Chapter 1

Introduction To The System

1.1 Layout of this manual

This manual is organised as follows
1.0 Introduction
#Rdefandconv# Definitions and Conventions
#Rimmediate# Immediate commands - tailoring the program
#Rcrysdatabase# The CRYSTALS database
#Rinitialdinput# Initial Data input
#Rreflectdata# Reflection data
#Ratomparams# Atoms and Parameters
#Rsfandls# Structure factors and least squares
#Rfouandpat# Fourier and Patterson maps
#Rgeomandres# Geometry and printing results
#Rgrcameron# Graphics - CAMERON
#Rtwinning# Twinned Data
#Rmatrixcalc# The Basic Matrix Calculator

This version of the manual, has been revised to coincide with the release of version 12.00 of CRYSTALS. This revision includes further enhancements to the graphical user interface. Opportunity has also been taken to revise some features of the underlying code. Brief details of the changes are in the web document #Uhttp://www.xtl.ox.ac.uk/fixes.html#. We have tried to mininise the impact on existing users, and we hope that the advantages brought about by the changes will become evident.

1.2 Status

This version of CRYSTALS continues to be based on a version (Issue 2) written by J.R. Carruthers in collaboration with J.S.Rollett during 1977-78, which was a rewrite of the 1975 CRYSTALS system implemented on the ICL 1900 series of computers. It contains significant contributions by Paul Betteridge, David Kinna, Lisa Pearce, Allen Larson, and Eric Gabe and many students and visitors to the Chemical Crystallography Laboratory, Oxford. The graphical user interface (GUI) was written by Richard Cooper as part of a Part II and D Phil project (supported by a CCDC studentship) in collaboration with Ludwig Macko and Markus Neuburger in Basel, who were working on a parallel Macintosh interface.
A suitable citation is:
Betteridge, P. W., Carruthers, J. R., Cooper, R. I., Prout, K., Watkin, D. J. (2003). J. Appl. Cryst. 36, 1487.

While CRYSTALS can still be executed in 'batch mode' (ASCII file in, ASCII file out), the major demand is now for the version running under Windows on Intel processors, with some small demand for the LINUX version. The GUI permits the user to continually see the structure as it develops, and to interact with it and the analysis through conventional windows features. The 'command line' and 'use file' modes have been retained for experienced users, or users wishing to explore new ideas. The 'SCRIPTing' language has been extended to enable full control and design of the user interface to be handled from ASCII files.

1.3 MAJOR CHANGES for WINDOWS

The majority of the changes (many thousand edits) are concerned with the GUI, the screen and file output, and the internal database.

Major changes are:

```
1        New .DSC file
2        Extended LIST 5 format (the model parameters)
3        New ADP handling
4        New weighting for Fsq
5        More items stored in LIST 30
6        Easier handling of twins
7        More robust creation of cifs
8        HTML versions of the manuals
```

New .DSC File
The internal format of some of the lists has been extended to accommodate information needed to meet current publication requirements. There is some degree of compatibility between old format .DSC files and the new CRYSTALS. A facility in /EDIT attempts to do the internal reformatting. There is no backwards compatibility. There is full forward compatibility at the ASCII level - the old LISTS 5 (the model parameters), 6 (reflections), 12 (constraints) and 16 (restraints) can be input into the new program.

New Temperature Factor (ADP) handling

In the original version of CRYSTALS, the *value* of Uiso was used to indicate whether it was a real usable value (e.g. 0.055), or a flag indicating that the atom was anisotropic (e.g. 0.00).

This location in LIST 5 (the model parameters) has now been changed to a proper flag, whose value indicates what kind of information is stored after the positional parameters. This has enabled us to have more complex models for the electron distribution.

```
original keys:

    type serial occ u[iso] x y z u[11] u[22] u[33] u[23] u[13] u[12] spare

revised keys

    type serial occ FLAG x y z u[11]  u[22] u[33] u[23] u[13] u[12] spare
                            U[ISO]                                 spare
                            U[ISO] SIZE                            spare
                            U[ISO] SIZE  DECLINAT AZIMUTH          spare
```

The value of 'FLAG' is used on input of atoms to indicate what kind of patameters will follow, and is used during calculations for the interpretation of the parameters.

FLAG interpretation

The following table shows the interpretation of the FLAG parameter.

```
FLAG   meaning      parameters
'old' types of atoms:

0    Aniso ADP  u[11]  u[22] u[33] u[23] u[13] u[12]
1    Iso ADP    U[ISO]

New 'special' shapes:

2    Sphere     U[ISO] SIZE
3    Line       U[ISO] SIZE  DECLINAT AZIMUTH
4    Ring       U[ISO] SIZE  DECLINAT AZIMUTH
```

The parameters have the following meaning for the new special shapes:

Special U[iso]

U[iso] is related to the 'thickness' of the line, annulus or shell.

Special SIZE

SIZE is the length of the line, or the radius of the annulus or shell.

Special DECLINAT

DECLINAT is the declination angle between the line axis or annulus normal and th *z* axis of the usual CRYSTALS orthogonal coordinate system, in degrees/100.

Special AZIMUTH

AZIMUTH is the azumuthal angle between the projection of the line axis or annulus normal onto the *x* - *y* plane and the *x* axis of the usual CRYSTALS orthogonal coordinate system, in degrees/100.

If either of these angles is input with a value greater than 5.0, it is assumed that the user has forgotten to divide by 100, which is thus done automatically.

Accessing 'Special shapes'

The special shape parameters can be accessed just like any traditional parameters. FLAG is not a refinable parameter, and should only be changed in /EDIT if corresponding changes are to be made to the actual parameter values. This is normally automatic if the FLAG is changed to 0 or 1. To facilitate the generation of special shapes from groups of atoms, the following directives are available in /EDIT.

SPHERE newserial atomlist

This creates a 'shell' shape from the specified atom list. The centre of the shell is at the centre of gravity, the size is the mean distance of the given atoms from the centre, and the occupancy is equal to the sum of the occupancies of the atoms listed. U[iso] is the mean of the U[iso] or Ueqiv of the listed atoms. The atom TYPE is QS, with the given serial number. The original atoms are not deleted, though they should be or their occupancy set to zero. The atom type, QS, should be changed to something appropriate.

RING newserial atomlist

This creates an 'annulus' shape from the specified atom list. The centre of the ring is at the centre of gravity, the size is the mean distance of the given atoms from the centre, and the occupancy is equal to the sum of the occupancies of the atoms listed. U[iso] is the mean of the U[iso] or Ueqiv of the listed atoms. The atom TYPE is QR, with the given serial number. The original atoms are not deleted, though they should be or their occupancy set to zero. The atom type, QS, should be changed to something appropriate. The DECLINATION and AZIMUTH are computed from the constituent atoms.

LINE newserial atomlist

This creates an 'line' shape from the specified atom list. The centre of the line is at the centre of gravity, the size is twice the mean distance of the given atoms from the centre, and the occupancy is equal to the sum of the occupancies of the atoms listed. U[iso] is the mean of the U[iso] or Ueqiv of the listed atoms. The atom TYPE is QL, with the given serial number. The original atoms are not deleted, though they should be or their occupancy set to zero. The atom type, QS, should be changed to something appropriate. The DECLINATION and AZIMUTH are computed from the constituent atoms.

REFORMAT

This command converts an old (non-FLAG) version of LIST 5 (atomic parameters) to the new format.

Currently, no action is taken by CRYSTALS for special shapes lying on positions of special site symmetry. The user must set up the appropriate constraints in LIST 12 (section #RLIST12#) (usually preventing the refinement of one or more parameters), and set up the correct occupancy in LIST 5 (atomic parameters).

Since the special shapes can coexist with normal atoms, it is possible to embed normal atoms in the special shapes to give a 'peakey' electron distribtion. The sums of the occupancies will need to be constrained.

New Weighting for Fsq refinement

Scheme 14 (Chebychev weighting) has been made more robust to ragged distributions of delta squared. The major influence is in Fsq refinement, though there is also a small (improved) effect on F refinement.

New items stored in LIST 30

LIST 30 (section #RLIST30#) has been extended because of changes in the new CIF dictionary.

