, all atoms of a type **not** specified will be converted. The *Atom* cards are then used to request that atoms of an omitted type be included. Up to 100 atoms may be entered with up to 10 identifiers per card.

Your *name.OUT* file will notify you if *Coord* and temperature factor cards have not been found for any atom on your *Atom* cards.

Input files needed

name.CON

name.CRD (if requested)

Output

name.CVT and *name.OUT*

Coord and temperature factor cards including transformed parameters, if any, in ORTEP format.

H Optimize (HOPT)

This program converts General Plane Fourier map coordinates to atomic x, y and z coordinates. It will also optimize the positions of from 2 to 9 atoms (i.e. hydrogen atoms of an ammonia group or carbon atoms of benzene or cyclopentadienyl) so as to form a regular polygon.

The program uses information in your latest *name.HOP* file (created by General Plane Fourier) as well as the input file composed of the following cards. Since *name.HOP* is the name of the file created by Fourier, you **must** use a different name for your input file (i.e. *name.HPT*).

Output will include a *name.CRD* file of **Crym** *Coord* cards, one for each atom.

Input cards

1)Map card—Form(I2,1X,I1,1X,A3,1X,A3,1X,A5,1X,A3,1X,F5.2,1X,2F5.1, 1X,A3,F12.8)
1–2 The number of the General Plane map to be read from your latest *name.HOP* file. (Default is *1*.)
4 The number of atoms to be converted from this map. (Default is *3*.)
6–8 Form factor label to be assigned to *Coord* cards. (Default is *HS*.)
10–12 First three characters of atom identifier printed on *Coord* cards. (Default is form factor label.)
14–18 ⟨Blank⟩/⟨*ALPHA*⟩ The last character in atom identifier is ⟨numerical $(1, 2, 3 \ldots)$⟩/ ⟨alpha $(a, b, c \ldots)$⟩. (Default is numerical.)
20–22 ⟨Blank⟩/⟨*OPT*⟩ ⟨Do **not**⟩/⟨Do⟩ optimize the coordinates around the center of the polygon. (Default is to optimize.)
24–28 Radius from the ring center to any atom in Å. (Default is 0.9Å. Only used if optimizing atoms.)
30–34 Horizontal coordinate of the center of the polygon **if different** from the center of the map. (Only used if optimizing atoms.)
35–39 Vertical coordinate of the center of the polygon **if different** from the center of the map. (Only used if optimizing atoms.)
41–43 ⟨Blank⟩/⟨*ISO*⟩ ⟨Do Not⟩/⟨Do⟩ include iso cards in .CRD file for each of these coordinates.
44–45 Iso temperature factor.

2)Atom cards—Format(2F5.1,I4)
One card per atom specified on map card.
1–5 Horizontal position of atom in map units.
6–10 Vertical position of atom in map units.
11–14 Weight for this atom. (Default is *1*. Only used if optimizing atoms.)

Cards 1) and 2) may be repeated as needed.

3)End card—Format(A3)
1–3 *END*

Description of individual cards

1)Map card

This card contains the number of the *General Plane* map to be read from your latest *name.HOP* file. The *name.HOP* file contains two lines per map of information concerning the map sizing. The map number is the first number printed for each map. **Note:** The program reads the latest version of *name.HOP* only. If an earlier version is wanted, it must be renamed to the highest version; i.e., *RENAME name.HOP;1 name.HOP;3*.

Also included on this card is the number of atoms to be converted from this map. The default is *3*. You need one *Atom* card (card 2) per atom specified here.

This card includes information to be included on the *Coord* cards created by the program. The three-character form factor is wanted. The default is *HS*. This form factor is needed if you plan to use the *Coord* cards in *Structure Factors* or *Least Squares*.

A four-character atom identifier is included on the *Coord* cards to distinguish between the different atoms. All programs that use *Coord* cards require the atom identifier. The program will assign the fourth character of the label to the atoms, in increasing order—or, if the atom positions are optimized, increasing in counterclockwise order around the polygon starting with the first atom on the list. You may specify the first three characters of the label which will be the same for all of the atoms from the map. You may also specify whether the fourth character is to be alpha $(A, B, C, \ldots)$ or numeric $(1, 2, 3, \ldots)$. Thus specifying "C 2" in columns 10–12 and leaving columns 14–18 blanks results in identifiers "C 21", "C 22", "C 23",...; "H11" in columns 10–12 and *ALPHA* in columns 14–18 gives "H11A", "H11B", "H11C",...

If you want to optimize the positions of the atoms so as to create a regular polygon, you must enter the radius from the ring center to any atom in Å (the default is *0.9*Å). The map center is included in the *name.HOP* file. Therefore, you need enter the ring center only **if it is different from** the center of the map. You may also request the inclusion of an ISO card in the name.CRD file for the calculated COORD cards. The specified isotropic temperature factor will be assigned to each of the coordinates.

2)Atom cards

One atom card per atom as specified on your *Map* card is needed. The cards contain the horizontal and vertical coordinates of the atoms in map units. If the atom positions are to be optimized, you may also enter a weight for the atoms. The default is *1*.

The *Map* and *Atom* cards may be repeated as necessary. Hence, if your *name.HOP* file contains several maps, atoms from each map can be converted.

Input files needed

name.HOP

Output Files

name.OUT
A small file which may be of some use if the program fails.

name.CRD
Crym format *Coord* cards for converted atoms (optimized if requested).

Miscellaneous Routines

CHEMDRAW

To convert from DISTAN file COORDS for some program from distances and angles and FFMURO. Replaces Hs with H.

Creates:name.CHD file

DELBACK

Reads edited name.OUT file (from BACKGROUND) selecting reflections with background GOF values x sigma below average value (absolute values)

Creates:name.UPD and 2 name.DEL files

DELBAD

Reads name.BAD file selecting reflections with F values x sigma below average F values

Creates:Sorted name.UPD and original (.BAD) ordered name.WOR file of offending reflections

HB

Reads name.TAB file calculating B values =1.2x B(eq)

Creates:name.HB

OPEN

Genreal introductory part of programs that gets information about file names, file name lengths, adds extensions to file names and opens files

TRAN

similar to OPEN

TWICE

Reads name.DAT file (from CAD-4) that had duplicate numbering of reflections–renumbers them in proper order

Creates:name.NEW

KAF

TRANS

Take the Least Squares parameter file and edit out all but the coordinate cards. Insert a card at the front with the name of the compound and a blank card at end for input

ADD

Program to put ADD the W(prime)'s from one tape to another after modifying mathematically

FIXHKL

Modifies HKL file and writes new file

BJW

ADD2

Program to put ADD the HKL's from one tape to another. Will write tapel + HKL of tape 2 to FOR010.dat. Will also rescale Fobs and weights of tape 2.

CLEAR

To clear terminal

CLEARVT

Clear VT125 terminal

DETERM

Something to do with determinants??

EDREM

Special version of EDMAG for REM (note: our only FORTRAN version of EDMAG)

HKLADD

Calculates 2theta, chi, phi from the orientation matrix and writes values on a HKL disk.

INTPLT

I don't know what it does, has copyright of 1968

LATCON

A General Program for the LS Refinement of Lattice Constants. One-pass, noninterative. Lots of information in comments.

LINE

Calculates number of lines per page

NEWPAT

NONIUS

NONIUSY

NUMSRCH

Search for numbers in character string

REMFIX

Read 2 Real numbers to put in columns as integers

RDHKL

Read HKL disk and multiply bweights by 1000

READIN

Read INT data, calculate weight from sigma, write short format data cards as input to INITIA

ROW

SIGTOWT

Changes entered SIG to weight

SPLICE

Splices columns n1–m1 of file1 with columns n2–m2 of file2

SPLINEFIT

Interpolates sorted data using Diverill's cubic spline

SQ

Read Short Format and write new file SQRT of FSQ*10

SUM

Add up columns

SUMUP

Compare and sumup Fobs and Fcalc from two separate files

TDASHL

List QMSplot features

WRTPE

Reads HKL.CRD file and writes to Tape (for FHH)

Creates:

About Crunch 1.5 (with source code and scripts)

by R. de Gelder, R.A.G. de Graaff & W.J. Vermin

BFSC Leiden Institute of Chemistry, Gorlaeus Laboratories, PO Box 9502, 2300 RA Leiden, The Netherlands. E-mail: rag@chem.leidenuniv.nl ; r.degelder@science.ru.nl

Place holder page. Report-proper starts on following page

About Crunch1.5

R. de Gelder, R.A.G. de Graaff & W.J. Vermin

BFSC Leiden Institute of Chemistry
Gorlaeus Laboratories
PO Box 9502, 2300 RA Leiden
The Netherlands

October 26, 2009

Email:
rag@chem.leidenuniv.nl (R.A.G. Graaff)
R.de Gelder@science.ru.nl (Rene de Gelder)

1 Introduction

Crunch uses the concurrent maximization of the determinants of small Karle–Hauptman matrices to get to phases that bear some relation to the structure that is being looked for. For the development of this concept the reader is refered to the papers mentioned below:

On the construction of Karle–Hauptman matrices.
Acta Crystallographica A46 (1990), 688-692,
R de Gelder, R.A.G. de Graaff & H. Schenk.
X-ray Department Gorlaeus Laboratories
PO Box 9502, 2300 RA Leiden
The Netherlands

Automatic Determination of Crystal Structures
using Karle-Hauptman Matrices.
Acta Crystallographica A49 (1993), 287-293,
R. de Gelder, R.A.G. De Graaff & H. Schenk.
X-ray Department Gorlaeus Laboratories
PO Box 9502, 2300 RA Leiden
The Netherlands
and
On the automatic extension of incomplete models by
iterative Fourier calculation.
Journal of Applied Crystallography 17 (1984), 364-366,
A.J. Kinneging & R.A.G. de Graaff.
Department of Biophysical Structural Chemistry
PO Box 9502, 2300 RA Leiden
The Netherlands

Looking at Crunch, there are fundamentally three stages which must be considered. All are discussed in the papers quoted.

1. Checking the input, calculating E's if necessary and the construction of a set of matrices containing a sufficient number of independent reflections to be phased. (Crunch.uni, thinkc and deter in Cruncher.uni)

2. Maximizing the product of the determinants of the matrices obtained as a function of the phases, starting from some point in reciprocal space. (pmf and deter, Cruncher.uni)

3. Evaluating the phaseset obtained by trying to find a complete model based on the map obtained by Fourier transformation of the phaseset (autofour, Cruncher.uni)

Re 1. At least N reflections should be phased where N is not less than twice the number of atoms to be found. This is why more than one matrix are needed. Deter constructs the matrices, maximizing the average E-value of the reflections present in the matrices. Also care is taken to ensure that reflections and their symmetry equivalents occur in the matrices a sufficient number of times. Thinkc determines whether the right input to deter, specifying the matrix to be used in phase refinement, has been found.

This section of Crunch is governed by the shell scripts Crunch.uni and Cruncher.uni. Crunch.uni also reads the reflection data, calculates E-values and generally sets the stage for the structure determination iterations which are carried out by the script Cruncher.uni. Both scripts are given in an appendix to this paper.

Re 2. This is an extension of the maximum determinant rule first formulated by Tsoucaris (Tsoucaris, 1970). Pmf calculates Patterson compliant starting sets, which according to Sheldrick provide a better starting point to the phase determination process. The maximum determinant rule states that the most probable set of phases of the reflections present in a Karle–Hauptman matrix maximizes the value of the determinant of the matrix. In deter the maximization is done following the method described in de Gelder, de Graaff and Schenk, 1993.

Re 3. The reflections present in the matrices represent a non-random selection from reciprocal space. This means that direct Fourier transformation after phasing usually produces a map which is not a good approximation of the true map. False peaks are present and many atoms may be missing. Autofour (Kinneging and de Graaff, 1984) takes care of this if the phases found by deter are good enough.

2 Phase refinement

In this section the source text of the key routine is presented, together with an explanation of what happens where and when.

```
module modopdet
use modunit
use moddet1
use modjaap1
use modschuif
implicit none
contains
SUBROUTINE OPDET(NHKLF,EMX,PMX,IJPA,IJQA,DETN,ICYCMX)
implicit none
integer,                     intent(in)    :: nhklf
real,    dimension(:,:,:), intent(inout) :: emx,pmx ! (nm,nm,nnrm)
integer, dimension(:,:),   intent(in)    :: ijpa    ! (7,nfs)
integer, dimension(:,:),   intent(in)    :: ijqa    ! (8,nijqa)
real*8,                      intent(out)   :: detn
```

```
      integer,                   intent(in)    :: icycmx

!
!  OPDET MAXIMIZES THE DETERMINANT
!  SEE ACTA CRYSTALLOGRAPHICA
!  LAPACK ROUTINES ZPOTRF, ZPOTRI ARE USED TO FACTORIZE AND INVERT
!  KH MATRIX
!
!  NNRM DETERMINANTS ARE MAXIMIZED CONCURRENTLY
!  ONLY REFLEXIONS WITH E>ELIM ARE CONSIDERED BY OPINIT
!
!  PARAMETERS
!
!
!  EMX(NM,NM,NNRM) = INPUT, output =
!  E-VALUES IN THE LOWER TRIANGLE
!
!  PMX(NM,NM,NNRM) = INPUT, OUTPUT =
!  INPUT: START PHASES IN RADIANS
!  OUTPUT: REFINED PHASES IN RADIANS
!
!  NM = INPUT =
!  DECLARED DIMENSIONS OF THE MATRICES
!
!  AMX(NM,NM,NNRM)
!  SCRATCH ARRAY, AFTER DET1, JAAP1 HAVE BEEN CALLED LOWER TRIANGLE CONTAINS
!  THE INVERSE OF KH MATRIX, THE UPPER CONTAINS ORIGINAL VALUES STILL
!
!  IJPA,IJQA: = INPUT = SEE OPINIT
!  W(NFS,7): SCRATCH
!  FAS: SCRATCH PHASES
!  GRAD: SCRATCH GRADIENTS
!  FDET(NNRM): SCRATCH, INDIVIDUAL DETERMINANT VALUES FOR EACH MATRIX
!
      COMPLEX*16, dimension(:,:,:), allocatable :: AMX
      real*8,     dimension(:),     allocatable :: fdet,grad
      real,       dimension(:),     allocatable :: fas
      real,       dimension(:,:),   allocatable :: w

      integer NNRM,NM,NIJQA,NFS,ifail
      INTEGER CONV,DIAG
      DIMENSION DT(9)
      REAL*8 V,VERM,DETO
      integer mk,nc,ma,lk,iry,iko,ifud,icycs
      integer j,icyc,icv,i,l
      real x,vfud,twopi,thpid2,pid12,pid2,dt
      real z,y,pi,hoek,fudmin,gstop,fudmax,dfud
      logical converged

!
!  DIAG = NUMBER OF TIMES THE ELEMENTS OF THE MATRIX HAVE BEEN
!  DECREASED BY 10% AT THE START, IF THE DETERMINANT IS NEGATIVE.
!  THIS IS DONE TO ENSURE THAT AT THE BEGINNING OF THE REFINEMENT
!  THE MATRIX IS POSITIV DEFINITE
!
!  THE PHASES
!
!     DATA IFAS /6/
!
!  THE FUDGE FACTORS
!
      DATA IFUD /6/
!
!  W(*,1-3)  SHIFTS OF THE LAST THREE ROUNDS
!
!  THE GRADIENTS
!
!     DATA IGRAD /4/
!
!  STORAGE OF PHASES FROM LAST CYCLE
!
      DATA ICV /5/
!
!  MAXIMUM OF ANY FUDGE FACTOR
!
      DATA FUDMAX /3.0/
!
!  MINIMUM OF ANY FUDGE FACTOR
!
      DATA FUDMIN /.01/
!
!  FUDGE FACTORS ARE MULTIPLIED BY VFUD IF DETERMINANT INCREASES
!
      DATA VFUD /1.1/
!
!  FUDGE FACTORS ARE DIVIDED BY DFUD IF NOT
```

```
!
      DATA DFUD /1.5/
!
!  THE ELEMENTS OF THE MATRIX ARE MULTIPLIED BY V IF DET < 0
!  UNTIL THIS IS NO LONGER TRUE. ON CONVERGENCE THIS IS REVERSED.
!  DIAG IS USED TO KEEP TRACK OF THIS.
!
      DATA V /0.9/
!
!  REFINEMENT STOPS IF THE CHANGE IN THE VALUE OF DET IS SMALLER THAN
!  GSTOP OVER THE LAST 10 CYCLES
!
      DATA GSTOP /0.01/
!     DATA GSTOP /0.005/

      nnrm = size(emx,3)
      nm   = size(emx,1)
      nfs  = size(ijpa,2)
      nijqa = size(ijqa,2)

      allocate(amx(nm,nm,nnrm))
      allocate(fdet(nnrm))
      allocate(w(nfs,7))
      allocate(fas(nfs))
      allocate(grad(nfs))

      PI     = 4.0*ATAN(1.0)
      TWOPI  = 2.0*PI
      PID12  = PI/12.0
      PID2   = PI/2
      THPID2 = 3*PID2

      VERM   = 1.0
      ICYCS  = 0
      DIAG   = 0

      nfs_eq_0: IF(NFS.eq.0) then
!
! NO REFINEMENT
!
        WRITE(LO,*)'NHKLF=  ',NHKLF
        IF(NHKLF.EQ.1)THEN
!
```

```
!  INPUT PHASES ARE USED AND CHECKED
!
          CALL DET1(EMX,PMX,AMX,
     1     IJPA,IJQA,FAS,DETO,FDET,IFAIL,VERM)
          IF(IFAIL.EQ.0)THEN
            WRITE(LO,3)DETO
    3       FORMAT(' OPDET: WITHOUT REFINEMENT THE DETERMINANT IS: ',
     1             E12.4)
          ELSE
            WRITE(LO,4)
    4       FORMAT(' OPDET: PHASES PUT IN RESULT ',
     1             'IN A NEGATIVE DETERMINANT')
          ENDIF
        ENDIF
      else

        convloop: do conv=0,1
          converged = .false.
          if(conv .eq. 1) then
            WRITE(LO,*)
     1      ' OPDET: NOW REFINE WITH SPECIAL REFLEXIONS RESTRICTED'
          endif
          DO I=1,NFS
!
!  GET ROW AND COLUMN OF VARIABLE I
!
            IRY=IJPA(1,I)
            IKO=IJPA(2,I)
            MA=IJPA(3,I)
!
!  CORRECT RESTRICTED PHASE
!
            Z=PMX(IRY,IKO,MA)
            MK=IJPA(6,I)
            IF(MK.ne.1) then
              Y=(MK-1)*PID12
              X=AMOD(Z-Y,TWOPI)
              IF(X.LT.0.0) X=X+TWOPI
              IF(X.GT.PID2.AND.X.LT.THPID2) Y=Y+PI
              Z=Y
            endif
            Z=AMOD(Z,TWOPI)
            IF(Z.LT.0.0) Z=Z+TWOPI
```

```
           FAS(I)=Z
         enddo
         LK=0
         VERM=1.0
!
!  CALCULATE DETERMINANT
!
         mainloop: do
           calcdet: do
             DO I=1,NFS
               W(I,IFUD)=0.3
             enddo
             DO MA=1,NNRM
               DO I=1,NM
                 EMX(I,I,MA)=1.0
               enddo
             enddo
             CALL DET1(EMX,PMX,AMX,
     1                IJPA,IJQA,FAS,DETO,FDET,IFAIL,VERM)
             IF(IFAIL.EQ.0) exit calcdet
             IF(LK.EQ.1) exit convloop

             DIAG=DIAG+1
             VERM=VERM*V
           enddo calcdet

           WRITE(LO,1) DETO
    1      FORMAT(' OPDET: INITIAL VALUE OF THE DETERMINANT=',E20.10)
           ICYC=1
           NC=0
           cycleloop: do
!
!  STORE PHASE IN W
!
             DO I=1,NFS
               W(I,ICV)=FAS(I)
             enddo
!
!  CALCULATE INVERSE MATRIX
!
             CALL JAAP1(AMX,IJPA,IJQA,GRAD,FDET)
!
!  THE LOWER TRIANGLES OF AMX CONTAIN THE INVERSE OF AMX
!  NOW CALCULATE SHIFTS WITHOUT FUDGEFACTORS
!

             CALL SCHUIF(AMX,W(:,1),W(:,ICV),GRAD,IJPA,
     1                    IJQA)

             loop: do

               if ( .not. any(w(1:nfs,ifud) .gt. fudmin)) then
                 exit cycleloop
               endif

               DO I=1,NFS
                 MK=IJPA(6,I)
                 IF(MK.eq.1.or.CONV.ne.1) then
                   HOEK=W(I,ICV)+W(I,1)*W(I,IFUD)
                   FAS(I)=AMOD(HOEK,TWOPI)
                   IF(FAS(I).LT.0) FAS(I)=FAS(I)+TWOPI
                 endif
               enddo
!
!  NOW RECALCULATE DETERMINANT USING NEW PHASES
!
               CALL DET1(EMX,PMX,AMX,
     1           IJPA,IJQA,FAS,DETN,FDET,IFAIL,VERM)
               if (ifail .eq. 0 .and. detn .gt. deto ) exit loop
!
!  DETERMINANT NEGATIVE OR SMALLER THAN THE OLD VALUE
!
               DO I=1,NFS
                 W(I,IFUD)=W(I,IFUD)/DFUD
               enddo
             enddo loop

             NC=MIN0(10,NC+1)
             if(nc .ge. 10) then
               X=(DETN-DT(1))/DETN
               IF(X.LT.GSTOP) exit cycleloop
               DO I=1,8
                 DT(I)=DT(I+1)
               enddo
               NC=NC-1
             endif
```

```
      DT(NC)=DETN
      if (icyc .ge. icycmx) then
                write(lo,*)' OPDET: TOO MANY CYCLES'
              else
                ICYC=ICYC+1
                ICYCS=ICYCS+1
                DETO=DETN
                DO I=1,NFS
                  W(I,IFUD)=AMIN1(W(I,IFUD)*VFUD,FUDMAX)
                enddo
              endif
            enddo cycleloop

            L=0
            IF(DIAG.GT.0) THEN
              LK=1
              DIAG=DIAG-1
              L=1
            ENDIF ! wwvv in original, this endif was placed after next endif
            IF(DIAG.EQ.0)THEN
              VERM=1.0
            ELSE
              VERM=VERM/V
            ENDIF
!
!  INCREASE THE OFF-DIAGONAL ELEMENTS OF THE MATRIX AND  REFINE AGAIN
!
            if (l .eq. 0) exit mainloop
          enddo mainloop

          converged = .true.
        enddo convloop

        if (converged) then
          WRITE(LO,*) ' OPDET: CONVERGENCE OBTAINED'
        else
          WRITE(LO,*)
     1     ' OPDET: NO CONVERGENCE, MATRIX IS NOT POSITIVE DEFINITE'
        endif

        WRITE(LO,10001) ICYCS,DETN
10001   FORMAT(' OPDET: NUMBER OF CYCLES =',I5,' DETERMINANT=',
     1           E20.10)
```

```
  DO MA=1,NNRM
    DO I=1,NM
      DO J=1,I
        PMX(I,J,MA)=AMOD(PMX(I,J,MA),TWOPI)
        IF(PMX(I,J,MA).LT.0.0) PMX(I,J,MA)=PMX(I,J,MA)+TWOPI
      enddo
    enddo
  enddo

endif nfs_eq_0$

deallocate(amx)
deallocate(fdet)
deallocate(w)
deallocate(fas)
deallocate(grad)

RETURN
END SUBROUTINE opdet
end module modopdet
```

3 Model extension

The program autofour, run by Cruncher.uni, evaluates and tries to extend the model obtained after phase refinement. The program uses iterative Fourier calculations to improve and extend the model. The individual atoms in the model are checked continuousy using a critrion based on R_2 Wrongly placed atoms are deleted from the model as soon as is possible, while the quality of the model is checked against the expected value of the criterion mentioned earlier. For a description of the procedure see Kinneging and de Graaff, 1984.

4 The scripts

Ths script crunch.uni is prepared during the installation of Crunch. Crunch.uni governs the iniial stages of the procedure as well as the communication with the user. E-values may be calculated in two different ways, things like cel contents and other input parameters are checked and stored. See for more information the manual which is appended.