Easier handling of twins

Although CRYSTALS has been able to handle twinned data since the program was first designed in the 1970's, the input was very general, and not easily understood. Now that area detector diffractometers are able to routinely handle TLS and TLQS twins, the input has been revised to handle these special sitautions routinely. The old format input, for twins with up to 9 components, is still available (see section #Rtwinning#)

Chapter 2
Definitions And Conventions

2.1 Syntax of Commands

Commands are given to CRYSTALS as small packages, rather like sentences. This enables the program to recognise when the user thinks that a piece of input is complete and then, after inserting any default values and checking for errors, perform the task.

Format of Commands

All command packages follow the same general format. The command is introduced with a backslash (or alternatively, a hash symbol) and ends with the word 'END'.

```
\COMMAND ([keyword=]value ) ...
(DIRECTIVE ([keyword=]value ) ...)
END
```

Items in round brackets '()' may be absent, items in square brackets '[]' are optional. Ellipsis '...' means the preceding item may be repeated.

Actual data on a COMMAND or DIRECTIVE line is input in free format, with at least one space (or sometimes an optional comma) terminating an item. Data items may either be preceded with the keyword and its '=' sign, or if the order given in the definition is strictly followed, just by the data values. COMMANDS, DIRECTIVES and KEYWORDS can be abbreviated to the minimum string which resolves ambiguity. Both types of identification can be intermingled.

The following examples are all identical to the program. They use the command \DISTANCES to compute all interatomic distances in the range 0.0-1.9 Angstrom, and all interatomic angles involving bonds in the range 1.6-2.1 Angstrom.

```
\DISTANCE
SELECT RANGE=LIMITS
LIMITS DMIN=0.0 DMAX=1.9 AMIN=1.6 AMAX=2.1
END
```

```
\DIS
SE RA=LI
LI 0.0 1.9 1.6 2.1
EN
```

```
\DIST
SEL RANGE=LIM
LIMI DMAX=1.9 1.6 2.1
END
```

Note that in the last example, the value for DMIN is omitted (its default value turns out to be 0.0) and the list of parameters starts with a DMAX: CRYSTALS knows that the next parameter is AMIN so it need not be specified.

2.2 Syntax of the manual

Those parts of the manual describing data and command input will generally be in the following format:

```
1. A summary of the complete command, with all directives and
   keywords. Ellipsis (...) may be printed to represent omitted
   but similar parameters.
2. An typical example. This may not demonstrate all the available
   options
3. A description of all the directives and keywords.
```

For a simple example, see LIST 1, section #RLIST01#.

2.3 Types of Commands:

Lists

These contain related data items, grouped together so that CRYSTALS can check that the data is complete. These lists are stored in the CRYSTALS database. Usually, input of a new list of a given type over-writes an existing list of the same type. In general, LISTS do not 'do' anything.

There are two types of syntax for LISTS:

Keyed LISTS

In these, CRYSTALS can know in advance how much and what kind of input to expect. Each element of data is identified by an optional keyword. See, for example, LIST 1 (section #RLIST01#).

Lexical LISTS

In these, CRYSTALS cannot know in advance what kind of data the user may wish to input. Each line of input is processed by a lexical scanner, and parsed to determine the action needed. See, for example, LIST 12 (section #RLIST12#).

For a list of all the lists, see the Lists overview (section #RLISTS#).

Commands
These cause CRYSTALS to 'do' something, for example, compute a Fourier map. There are two type of syntax for commands, similar to those used for LISTS:
   Keyed Commands
   Lexical Commands
Immediate Commands
These are special commands which are acted upon immediately they are issued. They are never more than one line long, and do not require an 'END'. They can be issued whenever the cursor is at the beginning of a line, even inside a COMMAND or LIST. They are not usually involved with the crystallographic calculation, but control some aspect of the way CRYSTALS works, such as hooking in an external data file, or changing the amount of output produced.

Comments
Any data line starting with a backslash or hash followed immediately with a space is ignored, and may thus be used as a comment, or to deactivate the line without deleting it from the file.

Continuation Lines
The directive CONTINUE indicates that the data on the current line is a continuation of the previous line.

2.4 Immediate commands

\FINISH
This command closes down CRYSTALS neatly.
\ ..... COMMENTS ....
This is a comment line. Column 1 contains the \ character and column 2 **must** be left blank. The remaining columns (3-80) may be used for a descriptive comment. Such a comment line may appear at any point in the input.
\TITLE ..... A title to be printed .....
This command changes the title that appears at the start of every operation. The characters \TITLE are terminated by a space in column 7 and the remainder of the line contains the title.
\TYPE 'filename'
The file 'filename' is typed on the users terminal without its contents being interpreted by CRYSTALS. Thus \ commands in this file are NOT acted on, giving the user a method of previewing a USE file.
\USE source
This command controls the source of commands for CRYSTALS. If 'source' is a filename then commands are read from that file. If 'source' is LAST ,the current USE file is closed and commands are read from the previous level USE file. If 'source' is CONTROL , all USE files open are closed and commands are read from the main control stream for the job , for example the terminal in an interactive job.
   One USE file may contain other USE commands. The maximum depth of USE files allowed will be installation dependent. Note that the USEd file need not be a complete list - it can be as little as only one line. An indirect file should end with '\USE LAST' , '\USE CONTROL' , or '\FINISH' . This command is only available in some implementations.
\SCRIPT filename
This command is only available in interactive mode, and passes control to the 'script' file, which tries to assist the user by prompting him for data and information. A separate manual describes the writing of user define scripts.
\SPAWN   'shell command'
A 'shell command' can be issued from inside CRYSTALS with this command. \SPAWN without a command spawns a sub-process and passes control of the sub-process to the terminal. Return to CRYSTALS by closing the sub-process.
$ 'shell command'
A 'shell command' can be issued from inside CRYSTALS with this command. Typical examples are: $dir, $notepad, $edit afile, $netscape something.html

\COMMANDS   command
This command, which takes other command-names as a parameter (without the \ ), produces a listing of the available parameters, keywords and defaults for those commands. The listing is derived directly from the 'command file', and is thus completely up to date for the program being run. This command will not operate correctly if the preceding command ended in error. Clear the error flag by performing some successful operation. The facility is an aide-memoire, and not intended to replace the manual. The full significance of the output is detailed in section #Rcrysdatabase# on the CRYSTALS database.
? text
This facility allows the user to make brief inquiries from the command file on the commands, directives, and parameters available at the current point in the job. If a command is not being processed, and '?' is

entered alone, a list of the commands is displayed. If '? command' is entered, a list of the directives available with that command is displayed. If '? command directive' is entered, a list of parameters for the given command and directive is displayed, and so on.

If a command is currently being processed, the behaviour is similar, but no command name is allowed. Then '?' alone gives a list of directives, while '? directive' gives a list of parameters, and so on. In this case care should be taken: After a '?', CRYSTALS loses track of the last parameter that was input so using CONTINUE will have unpredictable results. To work around this, specify the parameter explicitly on the command line, for example:

```
\EDIT
\  Start entering a new atom to be added to the model:
ATOM U 1.0
\  Refresh your memory as to the rest of the syntax:
?atom
\  Carry on entering the atom, but give the
\ parameter name, X, explicitly.
CONTINUE X=0.2 0.4 0.5
END
```

\MANUAL 'name'
The 'name' parameter is the name of the volume whose index is required. The special name 'INDEX' gives a list of subjects for each volume. The special name 'LISTS' gives a list of the function of each LIST.

\HELP 'topic'
The topic 'HELP' contains a list of topics for which help is given. This is likely to be very site-specific.

\OPEN devicename filename
This command is similar to RELEASE, except that a wider range of device names may be specified, and different messages are produced. An important facility available with this command is to open a named HKLI file, for the reflection input commands HKLI and LIST 6, using the device name HKLI. Similarly permanent files may be used in data reduction work by using the device names M32 and M33 overriding the default scratch files.

```
e.g.
\OPEN HKLI reflection.hkl
```

\CLOSE devicename
Any file on specified device is closed.

\RELEASE devicename filename
The file currently open on 'devicename' is closed, and a new file opened on that device if necessary. The file just closed can be examined using the \TYPE command. The filename parameter is optional. If it is specified, the new file will be opened with that name.

Useful devices currently available include PRINTER, PUNCH, LOG and MONITOR.

\APPEND devicename filename
Output to the specified device is appended to any output already in the specified file.

Device names
Devices are names used inside CRYSTALS to refer to files that it has opened. Deviecs recognised by CRYSTALS are:

DISCFILE - the database containing everything.
HKLI - an hkl file during input using \LIST 6 or \HKLI.
CONTROL - commands being input (can be a file if you type \USE filename, but is usually the command input line)
PRINTER - the listing file (bfilenn.lis by default)
PUNCH - the punch file (bfilenn.pch by default)
LOG - record of all commands issued (bfile.log by default)
MONITOR - obsolete, used to be text that appeared on the screen
SPY - obsolete, used to collect usage data (locally!).
NEWDISC - a second database, open during \PURGE commands.
COMMANDS - the file commands.dsc defines the syntax and data structures of all the commands and lists.
M32 M33 MT1 MT2 MT3 MTE - scratch files
SRQ - system request queue. Some high-level commands may issue other CRYSTALS commands using this file.
FORN1, FORN2 - output of data for 'foreign' (ie. external) programs.
SCPDATA - scripts can read directly from any file opened on this device.

SCPQUEUE - stores commands that scripts are building up.

Not all may be opened/closed by the user from a command prompt. Some are only available inside the initialisation file CRYSTSLS.SRC (See STORE below).

\PAUSE interval

This command causes the program to wait for 'interval' seconds before proceding. The maximum value of 'interval' is 200 seconds. It might be useful in a 'USE' file.

\BENCH nparam nref

This simulates sfls (structure factor least squares, i.e. a cycle of refinement, see #RSFLS#). to enable processor speeds to be compared. No real refinement is done, and the structure is not modified.

```
nparam defaults to 500
nref   defaults to 5000
```

Times for a Microvax 3800 (circa 1995) are printed for comparison.

\SET LIST   state

This command allows the user to control the monitoring level of transfer of lists to and from the database in conjunction with the SET WATCH command. There are four states available.

If state is OFF, no list logging information is produced.

If state is READ or WRITE, list logging information is only produced when lists are read or written respectively.

If state is BOTH, both reading and writing operations may be monitored. Note that the logging operation may be qualified by a list type specified by SET WATCH. The initial state is WRITE, with the specific watch set on list 5 (the model parameters), so that only operations creating or modifying list 5 will be monitored.

\SET WATCH number

This command is used in conjunction with SET LIST to control monitoring of list operations. If number is 0, operations on all list types may be monitored, according to the state set with SET LIST. If number is a positive integer, representing a list type, only operations on that list type may be monitored. The initial value for the list watch is 5, which in conjunction with the initial monitoring state means that operations creating or modifying list 5 will be monitored.

\SET FILE type

This command is used to control the case of file names generated by CRYSTALS. Possible values are:

```
LOWER       Filenames are converted to all lower case.
UPPER       Filenames are converted to all upper case.
MIXED       Filenames are left as input or defined.
```

\SET GENERATE state

This command is used to control the generation of output file names and pseudo-generation numbers on non-VMS systems. By default, CRYSTALS modifies the root of filenames for files which should not be overwritten (normally .LIS, .MON, .LOG).

OFF suppresses automatic name generation.

\SET EXPORT state

If 'state' is 'on' then LISTS 5 (atoms), 12 (constraints) and 16 (restraints) are copied to the PUNCH file when CRYSTALS closes down. These can be archived for safety, or exported to another computer.

\SET UEQUIV state

This controls the calculation of Uequiv. Both definitions are acceptable to Acta. The arithmetic mean of the principle axes is often similar to the refined value of Uiso. The geometric mean is more sensitive to long or short axes, and so is more useful in publications.

```
ARITH = (U1+U2+U3)/3
GEOM  = (U1*U2*U3)**1/3
```

\SET PAUSE value

This command sets a time, in seconds, for which the program will pause at the end of each screen full of output. It is only effective on DOS machines, and enables the user to use the 'pause' key to hold a selected screen. The maximum value of 'interval' is 200 seconds.

\SET LOG state

If state = ON then all user input commands are written to a log file.# If state = OFF then subsequent commands are not written to the log. Any change made to the log state applies only to the current level of USE file and any USE file called by it. Because the log file is a direct copy of the users commands, it may subsequently be used (probably after modification) as a control file.

\SET MONITOR state

If 'state' = ON, then all input is reflected in the monitor file. If 'state' = OFF, monitoring. is suppressed. Any change made to the monitoring state applies only to the current level of USE file and any USE file called by it.

\SET PAGE length

This command is used to change the length of the assumed 'page' when displaying files on the monitor channel, using the commands 'HELP', 'MANUAL', and 'TYPE'. The initial length is 20 lines. After the number of lines specified have been typed, the listing stops and a message indicates the program is waiting. A blank line or carriage return at this point will cause the listing to continue. Any other input is executed normally. If the length is set to zero, or a negative number, the feature is disabled.

\SET TERMINAL device

This command controls the display of SCRIPT menus on some terminals. Possible device types are

```
UNKNOWN  This is the default, and requires no special terminal.
VT52     For use on terminals with limited screen management facilities.
VT100    For use on advanced terminals.
VGA      For use on PC VGA terminals
WIN      For use under Win32 and X-windows.
```

\SET COMMUNICATION speed

This command is used to indicate to the program the speed of the communication line or terminal on which it is being run. This indication is used by some facilities to determine how much output to produce on the monitor channel. The possible values for the speed are "SLOW" and "FAST". These keywords are not associated with any particular terminal speed, and the appropriate value will depend on the user's patience. The initial value is "FAST"

\SET TIME state

This command is used to indicate to the program whether the timing messages usually produced at the end of each facility are produced. If 'state' is "OFF" the messages are not displayed. If 'state' is "ON" the messages are displayed.

\SET PRINTER state

This command is used to control output to the 'printer' file. The state OFF suppresses printer output.

\SET OPENMESSAGE state

This command is used to enable or disable file handling messages. OFF suppresses messages.

\SET MESSAGE state

This command is used to indicate to the program whether the command messages usually produced at the end of each facility are produced. If 'state' is "OFF" the messages are not displayed. If 'state' is "ON" the messages are displayed. Error messages are always displayed.

\SET SRQ state

This command is used to control mirroring of CRYSTALS internal commands. The normal state OFF suppresses the mirroring.

2.5 Files

When CRYSTALS runs it stores all crystallographic data in a single file, by default named crfilev2.dsc. This is a binary file and should not be opened with any other program.

CRYSTALS outputs results and analysis to several places:

The listing file:
this will be called bfilenn.lis where nn is a number from 00-99, incremented each time the program is run. When nn reaches 99 this file will be continuously overwritten every time CRYSTALS is run.# The listing file contains verbose output about all the calculations that take place.

The punch file:
called bfile.pch by default. The idea of 'punching' data is a throwback to the days of punched cards. This file is used to write out specific bits of data *as commands* that can then be read back into the program. For example, the parameters defining the crystallographic model are stored in CRYSTALS as a 'List 5'. Then 'Punching List 5':

```
\PUNCH 5
END
```

will result in a full \LIST 5 command being written to the file bfile.pch. You can then recover that version of the model parameters by typing

```
\USE bfile.pch
```

which causes the commands written in that file to be executed by CRYSTALS.

The monitor file:
called bfilenn.mon this is obsolete. If there is any output in it, it is from a bit of the program that hasn't been looked at recently.

The log file:
called bfilenn.log, this contains everything that you typed into CRYSTALS, and commands that were issued on your behalf by the menu system or script processor.

2.6 Errors and Warnings

CRYSTALS recognises the following run time error categories, in addition to any detected by the operating system.

Warnings

These will occur only for tasks which produce user-readable output, and do not write to the database. The current task is abandoned if necessary, and the next task fetched from the input stream.

Errors

The error is such that the current task must be abandoned. In batch or online modes, the job will be terminated as well. In interactive mode the current task is abandoned , and control is returned to the user at terminal. The processing of any 'USE' file will be abandoned.

Severe Errors

The error detected is such that it is not possible for the job to continue. These errors are usually caused by database management failures.

Catastrophic Errors.

The job is completely terminated. This is usualy the consequence of errors in reading or writing to the database.

Programming Errors

The program has detected an inconsistency either in the code, or in the command file. A dump and error report will be generated if possible. The error and the conditions that cause it should be reported to Oxford.