Crunch.uni:

```
#!/bin/bash
# *********************************************************|
# Attention: Specify the bash shell directory on the       |
#            1-st line of this script !!!                  |
# *********************************************************|
# Script:   crunch.uni                                     |
# Version:  Crunch 1.5      Date: 1/10/2009                |
# Language: Unix (Bash shell)                              |
# Purpose:  Input driver for CRUNCH                        |
# CRUNCH:   Automatic Determination of crystal structures  |
#           from single-crystal X-ray diffraction data     |
# *********************************************************|
#                                                          |
# CRUNCH:   Idea initially conceived at the University of   |
#           Leiden in the period 1988-1992.                |
#           For information or help please contact:        |

# Authors:  Dr. Rene de Gelder                             |
#           Department of Inorganic Chemistry              |
#           University of Nijmegen                         |
#           Toernooiveld 1                                 |
#           6525 ED Nijmegen                               |
#           The Netherlands                                |
#           e-mail: R.deGelder@science.ru.nl               |
#                                                          |
#           Dr. R.A.G. de Graaff                           |
#           Leiden Institute of Chemistry                  |
#           Gorlaeus Laboratories                          |
#           University of Leiden                           |
#           PO Box 9502                                    |
#           2300 RA Leiden                                 |
#           The Netherlands                                |
#           E-mail: rag@chemb0b.leidenuniv.nl              |
#                                                          |
# *********************************************************|
#   DETER (Multi-matrix)  <----------------------------    |
#     |                                                |   |
#     v                                                |   |
#   VEXFFT                                             |   |
#     |                                                |   |
#     v                                                ^   |
#   EXFFT80               ---                          |   |
#     |                  |Yes| ->- SEARCH80            |   |
#     v                   ---                          |   |
#   SEARCH80               ^                           |   |
#     |                    |                           |   |
#     v            -----------------      --           |   |
#   AUTOFOUR ---> |Structure solved?| -> |No| ---------    |
#                  -----------------      --               |
#   Koppiereit (c) 1992 Rene de Gelder Leiden University   |
# *********************************************************|
#   Last revision:
    datum="The Fall 2009"
#   by Rag de Graaff
#
#
#  You may change the default directory structure here
#  However, this should not be necessary
#
#  Root directory for Crunch:             dcru
#  Directory for Crunch-executables:      dpro
#  Directory for structure files:         dstr
#
#
#
# define some functions:
#
 echon ()
 {
   echo 1 | tr -d '\\012'
# end echon
 }
#
 do_help()
 {
   echo "CRUNCH supported syntax:"
   echo
   echo "Option 1: crunch   code    conhkl filename"
   echo "Option 2: crunch   code    run "
   echo "Option 3: crunch   code    run    start "
   echo "Option 4: crunch   code    run    start    end "
   echo "Option 5: crunch   code    run    start    end    peaks"
   echo "Option 6: crunch   code    run    start    end    peaks    num"
   echo "Options 7,8 : crunch help or crunch manual"
   echo
   echo "code  :  compound name "
   echo "conhkl:                -reflection file conversion (create .frefa)"
   echo "help, manual           -display the manual provided"
   echo "run   :  clear         -remove all files created by earlier runs"
   echo "                        excepting .frefa and .crysda files"
   echo "run   :  blessing      -run LEVY and EVAL only; calculate E-values"
   echo "run   :  pmf           -run pmf to create patterson compliant starts"
   echo "run   :  deter         -run DETER only; calculate phases"
   echo "run   :  peaks         -locate peaks in maps from DETER phases"
   echo "run   :  autofour      -run AUTOFOUR only; start from peak-file(s)"
   echo "run   :  try           -run CRUNCH until solution is found"
   echo "run   :  recycle       -run CRUNCH-recycling until solution is found"
   echo "run   :  collect       -run CRUNCH and collect solutions"
   echo "start :  1......end"
   echo "end   :  start....."
   echo "peaks :  all           -instead of largest fragment take all peaks"
   echo "peaks :  first         -instead of fragment take first num peaks"
   echo "peaks :  last          -instead of fragment take last num peaks"
   echo "num   :  1......all    -number of peaks selected form all peaks"
   echo
   exit
# end do_help
 }
#
do_prepare()
 {
#
# Check whether .crysda and .frefa files are present
#
   if [ ! -f "$dstr/$1.frefa" ]; then

     echo "The file" $1".frefa is not present."
     echo "You may create this file with the command:"
     echo " crunch $1 conhkl DATAFILE"
     echo "where DATAFILE is the name of your file with hkl and F values"
     exit
   fi
   if [ ! -f "$dstr/$1.crysin" ]; then
     csin="false"
   else
     csin="true"
     ln -s $1.crysin fort.12
   fi
   if [ -f "$dstr/$1.crysda" ]; then
     csda="true"
     ln -s $1.crysda fort.11
   else
     csda="false"
   fi
#
#   This was to check whether .crysin and/or .crysda files are present
#
   if [ ! "$csda" = "true" ] || [ ! "$csin" = "true" ]; then
     echo $1 > temp
     echo $csin >> temp
     echo $csda >> temp
     ln -s temp fort.10
#
#   crys is based on the Dirdif crysda program
#   .crysin is converted into .crysda file or vice versa
#   if required the user is prompted for data
#
     $dpro/crys
     if [ "$csin" = "false" ]; then
       mv fort.12 $dstr/$1.crysin
     fi
     if [ "$csda" = "false" ]; then
       mv fort.11 $dstr/$1.crysda
     fi
   fi
   rm -f fort.10
   rm -f fort.11
   rm -f fort.12
   rm -f temp
# end do_prepare
 }

do_conhkl()
 {
#
# Start of hkl-file conversion
#
   echo "========================================================"
   echo "*** CONHKL  conversion of reflection files"
#   Check input syntax
   if [ $# -ne 3 ]; then
     echo "========================================================"
     echo "==> syntax error!!"
     echo "==> syntax: crunch code conhkl inputfile"
     echo "========================================================="
     exit
   fi
   p1=$3
   p2=$1
# Check whether datafile is present
   if [ ! -f "$dstr/$p1" ]; then
     echo "========================================================="
     echo "==> datafile specified is not present!"
     echo "========================================================="
     exit
   fi
   echo "========================================================="
   echo
   echo "*** The first record on your reflection-file is:"
   echo
   head -1 $dstr/$p1
   echo "==> You must now specify your recordtype using keywords"
   echo
   echo "*** Example: hkl skip fobs sigma "
   echo
   echo "*** This means:"
   echo "*** The first 3 numbers are h,k,l, the 4th should be skipped,"
   echo "*** the next one is Fobs and the last one is sigma(Fobs)"
   echo "*** Typing f2obs instead of fobs tells the program your file"
   echo "*** contains F**2 rather than F"
   echo
   echon "==>  Specify your record: "
   read record
   echo "========================================================="
   echo $record   > $dstr/temp1
   cat $dstr/$p1 >> $dstr/temp1
   ln -s $dstr/temp1 fort.10
   ln -s $dstr/temp2 fort.11
   $dpro/conhkl
   echo ’FREFA  ’$p2 > $dstr/temp
   cat $dstr/temp $dstr/temp2 > $dstr/$p2.frefa
   rm -f $dstr/temp $dstr/temp1 $dstr/temp2
   rm -f fort*
   echo "*** conhkl ended"
   echo "========================================================="
   exit
# end do_conhkl
```

```
 }

do_clear()
 {
   (
     cd $PWD
     rm -f *.lis* temp* *.cout nohup.out *.cdat crunch* deter* $1.ref rm -f
     $1.red $1.d15  $1.mul $1.nor $1.nip $1.pub  $1.contents $1.matrix rm -f
     $1.report $1.par $1.nrl $1.aut $1.nrlbak $1.parbak $1.atoms $1.spf rm -f
     $1.pdb $1.xyz $1.phi* $1.dfs $1.exf $1.fou* $1.inf $1.lsq rm -f $1.map
     $1.mpa $1.pek* $1.fra $1.sym $1.symt $1.o $1.pjn rm -f $1.pls $1.fref
     $1.freffo multi.tpr autofour.* autofour.err hits.log hits
     rm -f output logfile nohup.out $1.ddlog dd.log $1.binfo IPR* CRYSDA
     rm -f CRYSIN $1.crysda FREFFO FREFA AT* CON* DDLOG XYZ*
     rm -f BINFO mogli.* pluton.* picture fragment* fort.* *.asc
     rm -f *.lp *.eee *.hkl eval.* *levy* *.refl random.seed resolution
   )
   exit
# end do_clear
 }
 do_pgrp()
 {
 cat $dstr/$p2.crysda | grep ISYST > fort.35
 cat $dstr/$p2.crysda | grep SPGR >> fort.35
 $dpro/pgrp
 }
#
# End of point group determination
#
 do_mklevy()
 {
 echo $p2 > levy.dat
 echo ' 6' >> levy.dat
 echo $dstr/$p2.frefa >> levy.dat
 cat fort.36 >> levy.dat
 cat $dstr/$p2.crysda | grep 'CELL ' > tmp
 read dummy cell < tmp
 echo $cell >> levy.dat
 rm tmp
 cat $dstr/$p2.crysda | grep 'Z ' > tmp
 read dummy Z dum < tmp
 echo $Z >> levy.dat
 rm tmp
 cat $dstr/$p2.crysda | grep 'WAVE' > tmp
 read  dummy Z dum< tmp
 if [ $Z = 'SY' ]; then
 Z='SYNCHROTRON'
 fi
 echo $Z >> levy.dat
 rm tmp
 cat $dstr/$p2.crysda | grep 'NTYPE' > tmp
 read dummy Z dum < tmp
 echo $Z >> levy.dat
 rm tmp
 cat $dstr/levy.con >> levy.dat
 echo '0' >> levy.dat
 echo '0.000 1.40' >> levy.dat
#echo '0.0 0.0' >> levy.dat
 rm -f fort.*
 }
#
# End of preparing input for Levy
#
 do_levy()
 {
 ln -s $dstr/levy.dat fort.10
 ln -s $dstr/levy.lp fort.60
 ln -s $dstr/$p2.refl fort.11
 $dpro/levy
 grep "BISO =   " $dstr/levy.lp > $dstr/biso.levy
 mv fort.12 eval.dat
 rm -f fort.*
 }
#
# End of levy run
#
 do_eval()
 {
 ln -s $dstr/eval.dat fort.10
 ln -s $dstr/eval.lp fort.60
 ln -s $dstr/$p2.refl fort.11
 ln -s $dstr/edata.hkl fort.13
 ln -s $dstr/edata.eee fort.15
 $dpro/eval
 rm -f fort*
 }
```

```
#
# End of E-value calculation following Blessing
#
#
# end of function definitions
#
#
# Start of executable statements
#

 dcru="/Users/rag/test/crunch1.5/drivers"
 dpro="/Users/rag/test/crunch1.5/programs"
#
# The next two lines enable you to run Crunch from
# any directory you happen to be in.
# It is strongly recommended not to change this.
#
 PWD=‘pwd‘
 dstr=$PWD
#
 rm -f  $PWD/fort.*
#
#   Show manual on screen if required
#
 case "$1" in
   help | manual | "" )
     if [ -f "$dcru/../manual" ]
     then
       more "$dcru/../manual"
       exit
     else
       do_help
       exit
     fi
   ;;
 esac
#
#   Store the type of run
#
 p1=$2
#
#   Check input syntax
#
```

```
case "$2" in
  conhkl   ) do_conhkl "$@" ;;
  clear    ) do_clear "$@" ;;
  deter | peaks | autofour | collect | recycle | try | blessing  | pmf ) : ;;
  *        ) do_help
  exit
  ;;
esac

do_prepare "$@"

case "$2" in
  blessing  )
    set $1 $2 1 1
  ;;
  deter | peaks | recycle )
    if [ "$3" = "" -o "$4" = "" ]
    then
      echo "Syntax error"
      echo
      echo "Syntax: crunch   code    run    start    end "
      echo "run  :  deter"
      echo "run  :  peaks"
      echo "run  :  recycle"
      echo "start:  1......end"
      echo "end  :  start....."
      echo "==> values for start and end must be specified "
      exit
    fi
  ;;
  autofour )
    if [ "$3" = "" -o "$4" = "" ]
    then
      set $1 $2 0 0
    fi
  ;;
  try | collect )
    if [ "$3" = "" -o "$4" = "" ]
    then
      # default try or collect 10 cycles
      set $1 $2 1 10
    fi
  ;;
```

```
 esac

 if [ "$3" -gt "$4" ]; then
   echo "Syntax error"
   echo
   echo "Syntax: crunch   code    run    start    end "
   echo "start:  1......end; you have specified: " $3
   echo "end  :  start.....; you have specified: " $4
   echo "==> end should be larger than start!?"
   exit
 fi
 if [ "$5" != "" -a "$5" != "all" -a "$5" != "last" -a "$5" != "first" ]
 then
   echo "Syntax error"
   echo
   echo "Syntax: crunch   code    run    start    end    peaks"
   echo "Possible values of peaks are the literals:  all, first or last"
   echo "==> you did not specify the value of peaks correctly"
   exit
 fi
#   end of input check
#
#   Read directory structure, otherwise install starting (dummy) structure
#
#   Check presence of directory structure file
#
#check directory structure, dstr is OK, because .crysin
#and .frefa are there.
#
 if [ ! -f "$dcru/cruncher.uni" ]; then
   echo "The crunchdirectory is wrongly specified."
   echo "Change this in crunch.uni"
   exit
 fi
 if [ ! -f "$dpro/autofour.exe" ]; then
   echo "The directory specified for the Crunch executables is incorrect."
   echo "Change this in crunch.uni"
   exit
 fi
#
#check automatic run
#
 p2=$1
 cat $dstr/$p2.crysda | grep  "ICENT"  > $dstr/$p2.cent
 read dummy icent center < $dstr/$p2.cent
 cat $dstr/$p2.crysda | grep "SPGR" > $dstr/$p2.lattice
 read dummy lattice s1 s2 s3 < $p2.lattice
 rm -f $dstr/$p2.lattice
 cat $dstr/$p2.crysda | grep "NLATT" > $dstr/$p2.nlatt
 read dummy nlatt dummy < $dstr/$p2.nlatt
 rm -f $dstr/$p2.cent
#
#Symmetryoperations in matrixform written
#as well as lattice translations and possible centre symmetry
#
 cat $dstr/$p2.crysda | grep "NSYMM" > $dstr/$p2.matrix
 cat $dstr/$p2.crysda | grep "SYMMAT" >> $dstr/$p2.matrix
 cat $dstr/$p2.crysda | grep "ILATT" >> $dstr/$p2.matrix
 rm -f $dstr/$p2.nlatt
 cat $dstr/$p2.crysda | grep "ISYST" > $dstr/$p2.syst
 read dummy isyst dummy < $dstr/$p2.syst
 cat $dstr/$p2.crysda | grep "NSYMM"> $dstr/$p2.syst
 read dummy nsym < $dstr/$p2.syst
 cat $dstr/$p2.crysda | grep "NSXYZ"> $dstr/$p2.syst
 read dummy nsxyz < $dstr/$p2.syst
 rm -f $dstr/$p2.syst
 echo $isyst > $dstr/$p2.symt
 echo $icent >> $dstr/$p2.symt
 echo $nsym  >> $dstr/$p2.symt
 echo $nsxyz >> $dstr/$p2.symt
 cat $dstr/$p2.crysda | grep "SYMMAT" >> $dstr/$p2.symt
 cat $dstr/$p2.crysda | grep "SYMIT" >> $dstr/$p2.symt
#
# Everything needed is stored in $p2.symt now
#
 ln -s $dstr/$p2.symt  fort.10
 ln -s $dstr/$p2.sym   fort.12
 $dpro/symt
 rm -f $dstr/$p2.symt
 rm -f fort.10
 rm -f fort.12
 if [ -f "$dstr/$p2.par" ]; then
   read p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14 < $dstr/$p2.par
   rm -f $dstr/$p2.par
 else
   cat $dstr/$p2.crysda | grep "IMULT" > $dstr/$p2.inft
   read dummy p6 dummy < $dstr/$p2.inft
   rm -f $dstr/$p2.inft
     if [ "$icent" -eq 2 ]; then
       center=Centrosymmetric
       p14=1
     else
       center=Acentric
       p14=0
     fi
     # clear
     echo "Crunch interactive, always give return to choose defaults"
     echo
     p3=Background
     #
     # This value of p3 runs Crunch in the background
     # Change into Foregrond if required
     #
     p4=Normal_start
     p5=Multi_matrix
     p9=0
     p10=0
     p11=Default
     p12=0
     p13=0
#
# Get celcontents now from .crysdafile
#
     echo $p6 > $p2.contents
     cat $dstr/$p2.crysda | grep VOLUM > $dstr/temp
     read dummy vol < $dstr/temp
     rm -f temp
     echo $vol >> $p2.contents
     echo "The contents of the unit cell:"
#    cat $dstr/$p2.crysda | grep CELLCO
#    Smart CRYSDA-file routine
     head -2 $dstr/$p2.crysda > $dstr/temp
     tail -1 $dstr/temp > $dstr/temp2
     read tit dummy < $dstr/temp2
     if [ "$tit" = "TITLE" ]; then
       head -7 $dstr/$p2.crysda > $dstr/temp
     else
       head -6 $dstr/$p2.crysda > $dstr/temp
     fi
     tail -2 $dstr/temp       > $dstr/temp2
     head -1 $dstr/temp2      > $dstr/temp3
     tail -1 $dstr/temp2      > $dstr/temp4
     read a b c d e f g h i j k l < $dstr/temp3
     if [ "$l" = "=" ]; then
       read m n o p q r s t u v < $dstr/temp4
       echo $a $b $c $d $e $f $g $h $i $j $k $m $n $o $p $q $r $s $t $u $v >> $dstr
       echo $a $b $c $d $e $f $g $h $i $j $k $m $n $o $p $q $r $s $t $u $v
     else
       echo $a $b $c $d $e $f $g $h $i $j $k
       echo $a $b $c $d $e $f $g $h $i $j $k >> $dstr/$p2.contents
     fi
     rm  -f $PWD/temp3 $PWD/temp4
#    End of Smart CRYSDA-file routine
#
#     The number of nonhydrogens is calculated here
#     The type of problem, equal or heavy atom is determined
#     Celcontents are checked
#
     ln -s $dstr/$p2.contents fort.40
     $dpro/contents
     mv fort.67 $dstr/levy.con
     read p7 p8 p9 p10 sure conok < fort.41
#    echo "$p7 $p8 $p9 $p10 $sure $conok"
     rm -f fort.40
     rm -f fort.41
     if [ ! "$conok" = "yes" ]; then
       echo
       echo "The celcontents specified in the .crysin file are probably incorrect"
       echon  "The number of non-hydrogen atoms is $p7, ok? [y/N]"
       read p77
       if [ ! "$p77" = "Y" ] && [ ! "$p77" = "y" ]; then
         echo
         echo "Change the contents by editing your .crysin file "
         echo "before running Crunch again"
         exit
       fi
     fi
#    echo "I stop now"
#    fi
#    fi
#    exit
#
```

```
#     Confirm structure type (p8)
#
     if [ "$p8" = "Equal_atom" ]; then
       lidwoord="an"
     else
       lidwoord="a"
     fi
     echo
     echo "The structure is $lidwoord $p8 structure"
     if [ ! "$2" = "autofour" ]; then
       if [  "$sure" = "notsure" ]; then
         echo "However, the heavy atom is relatively light."
         echon "Treat as an equal atom problem [y/N]? "
         read structype
       if [ "$structype" = "Y" -o "$structype" = "y" ]; then
         p8="Equal_atom"
         p9=‘ expr 0 ‘
         p10=‘ expr 0 ‘
       fi
     fi
   fi
#
#   Default matrices or user defined? (p11)
#
   if [  "$2" != "autofour" -a "$2" != "blessing" ]; then
     echo
     echon  "[D]efault or [U]ser specified matrix construction? "
     read pmatrix
     p11=Default
     if [ "$pmatrix" = "U" -o "$pmatrix" = "u" ]; then
       p11=User_specified
#
#     Specify the order and the number to be used
#
       echo
       echon  "Specify the number of matrices you want to use: "
       read p12
       echo $p12 > fort.40
       $dpro/mat < fort.40
       read p12 < fort.41
       rm -f fort.40
       rm -f fort.41
       echon  "Specify the order of the matrices you want to use: "
```

```
    read p13
    if [ ! "$p13" -gt 5 ]; then
     p13=‘ expr $p7 / 4 ‘
#
# The values put in by the user are checked and modified when required
#
    fi
   fi
  else
  p12=1
  fi
#
  echo $p2 $p3 $p4 $p5 $p6 $p7 $p8 $p9 $p10 $p11 $p12 $p13 $p14 \
     > $dstr/$p2.par
 fi
#
# .par file constructed
#

 p66=‘ expr $p6 ‘
 echo "0" > $PWD/quasi.cdat
 if [ "$p14" -eq 1 ]; then
  if [ "$2" != "autofour" ] || [ "$3" != "0" ]; then
#
#   The structure is centrosymmetric and the user is not just
#   running autofour with a centrosymmetric model
#   We use an acentric approach
#   Backup the .par file
#
   echo $p2 $p3 $p4 $p5 $p6 $p7 $p8 $p9 $p10 $p11 $p12 $p13 $p14 \
    > $dstr/$p2.parbak
   echo "1" > $PWD/quasi.cdat
   p14=0
   p9=‘ expr $p9 + $p9 ‘
#    center=Acentric
   p6=‘ expr $p6 / 2 ‘
   echo $center $p6 > $dstr/$p2.inf
  fi
 fi
 if [ ! -f "$dstr/$p2.nrl" ]; then
  cat $dstr/$p2.crysda | grep  "SPGR" > $dstr/$p2.lattice
  cat $dstr/$p2.crysda | grep  "CELL  " | grep -v RCELL > $dstr/$p2.cell
#  cat $dstr/$p2.crysda | grep  "CELLCO" > $dstr/$p2.contents
#  Smart CRYSDA-file routine
  head -2 $dstr/$p2.crysda > $dstr/temp
  tail -1 $dstr/temp > $dstr/temp2
  read tit dummy < $dstr/temp2
  if [ "$tit" = "TITLE" ]; then
   head -7 $dstr/$p2.crysda > $dstr/temp
  else
   head -6 $dstr/$p2.crysda > $dstr/temp
  fi
  tail -2 $dstr/temp       > $dstr/temp2
  head -1 $dstr/temp2      > $dstr/temp3
  tail -1 $dstr/temp2      > $dstr/temp4
  read a b c d e f g h i j k l < $dstr/temp3
  if [ "$l" = "=" ]; then
   read m n o p q r s t u v < $dstr/temp4
   echo $a $b $c $d $e $f $g $h $i $j $k $m $n $o $p $q $r $s $t $u $v > $dstr/$p2
  else
   echo $a $b $c $d $e $f $g $h $i $j $k > $dstr/$p2.contents
  fi
  rm -f $dstr/temp $dstr/temp2 $PWD/temp3 $PWD/temp4
#  End of Smart CRYSDA-file routine
  cat $dstr/$p2.crysda | grep  "ICENT"  > $dstr/$p2.cent
  echo "NORMAL80" $p2 >         $dstr/$p2.nrl
  echo "LINEAR" >>              $dstr/$p2.nrl
  read dummy lattice s1 s2 s3 < $dstr/$p2.lattice
  rm -f $dstr/$p2.lattice
  echo "LATTICE" $lattice >>    $dstr/$p2.nrl
  read dummy icent dummy < $dstr/$p2.cent
  if [ "$p14" -eq 1 ]; then
   echo "CENTRO" >>              $dstr/$p2.nrl
  else
   echo "NONCENTRO" >>           $dstr/$p2.nrl
  fi
  rm -f $dstr/$p2.cent
  read dummy cell < $dstr/$p2.cell
  rm -f $dstr/$p2.cell
  echo "CELL" $cell >>          $dstr/$p2.nrl
  cat $dstr/$p2.sym >>          $dstr/$p2.nrl
  if [ -f $dstr/norred.use ]; then
  read dummy bfac < $dstr/norred.use
  echo "BFAC" $bfac >>          $dstr/$p2.nrl
  fi
  echo "NREF 550"  >>           $dstr/$p2.nrl
  read dummy contents < $dstr/$p2.contents
  rm -f $dstr/$p2.contents
  echo "CONTENTS" $contents >>  $dstr/$p2.nrl
  echo "CREATE TAPE" >>         $dstr/$p2.nrl
fi
#
# end building of the .par and .nrl files
#
# Create the binary reflectionfile for autofour, .red
#
#
# use the .frefa file
#
  cat $dstr/$p2.nrl | grep NORMAL80 >  $dstr/$p2.nip
  cat $dstr/$p2.nrl | grep LINEAR   >> $dstr/$p2.nip
  cat $dstr/$p2.nrl | grep LATTICE  >> $dstr/$p2.nip
  cat $dstr/$p2.nrl | grep CENTRO   >> $dstr/$p2.nip
  cat $dstr/$p2.nrl | grep CELL     >> $dstr/$p2.nip
  cat $dstr/$p2.nrl | grep ","      >> $dstr/$p2.nip
  cat $dstr/$p2.nrl | grep NREF     >> $dstr/$p2.nip
  cat $dstr/$p2.nrl | grep BFAC     >> $dstr/$p2.nip
  cat $dstr/$p2.nrl | grep CONTENTS >> $dstr/$p2.nip
  cat $dstr/$p2.nrl | grep CREATE   >> $dstr/$p2.nip
#  echo "CARD 1" >> $dstr/$p2.nip
  echo $p2".frefa file used" >> $PWD/logfile
#
#  Henceforth $dpro/avrg is used to clean up the reflection file
#  Next the reflectionfile for Crunch is prepared
#
  ln -s $dstr/$p2.matrix fort.11
  ln -s $dstr/$p2.frefa  fort.10
  $dpro/avrg
  rm -f $dstr/fort.10
  rm -f $dstr/fort.11
  rm -f $dstr/$p2.matrix
  echo '(3I5,2X,F9.4,F7.2)' >> $dstr/$p2.nip
  cat $dstr/fort.12 | grep -v E >> $dstr/$p2.nip
  echo "    0    0    0   -90.0000   2.00" >> $dstr/$p2.nip
#
#  fort.12 is kept as input to levy (Blessing's E's)
#
  cp $dstr/fort.12 $dstr/$p2.refl
  rm -f fort.12
  echon  "Norred80 " >> $PWD/logfile
  ln -s $dstr/$p2.iscra  fort.8
  ln -s $dstr/$p2.mul    fort.9
  ln -s $dstr/$p2.lfour  fort.10
  ln -s $dstr/$p2.rec    fort.12
  ln -s $dstr/$p2.d15    fort.15
  ln -s $dstr/$p2.red    fort.37
  if [ ! -f "$dstr/$p2.aut" ]; then
    ln -s $dstr/$p2.aut    fort.38
  fi
  $dpro/norred80 < $dstr/$p2.nip > $PWD/norred.cout
  grep RESOLUTION $PWD/norred.cout > temp
  read d1 d2 d3 d4 d5 d6 d7 < temp
  rm temp
  echo  $d7 > $PWD/resolution
  rm -f $dstr/$p2.iscra
  rm -f $dstr/$p2.nip
#
# *.aut is prepared, *.red created
#
#
# Now run levy and if required eval
# If the resulting files are present they will be used
# by DETER and AUTOFOUR.
# Otherwise the Normal80 output will be used
#
  if  [ "$p1" = blessing ]; then
  do_pgrp
  do_mklevy
  do_levy
  do_eval
  rm -f $PQD crunch.*
  rm -f $dstr/$p2.d15
  rm -f $dstr/$p2.rec
  rm -f $dstr/$p2.o
  rm -f $dstr/$p2.sym
  rm -f $dstr/$p2.inf
  rm -f $dstr/$p2.par
  rm -f $dstr/$p2.aut
  rm -f $dstr/norred.cout
# rm -f $dstr/levy*
# rm -f $dstr/eval*
  rm -f $dstr/quasi.cdat
  rm -f $dstr/$p2.red
  rm -f $dstr/$p2.refl
  rm -f $dstr/$p2.mul
  rm -f $dstr/$p2.nrl
  exit
  fi
#
#  Write parameters p8, p9 and p10 to structure type file crunch.typ
#
 if [ "$p8" = Equal_atom ]
 then
   echo 0    >  $PWD/crunch.typ
 else
   echo 1    >  $PWD/crunch.typ
   echo $p9  >> $PWD/crunch.typ
   echo $p10 >> $PWD/crunch.typ
 fi
#
#   Write parameters p14, p6, p7, p12 and p13 to symmetry file crunch.det
#   Write number of start and end cycle to crunch.det
#
 if [ "$2" = recycle ]; then
   rec=1
 else
   rec=0
 fi
 echo $p14  > $PWD/crunch.det
 echo $p6  >> $PWD/crunch.det
 echo $p7  >> $PWD/crunch.det
 echo $p12 >> $PWD/crunch.det
 echo $p13 >> $PWD/crunch.det
 echo $3   >> $PWD/crunch.det
 echo $4   >> $PWD/crunch.det
 echo $rec >> $PWD/crunch.det
 echo 1    >> $PWD/crunch.det
#
#   Write parameters to log file $PWD/crunch.inf
#
 echo " " > $PWD/crunch.inf
 date >> $PWD/crunch.inf
 echo " " >> $PWD/crunch.inf
 echo "CRUNCH release " datum ", Bourne Shell version" >> $PWD/crunch.inf
 echo " " >> $PWD/crunch.inf
 echo "Please refer to:" >> $PWD/crunch.inf
 echo "Automatic Determination of Crystal Structures" >> $PWD/crunch.inf
 echo "using Karle-Hauptman Matrices." >> $PWD/crunch.inf
 echo "Acta Crystallographica A49 (1993), 287-293," >> $PWD/crunch.inf
 echo "R. de Gelder, R.A.G. De Graaff & H. Schenk." >> $PWD/crunch.inf
 echo "X-ray Department Gorlaeus Laboratories" >> $PWD/crunch.inf
 echo "PO Box 9502, 2300 RA Leiden" >> $PWD/crunch.inf
 echo "The Netherlands" >> $PWD/crunch.inf
 echo " " >> $PWD/crunch.inf
 echo "Compound name:                          "$p2 >> $PWD/crunch.inf
 echo $center                                       >> $PWD/crunch.inf
 echo "Spacegroup:                             "$lattice" "$s1" "$s2" "$s3 >> \
      $PWD/crunch.inf
 echo "Number of symmetry operations:          "$p66 >> $PWD/crunch.inf
 echo "Number of atoms in the unit cell:       "$p7 >> $PWD/crunch.inf
 p76=‘ expr $p7 / $p66 ‘
 echo "Number of atoms in the asymmetric unit: "$p76 >> $PWD/crunch.inf
 echo "Structure type:                         "$p8 >> $PWD/crunch.inf
 if [ "$p8" = Heavy_atom ]
 then
   echo "Number of heavy atoms in the asym. un.: "$p9 >> $PWD/crunch.inf
   echo "Scattering factor table number:         "$p10 >> $PWD/crunch.inf
 fi
 echo "Fore/Background:                        "$p3 >> $PWD/crunch.inf
 echo "Run type:                               "$p1 >> $PWD/crunch.inf
 echo "Strategy used:                          "$p5 >> $PWD/crunch.inf
 echo "Input parameters DETER:                 "$p11 >> $PWD/crunch.inf
 if [ "$p11" != Default ]
 then
   echo "Number of matrices:                     "$p12 >> $PWD/crunch.inf
   echo "Order:                                  "$p13 >> $PWD/crunch.inf
 fi
 echo " " >> $PWD/crunch.inf
 echo $p2 $p3 $p4 $p5 $p6 $p7 $p8 $p9 $p10 $p11 $p12 $p13 $p14 \
      > $PWD/crunch.par
 echo $p2 $p3 $p4 $p5 $p6 $p7 $p8 $p9 $p10 $p11 $p12 $p13 $p14 \
      > $dstr/$p2.par

#
#   Move original files .nrl and .par back
#
 if [ -f "$dstr/$p2.nrlbak" ]; then mv $dstr/$p2.nrlbak $dstr/$p2.nrl; fi
 if [ -f "$dstr/$p2.parbak" ]; then mv $dstr/$p2.parbak $dstr/$p2.par; fi
```

```
#
#   Log file:
#
 date >> $dstr/crunch.log
 echo "Phase cracking: CRUNCH "$p2 $2 $3 $4 $5 $PWD >> $dstr/crunch.log
 cat $dstr/$p2.par >> $dstr/crunch.log
 cat $dstr/$p2.nrl >> $dstr/crunch.log
#
#   Submit CRUNCH
#
 date
 rm -f $PWD/logfile
 driver=cruncher.uni
 case "$2" in
 recycle ) driver=recycle.uni ;;
 pmf ) driver=pmf.uni ;;
 esac
 if [ "$p3" = Background ]
 then
   echo "Crunch_1.5 to Background"
   rm -f $PWD/nohup.out
   nohup $dcru/$driver $PWD $2 $dstr $dcru $dpro $3 $4 $5 $6 2>>$PWD/logfile &
 else
   time  $dcru/$driver $PWD $2 $dstr $dcru $dpro $3 $4 $5 $6 2>>$PWD/logfile
 fi
 rm -f CRYSIN
 rm -f FREFA
 rm -f $dstr/$p2.dderr
 exit
```

Next is Cruncher.uni. Cruncher.uni gets its input from Crunch.uni. The only thing that still needs to be determined is the number of matrices that are needed for the current problem. This is done by a loop through thinkc and deter. Just running autofour (See the manual) is done by this script too.