Errors Detected During the Creation of a LIST

During all operations that create new versions of a list, either by input or internally, errors may be found that cause the new list not to be written to the database. To prevent the system using an old version of a list when the creation of the latest version has failed, the relevant list type is marked as an 'error list' in the 'list control table' (see below). This error flag is cleared when a new version of the list is created or by user action.

If a list which is marked as an error list is accessed, a message will be output, and the job terminated. However, for the printing of most lists, the error flags are not checked.

Chapter 3

The Crystals Database

It is quite possible to use CRYSTALS so that all the data is read from text files at the start of every job (as in SHELXL). However, for interactive working, it is preferable to establsh a database of crystallographic information which can be used whenever it is needed. This database is called the CRYSTALS 'disk' file. It is usualy a permanent file on the hard disk, and usually has the extension '.DSC'.

It is a direct-access binary file. DO NOT try to print or edit it.

external user-input lists. In modern terminology, these 'lists' would be called 'objects' or 'data structures'. Each list groups together related information, e.g. the cell parameters, the atoms, the reflections etc.
While in general input of a list will overwrite a previous version, for the atomic coordinate list (LIST 5, see section #RLIST05#) a new version is added to the database. In the event that a refinement goes wrong, the user can usually recover an earlier version of the structure.

In the CRYSTALS system, most of the data required to refine a crystal structure must be input to the computer as ASCII, translated into an internal format and stored on a random access disk file. On the disk, different types of data are recorded separately, in what are called LISTS. Each list holds only one type of crystallographic information and is identified by a number called the 'list type number'.

Normally, each structure uses a separate disk file, which is preserved between different jobs and updated whenever a program generates some new data. This means that several different versions of a given list may be produced during the course of a structure analysis. In order that each list may be uniquely identified, every list has associated with it a second number, called the 'list serial number'. To specify unambiguously a list that is stored in the database, it is necessary to know both the relevant list type number and the list serial number.

In most cases, however, the version of a given list that is required is the latest list of that type to be created. Accordingly, an index called the 'current list index' is maintained, which contains an entry for the latest version of each list. When a program requires information about the current version of a list, it accesses the current list index.

As well as an index of the current version of each list, a second index is kept, called the 'file index'. For each disk file, this latter index contains the information for every list that is present in the database. This index is always checked when a list is written to the database, to ensure that its list type number and list serial number refer to only one list.

For a user, the major advantage of splitting the input data into logical units in the database is that, for any run, only those lists which need to be changed have to be re-input to the database. New versions of each list that are generated by programs are automatically output to the database so that, provided the database file is not destroyed or erased, it is probable that each run will contain only control commands. For example, during a structure factor least squares calculation a new set of Fcalcs and a new normal matrix will be generated and stored in the database ready for further calculations.

Old versions of a list may be reused, provided that they have not been overwritten or deleted, by finding the relevant entry in the file index and copying it into the current list index.

The database contains two indices which control access to it.

Current List Index
This index contains the serial numbers of each list to be used in calculations. This index is updated whenever a list is written to database.

Disk Index
This is an index of all the lists contained in the database. When a new list is added to the database, its internal address is added to this index, and also inserted into the current disk index. It is possible to copy address from this index to the current disk index, thus changing the currently active version of a list.

3.1 General List-control Directives - \DISK

The list control table can be marked and used in various ways with this command.

```
\DISK
PRINT INDEX= PUNCH= LIST=
MARKERROR LIST= SERIAL= RELATIVE= ACTION=
RETAIN LIST= SERIAL= RELATIVE= ACTION=
DELETE LIST= SERIAL= RELATIVE= ACTION=
RESET LIST= SERIAL= RELATIVE=
USAGE LIST= SERIAL= RELATIVE= FLAG=
EXTEND RECORDS= FREE= TRIES= SIZE=
CHECK
END
```

For example:

```
\DISK
\  Print the current list index
PRINT INDEX=CURRENT
```

```
\ Print the index containing all the lists stored
\ on the disk
PRINT INDEX=DISK
\ Reset LIST 5 (the model parameters) to the one with serial number 4
RESET 5 4
\ Reset LIST 10 (Fourier peaks) to the 'current serial number - 1'
RESET 10 0 -1
\ Retain LIST 5 with serial number 6 when the disk
\ is purged
RETAIN 5 6
\ Delete current LIST 11 (normal/inverted least squares matrix)
DELETE 11
END
```

\DISK

This is the command which initiates the routines to modify the list control table. The directives MARKERROR, DELETE, RETAIN, USAGE, and RESET all accept the parameters SERIAL and RELATIVE. The parameters SERIAL and RELATIVE should not both be changed from their default settings on the same directive.

All directives are executed immediately after they have been entered.

PRINT INDEX=

This directive causes one of the indexes to be printed.

INDEX=

```
CURRENT  - default value
DISK
SUMMARY
```

If 'INDEX' is 'CURRENT', the current list index is printed (i.e. the index containing the current version of each list stored). If 'INDEX' is 'DISK', the index containing information about all the lists stored on the disk is printed. If 'INDEX' is 'SUMMARY', a summary of the usage of the disk file is printed.

PUNCH=

```
NO  - default value
YES
```

Writes a machine readable summary of whatever is selected by INDEX to the currently open PUNCH file.

LIST=

```
0 - default value
n
```

By default (0) all lists are printed or punched. If you specify an alternative value for the LIST parameter, then only information about that list number is output.

MARKERROR LIST= SERIAL= RELATIVE= ACTION=

This directive can either mark a specified type of list as an 'error list', or alternatively, such a mark can be removed from the list control table.

LIST=n

'n' is the list type number (there is no default value).

SERIAL=n

'n' is the list serial number. The default value is zero, which represents the serial number of the current list of this type.

RELATIVE=n

'n' is the offset applied to the serial. The default value is 0. If both SERIAL and RELATIVE are zero (the defaults) the current list of the specified type will be the one that is marked.

ACTION=

```
NO
YES  - default value
```

If ACTION is NO , the entry in the list control table for the list type specified is altered so that it is not marked as an 'error list'. If ACTION is YES, the entry in the list control table for the list type specified is altered so that it is marked as an 'error list'. If a program attempts to use such a list, an error is reported, and the job terminated.

RETAIN LIST= SERIAL= RELATIVE= ACTION=

With this directive, certain old versions of specified lists can be retained when the disk file is PURGED (see #RPURGE#).

LIST=n

'n' is the list type number (there is no default value).

SERIAL=n

See markerror definition above (#RMARKERROR#).
RELATIVE=
See markerror definition above (#RMARKERROR#).
ACTION=

NO
YES - default value

If 'ACTION' is 'YES', the list with the specified type and serial number will be retained when the disk file is PURGED. If 'ACTION' is 'NO' , the particular list will be deleted when the disk file is PURGEd (see section #RPURGE#).

DELETE LIST= SERIAL= RELATIVE= ACTION=
With this directive certain lists can be deleted from the file index.
LIST=
See markerror definition above (#RMARKERROR#).
SERIAL=
See markerror definition above (#RMARKERROR#).
RELATIVE=
See markerror definition above (#RMARKERROR#).
ACTION=

NO
YES - default value

If 'ACTION' is 'YES', the list with the specified serial number will be marked to be deleted from the file when the file is PURGED. If ACTION is 'NO' , the specified list will not be marked as one to be deleted when the disk file is PURGEd (see section #RPURGE#).

RESET LIST= SERIAL= RELATIVE=
This directive alters the entry for the list type in the current list index.
LIST=
This parameter is the list type number, for which there is no default value.
SERIAL=
See markerror definition above (#RMARKERROR#).
RELATIVE=
See markerror definition above (#RMARKERROR#).

USAGE LIST= SERIAL= RELATIVE= FLAG=
This directive alters the list write/overwrite flag.
LIST=n
'n' is the list type number (there is no default value).
SERIAL=
See markerror definition above (#RMARKERROR#).
RELATIVE=
See markerror definition above (#RMARKERROR#).
FLAG=

OVERWRITE
READY
UPDATE - default value.

EXTEND RECORDS= FREE= TRIES= SIZE=
This directive allows the user to extend the database by a specified number of records, and to control the auto-extension. On modern platforms the extension of the database is automatic by default.
RECORDS=
This parameter specifies a number of records to be added to the file. The default value is zero i.e. no records are added.
FREE=
This parameter specifies a number of records that must be available for use in the file. The file will be extended until there are at least 'FREE' records unused after the next free disk address.
TRIES=
This is the number of times the system may try adding 'SIZE' records to the disk file to achieve enough space for the current operation. Usual default is 1.
SIZE=
This is the size, in records, that the system will increase the disk by to try to accommodate the current operation. The usual default is 5. If the write still fails after TRIES x SIZE records have been added it produces an error. Setting SIZE to zero enables uncontrolled extension of the disk file. This is the default, but if the user gets carried away doing crystallography, they may fill their disk.

CHECK
This directive checks the integrity of the disk file. It takes no parameters.

Errors in the DISK file.

If CRYSTALS reports errors in the disk file, run this utility to get a list of those LISTS corrupted. Use \PUNCH 5, \PUNCH 12 and \PUNCH 16 to make ASCII copies of these important lists (by default output to the 'punch' file, bfile.pch - rename it after punching to e.g. 'savedlists.pch'). If the final version of a list is corrupt, use \DISK RESET (see above #RRESET#) to point the program to an earlier version and punch that instead. If the disk file becomes totally unusable, delete it, read in the reflections again, and then read in these 'punched' lists (\USE savedlists.pch).# If only certain lists are unusable, use the DELETE directive in \DISK to mark defective LISTS for deletion, and then use \PURGE NEW to create a new disk file (by default new.dsc). Rename new.dsc to crfilev2.dsc to make CRYSTALS open it by default when it starts in that directory.

The following lists can always safely be deleted since CRYSTALS creates new ones automatically.

11 Normal matrix (section #RLIST11#).
20 Saved geometrical transformation matrices (section #RLIST20#).
22 Constraints in internal format (section #RLIST22#).
24 Least squares shift list (section #RLIST24#).
26 Restraints in internal format (section #RLIST26#).
33 SFLS command in internal format (section #RSFLS#).
36 List dependency tracking (not documented).

See section #RLISTS# for all the list definitions.

Example 1

Imagine that the current versions of LIST 5 (the model parameters) and LIST 29 (element properties) have beome corrupt. We will also delete any recreatable lists.

```
\DISK
\  Check the index
CHECK
\  Delete current version of atomic params
DEL 5
\  Point index to the previous version
RESET 5 0 -1
\  Remove corrupt and safe-to-delete lists:
DEL 29
DEL 10
DEL 11
DEL 20
DEL 22
DEL 24
DEL 26
DEL 33
DEL 36
END
\  Open a new file save.dat on the 'PUNCH device'.
\  Release is equivalent to CLOSE followed by OPEN.
\RELEASE PUNCH SAVE.DAT
\  Output the important data lists:
\PUNCH 5
END
\PUNCH 12
END
\PUNCH 16
END
\  Write a new database called rescue.dsc with only the
\  current index of lists in it:
\PURGE RESCUE
\  Close CRYSTALS
\FINISH
```

At this point it would be useful on a stand-alone computer to run a system utility (e.g. SCANDISK on a PC) to check the integrity of the computers hard disk, since CRYSTALS rarely corrupts the disk itself.

One would then open a command prompt in the working folder and do this:

```
del crfilev2.dsc
ren rescue.dsc crfilev2.dsc
```

Now restart CRYSTALS in the same folder. Recover the important saved data by typing:

```
\USE save.dat
```

3.2 Elimination of Old Versions of LISTS - \PURGE

```
\PURGE FILE= INITIALSIZE= LOG= LIST=
END
```

\PURGE
   This command deletes all but the current version of each list, and any lists explicitly marked to be
   deleted, except for lists marked to be retained, and then rewrites the disk file so that the space occupied by
   the old deleted lists is overwritten. This operation does not normally shorten the file physically (see
   parameter FILE below for a method of doing this) , but frees a lot of space that can be reused. The
   optional parameters 'FILE' and 'INITIALSIZE' (available in some implementations only), allow the user
   to create a new file with only the current version of each list in it. This file may be smaller than the
   original disk file.
   FILE=

```
OLD - default value
name - The root to be used form the new database - name.DSC
DATE - The root name is generated by CRYSTALS from the date and
       time.
```

   This parameter controls whether a new file is created. If the value specified is not OLD, a new file
   will be created containing only the current versions of each list, and any which are marked to be
   retained. The program automatically extends the file to the size required for all the lists to be
   retained. This new file has the extension .DSC, and may be used in future CRYSTALS tasks.
   INITIALSIZE=
   This parameter is used to specify an initial size for a new file. If a new file is not to be created, the
   value is ignored. The default value of zero causes the new file to have the smallest size necessary to
   contain all the copied lists.
   LOG=

```
OFF - default value
ON
```

   If the value 'ON' is given for this parameter, the types and serial numbers of all lists deleted
   because they were old versions are listed. The types and serials of all lists not copied for other
   reasons, e.g. deleted lists, and lists which are marked as being updated, are always listed
   LIST = n
   'n' is the type of list to be purged from the database, all old versions of all other lists are preserved.
   If 'n' is 0 (the default value) lists of all types are purged.

3.3 LISTS in Current use

Lists marked * cannot be directly input by the User
List Number Type of data

1 Cell parameters (section #RLIST01#)
2 Unit cell symmetry (section #RLIST02#)
3 Atomic scattering factors (section #RLIST03#)
4 Weighting parameters (section #RLIST04#)
5 The model parameters (section #RLIST05#)
6 Reflection data (section #RLIST06#)
7 Reflection data not used for refinement (section #RLIST07#)
10 Peak coordinates from Fourier (section #RLIST10#)
11 Least squares matrix (section #RLIST11#)
12 Refinement directives (section #RLIST12#)
13 Crystal and collection data (section #RLIST13#)
14 Fourier directives (section #RLIST14#)
16 General Restraints (section #RLIST16#)
17 Special Restraints (section #RLIST17#)
20* Transformation matrices saved by \GEOMETRY (section #RLIST20#)
22* Refinement directives in internal format (section #RLIST22#)
23 Structure factor control list (section #RLIST23#)
24* Least squares shift list (section #RLIST24#)
25 Twin component operators (section #RLIST25#)
26* Constraints in internal format (section #RLIST26#)
27 Diffractometer scales (section #RLIST27#)
28 Reflection condition/filter list (section #RLIST28#)
29 Contents of asymmetric unit and elemental properties (section #RLIST2
30 General information (section #RLIST30#)
31 Cell parameter E.S.D.'s (section #RLIST31#)

33* Internal - Refinement control (last SFLS command, see #RSFLS#)
36* Tracking interdependencies of parameters, normal matrix, weights etc.
40 Bond forming/breaking directives (section #RLIST40#)
41* Bonds between atoms (section #RLIST41#)

3.4 Printed Summary of Data lists

```
\SUMMARY OF= TYPE= LEVEL=
END
```

For example:

```
\  Detailed list of model parameters:
\SUMMARY OF=LIST TYPE=5 LEVEL=HIGH
END


\  Again, but less typing:
\SUM L 5 HI
END


\  Summary of reflection data:
\SUM L 6
END
```

\SUMMARY OF= TYPE= LEVEL=

OF=
This parameter determines the extent of the data summary.

```
LIST
EVERYTHING
```

If 'LIST' is specified, a summary of a specific list whose type is given by the 'TYPE' parameter is produced. If 'EVERYTHING' is specified, a summary of all current lists for which summaries are available is produced. If 'OF = EVERYTHING' is specified, the value of 'TYPE' is ignored.
TYPE=
This parameter indicates the list type for which a summary is required. If a summary of this list type is not available, a warning message will be printed.

LEVEL=

```
LOW
MEDIUM
HIGH
```

This parameter has three values which indicate the level of detail required in the data summary. The parameter is ignored unless the list type is 5 or 10.

```
List       Level    Data printed
----       -----    ---- -------
5 or 10  LOW      Atom names
5 or 10  MEDIUM   Atom names and positional parameters
5 or 10  HIGH     All atomic and overall parameters
```

3.5 Element and Atom names

Element Names
Elements are specified by a name, called the atom **TYPE.** The 'TYPE' is used to associate the refined atoms with atomic properties, such as form factors, radii, etc. The TYPE **must** start with a letter, and is not case sensitive. REMEMBER that 'blank' is used as a delimiter in CRYSTALS and so must not appear in an atom TYPE. Elements in the form factor list (LIST 3 - see section #RLIST03#) and the atomic properties (LIST 29 - section #RLIST29#) are associated by their TYPE with atoms in the parameter list (LIST 5). Numeric properties of the elements are pre-prepared in the files SCATT, SCATCU, SCATMO and PROPERTI(.dat) in the SCRIPT directory (Often CRYSTALS\SCRIPT). The 'elements' C', C", H' and H" are in these files with the properties of C and H. You can define your own element names, but may not use +,- or *.