Cruncher.uni:

```
#!/bin/bash
# ********************************************************#
#   CRUNCHER.UNI: Automatic solution of structures        #
# ********************************************************#
#   DETER (Single matrix) <----------------------------   #
#     |                                                |  #
#     v                                                ^  #
#   DETER (Multi-matrix)  <----------------------------   #
#     |   (Optional: Using single matrix phases)       |  #
#     v                                                |  #
#   VEXFFT                                             |  #
#     |                                                |  #
#     v                                                ^  #
#   EXFFT80                 ___                        |  #
#     |                    |Yes| ->- SEARCH80          |  #
#     v                     ---                        |  #
#   SEARCH80                 ^                         |  #
#     |                      |                         |  #
#     v              -----------------      --         |  #
#   AUTOFOUR ---> |Structure solved?| -> |No| ---------   #
#                    -----------------      --            #
#   Koppiereit (c) 1996 Rene de Gelder R.U. Leiden.       #
# ********************************************************#
#
#   Last edit: Autumn 2009 by RAG de Graaff
#   --------------------------------------
 do_mkdetf()
 {
  if [ ! -f $dstr/deter.asc ]; then
 cp $dcru/deter.asc $dstr/deter.asc
 fi
 if [ ! -f $dstr/tempdeter ]; then
 if [ -f $dstr/$p1.drear ]; then
 #
 # check whether normal80 or blessing's E's are to be used.
 #
 echo "IBLES    2" >> deter.use
 else
 if [ -f $dstr/edata.eee ]; then
 echo "IBLES    1" >> deter.use
 fi
 fi
 if [ ! -f $dstr/deter.use ]; then
 echo "END   " > $dstr/deter.use
 fi
 fi
 ln -s $dstr/deter.asc $dstr/fort.36
 ln -s $dstr/deter.use $dstr/fort.37
 ln -s $dstr/resolution $dstr/fort.40
 $dpro/mkdetf
 read inp < $dstr/fort.39
 if [ $inp -eq 0 ]; then
 echo "Error in input to deter as specified in deter.asc" >> logfile
 exit
 fi
 mv $dstr/fort.38 deter.cdat
 rm -f $dstr/fort.36 $dstr/fort.37 $dstr/fort.38 $dstr/fort.39 temp
 }
#
# end of do_mkdetf
#
 do_thinkc()
 {
#
#  THINK: Adapt input of deter
#
 rm -f $PWD/deteri.cdat
 mv $PWD/deter.cdat $PWD/deteri.cdat
 ln -s $PWD/crunch.det  $PWD/fort.24
 ln -s $PWD/deteri.cdat    $PWD/fort.25
 ln -s $PWD/detero.cdat    $PWD/fort.26
 ln -s $PWD/deter.num    $PWD/fort.28
 ln -s $PWD/multi.cdat     $PWD/fort.30
 $dpro/thinkc
 rm $PWD/fort.*
 mv $PWD/detero.cdat $PWD/deter.cdat
 rm -f $PWD/deteri.cdat
 }
#
# End of do_thinkc
#
 do_init()
 {
#
#   INIT: Initialize input and checks
#
 cp $dcru/multi.cdat    $PWD/multi.cdat
 cp $dcru/check.cdat    $PWD/check.cdat
 cp $dcru/deter.num  $PWD/deter.num
 cp $dcru/deter.err    $PWD/deter.err
 cp $dcru/random.seed  $PWD/random.seed
 if [ -f deter.use ]; then
 cp deter.use tempdeter
 fi
 if [ -f autofour.use ]; then
 cp autofour.use tempauto
 fi
 chmod u+w $PWD/multi.cdat
 chmod u+w $PWD/check.cdat
 chmod u+w $PWD/deter.num
 ln -s $PWD/multi.cdat  $PWD/fort.2
 ln -s $PWD/check.cdat  $PWD/fort.3
 $dpro/init > $PWD/deter.num
 rm -f $PWD/fort.*
 nrset=0
 if [ "$6" != "" ]; then nrset=‘ expr $6 - 1 ‘ ; fi
 cycle=0
 nrhit=0
 solved=0
 if [ -f $PWD/hits ]; then rm $PWD/hits; fi
 }
#
# End of do_init
#
 do_detinit()
#
# Prepare input for deter, store toprow of the matrix
#
 {
 echon "deter "
  mulcheck=0
 if [ ! -f $dstr/multi.tpr ]; then
 do_mkdetf
 do_thinkc
 while [ $mulcheck = 0 ]
 do
 do_det
 do_thinkc
 read mulcheck < $PWD/multi.cdat
 done
 #
 # Toprow is found
 #
 echo "1" > $dstr/multi.cdat
 fi
 }
#
```

```
# End of do_detinit
#
 do_det()
#
# Build matrix and/or calculate phases
#
 {
 ln -s $PWD/deter.008    $PWD/fort.8
 ln -s $dstr/$p1.mul     $PWD/fort.9
 ln -s $dstr/$p1.con     $PWD/fort.11
 ln -s $PWD/deter.sho   $PWD/fort.12
 ln -s $dstr/edata.eee   $PWD/fort.14
 ln -s $dstr/$p1.d15     $PWD/fort.15
 ln -s $PWD/deter.num   $PWD/fort.16
 ln -s $dstr/$p1.drear   $PWD/fort.17
 ln -s $dstr/$p1.fas     $PWD/fort.21
 ln -s $dstr/$p1.dfs     $PWD/fort.22
 ln -s $PWD/multi.tpr    $PWD/fort.23
 ln -s $dstr/$p1.exf     $PWD/fort.24
 ln -s $dstr/$p1.030     $PWD/fort.30
 ln -s $PWD/deter.err    $PWD/fort.35
 if [ -f $PWD/coordinaten ]; then
 ln -s $PWD/coordinaten  $PWD/fort.68
 fi
 $dpro/deter.exe < $PWD/deter.cdat > $PWD/deter.cout
 rm $PWD/fort.*
 read core < $PWD/deter.err
 if [ $core -eq 1 ]; then
 echo "Not enough core in DETER!"  >> $PWD/logfile
 exit
 fi
 if [ $core -eq 2 ]; then
 echo "A suitable matrix cannot be  constructed." >> $PWD/logfile
 echo "Possibly the data are to weak" >> $PWD/logfile
 echo "You may try specifying the order and the number" >> $PWD/logfile
 echo "manually. Run crunch code clear " >> $PWD/logfile
 echo "First use the option 'blessing' to calculate E's" >> $PWD/logfile
 echo "Next respond by giving 'u' instead of a return " >> $PWD/logfile
 echo "when you are asked whether you'd like to use defaults" >> $PWD/logfile
 echo "Next follow instructions" >> $PWD/logfile
 exit
 fi
 }
```

```
#
# End of do_det
#
 do_vexfft()
 {
#
#   VEXFFT: Conversion of deter output file to exfft80 file
#
 ln -s $dstr/$p1.fou    $PWD/fort.10
 ln -s $dstr/$p1.phi    $PWD/fort.15
 ln -s $dstr/$p1.exf    $PWD/fort.24
 echon  "- Vexfft "
 $dpro/vexfft < $PWD/vexfft.cdat > $PWD/vexfft.cout
 rm $PWD/fort.*
 if [ "$pp2" = deter ]; then
 mv $dstr/$p1.fou $dstr/$p1.fou.$nrset
 mv $dstr/$p1.phi $dstr/$p1.phi.$nrset
 fi
 }
#
# End of do_vexfft
#
 do_exfft()
 {
#
#   EXFFT80: Fast fourier transform of structure factors
#
 if [ "$pp2" = peaks ]; then
 if [ -f $dstr/$p1.fou.$nrset ]; then
 cp $dstr/$p1.fou.$nrset $dstr/$p1.fou
 else
 echo "Error: no .fou file found with corresponding set number"
 exit
 fi
 fi
 ln -s $PWD/exfft.cdat $PWD/fort.1
 ln -s $dstr/$p1.fou   $PWD/fort.10
 ln -s $dstr/$p1.map   $PWD/fort.13
 ln -s $dstr/resolution $PWD/fort.40
 echon  "- Exfft "
 $dpro/exfft80 > $PWD/exfft.cout
 rm $PWD/fort.*
 }
```

```
#
# End do_exfft
#
 do_check()
 {
#
#   Check whether structure is solved, different for Crunch run
#   and autofour try
#
 read solved < $PWD/check.cdat
 if [ "$pp2" = autofour ] && [ "$6" = "0" ]; then
#
# autofour pure
#
 if [ $solved -eq 1 ]; then
 echon " Structure solved"
 tail $dstr/$p1.lsq | grep FINAL > $PWD/tmp.1
 echo "Structure solved by autofour " >> $PWD/tmp.2
 paste $PWD/tmp.2 $PWD/tmp.1 >> $PWD/hits
 rm $PWD/tmp.1 $PWD/tmp.2
 trial=2
 echo $trial > $PWD/multi.cdat
 else
 echo "Not successful"
 fi
 else
#
# Normal Crunch cycle or do_numauto
#
 echo "- end cycle " $cycle " set " $nrset
 if [ "$pp2" != deter ] && [ "$pp2" != peaks ]; then
 echo "autofour cycle" $cycle >> $PWD/hits
 if [ -f $PWD/deter.cout ]; then
 cat $PWD/deter.cout   | grep CYCLES >> $PWD/hits
 fi
 if [ -f $PWD/deter.cout.$nrset ]; then
 cat $PWD/deter.cout.$nrset   | grep CYCLES >> $PWD/hits
 fi
 cat $PWD/autofour.cout | grep Actual >> $PWD/hits
 if [ $solved -eq 1 ]; then
 nrhit=‘ expr $nrhit + 1 ‘
#
# keep input to autofour
```

```
#
 if [ -f $PWD/fragment.cout ]; then
 cp $dstr/fragment.cout   $dstr/fragment.okay
 fi
 tail $dstr/$p1.lsq | grep FINAL > $PWD/tmp.1
 echo "Structure solved in cycle " $cycle " set " $nrset >> $PWD/tmp.2
 paste $PWD/tmp.2 $PWD/tmp.1 >> $PWD/hits
 rm $PWD/tmp.1 $PWD/tmp.2
 if [ "$pp2" != "collect" ]; then
 trial=2
 echo $trial > $PWD/multi.cdat
 else
 solved=0
 fi
 else
 if [ $nrset -eq $7 ]; then
 trial=2
 echo $trial  > $PWD/multi.cdat
 fi
 echo "Not successful in cycle " $cycle " set " $nrset >> $PWD/hits
 fi
 fi
 fi
 }
#
# End of do_check
#
 do_final_search()
 {
 ln -s $PWD/search.cdat  $PWD/fort.1
 ln -s $dstr/$p1.map    $PWD/fort.13
 ln -s $dstr/$p1.pek    $PWD/fort.14
 ln -s $dstr/$p1.fra    $PWD/fort.15
 ln -s $dstr/resolution $PWD/fort.40
 echo "Final Search "
 $dpro/search80 > $PWD/searchf.cout
 rm $PWD/fort.*
 }
#
# End of do_final_search
#
 do_search()
 {
```

```
#
#   SEARCH80: Peak search program for Multan fourier files
#
 ln -s $PWD/search.cdat  $PWD/fort.1
 ln -s $dstr/$p1.map    $PWD/fort.13
 ln -s $dstr/$p1.pek    $PWD/fort.14
 ln -s $dstr/$p1.fra    $PWD/fort.15
 ln -s $dstr/resolution $PWD/fort.40
 echon  "- Search "
 $dpro/search80 > $PWD/search.cout
 rm $PWD/fort.*
 }
#
# End of do_search
#
 do_frag()
 {
#
#   FRAGMENT: Find largest fragment in Search80 peak-list
#   Prepare $dstr.pek file for autofour as specified in variable $8
#   Check presence of fragment file (struc.fra)
#
 if [ "$8" != all ] && [ "$8" != first ] && [ "$8" != last ]; then
 if [ ! -f $dstr/$p1.fra ]
 then
 cp $dcru/dummy.fra $dstr/$p1.fra
 fi
 if [ -f $dstr/$p1.fra -o "$p8" = Heavy_atom ]
 then
 cat $dstr/$p1.nrl | grep "CELL" > $dstr/$p1.cell
 read dummy c1 c2 c3 c4 c5 c6 < $dstr/$p1.cell
 echo $c1 $c2 $c3 $c4 $c5 $c6 > $dstr/$p1.cell
 ln -s $dstr/$p1.pek      $PWD/fort.1
 ln -s $dstr/$p1.fra      $PWD/fort.2
 ln -s $dstr/$p1.pk2      $PWD/fort.3
 ln -s $PWD/crunch.typ  $PWD/fort.4
 ln -s $dstr/$p1.cell     $PWD/fort.7
 ln -s $dstr/deter.atoms  $PWD/fort.8
 ln -s $dstr/deter.pdb    $PWD/fort.9
 echon  "- Fragment "
 $dpro/fragment > $PWD/fragment.cout
 rm $PWD/fort.*
 rm $dstr/$p1.cell
 rm $dstr/$p1.pek
 rm $dstr/$p1.fra
 mv $dstr/$p1.pk2 $dstr/$p1.pek
 wc -l $dstr/deter.atoms > $PWD/temp
 read length name < $PWD/temp
 rm $PWD/temp
 length=‘ expr $length - 1 ‘
 echo ’ATOMS ’$p1 > $dstr/ATOMS
 tail -$length $dstr/deter.atoms >> $dstr/ATOMS
 mv $dstr/ATOMS $dstr/deter.atoms
 fi
 else
 if [ "$8" = first ] && [ "$9" != "" ]; then
 head -$9 $dstr/$p1.pek > $dstr/$p1.tmp
 mv $dstr/$p1.tmp $dstr/$p1.pek
 fi
 if [ "$8" = last ] && [ "$9" != "" ]; then
 tail -$9 $dstr/$p1.pek > $dstr/$p1.tmp
 mv $dstr/$p1.tmp $dstr/$p1.pek
 fi
 fi
 if [ "$pp2" = peaks ]; then
 cp $dstr/$p1.pek $dstr/$p1.pek.$nrset
 fi
 if [ $nrset -eq $7 ]; then
 trial=2
 echo $trial > $PWD/multi.cdat
 fi
 }
#
# End of do_frag
#
do_mkautoinp()
 {
#
# Prepare input to autofour from $p1.aut and possible user input
#
 cp $dcru/autofour.err $dstr/autofour.err
 if [ ! -f $dstr/autofour.asc ]; then
 cp $dcru/autofour.asc $dstr/autofour.asc
 fi
 ln -s $p1.aut $dstr/fort.38
 ln -fs autofour.asc $dstr/fort.36
 ln -s autofour.use $dstr/fort.37
 $dpro/mkautf
 mv $dstr/fort.40 $p1.aut
 rm -f $dstr/fort36 $dstr/fort.37 $dstr/fort.38 $dstr/fort.40
 rm -f temp
 cp  $dstr/$p1.aut     $dstr/autofour.cdat
 cat $dstr/$p1.pek >> autofour.cdat
 cat $dcru/eind.txt >> $dstr/autofour.cdat
#
#  End of do_mkautoinp
#
 }
 do_auto()
 {
#
#   AUTOFOUR: Automatic Fourier program
#
 ln -s $PWD/autofour.cdat  $PWD/fort.1
 ln -s $dstr/$p1.mpa      $PWD/fort.8
 ln -s $dstr/$p1.red      $PWD/fort.9
 ln -s $PWD/check.cdat    $PWD/fort.10
 ln -s $dstr/$p1.xyz      $PWD/fort.11
 ln -s $dstr/$p1.pdb      $PWD/fort.12
 ln -s $dstr/$p1.lsq      $PWD/fort.15
 ln -s $dstr/$p1.drear    $PWD/fort.16
 ln -s $PWD/quasi.cdat    $PWD/fort.21
 ln -s $PWD/crunch.typ    $PWD/fort.22
 ln -sf $dstr/edata.eee   $PWD/fort.23
 ln -s $dstr/$p1.spf      $PWD/fort.31
 ln -s $PWD/autofour.err  $PWD/fort.35
 ln -s $PWD/resolution    $PWD/fort.40
 ln -s $PWD/random.seed   $PWD/fort.99
 ln -sf $PWD/biso.levy    $PWD/fort.14
 echon  "- autofour "
 $dpro/autofour.exe > $PWD/autofour.cout
 rm -f $dstr/fort.1 $dstr/fort.8 $dstr/fort.9 $dstr/fort.10 $dstr/fort.11
 rm -f $dstr/fort.12 $dstr/fort.14 $dstr/fort.15 $dstr/fort.16 $dstr/fort.23
 rm -f $dstr/fort.21 $dstr/fort.22 $dstr/fort.31 $dstr/fort.35
 rm -f $dstr/fort.40
 read core < $PWD/autofour.err
 if [ $core -eq 1 ]; then
 echo "Not enough core in AUTOFOUR!"  >> $PWD/logfile
 exit
 fi
 }
#
# End of do_auto
#
 do_final_fragment()
 {
#
#   FRAGMENT: Find largest fragment in Search80 peak-list
#
#   Check for presence of fragment file (struc.fra)
#
 if [ ! -f $dstr/$p1.fra ]
 then
 cp $dcru/dummy.fra $dstr/$p1.fra
 fi
 cat $dstr/$p1.nrl | grep "CELL" > $dstr/$p1.cell
 read dummy c1 c2 c3 c4 c5 c6 < $dstr/$p1.cell
 echo $c1 $c2 $c3 $c4 $c5 $c6 > $dstr/$p1.cell
 ln -s $dstr/$p1.pek      $PWD/fort.1
 ln -s $dstr/$p1.fra      $PWD/fort.2
 ln -s $dstr/$p1.pk2      $PWD/fort.3
 ln -s $PWD/crunch.typ  $PWD/fort.4
 ln -s $dstr/$p1.cell     $PWD/fort.7
 ln -s $dstr/$p1.atoms    $PWD/fort.8
 ln -s $dstr/fragment.pdb $PWD/fort.9
 echo "Final Fragment"
 $dpro/fragment > $PWD/fragment.cout
 rm $PWD/fort.*
 rm $dstr/$p1.cell
 rm $dstr/$p1.pek
 mv $dstr/$p1.pk2 $dstr/$p1.pka
 wc -l $dstr/$p1.atoms > $PWD/temp
 read length name < $PWD/temp
 rm $PWD/temp
 length=‘ expr $length - 1 ‘
 echo ’ATOMS ’$p1 > $dstr/ATOMS
 tail -$length $dstr/$p1.atoms >> $dstr/ATOMS
 mv $dstr/ATOMS $dstr/$p1.atoms
 echo " "
 echo "Check your structure"
 echo "A full report may be found in" $dstr"/"$p1".report "
 echo " "
 }
#
# End of do_final_fragment
#
 do_logaut()
 {
#
#   Create log file containing all necessary information
#   after a successful run of autofour
#
 cat $dstr/$p1.nrl >> $PWD/crunch.inf
 echo " "              >> $PWD/crunch.inf
 cat $PWD/autofour.cdat >> $PWD/crunch.inf
 echo " "              >> $PWD/crunch.inf
 cat $dstr/$p1.lsq     >> $PWD/crunch.inf
 echo " "              >> $PWD/crunch.inf
 date                  >> $PWD/crunch.inf
 echo " "              >> $PWD/crunch.inf
#
 mv $PWD/crunch.inf $dstr/$p1.report
 }
#
# End of do_logaut
#
 do_log()
 {
#
# Prepare log after successful Crunch cycle
#
#
#   Create log file containing all necessary information
#
#   Cat files
#
 cp $PWD/crunch.inf $dstr/$p1.pub
 echo "Input cards used for program NORMAL80:" >> $dstr/$p1.pub
 echo >> $dstr/$p1.pub
 cat $dstr/$p1.nrl   >> $dstr/$p1.pub
 echo " "            >> $dstr/$p1.pub
 echo "Resulting  (complete) model from CRUNCH:" >> $dstr/$p1.pub
 echo >> $dstr/$p1.pub
 cat $dstr/$p1.lsq >> $dstr/$p1.pub
 echo "Input cards used for program NORMAL80:" >> $PWD/crunch.inf
 echo >> $PWD/crunch.inf
 cat $dstr/$p1.nrl >> $PWD/crunch.inf
 echo >> $PWD/crunch.inf
 echo "Input cards used for program DETER:" >> $PWD/crunch.inf
 if [ -f $PWD/deter.sho ]; then
 cat  $PWD/deter.sho   >> $PWD/crunch.inf
 echo "Phases resulting from program DETER:" >> $PWD/crunch.inf
 echo >> $PWD/crunch.inf
 cat  $dstr/$p1.dfs     >> $PWD/crunch.inf
 echo >> $PWD/crunch.inf
 fi
 echo "Model (fragment) resulting from DETER-phases:" >> $PWD/crunch.inf
 echo >> $PWD/crunch.inf
 cat $PWD/autofour.cdat >> $PWD/crunch.inf
 echo >> $PWD/crunch.inf
 echo "Fragment resulting from final SEARCH80:" >> $PWD/crunch.inf
 echo >> $PWD/crunch.inf
 cat $dstr/$p1.atoms   >> $PWD/crunch.inf
 echo >> $PWD/crunch.inf
 echo "Complete peak-list resulting from AUTOFOUR:" >> $PWD/crunch.inf
 echo >> $PWD/crunch.inf
 cat $dstr/$p1.lsq     >> $PWD/crunch.inf
 cat $dstr/$p1.lsq     >> $dstr/crunch.log
 echo >> $PWD/crunch.inf
 cat $dstr/$p1.spf     >> $PWD/crunch.inf
 echo " "              >> $PWD/crunch.inf
 date                  >> $PWD/crunch.inf
 echo " "              >> $PWD/crunch.inf
#
 mv $PWD/crunch.inf $dstr/$p1.report
#
 rm -f $PWD/output
 cat  $PWD/norred.cout >> $PWD/output
 rm  $PWD/norred.cout
 if [ -f $PWD/deter.cout ]; then
 cat  $PWD/deter.cout >> $PWD/output
 rm  $PWD/deter.cout
 fi
 if [ -f $PWD/deter.cout.$nrset ]; then
 cat  $PWD/deter.cout.$nrset >> $PWD/output
 rm  $PWD/deter.cout.$nrset
 fi
 if [ -f $PWD/vexfft.cout ]; then
 cat  $PWD/vexfft.cout >> $PWD/output
 rm  $PWD/vexfft.cout
 fi
 if [ -f $PWD/exfft.cout ]; then
 cat  $PWD/exfft.cout >> $PWD/output
 rm  $PWD/exfft.cout
 fi
 if [ -f $PWD/search.cout ]; then
 cat  $PWD/search.cout >> $PWD/output
 rm  $PWD/search.cout
 fi
 if [ -f $PWD/fragment.okay ]; then
 cat  $PWD/fragment.okay >> $PWD/output
 rm  $PWD/fragment.okay
 fi
 rm  $PWD/fragment.cout
 if [ -f $PWD/autofour.cout.$nrset ]; then
 cat  $PWD/autofour.cout.$nrset >> $PWD/output
 rm  $PWD/autofour.cout.$nrset
 fi
 if [ -f $PWD/autofour.cout ]; then
 cat  $PWD/autofour.cout >> $PWD/output
 rm  $PWD/autofour.cout
 fi
 if [ -f $PWD/searchf.cout ]; then
 cat  $PWD/searchf.cout >> $PWD/output
 rm  $PWD/searchf.cout
 fi
 }
#
# End of do_log
#
 do_clean()
 {
#
#   Move original files .nrl and .par back
#
 if [ -f $dstr/$p1.nrlbak ]; then mv $dstr/$p1.nrlbak $dstr/$p1.nrl; fi
 if [ -f $dstr/$p1.parbak ]; then mv $dstr/$p1.parbak $dstr/$p1.par; fi
#
#   Delete structure files
#
 rm -f $dstr/fort*
 rm -f $dstr/$p1.030
 rm -f $dstr/$p1.d15
 rm -f $dstr/$p1.dfs
 rm -f $dstr/$p1.exf
#rm -f $dstr/$p1.mul
 rm -f $dstr/$p1.ref
 rm -f $dstr/$p1.pka
 rm -f $dstr/$p1.phi
 rm -f $dstr/$p1.fou
 rm -f $dstr/$p1.fra
 rm -f $dstr/$p1.lsq
 rm -f $dstr/$p1.map
 rm -f $dstr/$p1.mpa
 rm -f $dstr/$p1.pek
 if [ ! "$pp2" = "peaks" ] && [ ! "$pp2" = "deter" ] && \
 [ ! "$pp2" = "autofour" ]; then
 rm -f $dstr/levy.con
 rm -f $dstr/$p1.nrl
 rm -f $dstr/resolution
 rm -f $dstr/$p1.pub
 rm -f $dstr/$p1.xyz
 rm $PWD/check.cdat
 rm $PWD/crunch.typ
 rm $PWD/crunch.det
 rm $PWD/crunch.par
 rm -f $dstr/$p1.red
 rm -f $dstr/$p1.aut
 rm -f $PWD/autofour.cdat
 rm -f $PWD/vexfft.cdat
 rm -f $PWD/exfft.cdat
 rm -f $PWD/search.cdat
 rm -f $PWD/deter.cdat
 rm -f $PWD/deter.atoms
 rm -f $PWD/deter.pdb
 rm -f $PWD/fragment.pdb
 rm -f $PWD/deter.err
 rm -f $PWD/autofour.err
 rm -f $PWD/quasi.cdat
 rm -f $PWD/random.seed
 rm -f $PWD/$p1.phi.*
 rm -f $PWD/$p1.fou.*
 rm -f $PWD/$p1.pek.*
 fi
 rm -f $PWD/$p1.refl
 rm -f $dstr/$p1.o
 rm -f $dstr/$p1.sym
 rm -f $dstr/$p1.inf
 rm -f $PWD/deter.sho
 rm -f $PWD/deter.008
 rm -f $PWD/deteri.cdat
 rm -f $PWD/multi.tpr
 rm -f $PWD/multi.cdat
 rm -f $PWD/deter.num
 rm -f $PWD/norred.cout
 rm -f $PWD/autofour.asc
 rm -f $PWD/deter.asc
 if [ -f tempdeter ]; then
 mv tempdeter deter.use
 fi
 if [ -f tempauto ]; then
 mv tempauto autofour.use
 fi
 }
#
# End of do_clean
#
 do_justdet()
 {
 nrset=‘ expr $nrset + 1 ‘
 cycle=‘ expr $cycle + 1 ‘
 do_vexfft
 while [ $nrset -lt $7 ]
 do
 echon  "deter "
 do_det
 do_thinkc
 nrset=‘ expr $nrset + 1 ‘
 cycle=‘ expr $cycle + 1 ‘
 mv $dstr/deter.cout $dstr/deter.cout.$nrset
 do_vexfft
#
# A matrix of sufficient content is constructed
# prepare input to Exfft80
#
 done
#
```

```
# End of do_justdet
#
 }
 do_peaks()
 {
 nrset=‘ expr $nrset + 1 ‘
 cycle=‘ expr $cycle + 1 ‘
 while [ $nrset -le $7 ]
 do
 if [ ! -f $dstr/$p1.phi.$nrset ]; then
 echo "There is no file .phi of the number specified (" $nrset ")
 present!"
 echo "Use the option ’deter’ to calculate the phases or use the option
 ’try’ for an automatic run"
 exit
 fi
 do_exfft
 do_search
 do_frag $@
 nrset=‘ expr $nrset + 1 ‘
 cycle=‘ expr $cycle + 1 ‘
 done
 exit
 }
#
# End of do_peaks
#
do_strauto()
{
#
#  A straightforward run of autofour
#
 if [ ! -f $dstr/$p1.pek ];then
 echo "You did not create a .pek file in the structures directory!"
 echo "Bye, bye for now!"
 do_clean
 exit
 fi
 if [ -f $PWD/hits ]; then rm $PWD/hits; fi
 cp $dcru/check.cdat    $PWD/check.cdat
 do_mkautoinp
 do_auto
 do_check $@
 if [ ! $solved -eq 0 ];then
 do_logaut
 fi
 }
#
# End of autofour run
#
 do_numauto()
 {
#
# Execute autofour after doing options deter and peaks and check result
#
 set -vx
 nrset=‘ expr $nrset + 1 ‘
 cycle=‘ expr $cycle + 1 ‘
 if [ -f $PWD/hits ]; then rm $PWD/hits; fi
 if [ -f $dstr/$p1.drear ]; then
 echo "IBLES    2" >> autofour.use
 else
 if [ -f $dstr/edata.eee ]; then
 echo "IBLES    1" >> autofour.use
 fi
 fi
 if [ ! -f autofour.use ]; then
 echo "END   " > autofour.use
 fi
while [ $nrset -le $7 ]
 do
 if [ ! -f $dstr/$p1.pek.$nrset ]; then
 echo "There is no .pek file of the number specified (" $nrset ") present!"
 exit
 else
 cp $dstr/$p1.pek.$nrset $dstr/$p1.pek
 fi
 cp $dcru/check.cdat    $PWD/check.cdat
 do_mkautoinp
 do_auto
 do_check $@
 mv $dstr/autofour.cout $dstr/autofour.cout.$nrset
 nrset=‘ expr $nrset + 1 ‘
 cycle=‘ expr $cycle + 1 ‘
 done
 }
```

```
#
# End of autofour trials
#
do_try()
{
#
#   Begin of CRUNCH loop, trying to solve the structure automatically
#   Is executed for options collect and try
#
 if [ -f $dstr/$p1.drear ]; then
 echo "IBLES    2" >> autofour.use
 else
 if [ -f $dstr/edata.eee ]; then
 echo "IBLES    1" >> autofour.use
 fi
 fi
 if [ ! -f autofour.use ]; then
 echo "END   " > autofour.use
 fi
while [ $solved = 0 ]
 do
 cp $dcru/check.cdat $dstr/check.cdat
 if [ -f $PWD/fort.8 ]; then rm $PWD/fort.8; fi
 read trial < $PWD/multi.cdat
 if [ $trial -eq 2 ]; then
 echo $nrhit " solutions out of " $cycle " cycles" >> $PWD/hits
 if [ $nrhit -eq 0 ]; then
 echo "Not successful"
 else
 echo "Structure solved"
 fi
 do_clean
 exit
 fi
 nrset=‘ expr $nrset + 1 ‘
 cycle=‘ expr $cycle + 1 ‘
 do_vexfft
 do_exfft
 do_search
 do_frag $@
 do_mkautoinp
 do_auto $@
 do_check $@
 if [ $solved -eq 0 ] && [ ! $trial -eq 2 ]; then
 echon  "deter "
 do_det
 do_thinkc
 fi
 done
 echo $nrhit " solutions out of " $cycle " cycles" >> $PWD/hits
 if [ $nrhit -eq 0 ]; then
 echo "Not successful"
 else
 echo "Structure solved"
 cp $dstr/$p1.mpa $dstr/$p1.map
 do_final_search
 do_final_fragment
 do_log
 fi
 do_clean
#
# End of do_try
#
 }
 echon ()
 {
 echo $1 | tr -d ’\012’
 }
#
# End of definitions
#
# Read parameters from file
#
 PWD=$1
 rm -f $PWD/fort.*
 dstr=$3
 dcru=$4
 dpro=$5
 read p1 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14 < $PWD/crunch.par
 pp2=$2
#
# We do not allow for non-default inputfiles in structure-directory
#
 cp $dcru/exfft.cdat $PWD/exfft.cdat
 cp $dcru/vexfft.cdat $PWD/vexfft.cdat
 cp $dcru/search.cdat $PWD/search.cdat
```

```
#
#   Create input files for multi matrix run
#
 do_init $@
#
#   Check type of strategy and take appropiate action
#
 case "$pp2" in
  try | collect ) do_detinit
do_try $@ ;;
autofour )
if [ "$6" = "0" ]; then
do_strauto $@
else
do_numauto $@
if [ ! $solved -eq 0 ]; then
do_final_search
do_final_fragment
do_log
fi
fi
do_clean
;;
deter )
do_detinit
do_justdet $@
do_clean
;;
peaks )
do_peaks  $@
do_clean
;;
 esac
 exit
```

5 The manual

Crunch is a direct methods program developed for solving difficult small and medium - sized structures *ab initio*. Datasets should be of atomic resolution. The system has been designed to run under the Unix operating system. The current version has been tested on various multi-processorsystems using the Linux operating system or OS X 10.4 or better. Although Crunch is aimed in principle at difficult equal-atom problems, the program may be used effectively for routine structure determination of heavy-atom structures too. The program needs about 20 Mb disk space. If supported the program will use OpenMp and Gpu to speed up calculations. Memory allocation is dynamic and automatic.

The system consists of two main programs and assorted utilities. The first program, deter, determines the phases. The second program, autofour, evaluates the results of each deter cycle, trying to find the complete model based on the results obtained by deter. The contents of the unit cell should be given as accurately as possible. Especially the second principal program in the system, the section which tries to find the complete model based on the results of the first section, depends on the availability of reasonable estimates of B-overall and the scale. Most matrix and vector manipulations are done using the Blas and Lapack routines. If you do not have gfortran, or intel fortran + numerical library available on your machine it is of advantage to use processor specific optimized versions of this software. On Mac Os X systems use the libraries supplied by Apple. Usually the configure software used in the installation will take care of all this by setting the correct flags. Makefiles are s supplied for the program system. An include file 'flags' which defines the flags used in the makefiles is prepared during installation. This file may be adapted to the local situation.

Crunch supports just about any input of structure factors you can think of as long as the files contain hkl, F or F**2 and sigma(F or F**2), one reflection pro record and no records containing other items. The entries in the file must be separated by blanks, comma's or plus or minus signs. A record such as 12 3 4 1.2340.566 will not do. Crunch_1.1 or higher no longer requires the presence of the Dirdif system (inquiries Dirdif to Rene de Gelder)

In earlier versions of Crunch Dirdif was used to handle the crystallographic data such as cell contents, unit cell dimensions, symmetry etc. Crunch uses a reflection file containing the asymmetric unit in reciprocal space ONLY. However, your reflection file is cleaned of redundancies automatically. Atomic coordinates are produced in the .pdb and .spf formats. Visual inspection of the results is therefore straightforward, using the Pluton/Platon graphic suite written by Ton Spek (email: a.l.spek@chem.uu.nl). Unix scripts are written to be compatible with the Bourne/Korn shells. The scripts provided do not allow the running of more than one crunch-job from within one directory. However, multiple runs of Crunch on one computer from different directories and/or by different users are allowed.

Crunch was developed for the *ab initio* solution of the phase problem, which also covers the situation where a crystallographer doesn't know anything about his compound except that it is organic and its approximate atomic weight. This has proven to be extremely useful for the identification of unknown natural compounds.