Atom names
Individual atoms are specified by combining the TYPE with a SERIAL number. This is an integer in the range 1-9999. When combined with the TYPE, it must be enclosed in parentheses ().

```
e.g.   C(2) CL(123)
```

Symmetry equivalent atoms
Atoms specified as above correspond exactly to those in the refinable atom list (LIST 5). If a symmetry equivalent atom is required, up to 5 additional items are included inside the parentheses. The full specification of an atom is:

        TYPE(SERIAL,S,L,Tx,Ty,Tz)

S
is the serial number of the symmetry operator in the stored list of space group operators. If negative, it indicates that the atom coordinates are first negated before being operated upon. The default is 1

L
is the serial number of the non-centring lattice translation to be used. The default is 1.

Tx,Ty,Yz
are whole cell translations parallel to the cell axes. The default is 0.
   If the value of an item is its default, it may be omitted altogether, though its place must be marked by its associated comma. A series of trailing commas may also be omitted. For example:

        C(99,-1,1,0,0,0) - an atom related by inversion at the origin
        *   (assuming x,y,z is the first operator).*
        C(99,-1,1,,,0)   - same atom as above, omitting defaults.
        C(99,-1)   - same atom as above, omitting
        *   defaults and trailing commas.*

   For more detailed information, see Atomic and Structural parameters (section #Ratomparams#)

3.6 Foreign Program Links

\FOREIGN
This command provides links to 'FOREIGN' programs, that is, programs which are not part of CRYSTALS, but which provide useful functions in providing a complete system. These programs often come from other laboratories, and are only distributed with the authors permission unless they are public domain. Where practical, we make no changes to the original code. We offer little or no support in connection with these programs, though usually they are in frequent use in Oxford. The linking routine prepares data files for the foreign program, and may initiate a subprocess to execute the rogram.
   PROGRAM=

        SNOOPI - A plotting program using Tektronix 4010 devices.
        CAMERON - An interactive graohics program.
        MULTAN - Prepares files for MULTAN 84.
        SHELXS - Prepares files for SHELXS.
        SIRxx   - Prepares files for the SIR direct methods system

   MODE=
   This keyword controls the mode of operation of the foreign program. It currently only applies to SHELXS and SIR.

        NORMAL   - DEFAULT, prepares a default data file with recommended
        *   settings.*
        DIFFICULT   - Prepares a file with non-default settings.
        PATTERSON   - Prepares a default Patterson calculation (SHELXS only).
        SPECIAL   - Prepares a full SHELX data file (SHELX only)

Chapter 4
Initial Data Input

4.1 Scope of the Initial Data Input section.

The areas covered are:

```
Abbreviated startup command                QUICKSTART
Input of the cell parameters               LIST 1
Input of the unit cell parameter errors    LIST 31
Input of the space group symmetry information   SPACEGROUP
Alternative input of the symmetry information   LIST 2
Input of molecular contents                COMPOSITION
Input of the atomic scattering factors     LIST 3
Input of the contents of the unit cell     LIST 29
Input of the crystal and data collection details LIST 13
Input of general crystallographic data     LIST 30
```

4.2 Abbreviated startup command - QUICKSTART

The command QUICKSTART is provided to assist in migration from other systems to CRYSTALS. It requires that data reduction (section #RDATAREDUC#) has already been done or that a simple 4-circle Lp correction be suitable, and that the reflection data are available in a fixed format file with one reflection per line. This command expands the given data into standard CRYSTALS lists, as described elsewhere in the manuals. The user is free to overwrite LISTS created by QUICKSTART by entering new LISTS manually.

```
\QUICKSTART
SPACEGROUP SYMBOL=
CONTENTS FORMULA=
FILE NAME=
FORMAT EXPRESSION=
DATA WAVELENGTH= REFLECTIONS= RATIO=
CELL  A= B= C= ALPHA= BETA= GAMMA=
END
```

For example:

```
\QUICKSTART
SPACEGROUP P 21/n
CONTENT C 6 H 4 N O 2 CL
FILE CRDIR:REFLECT.DAT
FORMAT (3F3.0, 2X, 2F8.2)
DATA 1.5418
CELL 10.2 12.56 4.1 BETA=113.7
END
```

\QUICKSTART
None of the directives may be omitted, though some parameters do have default values. **CONTINUE directives may not be used.**

SPACEGROUP SYMBOL=
This directive generates symmetry information from the spacegroup symbol. The syntax is exactly as describe for the command SPACEGROUP, given in section #RSPACEGROUP#.
SYMBOL=
There is no default for the symbol, it should be a valid H-M space group symbol, e.g. 'P 21 21 21' or 'P 21/c' or 'I -4 3 m'. Use spaces to separate each of the operators.

CONTENTS FORMULA=
This directive takes the contents of the UNIT CELL (cf LIST 29 - section #RLIST29#) and generates scattering factors (LIST 3 - section #RLIST03#) and elemental properties (LIST 29 - section #RLIST29#).
FORMULA=
The formula for the UNIT CELL contents **(NOT ASYMMETRIC UNIT** - for compatibility with SIR92) is given as a list with entries of the type

```
'element name' 'number of atoms'
```

```
e.g. CONTENT FORMULA = C  24  H  36  O  8  N  4
```

The items in the list **must** be separated by at least one space. The number of atoms may be fractional or, if omitted, they default to 1.0.

FILE NAME=
This directive associates the file containing the reflections with the program. The special name *'COMMANDS'* causes reflection data to be read from the command stream. The reflections MUST then be terminated with an 'h' value of -512, otherwise the end-of-file is sufficient.
NAME=
The name of the file containing the reflections. The syntax of the name must conform to the computers operating system. See the **IMMEDIATE** command \SET FILE for case sensitive systems.

FORMAT EXPRESSION=
This directive controls the reading of the reflection list. The reflection file must contain the following items in the order given. Only one reflection is permitted per line. See \LIST 6 for more flexible input (section #RLIST06#)

```
    h k l F and optionally sigma(F)
```

F and sigma(F) may be replaced by I or F-squared.
EXPRESSION=
The expression is a normal FORTRAN format expression, **including the open and close parentheses.** The descriptor 'nX' may be used to skip unwanted columns. The indices may be I or F format. There is no default expression.

DATA WAVELENGTH= REFLECTIONS= RATIO=

WAVELENGTH=
The wavelength, in Angstroms, used in selecting elemental properties. The default is 0.7107 (Molybdenum K-alpha radiation).
REFLECTIONS=
A keyword to indicate whether the input data is F, F-squared or I.

```
FOBS      -  Default, indicating F values being input.
FSQUARED  -  Indicating F squared values being input.
I         -  Indicating intensity values being input.
```

If REFLECTIONS equals I, then an Lp correction is done assuming four circle geometry. Note that the reflections from modern diffractometers are unlikely to be stored as FOBS. Some old X-ray data and neutron data may still be given as FOBS.
RATIO=
The minimum ratio of I/sigma(I) to be used in selecting reflections. Default is 3.0

CELL A= B= C= ALPHA= BETA= GAMMA=
The real cell parameters. The angles default to 90.0 degrees.

4.3 Input of the cell parameters - LIST 1

Either the real cell parameters or the reciprocal cell parameters may be input and the three angles be given in degrees or as their cosines. A mixed form, containing both angles and cosines is not allowed.

```
\LIST 1
REAL A= B= C= ALPHA= BETA= GAMMA=
END
```

For example

```
\LIST 1
REAL 14.6 14.6 23.7 GAMMA=120
END
```

\LIST 1

REAL A= B= C= ALPHA= BETA= GAMMA=
This directive introduces the real cell parameters. If this directive is present, the directive RECIPROCAL will lead to an input error, and no new LIST 1 will be generated.
A=, B=, C=
These parameter are the real cell lengths along the A, B and C axes. There are no default values.
ALPHA=, BETA=, GAMMA=
These parameters give the real cell angles or their cosines. The default value is 90 degrees.