If you've used Crunch in any resulting paper please refer to:

Automatic Determination of Crystal Structures
using Karle-Hauptman Matrices.
Acta Crystallographica A49 (1993), 287-293,
R. de Gelder, R.A.G. De Graaff & H. Schenk.
X-ray Department Gorlaeus Laboratories
PO Box 9502, 2300 RA Leiden
The Netherlands

6 Installation

Copy crunch1.5.tgz into the directory where you would like to have Crunch installed. Next untar the archive by typing `'tar zxvf crunch1.5.tgz'`. Follow with `'cd crunch1.5'` and `'./configure'`. Next type `'make clean'` followed by `'make dep'` and `'make'`.

Crunch1.5 is being installed now

Mind that sufficient (> 20Mb) disk space is available before you start the installation. You need to have gunzip, the gnu unzipper, available on your system as well as a fortran90 compiler. Gfortran springs to mind. Another good choice is the Intel ccmpiler system. The configure step takes care of the compiler choice. OpenMp and Gpu calculation are supported. In the directory you've chosen for Crunch to be installed into, a new directory 'crunch1.5' should now be present. This directory should contain the following: A directory manual containing this manual - in the form of the files manual.aux, manual.toc and manual.tex - the files runit, flags and the directories drivers, programs, source, pytlud and rn001. Runit is a dynamic link to the script which actually runs the program. Move this to some suitable directory in your PATH, such as /usr/local/bin and chage it as required. Copy the directories rn001 and pytlud into the directory where you usually solve your structures. You may have to close and reopen your X-terminal to activate the link 'runit'.

You are now ready to proceed.

The configure script checks for the presence of a Fortran compiler on your system. The file configure.log contains details on the compilers etc. which are going to be used for the installation. The prefered option on Linux and OS X systems is GNU's gfortran. If this is not present I strongly advise you to get it installed on your system. For me this compiler works on Intel Macs too even though the current version is experimental.

7 Starting the first time

Open the directory rn001 which you have just created. Type `'crunch rn001 clear'`. This will leave you with just the files rn001.crysin, rn001.frefa and crunch.log. Now run Crunch by typing `'crunch rn001 try 1 5'`. Crunch will now proceed to solve the structure. On a modern computer this should take just a few seconds. The space group is P21, N=48, it is a simple steroid, an equal-atom structure.

When asked for input, choose default options by giving a return. When the program is finished your directory rn001 should contain the files rn001.report, rn001.pdb and rn001.spf, among others. Read the report file. If you wish to do so, plot the result using Ton Spek's program Platon, using the rn001.spf file. Compare the results, checking against the files present in the rn001 directory supplied. If you meet with any problems please contact the authors.

8 Solving a structure of your own

Open the directory where you would like to solve your structure. In the following 'code' stands for the name of your compound. This may be any suitable abbreviation, of course. If you already have a file in there 'code.crysin', containing the crystallographic data in the Dirdif format, so much the better. If not, Crunch will prompt you for the information required. The file 'code.crysin' will be created.

The file 'code.crysin' contains the usual crystallographic data such as cell constants, wavelength used, cell contents etc. It is important to give the cell contents as accurately as is possible. Next, you need a file containing h,k,l, F and sig(F), one reflection for each record, with proper separation between numbers (spaces, comma's or a plus or minus sign are required). The values for h,k,l should be consecutive on your file. However, the other variables may occur in any order. E.g. the hkl may be the last three numbers etc. In the record other numbers may be present, the program will skip them if required. Before you do anything else your data file must be converted into a 'code.frefa' file. Type `'crunch code conhkl 'filename'`. Here 'filename' is the full name of the file - NOT the full path - containing your reflections. Crunch will ask you a few questions and next the file will be converted to a Crunch-friendly one. The whole procedure is self explanatory. Now type `'crunch code try 1 5'`. Provide the information you are asked for.

Use the defaults indicated. Crunch determines automatically whether to treat the problem as a heavy atom structure or an equal atom one. In some cases you are given the option to treat for instance an organic compound containing chlorine as an equal atom problem. Basically you just give return a few times, Crunch will use sensible defaults and your structure will be solved. Try larger values of the last number given - 5 in this case - if you expect the solution to be difficult. Your structure should come out if all goes well. Crunch is a multi-solution method. In the example given, 5 random starts will be used to find the solution.

If you suspect your data is not truly of optimal quality, Crunch works best if you calculate E-values from your reflections using the option 'blessing'. See for details the section on syntax(Nr 7).

9 Using pmf

George Sheldrick, the author of the Shellx Program series, has pioneered the use of Patterson compliant starting sets instead of just random atomic positions to begin with. Crunch supports this as well. Use `'crunch code pmf 1 5'` to generate this type of start for Crunch. Next a normal run will use the starts generated by pmf automatically. The number of trials is limited by the number generated by pmf. See Crunch syntax for further detail. Note that using pmf is possible only if Blessings E's have been calculated. The program stops if this is not the case, telling you why.

10 A few helpful hints

1. Sometimes heavy atoms are disordered, for instance the chlorine in a perchlorate may be quite 'mobile'. If you expect something like that treat your structure as an equal atom one. Sometimes changing the cell contents - in 'code.crysin' - is needed in order to do this: a) If Crunch does not give you the option or b) you have a largeish organic molecule containing one or two sulfur or chlorine atoms. If Crunch has problems finding a solution, even if you told the system it is dealing with an equal atom problem, change the cell contents in your code.crysin file, replacing the heavier atoms by an equivalent number of oxygen atoms.

2. Atoms such as P, S and Cl are heavy in the context of Crunch. That is compared to first row elements such as C and O etc.

3. If you have a transition metal or an even heavier element in your compound DO NOT specify Cl and the like as heavy atoms.

4. If something has gone wrong, e.g. you've started Crunch while you didn't mean to and next interrupted the proceedings by 'Control c', ALWAYS start again by using the Crunch option 'clear': Type `'crunch code clear'`. This will clean up your directory, preventing otherwise inexplicable failures.

NB using the option 'clear' will delete the E-values you may have calculated using the option 'blessing'

11 Crunch' syntax

You may use any one of the options given below running the program. Always type your alias or the name of the dynamic link to the Crunch-script, followed by the complete specification of the run. The general syntax of a Crunch command line:

```
crunch code run [start end] [first n, last n, all]
```

Below the variables 'crunch', 'code' and 'run' have the following meaning throughout:

crunch: The name suggested for the link to or the alias of the script crunch.uni. In the distribution this link is named runit
code: compound name (e.g. rn001 in the test structure).
run: Defines the type of run which will be executed.
start: The number of the first trial do be done.
end: The number of the last attempt.

The last two values must be specified if one of the options `'first'` or `'last'` is to be used.

`conhkl filename` This type of run has a syntax different from the others. Use this run to convert your local structure factor file into a crunch-friendly (.frefa) format. The filename should be given with extension(s). See for more information the section Examples.

`clear` Most files created by earlier runs of Crunch are removed. Only the .crysin, deter.use, autofour.use, pmf.use and .frefa files are kept.

`blessing` E-values are calculated and saved for further use. The routines from the DREAR-suite, Levy and Eval are used. Reference: Blessing, R.H. & Smith, G.D. (1999). "Difference structure factor normalization for heavy-atom or anomalous scattering substructure determinations" J. Appl. Cryst. 32, 664-670. Until the option 'clear' is used to clean up, these E-values will be used rather than the ones calculated by Normal.

`pmf` If you have first calculated E's using the option blessing, the option pmf creates a file 'coordinates' containing partial structures derived from the patterson function. These will be used as starts to the phasing process instead of the random atomic positions which are used normally.

Values of start and end must be given to determine the number of starts to be generated.

`try` Crunch uses random starts from ’start’ until a solution is found or ’end’ is reached. If the structure is found various files such as code.pdb, code.spf and code.report are created. Some information on all trials done is given in the file ’hits’.

`collect` Scan all crunch-attempts from ’start’ to ’end’. The results are given in the file ’hits’.

`deter` Only the section calculating the phases - deter - is executed. The results are saved in files ’code.phi1...n’. As above, the calculations are done for the trials ’start’ to ’end’.

`peaks` The phases generated using the option ’deter’ are converted into possible fragments, using Exfft. This again is done for the trials ’start’ through to ’end’.

`recycle` This option implements a sort of ’shake and bake’ strategy. The trial ’start’ is run in the normal way. If a solution is obtained then the program stops as usual. If not, the best model obtained in the autofour section is used to calculate phases to be refined using deter. A new model is found etc. This process is continued until the structure is solved or ’end-start+1’ cycles are completed.

`autofour` This option may be used in two completely different ways:

1. You may have found a heavy atom or a small part of the structure by using some other method. If you then prepare a .frefa file as was described above, followed by creating a file ’code.pek’ containing the known part of your structure, you can try to extend the model to the complete structure by just typing `'crunch code autofour'`, not giving the values start and end. A description of the code.pek file is given below.

2. You have created files ’code.pek1...n’ by using the peaks option. Then type `'crunch code autofour start end'` to try to extend one or more of the models obtained.

The values `'start'` and `'end'` are optional except if you use `'recycle', 'deter', 'pmf' or 'peaks'`. The default values are 1 and 10 respectively.

Using the option `'try'`, which is normal when you are trying to solve a structure, usually the second step, model extension using autofour, is started with the largest consistent fragment present in the map. However, following the value of `'end'` you may specify the options `'first'` or `'last'`, followed by a number 'n', or `'all'`. The effect is that the first 'n', the last 'n' or all peaks found in the map, are used as input to autofour. Remember to specify the values `'start'` and `'end'` if you'd like to use the options just described. The script gets confused if you specify just the parameters `'first'` or `'last'`, interpreting them as values for `'start'` and `'end'`.

An example run: `'crunch rn001 collect 27 74'` means crunch rn001 and collect 'hits' from sets 27 to 74. Hits contains a brief report on this run. If you are using the option 'try' the results of the first successful run are summarized in a file code.report. Should you want to look at the results of a particular successful run collected in the file hits, use the command `'crunch code try n n'`. Further examples may be found in the designated section.

12 Required before running Crunch

You just need a structure factor file, the usual crystallographic data such as cell constants, cell contents, the wavelength used etc. Any structure factor file containing one reflection pro record will do, as long as h,k,l, Fobs (or F**2) and sigma(F) (or sig(F**2)) are there. Remove any records which contain "non intensity" information from the file. Convert the file to a 'code.frefa' file using the option conhkl. Do use an alias or the dynamic link provided to run the script crunch.uni. In the special case you want to use the syntax `'crunch code autofour'`, that is, you want to use crunch just to extend a model found some other way, you need to prepare one more file, 'code.pek'.

13 What to do in the case of problems?

In the case of failure of a standard run of Crunch - the structure does not come out - try changing some input parameters to deter. Look in the file deter.asc which should be present in the Crunch subdirectory 'drivers'. Create a file called deter.use in the directory you are trying to solve your structure from. This file should be of the same format as deter.asc, however you need to specify only the items you wish to change.

1. Increase the number of trials.

2. Use the option 'blessing' to calculate E-values and try to solve the structure again. This is particularly useful if the quality of your data is not completely up to scratch. In one test, collecting the results of 100 trials, using Blessing's E's six solutions were found, instead of two obtained using Normal80's E's.i

3. Try generating starts using pmf

4. In the subdirectory 'drivers' of the Crunch system files 'deter.asc, autofour.asc and pmf.asc' are present. These files contain defaults for the programs deter, dutofour and dmf. In the directory were you are trying to solve your structure create a file 'deter.use', of a similar format. You need to supply only those parameters you would like to give non-default values to. It is best not to change NNRM, INMX and NINMX in this way, as Crunch allows you to specify the number of matrices used and their orders when you start your efforts or after you've cleaned up the directory.

   Try different values for the following parameters:

   IBK the default value is calculate by the system, based on Crunch using approximately 256 Mb. However, sometimes smaller values give better results. Try something like 1500.

   NRL Increase or decrease the value a bit.

   ICRA You might like to try the value 2 here.

   NOTF This parameter defines the weight of forbidden reflections in the matrix construction process. The default is one, i.e. they are treated as valuable compared to very weak reflections. Sometimes treating them as statistically very weak works better: Change the value into zero. Changing the value into two is another option, increasing their importance.

   Try all this using the option `'crunch code deter 1 1'`. Inspect deter.cout and try to maximize the average E-value in the matrices and/or the number of strong, independent reflections.

5. For larger structures it is sometimes useful to use matrices of smaller dimensions than the default chosen by Crunch. Type `'crunch code clear'`. Next, type `'crunch code deter 1 1'` and now, when the system asks if you would like to use default values for the matrix construction, type `'u'`.

   Choose a different, preferably lower, number of matrices and/or a different order for the matrices. The minimum number allowed is 1. The number of matrices you may use is prescribed, not just any positive number will do. If you type a number here which is unacceptable Crunch will substitute the nearest acceptable one. The order of the matrices you may choose to be any odd number. Try to choose your parameters to fit the requirements. E-average should be high. The number of strong, independent reflections should be at least about 2.5 times the number of independent atoms you are looking for.

A significant number of symmetry equivalents - including Friedels - should be present. If you succeed in finding a set of matrices which looks promising, - high average E, enough strong independent reflections while containing a sufficient amount of redundancy - try to solve the structure again using the options `'try'` or `'collect'`.

6. Generate about a hundred solutions. Look in the file 'hits' for the solution which yields the lowest value for R2. Type: `'crunch code recycle n n+10'` or 20. n is the number of the trial chosen. If this does not work and other trials looking better than most others exist, try them in the same way.

7. As there is a file deter.asc, so files autofour.asc and pmf.asc are present in the directory 'drivers'. The format is the same as is its use: You may override the values given in *.asc by creating a file *.use in your working directory, specifying different values therein. If you suspect your reflection data is not of terribly good quality, try a different value for STOPR, something like 40. STOPR is the threshold value of R_2 in percentage points. For values of R_2 below STOPR autofour considers the structure to be solved. Should your file contain reasonable standard deviations, try using IWEIGHT=1. Or, if you're trying to solve a structure containing one or two relatively heavy atoms such as Cl or S, try giving IVAL some value, say 10 or 20. The latter is a good idea too if you use Crunch just to extend a small model based on a few (heavy) atoms.

If all this does not work give up. Crunch will not solve this structure, at least not for you. If you like, contact the authors, giving full details of what you have done so far.

14 The file code.pek

For each atom in your input model the file should contain a record as defined below.

1 - 10		May be used for atom name	Format: 10X
11 - 20	FI	Table number belonging to this atom	Format: F10.0
21 - 30	X	The fractional x-coordinate	Format: F10.3
31 - 40	Y	The fractional y-coordinate	Format: F10.3
41 - 50	Z	The fractional z-coordinate	Format: F10.3

Note: The records specify the format of the code.pek file, needed if you want to run autofour with a model found NOT using Crunch. Pay attention to the parameter FI, its value should tally with the input you have specified in your file 'code.crysin' For instance, you've found a Cl-atom and you would like to use this to find the other atoms. You specified Cl as the second atom type in your crysin file. Use FI=2.0 in this case.

15 Some useful files

flags This file contains the compiler and linking options during used during installation of the system. The adventurous user might want to play with this.

code.frefa The reflection file which must be present. However, you may convert a reflection file in a local format using the option provided. The .frefa file may be used in the Dirdif system as well.

code.crysin The Dirdif system file which contains the crystallographic data on your compound. This may be prepared interactively.

hits For each random start this gives the determinant value obtained as well as the values of R_2 and R_2 expected found by autofour.

crunch.log Contains a log of all Crunch's runs submitted from this directory.

deter.use Similar to 'drivers/deter.asc'. Values specified will override defaults in 'drivers/deter.asc'.

autofour.use As 'deter.use' but defining the input to autofour.

deter.cout The output of the program deter. Only the output of the last run is kept.

pmf.use As before, you may change parameters for pmf this way.

autofour.cout The output of autofour's last attempt.

The last two files are not present if the Crunch run was successful. The following files are present in the case of success only.

code.report This file contains relevant information on the proceedings leading up to the result.

code.spf The coordinates in spf format. This file may be used to prepare plots of the molecule(s) found using Pluton or Platon.

output The file contains all output generated during the last successful attempt.

code.pdb The coordinates in Protein data bank format.

16 Examples

1. Converting a local structure factor file named 'xtal.dat'

   In this example the user typed 'crunch hornef conhkl xtal.dat'. The following information appeared on the screen - the bit in [] was provided by the user - :

   ```
   ==================================
   ** CONHKL conversion of reflection files
   ==================================
   ** The first record on your reflection-file is:
   HKL 0 -2 0 66.23596 1.00171 1 SKIP
   == > You must now specify your record type using keywords
   ** Example:  hkl skip fobs sigma
   ** This means:
   ** The first 3 numbers are h,k,l, the 4th should be skipped,
   ** the next one is Fobs and the last one is sigma(Fobs)
   ** Typing f2obs instead of fobs tells the program your file
   ** contains F**2 rather than F
   == > Specify your record:  [skip hkl fobs sigma skip]
   ==================================
   ** conhkl ended
   ```

   Users of the package XTAL will recognize the record shown as typical input of ADDREF, the program in XTAL which reads in structure factors. Hornef is the name of the compound. The programs shows the first record of the file, followed by an explanation on how to proceed. Here the records consist of a bit of text, h,k,l, Fobs, sigma(Fobs) and some other information which Crunch does not need. By typing the record between brackets the user tells Crunch to skip the first item and to consider the next five numbers as h,k,l, Fobs and sigma respectively. The rest of the record is skipped again. Use lower case characters here! At the end of the run the file 'hornef.frefa' was prepared, ready for use by Crunch.

2. Trying to solve a structure the first time.

   The user next typed `'crunch hornef try'`. On the screen appeared:

   ```
   Crunch interactive, always give return to choose defaults
   CELLCO C 92 H 72 O 20
   The structure is an Equal_atom structure
   [D]efault or [U]ser specified matrix construction?
   ```

   The cell contents are shown on screen. Crunch has decided that hornef is an equal atom structure. The user gave a return on the next question as he wanted to try a default run of Crunch. In this case a maximum of 10 random starting sets is tried. If you next run Crunch you will not be asked these questions again. Should you want to change anything, for instance the way the Karle–Hauptman matrices are constructed, you will have to remove the file 'code.par', in this case hornef.par. You may choose to use the command `'crunch code clear'` instead.

3. The user wants to collect the results of a number of trials, to see whether there are any (non)solutions which look more promising than the others. See the section 'What to do in the case of problems?'. He typed: `'crunch hornef collect 1 25'`. After a few moments the Unix prompt reappears.

4. On completion the file 'hits' is inspected. Solution number 9 looks better than the others: A higher value of the determinant and a lower final value of R_2 are the criteria to look for. The user now typed: `'crunch hornef recycle 9 15'`. Again after a few seconds the Unix prompt reappears. Crunch will do the calculation for trial number 9 again. However, instead of proceeding to trial 10 in the case of failure, the best model obtained is used to calculate phases which are used as starting values for the maximization of the determinants. The refined phases are used to calculate a new start for the second section of Crunch. In this case this is done six times (15-9). Of course Crunch stops iterating if a satisfactory solution is obtained. Note that it is not necessary to use the options `'try'` or `'collect'` before using `'recycle'`. It is perfectly valid to try to solve your structure using a run such as: `'crunch hornef recycle 1 10'`.

GX Manual
(source code deposited on newsletter webpage)

by Ken Muir

Chemistry Department, University of Glasgow, Glasgow, G12 8QQ, United Kingdom. E-mail:
ken@chem.gla.ac.uk

Place holder page. Report-proper starts on following page

CONTENTS

The GX Package

A collection of compatible programs for crystal structure determination, by C.J. Gilmore, P.R. Mallinson, and K.W. Muir.

April 1985

Chemistry Department,
University of Glasgow,
Glasgow G12 8QQ,
Scotland.

DISCLAIMER

The authors and the University of Glasgow make no warranty with respect to the contents hereof. No liability is assumed for consequences of applying the methods described in this document.

Reprinted with corrections October 1985

1. INTRODUCTION TO GX

1.1 File and program organisation.

Crystal diffraction analysis has traditionally been carried out via batch processing on a mainframe computer, or by using a dedicated 12 or 16-bit minicomputer. Enlargements in the scale of circuit integration are making mini and micro-computers with 32-bit registers and larger program address spaces more cheaply available; these machines provide the power, scope and ease of programming of a mainframe while retaining the closely-interactive features of a mini. At the same time, the spread of data communications networks is making mainframes more widely accessible via interactive terminals. GX, the Glasgow X-ray crystallographic package (Mallinson and Muir, 1985; Gilmore, 1984) has been designed for interactive operation in the laboratory, employing a screen or text editor to monitor and manipulate files. The heart of the system is the CRYSTAL MODEL FILE, a free-format character file containing the parameters of the crystal structure model. It is described in Section 1.2 below.

Reflection data are logically separate from the crystal model, and program commands are logically separate both from the crystal model and from reflection data, for easy keyboard input during real-time interactive running. Commands are presented in one of two forms - dialogue or 'menu'. In a dialogue the user is prompted for specific items, and null responses (pressing the 'return' key) select suitable default values. A menu consists of a list of commands (with optional parameters), the commands being given in the order desired by the user. The underscore character '_' which appears in some command names in GX is used merely to assist readability; command names must contain no embedded spaces. Command parameters may be separated by spaces or commas, and a comma may be used to denote the default value of the corresponding parameter. The MITHRIL program (section 3) has slightly different conventions regarding command format; refer to the MITHRIL manual for details.

The package consists of independent programs that communicate via a standard set of data files, principally the model and reflection files. The operating system and its utilities are used for editing the crystal model, for control of the overall calculation sequence, and for file management; hence there is considerable flexibility for the programs to be installed and run in the best way to suit local conditions.

Sections 2 to 7 of this manual correspond to the main types of calculation performed during structure analysis. The Direct Phasing program MITHRIL is described in a separate manual as indicated in Section 3. The stages following structure solution in a typical determination are often those which account for most of the time spent by both computer and structure analyst. This phase includes the early refinement, location or placing of hydrogen atoms, checking of weighting schemes, postulation of disorder models, final refinement, geometry calculations, generation of numerical tables, and computer-graphical evaluation of both molecular and crystal structure. In these processes interactive operation produces results very quickly and conveniently. Figure 1.1 shows the relationship between the various programs and files in the GX system; circles represent files, oblongs represent programs and arrows indicate input and output. Sequences of operations corresponding to pathways in the diagram can generally be performed in real time. Section 8.4 suggests a systematic scheme for naming the files.

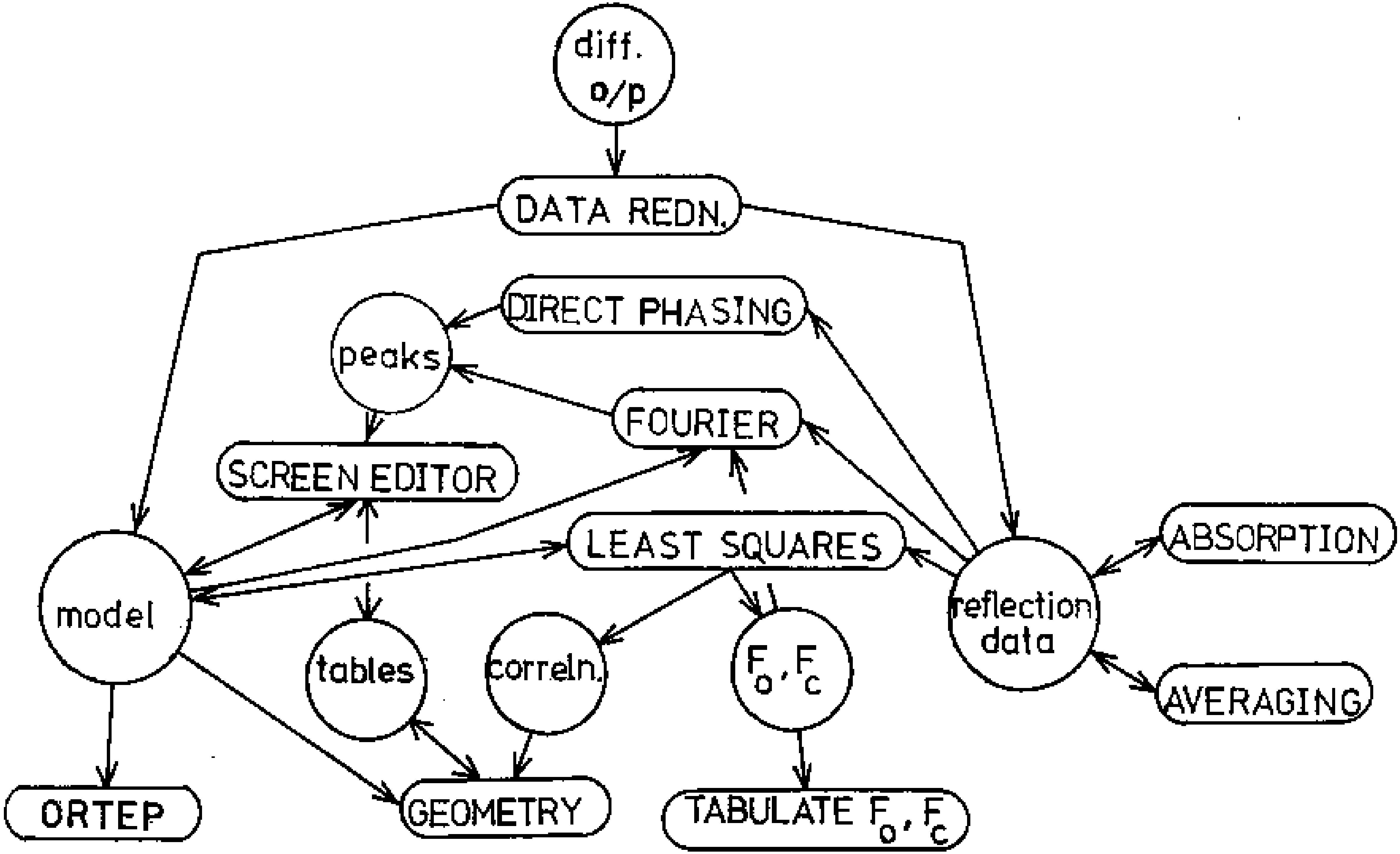


Figure i.1 The GX flow chart.

1.2 The crystal model file.

This file is a complete description of the experimental model of the crystal and contains all the parameters furnished by crystal structure analysis. It consists of a number of free-format entries each prefixed by a tag that identifies the type of information. The tag is followed by one or more parameters whose names appear in the specifications below. To continue an entry on the following line, put the symbol '=' (preceded by at least one space) at the end of the data in the current line. Only the first 72 characters in each line are significant. Items within a line may be separated by spaces or commas, and a comma may be used to denote the default value of the corresponding item.

TITLE information to identify the crystal
The title is written out by programs that read the model file.

CELL a b c α β γ
Unit cell parameters in Angstroms and degrees or cosines. Default angles are 90 degrees, default axis lengths 1 Å.

CELLSD σ(a) σ(b) σ(c) σ(α) σ(β) σ(γ)
Estimated standard deviations of the cell parameters in Angstroms and degrees or cosines. Default values are zero.

LAMBDA wavelength
The radiation wavelength is given in Angstroms. It is used in extinction corrections, and to select the appropriate anomalous scattering factors for Cu or Mo radiation from the system databank file - see SFAC below. Default value is zero, which selects scattering factors for Mo radiation.

LATT centrosymmetry flag lattice type
The centrosymmetry flag must be given as either A (non-centrosymmetric) or C (centrosymmetric). Lattice type may be P, I, R, F, A, B, or C. Rhombohedral lattices indexed on hexagonal axes (lattice type R) must be given as the obverse cell (-h+k+l=3n). Note that rhombohedral lattices indexed on rhombohedral axes have lattice type P. The default setting is 'LATT C P'.

SYMM general position coordinates
These are given exactly as in the International Tables, the three coordinates being separated by commas (spaces are insignificant here). Positions generated by a centred lattice or a centre of

symmetry should be omitted, and the origin must be at a centre of symmetry in centrosymmetric structures. The entry 'SYMM X,Y,Z' is always assumed and will be ignored if given. More than one position may be given within one SYMM entry, if desired, by placing a semicolon between each, e.g. for space group $I4_1$ (no. 80):

```
LATT    A   I
SYMM    -X, -Y, Z;    -Y, 1/2+X, 1/4+Z;    Y, 1/2-X, 1/4+Z
```

or a SYMM entry may be given for each position.

SFAC element a_1 b_1 a_2 b_2 a_3 b_3 a_4 b_4 c f' f'' μ r d
The coefficients a_i, b_i, and c fit the scattering factor curve for the given element to the analytic expression

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

The real part of the complex scattering factor is then f+f' and the imaginary part is f''. The total atomic cross-section is μ $\times 10^{-24}$ cm^2, the covalent and van der Waals radii are r and d in Å. The SFAC information for nearly all neutral species of atom is held in the system databank file, to which reference is automatically made in the absence of an SFAC entry for a particular element in the crystal. In that case the values of f' and f'' for Mo K_α radiation are used if the wavelength given in the LAMBDA entry is less than 1.0 Å, those for Cu K_α if $\lambda > 1.0$ Å. All scattering factors and absorption coefficients have been taken from International Tables for X-ray Crystallography (1974). If an element appears in both SFAC and CONTENTS entries, its SFAC entry must come first.

The sequence of SFAC entries, if there are any, is used as the sequence of type numbers for the optional SHELX form of atom parameter entry - see 'ATOM' below.

CONTENTS $n_1$element$_1$ $n_2$element$_2$ $n_3$element$_3$ ...
The n_i are the number of atoms of each type in the unit cell. Each n_i is followed without a space by a chemical element name, no more than 13 such parameters being allowed. The cell contents information may be used by the direct phasing program MITHRIL, and correct CONTENTS are required to evaluate F(000) in F_o Fourier synthesis and for μ in Gaussian absorption corrections.

The sequence of element types is used as the sequence of type numbers for the optional SHELX form of atom parameter entry - see 'ATOM' below. The CONTENTS entry may be just a list of element names with no n_i prefixes.

SCALE G_1 G_2 G_3 G_4
The F_c scale factors for up to four batches of reflection data. Default values are 1.0. Note that the BLOCK least-squares refinement program expects all the reflections to be on the same scale and uses only G_1. RBLS is always required when there is more than one scale factor.

EXTINCTN r^*
The isotropic extinction coefficient is defined by Larson, 1970. Default value is zero. If refinement by RBLS is desired, a suitable starting value is 0.1.

ETA eta
The coefficient of the imaginary component of the anomalous scattering factor is defined by Rogers, 1981. Default value is +1.0. The absolute configuration may be determined by allowing RBLS to refine this parameter; a suitable starting point is any non-zero value between -1.0 and +1.0.

Note two cases in which refinement can proceed to a false minimum in polar space groups (Cruickshank, 1967).
(i) as a result of the polar dispersion error when ETA is set to zero;
(ii) as a result of refining atom positions when the absolute configuration is undetermined and ETA is non-zero.

A safe procedure for polar space groups would be to refine initially with ETA=0, then calculate structure factors with ETA=1 and with ETA=-1 to establish the absolute configuration. Final positional refinement should then be carried out with the correct absolute configuration and ETA=1. Alternatively, one could refine initially with ETA=0, then finally with ETA=1 and ETA refinement switched on (see section 5.3). A change in the sign of ETA would indicate the opposite configuration, but the true minimum will have been reached.

ATOM label x/a y/b z/c occupancy U_{11} U_{22} U_{33} U_{12} U_{13} U_{23} [CN]n
The maximum number of atoms in the asymmetric unit is 200, all of which may be anisotropic. Isotropic atoms are distinguished by having only one temperature factor component. The label consists of an element name, optionally followed by any other characters enclosed in brackets, e.g. H(12A), CU(+2), NA, up to a total

label length of 8 characters. The element name is used to identify the scattering factor, so there must be a matching SFAC entry or a matching entry in the system file of scattering factors. The occupancy for an atom in a special position should be divided by the number of general positions which have coalesced into the special position, e.g. an atom on a two-fold axis usually has an occupancy of 0.5. The 'CN' parameter is optional; it defines a coordination number for the atom, which is used by a program for placing hydrogen atoms. The square brackets in the specification above should be omitted - they merely signify for documentation purposes that their content is to be reproduced literally, e.g. 'ATOM C CN4 ......' specifies a tetrahedral carbon atom. The CN parameter may appear anywhere within an ATOM entry.

There is an alternative form of atom parameter entry, provided for compatibility with SHELX (Sheldrick, 1976) as follows:

label type x/a y/b z/c occupancy U_{11} U_{22} U_{33} U_{23} U_{13} U_{12}

Note the omission of the ATOM tag and the different order of the U_{ij} parameters. Again, isotropic atoms have only one temperature factor. The 'type' parameter identifies the scattering factor - type n is the nth element in the CONTENTS entry. If there are SFAC entries, any corresponding CONTENTS items do not count in the sequence, and the SFAC entries count instead. The label can be any string up to 8 characters in length, e.g. H', C12. A 'CN' parameter may, optionally, appear anywhere after the 'type'. This SHELX form of parameter entry will be preferred if tables or illustrations are required without brackets in the atom labels. Both types of atom entry may be used in admixture if you wish.

Refinement flags and constraints do not appear in the crystal model file; that information is given in the form of a least squares program command. If the occupancy and U_{ij} parameters are omitted, the atom is given an occupancy of 1.0 and an isotropic temperature factor of 0.05 A^2. The order of atoms in the model file is immaterial, with the important exception of input for the Interactive ORTEP program, which assumes that atoms of the same type are grouped together.

ATOMSD label $\sigma(x/a)$ $\sigma(y/b)$ $\sigma(z/c)$ σ(occupancy) $\sigma(U_{ij})$
This entry gives the estimated standard deviations of the atom parameters, where the $\sigma(U_{ij})$ are in the same order as the U_{ij} under ATOM above, even if the corresponding atom entry is in

SHELX form. The label must match one in an atom entry of either type. The order of ATOMSD entries is immaterial, and they may be interspersed with atom entries if you wish.

2. OPERATIONS ON THE REFLECTION DATA AND CRYSTAL MODEL - PROGRAMS SORT, XYZ, ABSORB, DIFABS AND CAD4

2.1 SORT program.

The reduction of raw intensity data is laboratory-specific, often carried out on the diffractometer itself, and must precede input to GX. However, a program for processing the output of the Enraf-Nonius CAD4 instrument is provided (see section 2.5). Users of other types of diffractometer should refer to section 8.3 for help with connecting their files.

After the binary reflection file has been created it can be manipulated by the SORT program for index transformation, sorting and averaging. The program decides which reflections are equivalent on the basis of the point group implied by the positions in the model file, and merges them by assigning the average intensity. Systematically-absent reflections are rejected. The Miller indices of averaged reflections are chosen (and others may be changed) so that all reflections lie in the same hemisphere in reciprocal space. A new reflection file is written, in which the indices vary in ascending order of l, then k, then h. The options available in the dialogue allow you to specify an index transformation or to apply a cut-off at a particular value of $I/\sigma(I)$. The cut-off is applied at the output stage, after averaging and the accumulation of merging R-factor totals. Index transformation, if any, is applied on input, so the unit cell parameters and equivalent positions in the model file must refer to the transformed cell.

2.2 XYZ program.

The crystal model file can be originated and simple amendments made using the screen or text editor. More complex operations are carried out with program XYZ, which writes a new model file. The options are as follows.

(i) **TIDY**
Edit the model file by sorting so that atoms of the same type are grouped together in order of ascending sequence number. Atomic types are ordered according to the order of element names in SFAC (or CONTENTS) entries.

(ii) Carry out various crystallographic operations on the model, viz. inversion of configuration (**INVERT**), general transformation of coordinates (**CON**), construction of planar groups with idealised geometry (RING), addition of atoms (e.g. hydrogen) in theoretical positions (ADDH), addition of an atom at a centroid (CENTROID). Addition of atoms in theoretical positions proceeds according to the coordination number (CN) information in the model file, the number of atoms added to each existing atom being CN minus the number of existing bonds. The number of existing bonds is automatically deduced by searching for neighbouring atoms within the sum of the atomic radii (the search algorithm uses van der Waals radius information also). For organometallics with e.g. cyclopentadienyls it is usually advisable to set the metal covalent radii to zero.

2.3 ABSORB program.

ABSORB performs absorption corrections by Gaussian quadrature over the crystal volume. It requires for each reflection the direction cosines on the crystallographic axes of the incident and diffracted beams and also precise knowledge of the shape of the crystal. The mean X-ray path length is calculated and can be used to improve the refinement of the extinction parameter. The code is a straightforward implementation of the procedure described by Coppens, 1970.

The crystal model file is needed only to supply CELL information. It is advantageous to use the cell dimensions employed during data collection without alteration. If a CONTENTS card is supplied the linear absorption coefficient can be calculated from it using the system databank.

The input reflection file must contain direction cosines (see section 8.3). The program CAD4 described below supplies these correctly for all setting geometries if the DCOS option is in force. It is unwise to modify the processed data in any way before applying the correction.

The output reflection file contains the corrected $|F_o|^2$ and $\sigma(F_o{}^2)$ values and a calculated mean X-ray path length for extinction refinement. Command input is as follows. Sensible default values are supplied for all items except the crystal geometry, which must be defined by FACE commands.

FACE h k l d

For each face give its Miller indices and distance d mm from an origin within the crystal.

MU μ

Give μ cm^{-1} to override the value calculated from the model file.

GRID p
Flexible grid chosen to give an integration error of p%; the default is 1%. Alternatively, GRID NX NY NZ specifies a fixed grid with the given number of points along each axis. NX x NY x NZ cannot be greater than 10240.

LIST n
Print results for the first n reflections. If n is negative, the program returns to command menu input on completion of n reflections.

The output from the program lists the crystal vertices and edges; the calculation stops if faces + vertices - edges = 2, i.e. if the crystal is not a convex polyhedron. During processing, theta values are calculated from the direction cosines and cell dimensions. Any discrepancy greater than 0.2° causes the reflection to be printed with IERR = 1. If this happens check that the cell dimensions are compatible with those used to compute the direction cosines - reorientation during data collection can sometimes cause this problem.

It is very difficult to be certain that an absorption correction is free from error. The face indexing and crystal measurement should be done with great care. Check that the correction improves internal consistency between equivalent intensities and leads to flatter difference maps. If this does not happen look carefully again at the correction.

Remember to change the SCALE parameter in the model file - a new value is suggested by the program.

2.4 DIFABS program.

This program applies a correction for absorption/extinction by fitting a transmission surface to the differences between $|F_o|$ and $|F_c|$ at the end of isotropic refinement. It is a GX-compatible version of the program described by Walker and Stuart, 1983, and in this version it is assumed that the data were measured in bisecting mode on an Enraf-Nonius CAD4 diffractometer. The orientation matrix defining the reciprocal cell vectors on the diffractometer axes is required in response to a prompt at the terminal. The matrix is checked for consistency with the CELL data in the model file (the model file is used only for this). If the check fails the program re-prompts. The program also permits the theta dependence of the correction to be ignored if desired.

The input reflection file is an F_c file (see section 8.3). The reflection indices must be exactly as in data collection and must on no account be modified by sorting or merging, therefore the SORT program must not be applied to any reflection data which are going to be input to DIFABS. The $|F_c|$ values should correspond to a refined isotropic model - try to obtain as good an isotropic model as possible. It should be based on data which have been corrected for crystal decomposition (if necessary) and which have been sorted, averaged and are free of systematic absences. In principle the hydrogen scattering should be included, but omitting the hydrogen atoms does not usually give rise to problems. Once the isotropic refinement is complete, calculate structure factors using the file written by the diffractometer processing program and use this F_c file as input to DIFABS.

The output reflection file contains the corrected $|F_o|^2$ and $\sigma(F_o^2)$. Remember to set SCALE to unity in the model file after the correction.

2.5 CAD4 program.

This program corrects Enraf-Nonius CAD4 diffractometric intensity data for background, variable scan speed, Lp effects etc. and outputs a file containing h, k, l, $|F_o|^2$, and $\sigma(F_o^2)$, as detailed in section 8.3, which can be used by the GX programs. CAD4 is a modified version of an original program by Hursthouse and Davies, 1978. Command input options are as follows.

TITL a title of up to 60 characters

MONO θ_m

The monochromator Bragg angle (default 6.0817° suitable for a graphite crystal with Mo radiation). Positive θ_m values give Lp correction based on an ideally imperfect crystal monochromator whereas negative θ_m implies Lp correction based on a perfect monochromator crystal (mosaic spread about 0.5°). The 1/Lp factors applied to the intensities are, respectively:

$$\sin 2\theta(1 + \cos^2 2\theta_m)/(\cos^2 2\theta + \cos^2 2\theta_m)$$

and

$$\sin 2\theta(1 + |\cos 2\theta_m|)/(\cos^2 2\theta + |\cos 2\theta_m|)$$

ATEN attenuator factor

Defines the factor appropriate to the diffractometer and radiation (default 19.2).

SCAL scale factor

This factor is applied to the intensities (default 1.0).

NSET scale factor number

The GX system allows up to four different scale factors. Default NSET = 1.

PFAC fudge factor for intensity e.s.d.

This parameter is used in the calculation of $\sigma(F_o^2)$ and prevents excessive weight being given to large structure factors. The expressions used are:

$$\sigma(F_o^2) = [C + 4(B_1 + B_2) + (PFACxI)^2]^{1/2}/(t_cLp)$$

$$\sigma(F_o) = \sigma(F_o^2)/(2F_o)$$

$$F_o^2 = I/(t_cLp)$$

$$I = C - 2(B_1 + B_2)$$

where C is the total integrated count in time t_c. B_1 and B_2 are background counts, and Lp is the correction factor for Lorentz and polarisation effects. t_c is divided by the attenuator, decomposition ('fading') and scale factors, if any. The default value of PFAC is 0.03.

IMIN minimum intensity

Specifies an intensity to be assigned if experimental $I < 0$ (reflection code 'U'). Default IMIN is 1.0. For such reflections $\sigma(F_o^2) = 5F_o^2$.

FADE I0 I1 I2 N1 N2

Apply a linear crystal decomposition correction. The intensity at the start is I0 and at reflections N1 and N2 it is I1 and I2. The values are output at the terminal for checking before the calculation starts.

PROF

Intensity profiles are present in the diffractometer output file (default is that they are not).

REJP
Call subroutine REJECT supplied by user (default - do not call REJECT).

DCOS
Write direction cosines for Gaussian absorption correction program ABSORB. This uses extra disc space, so the default is to omit the cosines from the reflection file.

BGIN NREFA
Start processing at reflection number NREFA in the diffractometer output file. Default n=1.

SKIP
A previously-processed reflection file will be copied to the output reflection file before processing starts. This is necessary if data from several diffractometer output files are to be combined in the same reflection file. DCOS must be set in the same way for all files.

OUTP i
Print every ith reflection (default i=100). Reflections flagged in any way (e.g. because the attenuator was used) are always printed whatever the value of i.

END
Terminates the command input and starts processing.

DOTO NREFB
Process up to reflection number NREFB then re-display the menu to allow alterations in parameters (e.g. FADE, NSET) before continuing.

STOP
Used only after a DO TO command, so as to stop processing at NREFB. This allows a part of the data to be selected.

MENU
List the menu again.

Note that PROF, REJP, and DCOS are initially set to NO. Each time they are typed they change value from NO to YES or vice versa, e.g. DCOS twice would mean no direction cosines were to be written.

Print-out from the program includes a summary table for various classes of reflections, from which systematic absences may be detected. Intensity standards are written on a separate file for subsequent analysis.

STEREO separation
Causes the saved sequence to be executed twice, with stereoscopic rotations of the viewpoint. The 'separation' parameter is the required distance in inches between left and right-eye views, usually about 2.5 for a small-scale viewer. A suitable model scale, depending on the size of the subject, must be preset by a SCALE command. Alternatively, omitting the 'separation' parameter makes the program pause after drawing the left-eye view to allow a change of paper. The resulting large-scale stereo pair can then be viewed in a mirror viewer or reduced photographically.

NEW_PLOT view distance x origin
After pausing for permission, clears the screen and re-displays the menu if the MENU switch is on. This command prepares for a new drawing, for example after ROTATE, and is typically followed by EXECUTE. The origin point of the current graphical device will be shifted along the x axis if the 'x origin' parameter is given (in inches, positive or negative). The NEW command can also change the perspective view distance from its default value of 30 inches, if the 'view distance' parameter is given. A view distance of 0.0 signifies infinity, i.e. a parallel projection.

PLOTTER
Defines the current graphical output device to be the plotter. Is normally followed by EXECUTE or STEREO, to get a hard copy of a screen display.

PEN pen number
On multiple pen plotters, performs a change of pen under program control. By inserting PEN commands at suitable places in a saved sequence, red-green stereo pairs or different coloured atoms may be drawn without stopping the plotter.

VDU
Defines the current graphical output device to be the screen, which it is initially, and re-displays the menu if the MENU switch is on.

MENU ON/OFF
Inverts the setting of the menu switch (see NEW_PLOT and VDU above). Initially it is on.

NUMERICAL COMMANDS
Permits the original ORTEP instructions to be used (these are

described in ORNL-3794, Johnson, 1971). For example, 401 or 402 instructions combined with 406 or MOLECULE can be used to generate particular regions of the crystal structure, perhaps the environment of an included species. NUM suppresses checking of command syntax so it is best used only in a command file. This would normally be the case since the original instructions have fixed format and are more conveniently set up with an editor than typed while ORTEP is running.

EXIT
Causes ORTEP to stop and return control of the terminal to the operating system.

7.5 Examples.

7.5.1 Single molecule with line bonds and atom labels.

```
MOLECULE
BEST VIEW
SAVE
BONDS
C C
C S
S O
<blank line>
LABEL ALL
END
EXECUTE
PLOTTER
EXECUTE
```

7.5.2 Ball-and-spoke molecule with different atomic styles and colours.

```
MOLECULE
BEST VIEW
SAVE
BONDS 5 .04
C C
C S 2 .02
S O
<blank line>
```

```
ATOMS UNSHADED C
PEN 5
ATOMS SHADED S
PEN 2
ATOMS SHADED O
END
PLOTTER
EXECUTE
```

7.5.3 Red-green stereo view of unit cell contents.

```
CELL
BEST VIEW
ROT Z 175
SAVE SEQUENCE
BONDS
C C
C S
S O
<blank line>
OUTLINE CELL
LABEL AXES
PEN 2
END SAVED SEQUENCE
PLOTTER
PEN 6
STEREO
```

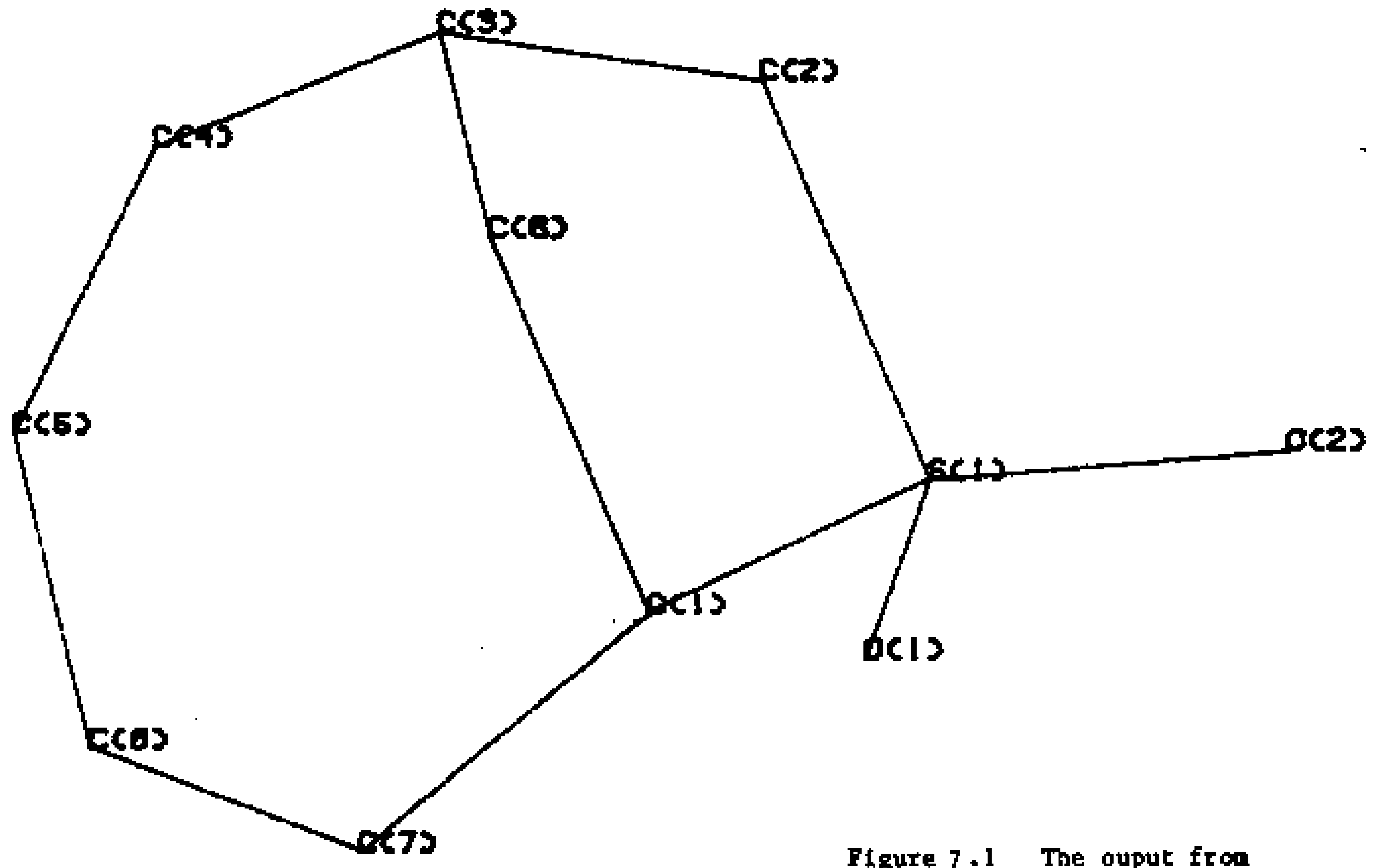
7.5.4 Small-scale side-by-side stereo pair. - This is the same as Example 6.5.3, except for the last two lines and the omission of the 'PEN' commands:

```
PLOTTER
SCALE 0.1
STEREO 2.5
```

All the above examples use the following crystal model file.

```
TITLE BRIDGED SULPHONE
CELL 11.256 5.951 13.099 90 114.44 90
SYMM 1/2-X, 1/2+Y, 1/2-Z
ATOM C(1) .70727 .10081 .58981 1.0 =
 .03661 .04154 .05204 .00197 .02104 .00559
```

```
ATOM C(2) .94398 .26707 .62216 1.0 =
 .04269 .05638 .05423 -.00725 .01353 -.00032
ATOM C(3) .82971 .35317 .51754 1.0 =
 .04785 .04186 .05031 .00423 .01791 .00608
ATOM C(4) .82059 .22760 .41675 1.0 =
 .05966 .08118 .04791 .01067 .02251 .00860
ATOM C(5) .74881 .04786 .36966 1.0 =
 .06549 .09628 .04481 .02541 .01034 -.01587
ATOM C(6) .66890 -.08695 .40603 1.0 =
 .05769 .05139 .05841 .01181 -.01541 -.01698
ATOM C(7) .65293 -.07583 .50016 1.0 =
 .03519 .03836 .08242 -.00106 -.00135 .00649
ATOM C(8) .70943 .33377 .54179 1.0 =
 .05225 .03474 .04931 .00836 .01633 -.00265
ATOM S .87792 .05506 .68127 1.0 =
 .04094 .05015 .03857 -.00239 .00397 .00555
ATOM O(1) .92284 -.16302 .66651 1.0 =
 .05252 .04313 .10529 .0108 -.00105 .0094
ATOM O(2) .90028 .12222 .79262 1.0 =
 .09104 .12328 .03611 -.02791 .01718 -.00596
```



<u>Figure</u> <u>7.1</u> The ouput from Example 7.5.1.

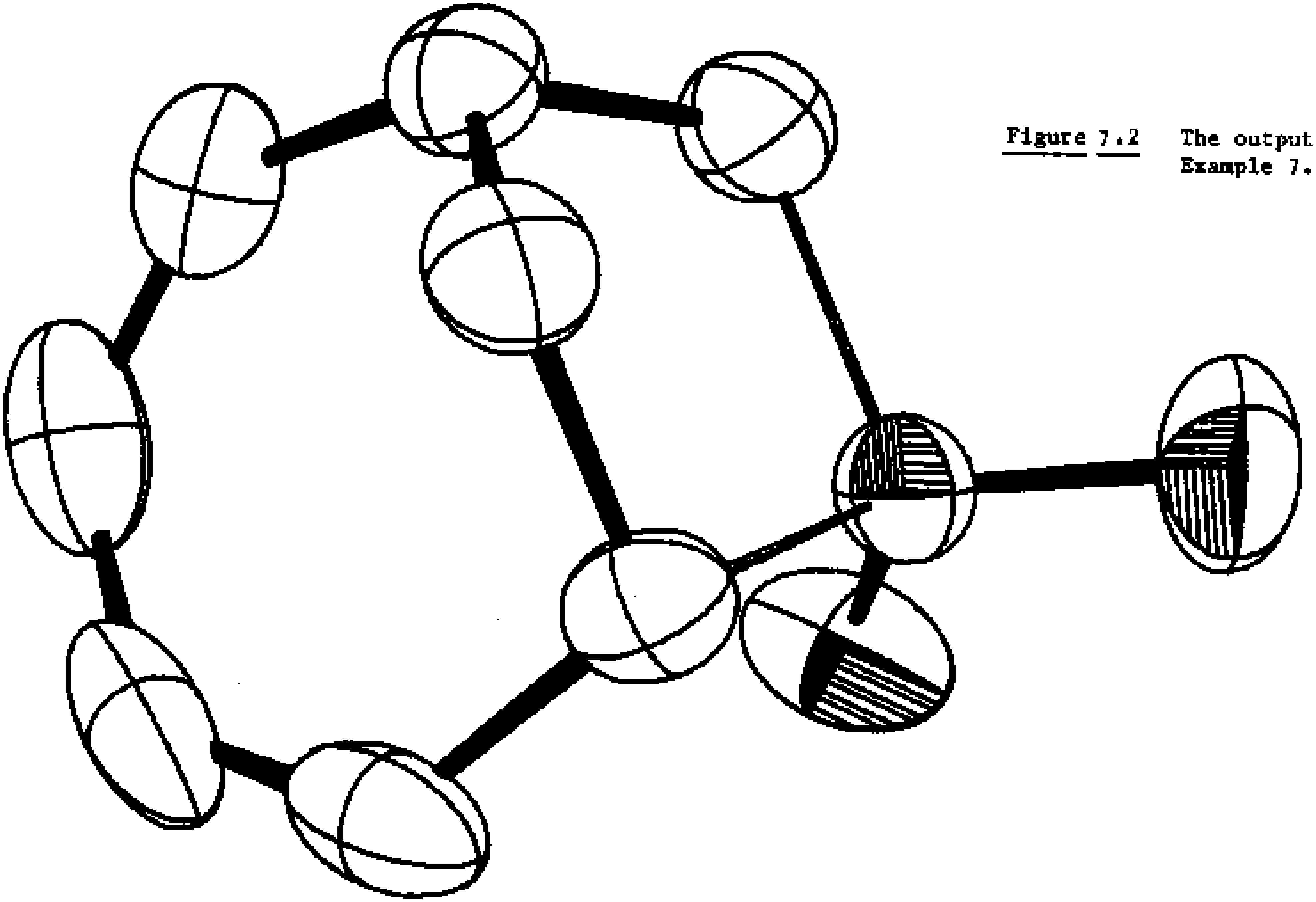


Figure 7.2 The output from Example 7.5.2.

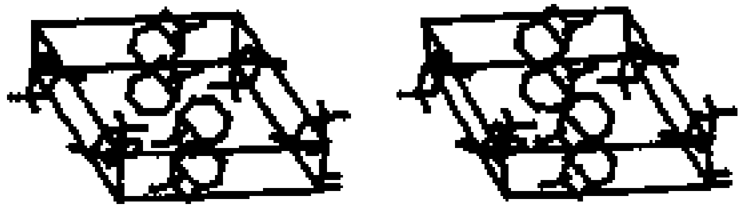


Figure 7.3 The ouput from Example 7.5.4.

8. INSTALLING, MODIFYING AND EXTENDING THE GX PACKAGE

8.1 The subroutine library.

A collection of subroutines is provided for use by most of the programs in the package, and by programs you may wish to add. It may conveniently be compiled into a library for access by your machine's linker program. The library contains routines for reading the crystal model file, general-purpose free-format input routines, and routines for basic numerical and matrix operations. Optionally, you may also wish to use the Tektronix-compatible graphics driver routines (the 'XY' series - see section 8.1.3).

8.1.1 Model file input/output routines. - All programs use these routines for reading and writing the common crystal model file of the package, by calling MDLIN(lfc) for reading or MDLOUT(lfc) for writing. MDLIN sets up two COMMON blocks named MODELC and MODEL, defined as follows:

```
       CHARACTER COMPID(72)
       CHARACTER*8 LABEL(200),FTYPE(13)
       INTEGER CN(200),CONTNT(13)
       COMMON/MODELC/COMPID,LABEL,FTYPE
       COMMON/MODEL/NATOM,NR,NT,NTYPE,NFTYPE(200),P(3,200),
      1CN,SOF(200),UIJ(6,200),EP(3,200),ESOF(200),EUIJ(6,200),
      2WAVEL,CELL(6),ECELL(6),R(24,3,4),T(3,4),ICENT,
      3SFAC(14,13),CONTNT,SCALE(4),ETA,NQ,EXTNCT
```

COMPID(72) - array of 1-byte title characters.

LABEL(200) - array of 8-byte atom labels.

FTYPE(13) - array of up to 13 8-byte scattering factor labels.

NATOM - number of atoms in the asymmetric unit. If the model file contains no ATOM entries, NATOM has the default value 0.

NR - number of equivalent positions, excluding those generated by lattice centring or centre of symmetry. If there are no SYMM entries in the model file, NR has the default value 1.

NT - number of lattice points (including origin). Default value 1, if there is no LATT entry.

NTYPE - number of atomic scattering factor types. Default 0 if no SFAC or ATOM entries.

NFTYPE(200) - array of pointers to the atomic scattering factors, e.g. atoms of the same type as the third SFAC entry have NFTYPE(I)=3.

P(3,200) - array of atomic fractional coordinates.

CN(200) - INTEGER array of atomic coordination numbers.

SOF(200) - array of atomic site occupation factors. Default 1.0 if no entry on ATOM entry.

UIJ(6,200) - array of atomic temperature factors. Isotropic atoms have one non-zero and five zero values. Default isotropic temperature factors of 0.05 Å^2 if no entries on ATOM entry.

EP(3,200) - array of atomic coordinate e.s.d. Default 0.0 if no ATOMSD entry.

ESOF(200) - array of site occupation factor e.s.d. Default 0.0 if none on ATOMSD entry.

EUIJ(6,200) - array of temperature factor e.s.d. Default 0.0 if none on ATOMSD entry.

WAVEL - X-ray wavelength in Angstroms. Default 0.0.

CELL(6) - unit cell constants in Angstroms and cosines. Default 90 degree angles if none on CELL entry. (The CELL entry on file may have angles as either degrees or cosines). If there is no CELL entry, the axis lengths are taken to be 1 Angstrom and the angles 90 degrees.

ECELL(6) - e.s.d. of the cell constants. Default 0.0 if no CELLSD entry.

R(24,3,4) - arrays of rotation matrices and translation vectors for up to 24 equivalent positions, excluding those generated by lattice centring and centre of symmetry. Default identity operator if no SYMM entries.

T(3,4) - coordinates of up to 4 lattice points, including origin.

Default one lattice point at origin if no LATT entry.

ICENT - 0 if centrosymmetric, else 1. Default 0 if no LATT entry.

SFAC(14,13) - array of scattering factor coefficients, absorption coefficients, covalent and van der Waals radii.

CONTNT(13) - INTEGER array of number of atoms of each type in unit cell. Default 0 if no CONTENTS entry.

SCALE(4) - array of batch scale factors applied to F_c. Default 1.0 if no SCALE entry.

ETA - coefficient of f''. Default 1.0 if no ETA entry.

NQ - number of batch scale factors. Default 1 if no SCALE entry.

EXTNCT - isotropic extinction coefficient. Default 0.0 if no EXTINCTN entry.

8.1.2 Free-format input routines. - The input routines used by MDLIN may be called independently, e.g. for decoding program commands. The library includes a set of four routines having the following specifications:

```
      SUBROUTINE PARSE(LINE,J,NPOS,LENGTH)
      CHARACTER LINE(148)
      DIMENSION NPOS(20),LENGTH(20)
```

PARSE scans the input array LINE (the first element of which is not used, so the actual input runs from the second element), and returns in J the number of fields, in array NPOS the starting element of each field, and in array LENGTH the length of each field.

```
      SUBROUTINE AFORMT(LINE,ISTART,LENGTH,STRING)
      CHARACTER LINE(148)
      CHARACTER*8 STRING
```

AFORMT copies into STRING the character contents of the field beginning at LINE(ISTART). The number of characters to be transferred is LENGTH. If LENGTH < 8, STRING is padded to the right with blanks.

```
      SUBROUTINE FFORMT(LINE,ISTART,LENGTH,VALUE)
      CHARACTER LINE(148)
```

FFORMT converts the contents of the field beginning at LINE(ISTART) to binary floating point format and returns the value in VALUE. LENGTH is the field width returned by PARSE.

```
      SUBROUTINE IFORMT(LINE,ISTART,LENGTH,IVALUE)
      CHARACTER LINE(148)
```

IFORMT converts the contents of the field beginning at LINE(ISTART) to binary integer format and returns the value in IVALUE. LENGTH is the field width returned by PARSE.