RECIPROCAL A*= B*= C*= ALPHA*= BETA*= GAMMA*=

> This directive introduces the reciprocal cell parameters. If this directive is present, the directive REAL will lead to an input error, and no new LIST 1 will be generated.
> **A*=, B*=, C*=**
>> These parameters are the reciprocal cell lengths.
> **ALPHA*=, BETA*=, GAMMA*=**
>> These parameters give the reciprocal cell angles or their cosines. The default value is 90 degrees.

4.4 Printing the cell parameters

\PRINT 1
> This command lists the cell parameters, and all the other information derived from them which is stored in LIST 1. The inter-axial angles are stored in radians in LIST 1, and printed as such.
>> There is no command to punch LIST 1.

4.5 Input of the unit cell parameter errors - LIST 31

This list contains the variance-covariance matrix of the unit cell parameters. The input consists of a multiplier which is applied to all input parameters, followed by the upper triangle of the variance-covariance matrix (21 Numbers). The units for the angles **MUST** be radians and those for the cell lengths are Angstroms.

```
\LIST 31
AMULT VALUE=
MATRIX V(11)= V(12)= .. V(16)= .. V(22)= .. V(26)= .. V(66)=
END
```

For example

```
\LIST 31
\ the values of the input matrix are to be multiplied
\ by 0.000001
AMULT 0.000001
\ the cell is trigonal,
\ with errors of 0.002 along 'a' and 'b', and 0.004 along 'c'
MATRIX 4 4 1 0 0 0
CONT      4 1 0 0 0
CONT       16 0 0 0
CONT          0 0 0
CONT            0 0
CONT              0
END
```

\LIST 31

> **AMULT VALUE=**
>> This directive gives the value by which all the subsequent terms are to be multiplied, and has a default of 1.0.
>> **VALUE=**

> **MATRIX V(11)= V(12)= . . V(16)= V(22)= . . V(66)=**
>> This directive is used to read in the variance-covariance matrix.
>>> If you only have the parameter e.s.d's, input the square of these for V(11), V(22) etc.
>> **V(11)= V(12)= . . V(16)= V(22)= . .V(66)=**
>>> V(11) is the variance of A , V(12) is the covariance of A and B , V(16) is the covariance of A and GAMMA , V(22) is the variance of B , and V(66) is the variance of GAMMA . The default values for V(11), V(22) and V(33) correspond to axis e.s.d's of .001 A, V(44), V(55) and V(66) to angle e.s.d's of .01 degree.

4.6 Printing the cell variance-covariance matrix

\PRINT 31
> This prints list 31. There is no command for punching LIST 31.

4.7 Space Group input - \SPACEGROUP

The spacegroup symbol interpretation routines in CRYSTALS are derived from subroutines developed by Allen C. Larson and Eric Gabe. It is distributed with their permission. Standard CRYSTALS command input, error handling, data storage, and output has been added to the basic routines. In addition a more flexible method of specifying the unique axis in a monoclinic spacegroup is used. The routine generates a LIST 2 (symmetry information - section #RLIST02#), and a LIST 14 (Fourier and Patterson asymmetric unit

limits - section #RLIST14#).

```
\SPACEGROUP
SYMBOL EXPRESSION=
AXIS UNIQUE=
END
```

For example

```
\ Input the symbol for a cubic spacegroup
\SPACEGROUP
SYMBOL F d 3 m
END
```

```
\ Input the symbol for a common monoclinic spacegroup
\SPACEGROUP
SYMBOL P 21/c
END
```

```
\ Input the symbol for a triclinic spacegroup
\SPACEGROUP
SYMBOL P -1
END
```

\SPACEGROUP

SYMBOL EXPRESSION=
This directive is used to specify the space group symbol.
EXPRESSION=
The value of this parameter is the text making up the spacegroup symbol. At least one space character should appear between each of the axis symbols in the spacegroup symbol. e.g.

```
Use  P 21 3 rather than P 213, P2 1 3, or P2 13
```

Failure to put spaces in the correct place in the symbol will lead to misinterpretation.
Rhombohedral cells are always assumed to be on hexagonal indexing.

AXIS UNIQUE=
This directive specifies the unique axis orientation for monoclinic spacegroups where the symbol specified contains only one axis symbol (short symbol). In other cases any information specified with this directive is ignored.
UNIQUE=

```
A
B
C
GENERATE - the default value.
```

When UNIQUE has the value A, B, or C the program uses the 'a', 'b', or 'c' axis respectively as the unique axis. When UNIQUE has the value GENERATE, the program will attempt to select the unique axis on the basis of the cell parameters currently stored in LIST 1. If this is not possible, because the angles in LIST 1 are all close too 90 degrees or there is no valid cell parameter information, the program will assume that the unique axis is 'b'.

Further examples.

```
\LIST 1
REAL 10.2 11.3 14.1 88.3 90 90
END
\ Input symmetry - the program will  automatically select 'a' as the
\ unique axis based on the cell parameters.
\SPACEGROUP
SYMBOL P 21/M
END
```

```
\ Explicitly specify 'c' unique by giving the full symbol.
\SPACEGROUP
SYMBOL P 1 1 21/M
END
\
\ Explicitly specify 'c' unique by using the UNIQUE parameter.
\SPACEGROUP
SYMBOL P 21/M
AXIS UNIQUE=C
END
```

4.8 Input of the symmetry data - LIST 2

The result of inputting a \SPACEGROUP command (section #RSPACEGROUP#) is the automatic generation of a 'LIST 2' containing the explicit symmetry operators and other information that defines the spacegroup.

Direct input of this list enables the user to specify explicitly the symmetry operators to be used. The advantage of this is that they need not comply to any standard convention - the only check made by the program is to ensure that the determinant is not zero. For example, this technique may be used to enter a set of symmetry operators that contains a translation of a half along an axis - normally that cell length would be halved instead, but it may be useful in order to work consistently with a structure that undergoes a cell-doubling phase transition.

```
\LIST 2
CELL NSYMMETRIES=  LATTICE=  CENTRIC=
SYMMETRY  X=   Y=   Z=
SPACEGROUP LATTICE= A-AXIS= B-AXIS= C-AXIS=
CLASS NAME=
END
```

For example:

```
\ the space group is B2/b
\LIST 2
CELL NSYM= 2, LATTICE = B
SYM X, Y, Z
SYM X, Y + 1/2,  - Z
SPACEGROUP B 1 1 2/B
CLASS MONOCLINIC
END
```

The CELL directive defines the Bravais lattice type, the number of equivalent positions to be input, and whether the cell is centric or acentric. The equivalent positions are defined by SYMMETRY directives, which contain one equivalent position each, and must follow the CELL directive. The equivalent positions input should not include those related by a centre of symmetry if the lattice is defined as centric, and should not include those related by non-primitive lattice translations if the correct Bravais lattice type is given. Positions generated by the last two operations are computed by the system. The unit matrix, defining x, y, z, **MUST ALWAYS** be input. If a centric cell is used in a setting which does not place the centre at the origin, then ALL the operators must be given and the cell be treated as non-centric. This will of course increase the time for structure factor calculations.

Rhombohedral cells can be treated in two ways. If used with rhombohedral indexing (a=b=c, alpha=beta=gamma), the lattice type is P, primitive. If used with hexagonal indexing, the lattice type is R.

\LIST 2

CELL NSYMMETRIES= LATTICE= CENTRIC=
NSYMMETRIES=
This defines the number of SYMMETRY directives that are to follow. There is no default.
LATTICE=
This defines the Bravais lattice type, and must take one of the following values :

```
P  -  Default value.
I
R
F
A
B
C
```

CENTRIC=
This parameter defines whether the cell is centric or acentric, and must take one of the values :

```
NO
YES  -  The default value.
```

SYMMETRY X= Y= Z=
This directive is repeated NSYMMETRIES times, and each separate occurrence defines one equivalent position in the unit cell. The parameter keywords X , Y and Z are normally omitted on this directive, and the equivalent position typed up exactly as given in international tables. The expressions may contain any of the following :

```
+X or -X
+Y or -Y
+Z or -Z
+ or - a fractional shift.
```

The fractional shift may be represented by one number divided by another (e.g. 1/2 or 1/3) or by a true fraction (0.5 or 0.33333...). Apart from terminating text, spaces are optional and ignored. The terms for the new x, y and z must be separated by a comma (,) , and the whole expression may be terminated by ; if required.