Example. Subroutine MDLIN identifies the type of an entry in the model file by the following coding:

```
      CHARACTER LINE(148)
      CHARACTER*8 TAG
      INTEGER FIRSTCH(20),FWIDTH(20)
 1    READ (LFC,100) (LINE(I),I=2,73)
 100  FORMAT (72A1)
      CALL PARSE(LINE,NFIELDS,FIRSTCH,FWIDTH)
      IF (NFIELDS.EQ.0) GO TO 1
C**** EXTRACT THE FIRST FIELD
      CALL AFORMT(LINE,FIRSTCH(1),FWIDTH(1),TAG)
      IF (TAG.EQ.'REMARK') GO TO 1
      etc.
```

8.1.3 Low-level interface routines for Tektronix 4010-type devices. The names of these all start with the letters XY; they are called by the optional subroutines CRT, PLOTS, PLOT, SYMBOL, NUMBER and NEWPEN in ORTEP (see section 8.2.3). Machine-specific run-time library routines are required by the 'XY' routines; these must be provided at the user's installation, and are specified below for implementation purposes. If multi-user concurrent access to an on-line plotter is not required, M:ALOC3 and M:DALOC may be dummy routines.

```
      SUBROUTINE BUFFEROUT (U,M,A,W)
      INTEGER U,M,W
      DIMENSION A(3)
U - logical unit number.
```

M - mode of operation. Always used in binary mode (M=1), in which no control information is prefixed to the data, and the data are not broken up into physical records (i.e. there are no embedded carriage returns or line feeds).

A - the name of a buffer containing character data to be output on unit U.

W - the number of words to be output (each containing four characters).

```
      SUBROUTINE WAIT (J,K,M)
```

J - time interval, in units as specified by K, before returning to the calling routine.

K - specifies units of time, here always milliseconds (K=1).

M - return status flag; M=1 indicates successful completion.

```
      SUBROUTINE M:WAIT (LFC)
```

LFC - logical unit number (file code) on which to wait for completion of asynchronous output before continuing execution.

```
      SUBROUTINE M:ALOC3 (LFC,DT,CHANNEL,PARAM,LAB,MO,VOL,
                          BLOCKED,WAIT,ERRSTAT,DA)
      INTEGER LFC,DT,CHANNEL,PARAM,MO,VOL,ERRSTAT,DA
      LOGICAL BLOCKED,WAIT
```

LFC - logical unit number to which an assignment is made at the time of calling this routine.

DT - specifies, together with CHANNEL and DA, the physical device (in this context the hard-copy device), which is assigned to LFC.

CHANNEL - see DT.

PARAM - not used.

LAB - statement label to which control is returned if assignment fails because device is unavailable.

MO - not used.

VOL - not used.

BLOCKED - has the value .FALSE. so that the device is assigned unblocked.

WAIT - has the value .FALSE. so that return from the call does not wait for successful assignment.

ERRSTAT - return status code. Not used, but could be used instead of LAB to indicate assignment unsuccessful.

DA - see DT.

```
      SUBROUTINE M:DALOC(LFC)
LFC -     logical unit number to de-allocate from its currently
          assigned physical device.
```

Some plotters, e.g. Tektronix 4662, have an extension socket that allows the VDU to be 'looped through' so as to share the plotter's computer port. The VDU and plotter are then selected by means of 'plotter on' and 'plotter off' signals in the data. When the plotter is logically on, no data are passed through to the VDU. This arrangement could easily be supported by adding an XY library routine to send the 'plotter on' and 'plotter off' signals, and calling the routine at appropriate times. In ORTEP, the plotter on/off routine would be called in subroutine PLOTS. In another possible arrangement the plotter is connected to the extension socket of the VDU, and manually enabled. In this case 'plotter on' will generally not be required (one exception being a Tektronix 4662 in non-copy mode). Careful study of the plotter manual should indicate the necessary software functions.

8.2 Compiling and linking.

The coding of all programs (except MITHRIL, see below) is in the current ANSI standard language, Fortran 77. RBLS and GEOM contain optional code relevant to the use of extended memory on Gould-S.E.L. computers; this code appears in the form of comments in distributed versions of those programs. The subroutine library described in section 8.1 will be needed by your machine's linker program to produce loadable programs. Therefore, prior to compilation of the programs the subroutine library source file should be compiled into an object module library file that can be searched as required by your local linker. The direct methods program MITHRIL is a completely self-contained program written in Fortran 66, therefore these remarks do not apply to it.

Some Fortran 77 run-time environments require array bound checking to be explicitly suppressed by calling an installation-specific routine for that purpose. The need for this may arise in the Fourier program FFT.

8.2.1 Setting the physical constants of ORTEP's graphical devices. - For the purposes of positioning and scaling, the program needs to know the dimensions of the screen display and the plotter. These are set in the main program under labels (not line numbers) 4036 and 4090 respectively. Variables AIN(1) and

AIN(2) are the x and y dimensions in inches. AIN(4) is the required border width in inches. The border must be wide enough to accomodate the ellipsoid circumferences and/or labels of the atoms, since scaling is performed with respect to the atom centres and the inside edge of the border. Ellipsoids with straight-line clips occur when part of the circumference falls outside the drawing boundary.

Unless you are using locally-available CALCOMP software (see sections 8.2.2 and 8.2.3), the program also needs to know the resolution or incremental step size of the graphical devices. This is held in the variable called SCALE defined in subroutines PLOTS and CRT. The value is given as the number of addressable points per inch.

8.2.2 Using ORTEP with CALCOMP software . - Interactive ORTEP is designed for use either with a conventional CALCOMP-type plotting system or with the addition of a graphics display 'on-line', i.e. accessible while the program is running. In the former case, subroutine CRT near the end of the source file should be substituted by a dummy subroutine CRT. Subroutines PLOTS, PLOT, SYMBOL, NUMBER and NEWPEN should be removed from the source file. The program can then be compiled and linked with the subroutines available in your local CALCOMP software, and output channels 7 and 8 (section 8.4.10) will not be used. Note that the ORTEP command PLOTTER must always be given to initialise the plotter, since the default output device is a graphics display. When using a plotter with continuous stationery the NEWPLOT command (with a suitable origin parameter) must be used to advance the paper between successive plots in the same run, and the STEREO command must have a suitable separation parameter. The logic of the STEREO command must be slightly modified when using standard CALCOMP routines, to enable correct frame advance between the two views. This requires the removal of two lines of code about 195 lines from the beginning of the main program, indicated by comments in the source immediately preceding statement label 5090.

The specifications of the CALCOMP routines are as follow.

SUBROUTINE PLOTS(I)

Initialises the plotter and sets the coordinate units to inches (I=1).

SUBROUTINE PLOT(X,Y,IC)

Moves the pen from its current position to a position with coordinates (X,Y) in inches. |IC|=1 signifies no change in pen mode since the last call of plot; |IC|=2 signifies a move with pen down, |IC|=3 a move with pen up. |IC|=999 terminates the current run. If IC>0, the origin is unchanged; if IC<0 a new origin is established at (X,Y).

SUBROUTINE SYMBOL(X,Y,SIZE,IBCD,THETA,N)

Draws N characters stored in integer array IBCD and of height SIZE at an angle THETA degrees to the X-axis, starting at a point (X,Y).

SUBROUTINE NUMBER(X,Y,SIZE,FPN,THETA,N)

Writes the real variable FPN in character form starting at location (X,Y) at an angle THETA degrees to the X-axis, using digits of height SIZE. N is the number of digits to the right of the decimal point, after rounding. ORTEP uses this subroutine only for the annotation of bonds with their lengths.

SUBROUTINE NEWPEN(I)

Changes to pen number I.

8.2.3 Using ORTEP with a graphics display. - The system is distributed with a Fortran source file of graphical driver routines suitable for Tektronix 4010-compatible displays and Tektronix 4662-compatible plotters. These require some machine-specific routines whose specifications are given in section 8.1.3. If you wish to use a Tektronix-compatible display and plotter 'on-line' (and there is no locally-provided means of doing so), compile these driver routines into the system library and compile ORTEP as distributed, i.e. including subroutines CRT, PLOTS, PLOT, SYMBOL, NUMBER and NEWPEN. Channels 7 and 8 will then carry the graphical output (in the form of ASCII characters) for the display and plotter respectively.

8.2.4 Changing the dimension of the normal matrix. - There is a large array (AM in RBLS, PM in GEOM) used for storing the least-squares normal matrix in RBLS and the corresponding correlation matrix in GEOM. You may wish to reduce the overall memory requirement of these programs by decreasing the dimension of the array. Its size is given by N(N+1)/2, where N is the maximum allowed number of refined parameters; in the distributed versions of the programs the dimension is 61425 (=350x351/2). In RBLS there is a trap on the number of refineable parameters being exceeded (label 21 in subroutine RBLSA), so it would be desirable to modify the constant in the trap to correspond to the new dimension of AM. In GEOM, subroutine TORSN contains an array called DUMMY whose dimension must be set to N(N+1)/2-30141.

8.3 The reflection and F_c files.

These are packed binary files giving optimum disc space utilisation. Each record contains one reflection. The record contents for the reflection file are:

word 1 - packed indices and batch number (NSET), viz. NSETx256^3 + (h + 128)x256^2 + (k + 128)x256 + (l + 128)

word 2 - packed path length (TBAR cm) and reflection sequence number (NREFL), viz. IFIX(TBARx10^4)x2^{16} + NREFL

word 3 - unscaled F_o^2

word 4 - $\sigma(F_o^2)$

words 5 to 10 - direction cosines l_1, l_2, m_1, m_2, n_1, n_2 where (l_1 m_1 n_1), (l_2 m_2 n_2) are the direction cosines of the reversed incident and diffracted rays respectively on the reciprocal axes. These are required only by the Gaussian absorbtion correction program ABSORB.

On the GX distribution tape, a small Fortran program is provided for creating the binary reflection file from the user's input. The program can be used by itself or incorporated into your local data reduction program.

The record contents for the F_c file are:

words 1 to 4 - as above, with F_o^2 on absolute scale and $\sigma(F_o^2)$ = $2F_o/w^{1/2}$, corresponding to the weights used in refinement.

word 5 - A_c

word 6 - B_c

word 7 - $\sin^2\theta/\lambda^2$

8.4 Assigning logical unit numbers (logical file codes) and running the programs.

Most operating systems provide a job or task control macro facility. A convenient arrangement is to have a macro called, perhaps, GX to run the package. The macro will contain all the installation-specific operating system commands required for creating and assigning files, allocating devices to the appropriate logical channels, and for loading and running the programs in either batch or interactive mode. It is a good idea to adopt a file-naming convention so that your GX macro can automatically create files. A suggested scheme is the following, in which 'name' is a name to identify the compound to which the files refer:

name.MA input model file (see section 1).

name.MB output model file.

name.IN instruction file, containing program commands. If this file exists, programs optionally read commands from it rather than expecting real-time command input from the keyboard. The .IN file may hold instructions for more than one program, since each will select those relevant to itself.

name.FO binary reflection file (see section 8.3).

name.FC binary structure factor file (see section 8.3).

name.TB GEOM tables file (see section 6.2).

name.MX correlation matrix file (binary), created by RBLS and optionally used by GEOM.

name.OC orthogonal coordinates file, created by ORTEP. It enables ORTEP's figure-subject list of atoms to be passed to other programs, e.g. van der Waals surface or space-filling model.

name.TF temporary file (normally deleted automatically).

name.BK backup file created when a file is re-named, e.g. the command

```
GX name RENAME MB MA
```

re-names name.MB to become name.MA and puts the old version of name.MA (if there was one) in name.BK. After running any program that writes a new model file, the .MB file has to become the .MA file for input to the next program. For the Patterson option (FMAP FSQ) in FFT, re-name .FO to .FC.

The following sections define the logical channels for each program in the system.

8.4.1 SORT

1 - name.MA

2 - name.FO

3 - a temporary (scratch) file.

4 - name.TF, the output binary reflection file (sorted and intensity-merged) to be renamed name.FO

5 - dialogue input; assigned to the keyboard.

6 - dialogue output; assigned to the VDU.

7 - merging-statistics listing; assigned to a line printer or to the VDU.

10 - the system databank file (see section 1.2).

8.4.2 XYZ

1 - name.MA

2 - name.MB

5 - dialogue input; assigned to the keyboard.

6 - dialogue output; assigned to the VDU.

7 - results listing; assigned to a line printer or to the VDU.

10 - the system databank file (see section 1.2).

8.4.3 ABSORB

1 - name.MA

2 - name.FO

4 - name.TF, the output binary reflection file (absorption-corrected) to be renamed name.FO

5 - name.IN, or the keyboard

8 - results listing, assigned to the printer or VDU

10 - the system databank file (see section 1.2).

8.4.4 DIFABS

1 - name.MA

2 - name.FC

3 - name.TF, the output binary reflection file (absorption-corrected) to be renamed name.FO

4 - a temporary (scratch) file

5 - dialogue input, assigned to the keyboard

6 - results listing on printer or VDU

10 - the system databank file (see section 1.2).

8.4.5 CAD4

1 - name.FO

2 - message output and menu display

3 - existing intensity standard file, for use with SKIP command

4 - CAD4 diffractometer output reflection file

5 - command input, usually keyboard

6 - results listing (printer or VDU)

7 - output intensity standard file

8 - existing name.FO file, for use with SKIP command

8.4.6 FFT

5 - name.IN, or, usually, the keyboard.

6 - message output and menu display; normally assigned to the VDU.

7 - results listing; normally assigned to a line printer.

8 - a temporary (scratch) file.

9 - name.MA

10 - the system databank file (see section 1.2)

11 - name.FC (re-named from name.FO for a Patterson map)

13 - name.TF, a binary file for output of the map, used for input to SEARCH.

8.4.7 SEARCH. - The GX macro should be set up to run SEARCH immediately after FFT, the map (.TF) file being passed from FFT to SEARCH and finally deleted.

1 - name.MA

5 - name.IN, or, usually, the keyboard.

6 - message output and menu display; normally assigned to the VDU.

7 - results listing, normally assigned to the line printer.

10 - the system databank file (see section 1.2).

13 - name.TF (input binary map file from FFT).

14 - name.MB (output file for peak coordinates).

8.4.8 BLOCK

2 - name.MA

3 - results listing; assigned to a line printer or to the VDU.

4 - name.MB

5 - name.IN (assigned to the keyboard when running interactively, or to a command file).

6 - message output and menu display; normally assigned to the VDU.

10 - the system databank file (see section 1.2).

17 - name.FC

50 - name.FO

8.4.9 RBLS

2 - name.IN (assigned to the keyboard when running interactively, or to a command file).

3 - results listing; assigned to a line printer or to the VDU.

4 - name.MB

6 - message output and menu display; normally assigned to the VDU.

8 - name.MA

10 - the system databank file (see section 1.2)

16 - name.FO

17 - name.FC

20 - name.MX

8.4.10 WTANAL

3 - results listing; assigned to a line printer or to the VDU.

5 - dialogue input; assigned to the keyboard.

6 - dialogue output; assigned to the VDU.

17 - name.FC from RBLS or BLOCK.

8.4.11 FTAB

2 - dialogue input; assigned to the keyboard.

3 - results listing; assigned to a line printer or to the VDU.

6 - dialogue output; assigned to the VDU.

17 - name.FC from RBLS or BLOCK.

18 - a temporary (scratch) file.

8.4.12 GEOM

1 - name.IN (assigned to the keyboard when running interactively, or to a command file).

2 - name.TB, tables file for editing and printing

3 - printed output of results; assigned to a line printer.

6 - message output, menu display, and screen output of results; normally assigned to the VDU.

10 - the system databank file (see section 1.2)

16 - name.MA

20 - name.MX from RBLS. Not required unless CORRELATIONS GIVEN command is used.

8.4.13 ORTEP

1 - name.OC

3 - a temporary (scratch) file used to store saved sequences; this could be a permanent file if you wish to keep a saved sequence for future sessions.

5 - name.IN (assigned to the VDU when running interactively).

6 - message output and menu display; normally assigned to the VDU.

7 - graphical output to the VDU.

8 - graphical output to the plotter. Channel 8 may be assigned dynamically (see Section 8.1.3), to allow sharing of an on-line plotter between concurrent users of ORTEP in a multi-user environment. In 'daisy-chained' configurations (plotter connected to the VDU extension socket or vice-versa), channel 8 should be assigned to the VDU.

9 - print-out of commands executed; may be assigned to a line printer, but usually to the null device.

10 - the system databank file (see section 1.2).

16 - name.MA

9. REFERENCES

COPPENS, P. (1970). In Crystallographic Computing, edited by F.R. Ahmed, p.255. Copenhagen: Munksgaard.

CRUICKSHANK, D.W.J. (1970). In Crystallographic Computing, edited by F.R. Ahmed, p. 187. Copenhagen: Munksgaard.

CRUICKSHANK, D.W.J. and McDONALD, W.S. (1967). Acta Cryst. 23, 9.

DARLOW, S.F. (1960). Acta Cryst. 13, 683.

DOEDENS, R.J. (1970). In Crystallographic Computing, edited by F.R. Ahmed, p.198. Copenhagen: Munksgaard.

DUNITZ, J.D. and SEILER, P. (1973). Acta Cryst. B29, 589.

FORDER, R.A. and SHELDRICK, G.M. (1971). J. Chem. Soc. (A) 1971, 1107.

GILMORE, C.J. (1984). J. Appl. Cryst. 17, 42.

HUBER-BUSER, E. and DUNITZ, J.D. (1961). Helv. Chim. Act. 44, 2027.

HURSTHOUSE, M.B. and DAVIES, J. (1978). Personal communication.

International Tables for X-ray Crystallography (1974). Vol. IV, Tables 2.1B, 2.2B, and 2.3.1. Birmingham: Kynoch Press.

JOHNSON, C.K. (1971). ORTEP II. A Fortran Thermal-Ellipsoid Plot Program for Crystal Structure Illustrations. Report ORNL-3794 (2nd revision) and supplementary instructions. Oak Ridge National Laboratory, Tennessee.

LARSON, A.C. (1970). In Crystallographic Computing, edited by F.R. Ahmed, p.291. Copenhagen: Munksgaard.

MAIN, P., FISKE, S.J., GERMAIN, G., HULL, S.E., DECLERCQ, J.-P., LESSINGER, L. and WOOLFSON, M.M. (1980). MULTAN80. A System of Computer Programs for the Automatic Solution of Crystal Structures from X-ray Diffraction Data. Universities of York, England and Louvain, Belgium.

MALLINSON, P.R. and MUIR, K.W. (1985). *J. Appl. Cryst*. **18**, 51.

ROGERS, D. (1981). *Acta Cryst*. **A37**, 734.

ROLLETT, J.S. (1965). In *Computing Methods in Crystallography*, p.25. Oxford: Pergamon Press.

SHELDRICK, G.M. (1976). SHELX. A Program for Crystal Structure Determination. Institut fur Anorganische Chemie der Universitat Gottingen.

STANFORD, R.H. and WASER, J. (1972). *Acta Cryst*. **A28**, 213.

WALKER, N. and STUART, D. (1983). *Acta Cryst*. **A39**, 158.

CAOS User Manual
(source code deposited on newsletter webpage)

by R. Spagna

Istituto di Cristallografia - CNR, Sede di Monterotondo, Area della Ricerca Roma 1, Via Salaria Km.29, 00016 Monterotondo Stazione (Roma), Italy. E-mail: riccardo.spagna@ic.cnr.it

Place holder page. Report-proper starts on following page

```
  @@@@@@         @@@@@@         @@@@@@         @@@@@@
 @@@@@@@@       @@@@@@@@       @@@@@@@@       @@@@@@@@
@@@    @@@     @@@    @@@     @@@    @@@     @@@    @@@
@@@            @@@    @@@     @@@    @@@      @@@   @@@
@@@            @@@    @@@     @@@    @@@       @@@  @@@
@@@            @@@    @@@     @@@    @@@        @@@ @@@
@@@            @@@@@@@@@@     @@@    @@@     @@@ @@@
@@@            @@@@@@@@@@     @@@    @@@     @@@  @@@
@@@            @@@@@@@@@@     @@@    @@@     @@@   @@@
@@@    @@@     @@@    @@@     @@@    @@@     @@@    @@@
 @@@@@@@@      @@@    @@@      @@@@@@@@       @@@@@@@@
  @@@@@@       @@@    @@@       @@@@@@         @@@@@@
```

CRYSTAL ANALYSIS OPERATING SYSTEM

*** C. A. O. S. ***

(vers 2009)

USER MANUAL

by

Camalli M., Capitani D., Cerrini S., Spagna R.
Istituto di Cristallografia - CNR.
Sezione di Roma
Area della Ricerca di Roma
C.P. 10
00016 Monterotondo Stazione
(Roma) Italy

Cascarano G., Giacovazzo C.
Istituto di Cristallografia – CNR
Sede di Bari
Campus Universitario
Via Orabona 4
70125 Bari
Italy

Italian Patent No. 35403c/86. CAOS user manual, Istituto di Strutturistica Chimica CNR.

Abstract

CAOS is a crystallographic package for crystal structure determination from single crystal diffraction data. The first version of the CAOS program was written at the early years of the 70's. (Cerrini S. & Spagna R. (1977) *Crystallographic software for a mincomputer,* IV Eur. Crystallgr.Meet., Oxford, UK, Abstract A-212). At the present, CAOS covers most of the crystallographic calculation: data reduction, intensity statistics, Fourier and Patterson synthesis, automatic Patterson solution, peaks interpretation, structure factor and least squares calculation with user defined blocked matrix, geometrical calculation, molecular display, publication tables, CIF files. Space group restrictions are handled automatically and it is possible introduce restraints in the refinement and many weighting schemes are available. Furthermore, hydrogen atoms can be calculated from model.

Please, any publication arising from use of CAOS should include the reference:

Camalli M. & Spagna R. (1994) J.Appl. Cryst. 27, 861-862.

Some general information

The input necessary for all jobs consists of a sequence of commands, data and directives which determine the course that the calculation should take. The commands are headed by '#' or '%' character, directives and data follow the related command.

CAOS recognises the following commands:

#INITIALIZE		Initialize the direct access file
#FILE	binary_file	give the file name of the direct access binary file
#BACKUP		backup the binary file (create binary_file.SAV) writing the last current lists stored
#STRUCTURE	binary_file	like #FILE
#TITLE	title	give the title of the structure
#JOB	title	like #TITLE
#INPUT	input_file	give the input file name (if no file name is given, the input is the keyboard)
#OUTPUT	output_file	give the output file name (if no file name is given, the output is on the screen)
#CLOSE	output_file	close the output file and return on the screen
#READ	list_no	read the data of the list_no list
#LIST	list_no	like #READ
#PRINT	list_no	print the data of the list_no list
#UPDATE	list_no	update the list_no list
#DUMP	list_no	dump on requested file the list_no list
#PUBLICATION	list_no	print the list_no list as tables for publication
#DATA		read reflns and obtain set of unique reflections put on Wilson scale
#EXTINCTIONS		analysis of the extinctions of the reflections
#SAVE		save list 5 and list 12 as saved lists
#RESTORE		restore the saved lists 5 and 12
#SORTING		sort the reflections by h,k,l
#MERGE		merge equivalent reflections
#NORMAL		calculate Wilson plot
#FOURIER		calculate a Fourier map
#PATTERSON		calculate a Patterson map
#SEARCH		calculate connectivity from list of atoms (list 5) and produce a list 35
#LSQ		Least Squares routine
#AGREEMENT		analysis of Fo Fc
#DISTANCE		calculate distances
#ANGLE		calculate angles
#SHORT		calculate short distances
#LONG		calculate long distances

#HYDROGEN	calculate geometrical positions of H atoms
#GEOMETRY	geometrical calculation routine
#PLOT	general command to draw the structure
#SKETCH	simple command to draw a sketch of the structure
#BALL	simple command do draw the structure
#ELLIPSOIDS	simple command to draw the structure
#INFO	print information about the structure data
#STATISTICS	print information about the reflections
#CIF	produce a file CIF
#END	End of the program

All commands and directives are in free format (between columns 1-80) and are case independent. Only the first four characters are significant. If the first character following the header character '#' is a blank character, then the record is interpreted as a comment.

CAOS uses a direct access binary file to store the data of the crystal structure and the name of this file is the first information requested by the program. We remind that the binary file must be indicated without the extension .BIN. The program inserts (or replaces) the extension

The input_file is a text file containing directives and data (created previously with an editor)

The output_file is file containing the results.

Windows platform: at the beginning, the program opens a window asking:

1) the working directory where are stored the files of the structure under examination,
2) the name of the binary_file;
3) the names of input_file ;
4) the name of output_file.

In the last two questions, if no file names are given, the interactive mode is selected

Input/output of lists

The data required to solve a crystal structure must be input to the computer, translated into an internal format and stored on the binary file.

On the disk, different types of data are recorded separately, in what they are called 'LISTS', each list holds only one type of crystallographic information and it is identified by a number, called 'LIST TYPE NUMBER', and by a alpha-numeric string (the first four characters are significant), called IDENTIFYING STRING

LIST TYPE NUMBER	IDENTIFYING STRING	TYPE OF DATA
1	CELL	cell parameters
2	SYMMETRY	space group symmetry
3	FORM	form scattering factors and cell content
4	WEIGHTING	weighting parameters
5	ATOMS	atomic parameters
6	REFLECTIONS	reflections
12	REFINEMENT	refinement instructions
14	LIMITS	fourier directives
15	PEAK	peaks localised from the map
18	CONDITIONS	min & max values of the data
35		peaks interpreted by the program.

Commands to control the lists are:

```
#READ   (list number or string)
#LIST             "              (equivalent to READ)
#PRINT            "
#DUMP             "              (valid only for 5,6,12,15)
#UPDATE           "              (valid only for 1,5,6,12)
#PUBLIC           "              (valid only for 5,6)
```

Input/output of all the lists

To read all the experimental data, the following command can be used:

#READ FILE [keyword]
The allowed keywords are:

CAOS (default)
The input data are:

a b c α β γ	(see List 1)
σ(a) σ(b) σ(c) $\sigma(\alpha)$ $\sigma(\beta)$ $\sigma(\gamma)$ σ(Vol)	
string	(symbol of the space group, see List 2)
cell content	(see List 3)
h k l Fo σ(Fo) Fc Phase	(see List 6, DATA directive)

SHELX
the program prompts the request of the first component of the filename for the ShelX files:
structure_name.ins & structure_name.hkl
At the present, the routine is unable to resolve the symmetry cards and they were jumped. The user have to give this information before *via* CAOS directive:
#read symm [symbol of the S.G.] (see list 2)

SIR
The program asks for the SIR file_name.sir and it will read the list 1, 2 & 3 from this file and the reflections from file defined in the %reflections directive (the directive %format is not used).

To create a file with all the data, the following command ca be used:

```
#DUMP   FILE     [keyword]
```

The program asks for the file name to dump the data. The allowed keywords are:

CAOS (default)
SHELX
SIR
PARST Useful to run the program PARST by M. Nardelli (J.Appl.Cryst.(1995). 28,659)

Examples:

```
…
#init
#title  AP4  (STRUCTURE TEST)
#read file
8.278 8.856 12.814 103.89 99.12 105.62
p -1
hg 2 s 2 b 2 n 16 c 26 h 30   end
   1   0  -7       22.58745        0.18191
   1   0  -7       22.47132        0.18288
…

#initialize
#read  2  p -1
#read file shelx
name_of_structure
…

#read  file  sir
name.sir

….
```

Cell parameters   List 1

```
#READ  CELL
a   b   c     α  β  γ
σ(a)  σ(b)  σ(c)  σ(α)  σ(β)  σ(γ)  σ(Vol)
```

cell dimensions a, b, and c are in angstroms, α β and γ in degrees, and their standard uncertainty and the standard uncertainty of the volume.

```
#read cell
 10.818 9.062 11.822 90 91.87 90
  0.002  0.002  0.003   0.05  0.05  0.05   0.1
```

Space group symmetry   List 2

```
#READ  SYMMETRY    string
```

String is the symbol of the space group, according to International Tables (1974). Blanks are necessary among the terms constituting the space group symbol .

#read symm P 21 21 21

#read 2 P 21/c

Form scattering factors and Cell content List 3

#READ FORM
Species₁ n₁ Species₂ n₂ Species₃ n₃ ....
Keywords

Species$_i$ is the chemical symbol of atomic type *i*, n_i is the corresponding number of atoms in the unit cell (to a maximum of 8 atomic types). For each chemical element up to Cf (Z=98) X-ray and electrons scattering factor constants are stored, together with information on the atomic number and weight, covalent and Van-der-Waals radii., etc. in the file CaosWare.xen.
The scattering factors are:

$\mathbf{f}_r = \sum a_i \; e^{-b_i \sin(\theta/\lambda)^2} + c + \mathbf{f}'$ (real component)

$\mathbf{f}_i = \mathbf{f}''$ (imaginary component)

The structure of the file CaosWare.xen is:

```
(Header record)   |06.96   ---- scattering factors ----
(no. of records)  |    9
                  |#  Atomic  Weight   coval_radius   vanderWaals_radius
(x_ray)           |#      a(1)   b(1)      a(2)   b(2)   a(3)   b(3)
                  |#           a(4)   b(4)      c
                  |#         f'(Mo)  f"(Mo)   mu/rho
                  |#         f'(Cu)   f"(Cu)   mu/rho
(electr.)         |#      a(1)   b(1)      a(2)   b(2)   a(3)   b(3)
                  |#           a(4)   b(4)      c
(neutrons)        |#      fact
                  |#   Name_specie
                  |h    1   1.008  .320        1.33
                  |h        .48992  20.65930   .26200   7.74039   .19677  49.55190
                  |h        .04988   2.20159   .00130
                  |h        .00000    .00000   .38000
                  |h        .00000    .00000   .43500
                  |h        .37540  15.49460   .14080   4.12610   .02160   .02460
                  |h       -.10120  46.88400   .00000
                  |h      -0.37200
                  |h  Hydrogen
                  |etc...
```

Accepted keywords:

<u>FILE</u> file name containing the information of the atomic species (default = CaosWare.xen)

<u>WAVE</u> (value or code) value of the λ or symbol (Mo and Cu)

<u>DIFFRACTION</u> (code) code indicate the type of data:
x-ray

electron
neutron

ANOMALOUS species f' f" — give the anomalous dispersion factors for the atomic species if different from what is stored in the CaosWare.xen file.

SFACTOR specie a(1) b(1) a(2) b(2) a(3) b(3) a(4) b(4) c f' f" mu/rho r wt
(like SHELX instruction)

UNIT n1 n2 n3 … — number of atoms of each type in the unit cell, in the directive SFACTOR order, (like SHELX instruction)

END — ends the command.

```
#read form
hg 1 p 2 c 15 h 22
end
```

```
#read  3
o 1  c 21   h 24
wave  cu
end
```

Weighting schemeList 4

#READ WEIGHTING
type npar par1 par2 ...

type — number of the weighting scheme requested (1 - 18)
npar — number of parameters (drefaulted)
par1 par2 ... — npar parameters of the weighting scheme (defaulted)

The available weighting schemes are the following :

1 w = Fo/p(1) if Fo ≤ p(1)
w = p(1)/Fo if Fo ≥ p(1)
default ; p(1)=4*Fo(min)

2 w = 1/(p(1)+p(2)*Fo+p(3)*Fo**2)
default: p(1)= 2*Fo(min); p(2)=1.0; p(3)=2./Fo(max)

3 w = (sin(ϑ)/λ)**p(1)
default: p(1)=1.0

4 w = (λ/sin(ϑ))**p(1)
default: p(1)=1.0

5 w = p(1)/(σ(Fo))**2

6 w = p(1)
default ; p(1)=1.0

7 w = 1/(1.+((Fo-p(2))/p(1))**2)
default: p(1)= Fo(max); p(2)=Fo(min)

8 w = 1.0 if Fo ≤ p(1)
w = p(1)/Fo if Fo ≥ p(1)
default ; p(1)=4*Fo(min)

9 w = p(1)*σ(Fo)

10 w = p(1)/σ(Fo)

11 w = (sin(ϑ)/λ)**p(4) / (p(1)+p(2)*Fo+p(3)*Fo**2)
default: p(4)=1.0; p(1)= 2*Fo(min); p(2)=1.0; p(3)=2./Fo(max)

12 w = 1./(p(1)*σ(Fo)**2+p(2)*P+p(3)*P**2)
where P=(Fo+2Fc)/3 (shelx scheme)
default: p(1)=1.0; p(2)=2.*P(min); p(3)=2./P(max)

13 w = 1/ (p(1)+p(2)*Fo+p(3)*Fo**2+p(4)*(sin(ϑ)/λ))
default: p(1)= 2*Fo(min); p(2)=1.0; p(3)=2./Fo(max); p(4)=1.0

14 w = 1/(p(1)+p(2)*Fo+p(3)*Fo**2)
default: p(1)= 2*Fo(min); p(2)=1.0; p(3)=2./Fo(max)
as scheme 2, but #WEIGHT uses different algorithm to find the best coefficients

15 w = 1/(p(1)+p(2)*(sin(ϑ)/λ)+p(3)*(sin(ϑ)/λ)**2)
default: p(1)= (sin(ϑ)/λ)(min); p(2)=1.0; p(3)=(sin(ϑ)/λ)(max)

16 w = 1/[$\sum a_i T^*_i$(F)]
no default, the Chebyshev coefficients a_i must be calculated using #WEIGHT

17 w = (sin(ϑ)/λ)**p(3) / (p(1)*σ(Fo)**2+p(2)*Fo**2)
default: p(3)=1.0; p(1)=1.0; p(2)=2./Fo(max)

18 w = 1/((p(4)*(sin(ϑ)/λ)**p(5)+ p(1)+p(2)*Fo+p(3)*Fo**2)*σ(Fo)**2)
default: p(4)=1.0; p(5)=1.0; p(1)= 2*Fo(min); p(2)=1.0; p(3)=2./Fo(max)

The weight is multiplied by a scale factor (SC) to achive:
$\sum w\Delta^2$ /(Nref-nparam) = 1.0
The value of SC is set = 1.0 when the weighting scheme is input.

#read weight
2
(Cruickshank weighting scheme using default values)

Atomic parameters List 5

The atomic parameters are:

Species Serial_number x y z Thermal_factor Occupation

where Species is the chemical symbol of the atom and it must be given in the list 3; Serial_number is a sequence number of that species; the Thermal_factor is B(iso) for an isotropic atom (the exponential factor takes the form exp[–B(sin(ϑ)/λ)^2] and the Thermal_factors are (β_{11} β_{12} β_{13} β_{22} β_{23} β_{33}) for an anisotropic atom (the exponential factor takes the form exp(-$\sum_i \sum_j \beta_{ij} h_i h_j$). (in the test b≡β). The individual atom is identified by the *(name)* (see later) and all the instructions working on the atom list use this *(name)* to select the atoms.

#READ ATOMS

Scale B(overall) Extinction Polarity scale to be applied to Fc; B(overall) is used in the refinement block-diagonal approximation, coeff. of extiction, coeff. of polarity

[*projection* optionally, this keyword gives the matrix to draw the structure.
a11 a12 a13
a21 a22 a23
a31 a32 a33]

The record(s) for each atom are the following:

1. Species serial x y z [b(iso)] [(occ)] [(idev)] [(imix)] [(name)]

if *b(iso)*=0, read a record containing the *b*(i,j):

2. Species b(11) b(12) b(13) b(22) b(23) b(33)

if *idev*=1, read a record containing the standard deviations on the parameters :

3. *Species σ(x) σ(y) σ(z) σ(b(iso)) σ(occ)*

if *b(iso)*=0*,* read the a new record containing the standard deviations on *b*(i,j): :

4. Species σ(b(11)) σ(b(12)) σ(b(13)) σ(b(22)) σ(b(23)) σ(b(33))

if *imix*>0*,* read the a new record containing *Species* & *occ* for *imix* times

5. *Species1 occ1 Species2 occ2 …*

If *b(iso)* is not present, the default value is *B(overall);* default value for *occ* is 1.00; *Idev* and *imix* if not present assume the value 0, the maximum value for *imix* is 4*;* the *name* is a string identifying the atom (max 8 characters), if it is not present, a string will be built *Species(serial)*.

The sequence of atoms can ends with the connectivity :

nbond number of bonds

n1 n2 code for each bond : *n1 and n2* are the sequence number in the list 5 of the atoms; *code* is = 0.

The list 5 ends with : ##

If the connectivity is not present, the program builds it calling the interpretation routine.

Examples:

```
#list 5
   0.80185   0.47029   0.00526   0.00000
project
0.96764  0.18366  -0.17298
-0.19308  0.98039  -0.03921
0.16240  0.07133  0.98415
 cu      1   0.05005   0.01671   0.32102   0.00000   1.00000  1 0  cu(1)
 cu          0.00537   0.00162   0.00839   0.00467  -0.00128   0.00519
 cu          0.00012   0.00012   0.00012   0.00000   0.00000
 cu          0.00010   0.00017   0.00017   0.00011   0.00016   0.00010
 p       1   0.11986  -0.10350   0.27318   0.00000   1.00000  1 0  p(1)
 p           0.00369   0.00010   0.00559   0.00378  -0.00127   0.00378
 p           0.00022   0.00025   0.00022   0.00000   0.00000
 p           0.00017   0.00031   0.00027   0.00021   0.00029   0.00015
…..

#read atoms
 1.035  5.63  0.00  0.00
 o  1   -.99299   -.37159   -.36338    4.33    1.0
....
##
```

The directive can be followed by a keyword:

#READ ATOMS [*keyword*]
the following keywords are allowed:
CAOS (default)

SHELX
CRYSTALS
PDB
U

The keyword U informs the program that the thermal_factor is expressd by U or U_{ij} term(s) instead of B or β_{ij} .

For SHELX, CRYSTALS, PDB keywords, the program prompts the request of the file name where the co-ordinates are.

Example of Shel-X file (file.ins o file.res):

```
TITL                          calix
CELL   .71069    28.181   11.769   15.838   90.000   90.000   90.000
ZERR       4      .005     .005     .005     .005     .005     .005
LATT -1
SYMM -X,-Y,  .50000+Z
SYMM   .50000-X,  .50000+Y,  .50000+Z
SYMM   .50000+X,  .50000-Y,Z
SFAC    C   H   O   CL
UNIT   228 192  32   8
MERG     2
OMIT     4
L.S.     4
LIST     1
FMAP     2
PLAN    25
FVAR    1.0
O1      3   1.09077    .03784    .66737  11.00000    .02816    .06526 =
        .04829   -.00486   -.00418   -.00302
O2      3   1.04664    .16432    .78076  11.00000    .01328    .07291 =
        .05731   -.00699   -.00204   -.00059
O3      3   1.13441    .21943    .60461  11.00000    .03621    .06610 =
…and so on……..
H48     2    .88080    .19495    .17274  11.00000    .14491
H49     2    .75822    .24202    .94141  11.00000    .15600
H50     2    .78244    .12363    .92712  11.00000    .15600

HKLF     4
END
```

Example of PDB file:

```
HEADER                             calix
CRYST1   18.310   26.681   16.809 101.45 110.20  72.86 P -1          2
ORIGX1      1.000000  0.000000  0.000000        0.00000
ORIGX2      0.000000  1.000000  0.000000        0.00000
ORIGX3      0.000000  0.000000  1.000000        0.00000
SCALE1      0.054615 -0.016843  0.018385        0.00000
SCALE2      0.000000  0.039222  0.004256        0.00000
SCALE3      0.000000  0.000000  0.063763        0.00000
HETATM    1  O                -0.805   2.562  12.204  1.00  5.39
HETATM    2  O                -6.527   7.429  14.457  1.00  4.80
HETATM    3  C                 3.435   9.031  14.613  1.00  4.32
HETATM    4  O                -4.108   1.454  14.751  1.00  4.48
HETATM    5  C                -7.018   3.854  13.997  1.00  5.72
.....
```

```
.....
HETATM  327  C              -9.432   6.763  12.808  1.00 40.50
HETATM  328  C              -6.725   6.674  10.124  1.00 18.47
HETATM  329  C               1.316  -3.378  14.447  1.00 26.34
HETATM  330  C              -8.116   5.234   7.666  1.00 25.02
CONECT    1   21   45
CONECT    2   48   75
CONECT    3   54   74  186  285
CONECT    4   17   79
CONECT    5   11   35  275
CONECT    6   27   93
.....
.....
CONECT  279  284
CONECT  289  319
CONECT  290  310
CONECT  292  311
CONECT  314  323
END
```

Example of PDB file:

```
REMARK Written by O version 6.2.1
REMARK Fri Oct 23 16:33:54 1998
CRYST1  110.824  110.824  136.445  90.00  90.00 120.00
ORIGX1      1.000000  0.000000  0.000000        0.00000
ORIGX2      0.000000  1.000000  0.000000        0.00000
ORIGX3      0.000000  0.000000  1.000000        0.00000
SCALE1      0.009023  0.005209  0.000000        0.00000
SCALE2      0.000000  0.010419  0.000000        0.00000
SCALE3      0.000000  0.000000  0.007329        0.00000
ATOM      1  CB  SER A   4     -13.135  87.310  34.647  1.00 40.24   6
ATOM      2  OG  SER A   4     -14.273  87.097  33.827  1.00 41.79   8
ATOM      3  C   SER A   4     -11.189  85.881  35.294  1.00 39.68   6
ATOM      4  O   SER A   4     -11.596  84.836  35.803  1.00 40.58   8
ATOM      5  N   SER A   4     -12.413  85.458  33.178  1.00 40.50   7
ATOM      6  CA  SER A   4     -11.941  86.512  34.126  1.00 40.79   6
....
....
ATOM   4198  N   THR B 535      18.270  34.250  65.285  1.00 44.15   7
ATOM   4199  CA  THR B 535      17.709  33.497  66.407  1.00 46.23   6
ATOM   4200  CB  THR B 535      16.249  33.066  66.123  1.00 46.88   6
ATOM   4201  OG1 THR B 535      16.165  32.489  64.814  1.00 48.35   8
ATOM   4202  CG2 THR B 535      15.309  34.263  66.214  1.00 46.49   6
ATOM   4203  C   THR B 535      18.528  32.240  66.710  1.00 46.36   6
ATOM   4204  OT1 THR B 535      19.692  32.381  67.148  1.00 46.94   8
ATOM   4205  OT2 THR B 535      17.994  31.127  66.510  1.00 45.69   8
END
```

#PRINT ATOMS
the following keywords are allowed:

STANDARD	print co-ordinates and standard deviations
FULL	print the b(ij) for anisotropic atoms (default the B(equiv) will be printed)
UIJ	analysis of b(ij)
ORTHOGONAL	give orthogonal co-ordinates of atoms
END	ends the command.

in the output list the value of B(eq) is $4/3\sum\sum\beta_{ij}\, a_i.a_j$

#UPDATE ATOMS
the allowed keywords are:

RESET	copy the list 35 into the list 5.

The directive MATRIX must follow the ATOM instructions (one or more).

ATOM *(atom)*	the (atom) define which atoms are used
MATRIX *a11 a12 .. a33 t1 t2 t3*	apply the matrix to the atoms indicated before.
ANISO *(atom)*	transform the thermal factor of the (atom) atoms from isotropic to anisotropic.
HANISO *(atom)*	(as ANISO, not considering the hydrogen atoms)
ISO *(atom)*	from anisotropic to isotropic.
DELETE *(atom)*	delete (atom) atoms from list 5 .
ADD *species serial x y z b(iso) occ*	add the atom to the list 5
SPECIES *(atom) species*	assign the species to the (atom) atoms
SERIAL *(atom) serial_no.*	assign the serial number to the (atom) atom
HEIGHT *(atom) a*	assign the height a to the (atom) atom
BMEDIUM *a*	assign the value a to the B(overall)
SCALE *a*	assign the value a to the Scale factor
BOND *(atom1) (atom1)*	the atoms (atom1) and (atom2) are bonded
NOBOND *(atom1) (atom1)*	the atoms (atom1) and (atom2) are not bonded
CONNECTIVITY	the program builds the connectivity, having fixed the coordinates of the atoms defined in previous ATOM directive(s)
SORT	sort the atom list for species and serial number
INCLUDE	add to the list 5 the atoms from a file containing a list 5:
file_name	*#list 5* (this card will be skipped) *scale b(overall)*… (this card will be skipped) *atom1…* *atom2…* ….. ##
PROJECTION *a11 a12 a13* *a21 a22 a23* *a31 a32 a33*	read the matrix to draw the structure
END	end the command.

#PUBLICATION ATOMS
the following keywords are allowed:

CIF	write a file in *CIF* format of atomic parameters;
WORD	write a table using the character TAB to put in columns the co-ordinates (useful if a word processor like WORD will be used);
SKIP *n*	*n* is no. of initial blank (default n = 1) ;
NO	the decimal point will not be printed (default : the decimal point is used) ;
COMPRESS *(OFF)*	single spacing (default condition), if the option OFF is present, double spacing is required ;
XFIGURE *n*	*n* is the number of digits for x, y, z (default n = 15) ;
XDECIMAL *n*	*n* is the number of decimals for x, y, z (default n = 4) depending on st. dev.;
BFIGURE *n*	*n* is the number of digits for b(i,j) (default n = 14) ;
BDECIMAL *n*	*n* is the number of decimals for b(i,j) (default n = 4) depending on st. dev.;
IFIGURE *n*	*n* is the number of digits for B(iso) (default n = 12) ;
IDECIMAL *n*	*n* is the number of decimals for B(iso) (default n = 1) depending on st. dev.;
BEQUIV	calculate from b(i,j) the B(equiv) ;

UEQUIV	calculate from b(i,j) the U(equiv) ;
BIJ	calculate from b(i,j) the B(i,j) ;
UIJ	calculate from b(i,j) the U(i,j) .
U	write a table with U(equiv) and a table with the U(ij);
B	write a table with B(equiv) and a table with the B(ij);
NDEVIATION *n*	*n* is the number of digits of the standard deviations;
FIXED	the number of decimals is fixed,
LIST	list the allowed keywords.
END	ends the command.

in the table the value of B(eq) is $1/(6\pi^2) \sum\sum\beta_{ij}\, a_i.a_j$

#DUMP ATOMS [*keyword*]
the following keywords are allowed:

CAOS	(default)
NOH	(dump no-hydrogen atoms only)
SHELX	
MOLDRAW	
CHEM-X	
SYBYL	
PDB	
PDB_NOH	(dump no-hydrogen atoms only)

The output file can be input to the corresponding program.

Examples:

```
...
#update  5
aniso
end
 ...
#update  5
aniso  c o
end
 ...
#update  5
atom sn
matrix   -1 0 0   0 -1 0   0 0 -1   0  0  0
end

#publ  5
end

#publ  5
b
fixed
end
```

Reflections *List 6*

#READ REFLECTIONS
the following keywords are allowed:

FILE	the program prompts the request of the file name:
file_name	(the file where the reflections are stored)

FOSQUARE	the Fo^2's are read and the Fo are calculated and stored on binary file
CARD m n	m is number of reflections for each record n is the number of data to be read (default n=7)
SEQUENCE (codes..)	which data are read. Codes are: H K L FO SIGMA FC PHASE CODE
FORMAT format	format to be used (remind: the indices are read as floating point numbers)
RADIANT	the phase are expressed in radians
END	ends the command.

If the keyword FILE is not given, the reflections follow the END. The default quantities read in free format for each reflections are:

h k l Fo σ(Fo) Fc Phase

and it is not necessary that all the information are present. The dummy reflections ends the list.

512 512 512

The list 18 is automatically created.

#PRINT REFLECTIONS
the allowed keywords are:

ALL	print all the reflections (default).
FIRST n	print the first n reflections.
LAST n	print the last n reflections.
SCALE	print the Fo on absolute scale.
COLUMNS n	number of columns.
LINES n	number of rows per page
STATISTIC	print only some information on list 6.
END	ends the command.

#UPDATE REFLECTIONS
the allowed keywords are:

OMIT *[h k l]* *[h k l* *....* *h k l* *127 127 127]*	flag as deleted the followed reflection(s) *h,k,l*
UNOMIT	reset the flag of the all the 'OMIT' reflection(s).
CHOP *a*	flag as unobserved the reflection having Fo<a*σ(Fo).
CUTOFF	(as CHOP)
SYSTEMATIC *n*	delete the systematic absences and assign the same indices to all the equivalent reflections. If n=0 applies the centre of symmetry, if n=1 , the centro-symmetric reflections are left as they are.
MATRIX *a11 a12 .. a33 t1 t2 t3*	apply the operator to the indices : H' = (H*a11)+(K*a12)+(L*a13)+t1 K' = (H*a21)+(K*a22)+(L*a23)+t2 L' = (H*a31)+(K*a23)+(L*a33)+t3
CONDITIONS	the quantities of each reflection are compared with the limit stored in list 18 and not more used in the subsequent calculations. The reflections can be reused changing the limits in list 18 and update again the reflections.

SORTING *ilent* *ivelox*	are the code of the sorting
OBSERVED	the unobserved reflections are definitively delete from the list 6.
SCALE *a*	F(obs) are multiplied by a .
MODIFY	calculate the equivalent indices and preserve hkl in the limits of list 18.
GENERATE	Generate a file with all the equivalent reflection without the centre of symmetry.
END	ends the command

#PUBLICATION REFLECTIONS
the allowed keywords are:

CIF	write a file in *CIF* format ;
LASER	write a file in a format according with the laser printer; (default: columns=3; lines=62; spaces=3; skip=5)
COLUMNS n	no. of columns *per* page (default = 4) ;
LINES n	no. of lines *per* page (default = 57) ;
SPACES n	no. of spaces between columns (default = 3) ;
SKIP n	*n* is no. of initial blank (default n = 10) ;
SCALE a	a multiplies Fo and Fc (default = 1.0) .
END	ends the command.

The reflections must be sorted by indices (use #SORTING)

#DUMP REFLECTIONS [keyword]
the allowed keywords are:

ALL	(default, CAOS format)
OBSERVED	(only the observed structure factors are dumped, CAOS format)
SHELX	

Example:

```
#read reflections
end
0 0 1 372.9
0 0 2 387.7
0 0 3 1224.2
0 0 4 640.3
 ...
12 4 1 29.2
13 1 -1 27.6
13 1 0 61.2
127 127 127

#print 18
#end..

#print  6
statis
end
#end
```

Refinement directives List 12

#READ REFIN

The following directives define the type of matrix to accumulate the least squares derivatives. Two type of directives exist, type A and type B ; the first directive must be of type A and only the directive BLOCK can be repeated to start the user-defined blocks approximation.

The label of an atom is Species(Serial_number). The atoms involved into the refinement process can be indicated as:
1) no indication (all the atoms are considered);
2) Species (all the atoms belonging to that species are considered)
3) Label (only that atom is considered)
4) Label1 until Label2 (the atoms in the list starting from Label1 until Label2 are considered)
A mixed way can be used.
Examples:

c	(only carbon atoms are indicated)
c o	(carbon and oxygen atoms are indicated)
c(1)	(only this atoms is indicated)
c(1) until c(5)	(if c(2), c(3) and c(4) are present in the list between c(1) and c(5), all these atoms are considered)
c(1) c(12)	(only c(1) and c(12) are considered)
c(2) c(7) until c(15)	(the atoms c(2) and c(7),c(8),c(9)...,c(15) are considered)

To indicate the parameters, the following codes are used:

nothing	x y z and b[iso] for isotropic atom x y z and $\beta_{ij}$for anisotropic atom
x y z	positional co-ordinates
x's	to indicate all three co-ordinates x, y and z
b[iso]	to indicate the isotropic thermal parameter
11 12 13 22 23 33	β_{ij} anisotropic thermal parameters
ij ii	to indicate the off- and the diagonal thermal parameters
occ	occupation (the main species for mixed site)
occ1	occupation of 1st species for mixed site
occ2	occupation of 2nd species for mixed site
occ3	occupation of 3rd species for mixed site
occ4	occupation of 4th species for mixed site
scale	overall scale
boverall	the overall temperature factor
extinction	extinction coefficient
polarity	polarity coefficient (not used at the moment)

Type A:

FULLMATRIX	atoms params	The parameters of the indicated atoms are included in a full matrix (the scale factor is automatically included). In the default way, all the atoms are included.
DIAGONAL	atoms params	The parameters of the indicated atoms are refined using diagonal matrix approximation, one block per atom ; an extra block is added for the scale factor and the overall temperature factor. In the default way, all the atoms are included.
BLOCK	atoms params	The parameters of the indicated atoms are included in the same block. The directive BLOCK has to be repeated for each block.
DEFAULT	FULL	
DEFAULT	DIAG	To set the default refinement conditions.

Type B:

PLUS atoms params
The parameters of the indicated atoms are added to those defined by the previous instruction of type A

FIXED (or FIX) atoms params
To fix the parameters of the indicated atoms.

EQUIVALENCE atom1 param1 atom2 param2
To equivalence parameters of the selected atom(s). If atom2 is omitted, param1 and param2 are referred to the atom1 . Using this instruction, param1 is constrained to param2 in the least squares process and only one parameter is refined . This may be due to symmetry or any other constraint process. The obtained shift, multiplied by proper factor1 and factor2 respectively (see next instruction), is applied to param1 and param2 . The default value for factor1 and factor2 is 1.00.

FACTOR atom param coef
To set the value of the factor *coef* which multiplies the shift of the parameter of the indicated atom.

RIDE atoms1 atoms2
Normally the atoms2 indicates hydrogen atoms and atoms1 one of the species carbon, oxygen, nitrogen and boron if they are present in the content of the cell. The hydrogen co-ordinates 'ride' on the atoms to which they are attached.

DIST target sigma atom1 atom2
The distance between atom1 and atom2 is restrained to the target value with the estimated standard deviation sigma.

ANGLE target sigma atom1 atom2 atom3
The angle is restrained to the target value with the e.s.d. sigma. The angle is defined as the angle subtended at atom2 by atom1 and atom3.

PLANAR [sigma] atoms
The program computes the mean plane trough the atoms and then restrains each of the atoms to lie in the plane (default value for sigma=0.001Å).

SUM [sigma] [atoms] params
This restraint holds the sum of the parameters. This is the default case for polar space groups, where the singularity up the polar axis may be removed by holding the sum of all the co-ordinates up that axis constant (default value for sigma=0.00001)

LIMIT X_Sigma_value B(iso)_Sigma_value B(ij)_Sigma_value Atoms
The l.s. imposes restraints against excessive shifts for the given atoms. The _Sigma_values are the permissible values for the distributions of shifts for positional, isotropic and anisotropic thermal parameters.

END
To end the directives input.

#PRINT 12

#DUMP 12

Example:

```
#read refin
default  full
end

#read refin
diag  o  c
ride  c h
##
...
#read refin
full  br(1) until c(15)  x's  ij
fix br(1) z
```

```
ride  c h
end
...
#read refin
block first until last x's
block o c ij
plus scale
##
...
#read refin
block o
block c
plus scale
ride c h
end
...
#read refin
block o c
plus scale
equiv o x y
end
...
#read refin
full
plus  c(1)   occ
equiv  c(1) occ c(11) occ
factor  c(11)   occ    -1.0
end
...
#read refin
block c(1) until c(48) x's ij
plus h(1) until h(483) x's
plus scale
ride    c(1) until c(48) h(1) until h(483)
block c(101) until c(148) x's ij
plus h(11) until h(1483) x's
ride c(101) until c(148) h(11) until h(1483)
end
...
#read refin
diag br(1) until c(12)
ride  c h
end

#list 12
full
fix   om1(1)   y
fix   om1(1)    23
fix   om1(1)    12
fix   o(6)     y
fix   o(6)      23
fix   o(6)      12
fix   mgp2(1)  x   z
fix   mgp2(1)   23
fix   mgp2(1)   12
fix   mgp2(2)  x   z
fix   mgp2(2)   23
fix   mgp2(2)   12
fix   mgp2(3)  x y z
fix   mgp2(3)   23
fix   mgp2(3)   12
fix   nap1(1)  x   z
fix   nap1(1)   23
fix   nap1(1)   12
fix   nap1(2)  x y z
fix   nap1(2)   23
fix   nap1(2)   12
fix   nap1(3)  y
fix   nap1(3)   23
fix   nap1(3)   12
fix   nap1(4)  x   z
fix   nap1(4)   23
fix   nap1(4)   12
fix   h(1)     y
fix   fep2(3)  x   z
ride  o h
fix h b[iso]
fix o(1) ij
plus o(1) occ occ1
sum o(1) occ  occ1
fix o(2) ij
plus o(2) occ occ1
sum o(2) occ  occ1
fix om1(1) ij
plus om1(1) occ occ1
sum om1(1) occ  occ1
fix o(3) ij
plus o(3) occ occ1
sum o(3) occ  occ1
fix o(4) ij
plus o(4) occ occ1
sum o(4) occ  occ1
fix o(5) ij
plus o(5) occ occ1
sum o(5) occ  occ1
fix o(6) ij
plus o(6) occ occ1
sum o(6) occ  occ1
fix al(1) ij
plus al(1) occ occ1  occ2   occ3
sum al(1) occ  occ1  occ2  occ3
fix si(2) ij
plus si(2) occ occ1
sum si(2) occ  occ1
fix mgp2(1) ij
plus mgp2(1) occ occ1
sum mgp2(1) occ  occ1
fix mgp2(2) ij
plus mgp2(2) occ occ1  occ2
sum mgp2(2) occ  occ1  occ2
fix mgp2(3) ij
plus mgp2(3) occ occ1
sum mgp2(3) occ  occ1
fix nap1(1) ij
plus nap1(1) occ occ1
sum nap1(1) occ  occ1
```

```
end
#end
```

Map limits List 14

The list 14 gives information to calculate a map.

#READ LIMITS
The allowed keywords are:

X *Min Max Step Div*	minimum, max, step and unit for the a axis
Y *Min Max Step Div*	for the b axis;
Z *Min Max Step Div*	for the c axis .
PEAK *n*	Number of peaks requested
LEVEL *a*	Threshold for peak search routine;
MULTIPLY *a*	a is the scale for electronic density. The program multiplies in any case for 10.0
F000 *a*	give the F(000) (default : a= 0.0).
ORIENT *ioriz ivert*	the orientation of the map on the printout giving the horizontal axis (*ioriz*) and the vertical axis (*ivert*) ; (remind: 1= axis X ; 2= axis Y ; 3= axis Z) .
DEFAULT	produce a list 14 using default values.
END	

Examples:

```
#list 14
x  0.0 0.5
y 0.0 0.5
z 0.0 1.0
peak 25
level 20.
orient 1 3
end
 ...
#list 14
default
peak  50
end
 ...
#list 14
x 0 50 2 100
y 0 50 2 100
z 0 100 2 100
orient  1 3
end

#list 14
default   diag
end
```

Reflections conditions List 18

The list 18 has two functions: 1) to keep the min and max values of some quantities of the reflections; 2) to modify the reflections to be used in the calculations changing the min and max values and updating the list 6.

#READ CONDITIONS
The allowed keywords are:

H	*min*	*max*	min and max values for *h* ;
K	*min*	*max*	min and max values for *k* ;
L	*min*	*max*	min and max values for *l* ;
FO	*min*	*max*	min and max values for *Fo*;
SIN(	*min*	*max*	min and max values for $sin(\theta/\lambda)^2$;
FC	*min*	*max*	min and max values for *Fc* ;
FO/F	*min*	*max*	min and max values for *Fo/Fc* ;
(FO-	*min*	*max*	min and max values for *[Fo-Fc]/Fo* ;
2THETA	*min*	*max*	min and max values for 2θ;
RESOLUT	*min*	*max*	min and max values for the *resolution*;
RESET			update the user values using the stored experimental values;.
SET			calculate all the value from the experimental data of list 6.
HKL	*a b c d e f*		coefficients for defining validity conditions to accept reflections: f > H*a + K*b + L*c + d > e
INTERNAL			condition internal the range
EXTERNAL			condition external the range
CLASS	*code class*		flag the class of reflections (compatibility with SIR2002)
OMIT	*[h k l]*		store the reflections to be deleted
[h k l			
....			
h k l			
127 127 127]			
END			ends the command

code class are:

GGG	GGU	GUG	GUU	UGG	UGU	UUG	UUU						
H00	0K0	HK0	00L	H0L	0KL	HKL							
G00	U00	0G0	0U0	GK0	UK0	HG0	HU0	00G	00U	G0L	U0L	H0G	H0U
0GL	0UL	0KG	0KU										

Examples:

```
#list 18
set
(fo-fc)/fo  0.0  0.3
end
#update reflecti
conditions
end

#list 18
fo 2.0   300.
end
#update reflec
condit
end
```

```
#list 18
reset
end
#update  6
condit
end
```

Data reduction

#SORTING *il iv ia* — where il is the slowest varying index , iv is the fastest index and ia is a code: if ia=1 the sorting is done using the absolute values of the indices.

#MERGE — average the close equal reflections

#DATA — follow the reflections, like list 6. The results of this command is:
- read and store list 6
- the equivalent reflections are reported with the same indices
- sorting
- merge
- wilson statistic
- put the fo on absolute scale
- create the list 4, 14, 12, 18 using default values.

Examples:

```
#title dibromo
#list  1
12.268   12.268   9.923  90. 90. 90.
  0.003     0.004    0.002  0.05  0.06  0.07  0.9
#list  2   p  41
#list  3
br 8 cl 12 o 4 n 8 c 64 h 52
end
#list 6
end
   0   0   2        1.68536        6.29010
   0   0   3        9.50451        3.09145
   0   0   4      195.01208        2.37237
   0   0   5        2.48926        9.29041
   1   0   6      208.44153        2.53291
    .....
    .....
  18   3   1        5.03256       19.05841
  18   3   2       33.41090        5.80577
  18   4   0        5.12315       19.12062
  18   4   1        5.14043       19.18509
 127 127 127    5.14043       19.18509
#update 6
syst  0
end
#list 18
set
end
#sort
#merge
```

```
#list 18
set
end
#end
```

The following example is equivalent to the above example:

```
#title  dibromo
#list  1
12.268   12.268   9.923  90. 90. 90.
  0.003     0.004    0.002  0.05  0.06  0.07  0.9
#list  2   p 41
#list  3
br 8 cl 12 o 4 n 8 c 64 h 52
##
#data
   0   0   2       1.68536       6.29010
   0   0   3       9.50451       3.09145
   0   0   4     195.01208       2.37237
   0   0   5       2.48926       9.29041
   1   0   6     208.44153       2.53291
    .....
18   3   1       5.03256        19.05841
  18   3   2       33.41090        5.80577
  18   4   0        5.12315       19.12062
  18   4   1        5.14043       19.18509
 127 127 127    5.14043       19.18509
#end
```

Patterson and Fourier map

The commands allows some keyword to override the value stored in list 14.

#PATTERSON
The allowed keywords are:

AUTOMATIC		automatic solution of a Patterson map.
OBSERVED		use Fo (default conditions)
CALCULATED		use Fc
PRINT		print output the map
LEVEL	a	threshold of the electron density for peak searching
PEAK	n	number of requested peaks
F000	a	a is the F(000) to be added
MULTIPLY	*a*	a is the scale for electronic density. The program multiplies in any case for 10.0
END		ends the command

#FOURIER
The allowed keywords are:

OBSERVED		use Fo (default conditions)
CALCULATED		use Fc
DIFFERENCES		use Fo-Fc
PRINT	k	k=1 print the localized peaks; k=2 print the map
LEVEL	a	threshold of the electron density for peak searching

PEAK n — number of requested peaks

F000 a — a is the F(000) to be added

MULTIPLY *a* — a is the scale for electronic density. The program multiplies in any case for 10.0

ATOM (atom) — the (atom) define which atoms are used

FREE — allow the atoms defined in previous ATOM directive(s) to be moved in different asymmetric units in the interpretation of the map.

END — ends the command

Examples:

```
#patter
automatic      (if the automatic procedure PAUTO has to run)
end

#patt
end
or
#patt    #      (the character # after the directive means to use the default values)

#fourier  #

#fourier
peak 10
level 40
end

#fourier
diff
pe 30
le 5
end
```

LSQ routine

The structure factors F_{obs} are used in the least squares refinement process and it is possible to ignore data with F_{obs} less than *chop_value* multiply $\sigma(F_{obs})$.
A variety of weighting schemes is available and the user can choose a scheme which does not give marked systematic trends with the magnitude of F_{obs} or of resolution. If the Cruickshank weighting scheme is selected, at the end of the refinement process a routine calculates the best coefficients of the function. In this way, the scheme would give constant $<w\Delta^2>$ when the set of $w\Delta^2$ values is analysed in any significant systematic fashion and a value of the Goodness of Fit equal to the unit.
A damping factor can be set by the user to improve the convergence properties of the refinement.
The atomic parameters are refined using various type of matrix to accumulate the least squares derivatives. It is possible to use any user-defined blocks approximation starting from the full matrix type until the block diagonal *per* atom (4x4 for isotropic atom and 9x9 for anisotropic atom).
The scale factor is also refined and the constraints due to the space group are automatically taken into account. Nevertheless, the user can fix any parameters and set any equivalence between refined parameters.
If hydrogen atoms are present, their positional co-ordinates ride on the atoms to which they are attached and their isotropic thermal parameters are fixed equal to that of the attached atom.

#LSQ
The allowed keywords are:

NUMBER n — The number of the first calculation for this run.

PRINT — For each reflection the output of structure factor calculation is printed

VERBOSE n — code to determine the level of printing:
n=0 (default conditions), the values of R% , wR% , S and some other general statistics
n=1 the atomic parameters
n=2 details of the refinement

ANOMALOUS code — the anomalous coefficients are taken into account (code = ON) (default condition), code = OFF no anomalous scattering factor is introduced.

CYCLES n — the number of requested cycles

CALCULATION — only one structure factor calculation is performed

WEIGHTING type npar par1 par2 ... (see #LIST 4)
type number of the weighting scheme requested (1 - 12)
npar number of parameters
par1 par2 ... npar parameters of the weighting scheme

CUTOFF chop_value — To ignore data with Fo less than chop_value multiplied by σ(Fo).

CHOP chop_value — like CUTOFF

DAMPING damp — The calculated shifts are multiplied by damp before being added to the parameters

SCALE — To refine the scale only.

ISO atoms — The thermal factors of the atoms here indicated are transformed into isotropic ones

ANISO atoms — The thermal factors of the atoms here indicated are transformed into anisotropic ones

CHECK *on/off* — to check or not the list 12 (default: not check)

NOREFL — do not use the reflections (refine the coordinate using the restraints)

RESISTANCE *s(x) s(B) s(bij)* — resistance to excessive shifts. the input values are the permissible breadths of shifts for positional, B(iso) and anisotropic thermal parameters.

END — ends the command

Examples:

To run a least squares refinement:

```
#lsq
end       or
#lsq   #
```

the run ends when convergence is reached.

```
#lsq
cycle 4
end
```

only 4 cycles are requested.

```
#lsq
scale      (refinement only of the scale)
end
```

```
#lsq
calculat    (structure factors calculation only)
end
```

Optimized weighting function

#WEIGHT
The allowed keywords are:

VERBOSE
END

Examples:

#weight #

#weight
verbose
end

Agreement analysis

#AGREE
The allowed keywords are:

VERBOSE			
PRINT	a	print the reflection(s) having |Fo-Fc|> a*Fo	
SELECT	*Keyword min max*	print selected reflections defined by the keyword. Allowed keywords are:	
		RESOL min max	(print the reflections within the range)
		FO min max	(print the reflections within the range)
		GGG	print the reflections with hkl of type ggg
		GGU	etc..
		GUG	
		UGG	
		GUU	
		UGU	
		UUG	
		UUU	
		H00	
		0K0	
		00L	
		HK0	
		H0L	
		0KL	
		HKL	
END			

Examples:

#agree #

#agree
select 00l
end

#agree
verbose

```
print   0.3
end
```

Distances and angles

#DIST (dmin) (dmax) — calculate the distances between atoms in the range (dmin) and (dmax)

#DIST (dmin) (dmax) (dmin) (dmax) — calculate the distances and angles between atoms in the range (dmin) and (dmax)

#SHORT (dmin) (incr) — calculate the distances larger than (dmin) taking into account the atomic radii +/- (incr)(default incr=0.3)

#LONG (dmin) (incr) — calculate the long distances greater than (dmin) taking into account the van der Walls readii +/- (incr)(default incr=0.3) between atoms of different asymmetric units

#ANGLE (dmin) dmax) — calculate ancles between atoms in the range (dmin) and d(max)

Geometrical calculation of hydrogen atoms

#HYDROGEN
The allowed keywords are (*atoms* define the atom(s) which the hydrogen atoms are bonded to):
- instruction

33 *atoms* — three atoms bonded to carbon atom sp3

32 *atoms* — two atoms bonded to carbon atom sp3

31 *atoms* — one atom bonded to carbon atom sp3

22 *atoms* — two atoms bonded to carbon atom sp2

21 *atoms* — one atom bonded to carbon atom sp2

11 *atoms* — one atom bonded to carbon atom sp1

40 *atoms* — the program finds the hybridization and calculates the hydrogen positions bonded to the given atoms.

30 *atoms* — the program calculates the possible hydrogen positions bonded to the given sp3 atoms.

20 *atoms* — the program calculates the possible hydrogen positions bonded to the given sp2 atoms.

10 *atoms* — the program calculates the possible hydrogen positions bonded to the given sp1 atoms.

AUTOMATIC — the program finds automatically the hybridization and calculates the hydrogen positions bonded to carbon atoms (default conditions)

INTERACTIVE — to check and modify the hybridisation of the N C B O atoms found by the program. The information will be saved for subsequent automatic calculation.

SERIAL *n* — serial number relative to the first H atoms

NAME *labe*l — species for the generated atoms (default h)

B[ISO] *value* — b(iso) for the atoms (default the b(iso) is equal to the bonded atom)

DIST *a* — value of bond distance (default 0.96)

END — ends the command

Example:

```
#hydrog  #

#hydr
```

```
33   c(1)
32   c(11)  c(12)   c(13)

#hydrogen
atom  c(1)
33
atom c(11)
atom c(12)
atom  c(13)
32

….
```

In the first example, the program calculates automatically the positions of H atoms. In the second (and the third is the same), the program calculates and adds to the list 5 the co-ordinates of the three hydrogen atoms bonded to c(1) and two H atoms bonded to c(11), to c(12) and to c(13) .

Geometry

It is possible to calculate in this module some geometrical quantities. The esds are estimated from the diagonal elements of the correlation matrix. When the calculation of the coefficients of a plane is requested, they are represented by an identification label given by the user. In this way, subsequent calculations can be done recalling the plane *via* this label. Furthermore, it is possible to indicate a line passing through two atoms and to compute distances and angles using this vector. For bond lengths and angles calculation, the results can be obtained as a table ready for publication or a *Crystallographic Information File* file, named *structure_name*.cif. The torsion angles calculated follow the conventions defined by W. Klyne and V. Prelog, Exper. (1960)16,52.

The atoms involved in the calculations are indicated in a general way:

Species(Serial_number,[Sym,Npl,Tx,Ty,Tz])

where Species and Serial_number are the usual label of the atom; (Sym is the number of the symmetry operator of the space group and the leading sign minus informs that the centrosymmetric operator is requested; Npl is the number of the lattice operator (see table below); Tx, Ty and Tz are positive or negative numbers indicating a lattice translation along axis a, b and c respectively.
The requested atom list in the instructions follows the rules given in LSQ module.

Lattice	no. of operators	Npl = 1	Npl = 2	Npl = 3	Npl = 4
P	1	0 0 0			
I	2	0 0 0	1/2 1/2 1/2		
R	3	0 0 0	1/3 2/3 2/3	2/3 1/3 1/3	
F	4	0 0 0	0 1/2 1/2	1/2 0 1/2	1/2 1/2 0
A	2	0 0 0	0 1/2 1/2		
B	2	0 0 0	1/2 0 1/2		
C	2	0 0 0	1/2 1/2 0		

The command is:

```
#GEOMETRY  STANDARD
```

One or more of the three following instruction must precede each request of geometrical calculation.

ATOM *atoms* The named atoms are considered
FROM *atoms* The program calculates distances and angles from the named atoms
TO *atoms* The program calculates distances and angles to the named atoms
LINE *atom1 atom2* The two atoms identify a straight line
PLANE *label* The label identifies the plane calculated before in the same run.

The following instructions allow the geometrical calculation.

SYMM apply the symmetry operators (DIST and ANGLE calculations)
P3P *label* A plane for three atoms is calculated.
PQM *label* A least-squares mean plane is calculated. Deviations of the atoms from the plane are listed.
PNR *label* The coefficients of the plane perpendicular to a line and passing through an atom are calculated
DIST *dmin dmax* Distances are calculated for the indicated atoms defined by the instruction FROM, TO or ATOMS in the range dmin – dmax. If *dmin* (and *dmax*) is not given, the program takes into account the covalent radius to decide if the atoms are bonded. If the directive does not follow the instructions FROM, TO or ATOM (i.e., the user does not indicate any atoms) the distances are calculated from the connectivity.
VAN Distances are calculated from a.u. to equivalent unit and contiguous cells larger than the sum of covalent radii and smaller than sum of Van der Waals radii.
ANGLE *dmin dmax* Bond angles are calculated for the indicated atoms having distances in the range dmin - dmax. likewise the instruction DIST
TOR The atoms define a chain of at least four atoms. A list of torsion angles is generated for all torsion angles defined by this chain.
APP The angle between two planes is calculated.
ARP The angle between a line and a plane is calculated
ALL The angle between two lines is calculated
DPA Distances of atoms from a plane are listed.
DRA Distances of atoms from a line are listed.
PM Calculate the barycentre of the indicates atoms.
CALC ±*dist* Calculate the coordinates of a point along a line between two given atoms having a distance *dist* (positive or negative) from the first atom.
SAVE The subsequent bond length and angle calculation is stored.
CIF A *Crystallographic Information File* file '*structure_name*.cif' is created containing the bond lengths and angles stored in memory in the self-defining STAR format.
TABLE *nc ndev nfix* The tables of bond lengths and angles previously stored are printed, ready for publication.
nc number of columns (default=2)
ndev number of digits for esd (default=1)
nfix = 0 decimal digits depending on esd (default)
= 1 number (3) of decimal digits is fixed
END end the directives input.

Examples :

#geometry
or
#geometry standard
atom all the atoms are considered

```
dist                            bond distances in the default range 0.2-1.7
atom                            all the atoms
angle                           angles between atoms in the default distance range 0.2-1.7
atom c(1)
atom c(2)
atom c(5)
atom c(6)
atom o(6)                       the sequence c(1), c(2),..,o(6) defines a chain
torsion                         torsion angles are computed
atom                            all the atoms are selected
pqm  p1                         l.s. mean plane labelled 'p1' is calculated
atom c(1) until c(5)
atom o(5)                       the atoms between c(1) and c(5) plus o(5) are selected
pqm p2                          l.s. mean plane labelled 'p2'
plane p1
plane p2
app                     the angle between the planes 'p1'^'p2' is calculated
line o(1) c(1)                  the line through the atoms o(1) and c(1) is defined
plane p2
arp                             the angle between the line and the plane 'p2' is calculated
atom o(1)
plane p2
dpa                     the distance of o(1) from the plane 'p2' is computed
line o(1) c(1)
atom c(7)
dra                             the distance of c(7) from the line o(1)-c(1) is computed
atom c(1)
atom c(2)
atom o(5)
p3p  p3                         the plane 'p3' through c(1),c(2) and o(5) is computed
line o(1) c(1)
plane p3
arp                             the angle between the line o(1)-c(1) and the plane 'p3' is calculated
line o(1) c(1)
line o(1) c(7)
all                             the angle between lines o(1)-c(1) and o(1)-c(7) is computed
line o(1) c(1)
atom c(1)
pnr  p4                         the plane perpendicular to the line o(1)-c(1) and
                                passing through the atoms c(1) is computed
plane p4
plane p3
app                     angle between planes 'p4' and 'p3' is calculated
save                            the following distance and angle calculated are stored
atom br(1) until c(12)          the atoms between br(1) and c(12) are selected
dist 0.2   2.2                  distances in the range 0.2-2.2 are calculated
atom c(1) until h(10)
dist 0.1   1.1                  distances in the range 0.1-1.1
atom br(1) until c(12)
angle 0.2  2.2                  angles between atoms in the distance range 0.2-2.2
atom c(1) until h(10)
angle 0.1  1.1
table                   distance and angle tables are printed
end
#end
```

Molecular drawing

To plot the structure on the screen (Microsoft Windows platform only) , on a HP-GL plotter or on a POSTSCRIPT file. Only for the Windows platform, the default is the output on the screen. Strike any key to exit from this graphic screen. For Unix and OpenVMS systems, the default is the output on a POSTSCRIPT file having the name *structure_file_name.ps*

#SKETCH
The command does not require keywords

#BALLS
The command does not require keywords

#ELLIPSOIDS
The command does not require keywords

#PLOT
This is the main directive to draw the structure.
To choose the output not on the screen, you must given the name and the type of the output file to be plotted using one of the following two options

To obtain a file in HP-GL format to print on a plotter, the instruction is:

PLOTTER
file_name (without extension, the program inserts (or replaces) the extension ".hpg")

To obtain a file in POSTSCRIPT format to print on a laser printer, the instruction is:

POSTSCRIPT
file_name (without extension, the program inserts (or replaces) the extension ".ps")

Follow the sections:

SCENA (optional) (define the object to draw)
ORIENT (optional) (give the plane of drawing)

If these sections (SCENA and ORIENT) are not given:

1) the default condition is that all the atoms are considered and the projection plane is taken from the projection matrix stored with the list of atoms (the default projection is the plane *a^b*).
2) It is possible apply rotations around the cell axis with the following keywords:

   XROTATION *a* apply a rotation around the x axis of an angle *a*, the new projection plane is stored on list 5 and available in subsequent runs.
   YROTATION *a* as above for y axis
   ZROTATION *a* as above for z axis

3) the following keywords can be given:
   LINE *ngen xrm* draw black bonds modifying the number of lines between the atoms: *ngen* is in the range 1 ÷ 15 (default 6) ; *xrm* gives the ratio between the average radius of the bond respect to the radii of the bonded atoms (default 0.25) ;
   SCALE *a* apply a scale to the drawing
   STERESCOPIC to obtain a stereoscopic vision of the structure
   PACKING the view of the cell content

Follow the sections:

SKETCH	simple sketch
BALLS	sticks and balls
ELLIPSOIDS	sticks and ellipsoids
LABEL	write the label of atoms
HYDROGEN	allow to draw particular bonds

As the following table.

Sections

SCENA 0ptional

⇓

ORIENT Optional

⇓

SKETCH or BALLS or ELLIPS

⇓

LABEL Optional

⇓

HYDROGEN Optional

SCENA section:

SCENA
The allowed keywords are:

ATOMS		the named atoms are considered, this instruction must precede the following keywords
TRANSFER		transfer the indicated atoms from list 5 to a temporary list for drawing;
DELETE		do not consider the indicated atoms;
ONLY		consider only the indicated atoms;
MOLECULE	*d(min) d(max)*	starting from a fragment of atoms (indicated using the above instructions) change the co-ordinates of the atoms present in list 5 in order to build the molecule. Two atoms are considered bonded if the distances is in the range between *d(min) e d(max)* .
PACKING		apply the symmetry operators to obtain the cell content.
SOPERATION	*a11 a12 ... a33 t1 t2 t3*	apply the matrix to the indicated atoms;
SCODE	*nsym npl t1 t2 t3*	apply to the indicated atoms the *nsym* operator (see list 2) , the *npl* lattice operator (see Geometry) and the shift vector (*t1 t2 t3*) ;
ADDITION	*n*	this keyword does not need ATOMS keyword, but n records containing co-ordinates must follow: Ty(1) x y z b :::::::::::: Ty(n) x y z b
NEXT		ends the section:

ORIENT section:

ORIENT
The allowed keywords are:

IN	*[atom]*	the named atoms are included
OUT	*[atom]*	the named atoms are excluded
PLANE	*at(1) at(2) at(3)*	chose the projection plane:

		u = AT(1) - AT(2) v = AT(1) - AT(3) the orthogonal system is: x parallel to u z normal to the plane u^v y normal to the plane x^z
PROJECTION	*code*	CODE defines the projection plane: CODE = XY plane a^b XZ plane a^c YZ plane b^c ;
XROTATION	*a*	apply a rotation around the x axis of an angle a ;
YROTATION	*a*	as above for y axis ;
ZROTATION	*a*	as above for z axis ;
VROTATION	*a at(1) at(2) at(3)*	apply a rotation around the vector defined by the atoms (AT(1)-AT(2)) having the origin in the atom AT(3) ;
SCALE	*k*	apply a scale factor in cm/Å;
DSCALE	*k*	reduction factor (default = 0.8) .
NEXT		ends the section:

SKETCH section:

SKETCH

The default condition is: all the atoms are considered and the bonds to be drawn are defined by the connectivity present in the list 5. The allowed keywords are:

IN	*[atom]*	the named atoms are included
OUT	*[atom]*	the named atoms are excluded
BOND	*d(min) d(max) Ty1 Ty2*	define the bonds to be drawn in the range d(min) and d(max): if Ty1 e Ty2 are not present, the program takes into account all the atoms considered; if Ty2 is not present, only atoms having species Ty1 are considered ; if Ty1 and Ty2 are present, (indicating only chemical species) the bonds are between atoms of Ty1 species and atoms of Ty2 species; if Ty1 e Ty2 indicate two specific atoms, only that bonds will be drawn.
NOBOND	*Ty1 Ty2*	bond involving atoms of species Ty1 (and Ty2 if present) are not drawn; if Ty1 e Ty2 indicate two specific atoms, that bond will be not drawn.
EBOND	*Ty1 Ty2*	draw the bond between the atoms Ty1 e Ty2 .
NEXT		ends the section:

BALLS section:

BALLS

The default condition is: all the atoms are considered and the bonds to be drawn are defined by the connectivity present in the list 5. The allowed keywords are:

IN	*[atom]*	the named atoms are included
OUT	*[atom]*	the named atoms are excluded
BOND	*d(min) d(max) Ty1 Ty2*	define the bonds to be drawn in the range d(min) and d(max): if Ty1 e Ty2 are not present, the program takes into account all the atoms considered; if Ty2 is not present, only atoms having species Ty1 are considered ; if Ty1 and Ty2 are present, (indicating only chemical species) the bonds are between atoms of Ty1 species and atoms of Ty2 species; if Ty1 e Ty2 indicate two specific atoms, only that bonds will be drawn.
NOBOND	*Ty1 Ty2*	bond involving atoms of species Ty1 (and Ty2 if

present) are not drawn; if Ty1 e Ty2 indicate two specific atoms, that bond will be not drawn.

EBOND *Ty1 Ty2* — draw the bond between the atoms Ty1 e Ty2 .

RADIUS *r Ty* — assign a radius r to all the atoms , if Ty is not present; only to the atoms having species Ty ; to the atom Ty if a specific atom is indicated; (default: r=3.0 for H, r=7.0 for species until Z=13, r=9.0 for the remaining species)

BISOTROPIC *Ty* — assign a radius calculated from the thermal factor B(iso) to all the atoms , if Ty is not present; only to the atoms having species Ty ; to the atom Ty if a specific atom is indicated;

LINE *ngen xrm* — draw black bonds modifying the number of lines between the atoms: *ngen* is in the range 1 ÷ 15 (default 6) ; *xrm* gives the ratio between the average radius of the bond respect to the radii of the bonded atoms (default 0.25) ;

NEXT — ends the section:

ELLIPSOIDS section:

ELLIPSOIDS

The default condition is: all the atoms are considered and the bonds to be drawn are defined by the connectivity present in the list 5. The allowed keywords are:

IN *[atom]* — the named atoms are included

OUT *[atom]* — the named atoms are excluded

BOND *d(min) d(max) Ty1 Ty2* — define the bonds to be drawn in the range d(min) and d(max): if Ty1 e Ty2 are not present, the program takes into account all the atoms considered; if Ty2 is not present, only atoms having species Ty1 are considered ; if Ty1 and Ty2 are present, (indicating only chemical species) the bonds are between atoms of Ty1 species and atoms of Ty2 species; if Ty1 e Ty2 indicate two specific atoms, only that bonds will be drawn.

NOBOND *Ty1 Ty2* — bond involving atoms of species Ty1 (and Ty2 if present) are not drawn; if Ty1 e Ty2 indicate two specific atoms, that bond will be not drawn.

EBOND *Ty1 Ty2* — draw the bond between the atoms Ty1 e Ty2 .

EMPTY *Ty1 Ty2* — draw the bond between the atoms Ty1 e Ty2 showing only the external lines

RADIUS *r Ty* — assign a radius r to all the atoms , if Ty is not present; only to the atoms having species Ty ; to the atom Ty if a specific atom is indicated;

LINE *ngen xrm* — draw black bonds modifying the number of lines between the atoms: *ngen* is in the range 1 ÷ 15 (default 6) ; *xrm* gives the ratio between the average radius of the bond respect to the radii of the bonded atoms (default 0.25) ;

BEQUIVALENT *c* — give the coefficient defining the percentage of thermal ellipsoids (D.B. Owen, Handbook of Statistical Tables, Addison-Wesley, Reading, Mass., 1962);
c = 1.5382 correspond to 50% ;
c = 1.1932 correspond to 30% .

NEXT — ends the section:

LABEL section:

LABEL

The default condition is: the label of all the atoms are drawn using h=0.2cm. The allowed keywords are:

CAPTION *h iot Ty* — h is the height (cm) of the characters, (acceptable values are in the range 0.2 ÷ 0.5 cm); iot = 0 means that the Ty chemical species only is drawn ; iot = 1 , the label of the atoms having Ty chemical species is drawn.

SYMBOL *h CODE Ty* — h is the symbol height (cm) to indicate the Ty atoms, (acceptable values are in the range 0.02 ÷ 0.05 cm); keyword CODE can be:
STAR
SQUARE
TRIANGLE
HEXAGON
OCTAGON
CIRCLE

NEXT — ends the section:

HYDROGEN section:

HYDROGEN

The allowed keywords are:

IN *[atom]* — the named atoms are included

BOND *Ty1 Ty2* — define the bonds to be drawn

LINE *i* — to dash the bond lines, i is a integer number in the range 1 ÷ 6 ; i=1 for postscript output file.

PEN *j* — j is the position number of the pen in the carousel of the HP plotter in the range 1 ÷ 6;

NEXT — ends the section:

END — end the command.

Examples:

```
#BALLS

#ELLIPSOIDS

#PLOT
POSTSCRIPT
(file_name)
BALL
END
#END

#PLOT
BALL
NEXT
LABEL
END
#END

#PLOT
ELLIPSOIDS
END
#END
```

```
#PLOT
STEREO
BALL
END
#END

#PLOT
SCENA
ATOM  O(1) UNTIL O(4)
TRANSF
ATOM N(1)
TRANSF
ATOM C(1)  UNTIL C(37)
TRANSF
ATOM O(101)  UNTIL C(137)
TRANSF
NEXT
ORIENT
PLANE  C(6) C(3) C(7)
XROT  -65.0
XROT  -15.0
SCALE
DSCALE  0.8
NEXT
BALL
BOND 1. 1.6
RADIUS  8.  O
RADIUS  6.  N
RADIUS  4.  C
LINE  4   0.03
NEXT
LABEL
CAPTIONS  0.3  1  O
CAPTIONS  0.3  1  N
NEXT
END
#END

#PLOT
SCENA
ATOM  H
DELETE
ATOM
PACKING
ADD  4
V(1)  0. 1. 0.   1.
V(2)  1. 1. 0.   1.
V(3)  0. 1. 1.   1.
V(4)  1. 1. 1.   1.
NEXT
ORIENT
PROJECTION   XZ
ZROT  90.
SCALE
DSCALE  0.9
NEXT
SKETCH
BOND  1.1  1.6   C
BOND  1.1  1.5   C  O
BOND  1.3  1.5   C  N
BOND  1.4  1.8   N
EBOND V(1)  V(2)
EBOND V(1)  V(3)
EBOND V(4)  V(3)
EBOND V(4)  V(2)
NEXT
LABEL
CAPTION  0.2    V
SYMBOL   0.05  STAR  O
SYMBOL   0.1   TRIA  N
SYMBOL   0.01  CIRC  V
NEXT
END
#END

#PLOT
PLOTTER
(file_name)
SCENA
ATOM  GA(1) UNTIL S(3)
PACKING
ADD 4
V(1) 0.5 0.0 0.0 3.0
V(2) 0.0 0.5 0.0 3.0
V(3) 0.5 0.0 1.0 3.0
V(4) 0.0 0.5 1.0 3.0
NEXT
ORIENT
PLANE V(1) V(2) V(3)
XROT 0.4
ZROT 90.
SKALE 1.11111
NEXT
BALL
OUT V
RADIUS  12.0 FE
RADIUS   8.0  GA
RADIUS   24.  S
BOND   2. 3. GA S
BOND   2. 3. FE S
NEXT
LABEL
SYMB  0.01 CIRCLE V
NEXT
END
#END

#PLOT
PLOTTER
(file_name)
SCENA
ATOMS   O(1) until C(41)
TRANSF
NEXT
ORIENT
PLANE N(2) C(15) O(4)
```

```
DSCALE 0.8
# XROT 0.0
XROT 45
# YROT 0.0
YROT 20
NEXT
BALL
BOND .8 1.6 C
BOND .8 1.6 C N
BOND .8 1.6 N O
BOND .8 1.6 C O
BOND .8 1.6 N
# BEQUIV 1.5382
RADIUS 5.
LINE 10  0.04
NEXT
LABEL
CAPTIONS 0.3 1 K
CAPTIONS 0.3 1 B
CAPTIONS 0.3 1 O
CAPTIONS 0.3 1 N
CAPTIONS 0.3 1 C
END
#END
```

```
#PLOT
SCENA
ATOMS   HG(1) until C(13)
TRANSF
NEXT
ORIENT
PLANE N(2) N(4) N(5)
DSCALE 0.8
XROT 26
YROT 68
NEXT
BALLS
BOND .8 1.6 C
BOND .8 1.6 C N
BOND .8 1.6 B N
BOND .8 1.9 C S
BOND .8 1.6 N
BOND .8 2.5 HG N
BOND .8 2.5 HG S
RADIUS 9. HG
RADIUS 5. N
RADIUS 6. S
RADIUS 5. C
RADIUS 5. B
LINE 10  0.04
NEXT
END
#END
```

On Windows platform

- copy the file CAOS.ZIP in a new directory (i.e. c:\caos)
- unzip CAOS.ZIP

- install Caos program on your DeskTop using the standard link mechanism of Windows
- to run the program click on Caos icon

                        - @caosfor
- create the executable file using
                        - @caoslink
- associate the Caos program directory with the logical name Caos$root
- associate the symbol Caos with the command line @Run_caos.com

Call for Contributions to the Next CompComm Newsletter

The next issue of the Compcomm Newsletter is expected to appear around October of 2010 with the primary theme of Age Concern relevant to Protein Crystallography and/or Powder Diffraction. If no-one is else is co-opted, the newsletter will be edited by Lachlan Cranswick.

Contributions would be also greatly appreciated on matters of general interest to the crystallographic computing community, e.g. meeting reports, future meetings, developments in software, algorithms, coding, historical articles, programming languages, techniques and other news.

Please send articles and suggestions directly to the editor.

Lachlan M. D. Cranswick
Canadian Neutron Beam Centre (CNBC),
National Research Council of Canada (NRC),
Building 459, Station 18, Chalk River Laboratories,
Chalk River, Ontario, Canada, K0J 1J0
Tel: (613) 584-8811 ext: 43719
Fax: (613) 584-4040
E-mail: lachlan.cranswick@nrc.gc.ca