SPACEGROUP LATTICE= A-AXIS= B-AXIS= C-AXIS=
This directive inputs the space group symbol, and is optional for the correct working of CRYSTALS. However, some foreign programs need the symbol as input data, and they will extract it from this record. The keywords LATTICE, A-AXIS etc are normally omitted, and the full space group symbol given with spaces between the operators, e.g.

```
SPACEGROUP P 1 21/C 1
```

CLASS NAME=
This directive inputs the crystal class. It is not used by CRYSTALS, but is required for cif files.

4.9 Printing the symmetry information

\PRINT 2
This prints LIST 2. There is no command for punching LIST 2.

Further examples.

```
\ THE SPACE GROUP IS P1-BAR.
\LIST 2
CELL NSYM= 1
SYM X, Y, Z
SPACEGROUP P -1
END


\ THE SPACE GROUP IS P 321
\LIST 2
CELL CENTRIC= NO, NSYM= 6
SYM X, Y, Z
SYM -Y, X-Y, Z
SYM Y-X, -X, Z
SYM Y, X, -Z
SYM -X, Y-X, -Z
SYM X-Y, -Y, -Z
END
```

```
\ THE SPACE GROUP IS P 6122 (note alternative notation for fractions)
\LIST 2
CELL NSYM= 12, CENTRIC= NO
SYM X,Y,Z
SYM -X    ,    -Y  ,Z+.5
SYM +Y, +X,1/3-Z
SYM -Y,-X,5/6-Z
SYM -Y, X-Y, .333333333+Z
SYM Y, Y-X, Z+10/12
SYM -X, Y-X, 4/6-Z
SYM X, X-Y, 1/6-Z
SYM Y-X, -X, Z+4/6
SYM  X-Y, X, Z+1/6
SYM X-Y, -Y, -Z
SYM Y-X, Y ,   -Z+.5
SPACEGROUP P 61 2 2
END
```

4.10 Input of molecular composition \COMPOSITION

This command takes the contents of the asymmetric unit, searches the specified data files for required values, and then internally creates normal scattering factors (LIST 3 - section #RLIST03#) and elemental properties (LIST 29 - section #RLIST29#). **NOTE LISTS 1** (see #RLIST01#) **and 13** (see #RLIST13#) **must have been input beforehand.**

```
\COMPOSITION
CONTENTS FORMULA=
SCATTERING FILE=
PROPERTIES FILE=
END
```

For example:

```
\COMPOSITION
CONTENT C 6 H 5 N O 2.5 CL
SCATTERING CRSCP:SCATT.DAT
PROPERTIES CRSCP:PROPERTIES.DAT
END
```

\COMPOSITION

There are three directives, none of which have default values.

CONTENTS FORMULA=
FORMULA=
The formula for the UNIT CELL (NOT ASYMMETRIC UNIT) is given as a list with entries

```
'element TYPE' 'number of atoms'.
```

The items in the list MUST be separated by at least one space. The number of atoms may be omitted, when they default to 1.0, and may be fractional.

The element TYPE must conform to the TYPE conventions described in the atom syntax, section #RATOMSYNTAX#.

SCATTERING FILE=
This directive gives the name of the file to be searched for scattering factors, and must conform to the syntax of the computing system. A file CRSCP:SCATT.DAT is provided for some implementations, and contains all the scattering factors listed in Volume IV, International Tables.

PROPERTIES FILE=
This directive gives the name of the file to be searched for elemental properties, and must conform to the syntax of the computing system. A file CRSCP:PROPERTIES.DAT is provided for some implementations, and contains values gleaned from various sources. The file contains references.

4.11 Input of the atomic scattering factors - \LIST 3

This list contains the scattering factors that are to be used for each atomic species that may appear in the atomic parameter list (LIST 5) - see the section of the user guide on Atom and Element names.

```
\LIST 3
READ   NSCATTERERS=
SCATTERING TYPE= F'= F''= A(1)= B(1)= A(2)= . . . B(4)= C=
END
```

For example

```
\LIST 3
READ 2
SCATT C     0     0
CONT  1.93019  12.7188  1.87812  28.6498  1.57415  0.59645
CONT  0.37108  65.0337  0.24637
SCATT S 0.35 0.86  7.18742  1.43280  5.88671  0.02865
CONT                  5.15858  22.1101  1.64403  55.4561
CONT                 -3.87732
END
```

The scattering factor of an atom in LIST 5 (the model parameters) is determined by its TYPE, an entry for which must exist in LIST 3.

The form factor is calculated analytically at each value of sin(theta)/lambda, s , from the relationship :

```
f = sum[a(i)*exp(-b(i)*s*s)] + c        i=1 to 4.
```

The coefficients a(1) to a(4), b(1) to b(4) and c and the real and imaginary parts of the anomalous dispersion correction are input for each element TYPE.

\LIST 3
This is the normal calling command for the input of LIST 3.

READ NSCATTERERS=
NSCATTERERS=
This must be set to the number of atomic species to be stored in LIST 3, and thus the number of SCATTERING directives to follow. There is no default value.

SCATTERING TYPE= F'= F''= A(1)= B(1)= A(2)= . . . B(4)= C=
This directive provides the form factor details for one atomic species. This directive must be repeated NSCATTERERS times.
TYPE=
The element TYPE must conform to the TYPE conventions described in the General Introduction. The values used for TYPE in LIST 3 will have their counterparts in the TYPEs stored for atoms in LIST 5 (the model parameters), and in the TYPEs stored for atomic species in LIST 29 (see section #RLIST29#). There is no default for this parameter.
F'= F''=
These define the real and imaginary parts of the anomalous dispersion correction for this atomic species at the appropriate wavelength. A default value of zero is assumed if these parameters are omitted.
A(1)= B(1=) A(2=) B(2)= A(3)= B(3)= A(4)= B(4)= C=
These define the coefficients used to compute the scattering factor for this atomic species. There are no default values.

```
 For neutrons, all the A(i) and B(i) are set to zero, and C is set to
 the scattering length.
```

4.12 Printing the scattering factors

\PRINT 3
This prints LIST 3. There is no command for punching LIST 3.

4.13 Input of the crystal and data collection details - LIST 13
LIST 13 contains information about those experimental details which may be needed during structure analysis. Information only required for the generation of a cif are held in LIST 30 (section #RLIST30#).
If no LIST 13 has been input and one is required, a default list is generated.

```
\LIST 13
CRYSTAL FRIEDELPAIRS= TWINNED= SPREAD=
DIFFRACTION GEOMETRY= RADIATION=
CONDITIONS WAVELENGTH= THETA(1)= THETA(2)= CONSTANTS . .
MATRIX R(1)= R(2)= R(3)= . . . R(9)=
TWO H= K= L= THETA= OMEGA= CHI= PHI= KAPPA= PSI=
THREE H= K= L= THETA= OMEGA= CHI= PHI= KAPPA= PSI=
REAL COMPONENTS= H= K= L= ANGLES=
RECIPROCAL COMPONENTS= H= K= L= ANGLES=
AXIS H= K= L=
```

For example:

```
\LIST 13
DIFF GEOM= CAD4
COND WAVE= .7107
MATRIX
END
```

\LIST 13
   This directive describes properties that relate to the whole crystal.

CRYSTAL FRIEDELPAIRS= TWINNED= SPREAD=

FRIEDELPAIRS=
   This parameter defines whether Friedel's law should be used during \SYSTEMATIC in data reduction. It should be set to NO for high accuracy or absolute structure determinations. If omitted, Friedel's law will be used.

```
YES  -  default value.
NO
```

TWINNED=
   This parameter is used during refinement to indicate whether the twin laws should be used. It is automatically updated if twinned reflection data is input.

```
NO  -  Default value.
YES
```

SPREAD=
   This parameter defines the type of mosaic spread in the crystal. This information is used during the calculation of an extinction correction.

```
GAUSSIAN   -  Default value. Suitable for X-rays
LORENTZIAN -  Suitable for Neutrons
```

DIFFRACTION GEOMETRY= RADIATION=
   This directive defines the experimental conditions used to collect the data.
GEOMETRY=
   This defines the type of data collection method used to measure the raw intensities, and determines the type of Lp correction.

```
NORMAL   -  Normal beam Weissenberg geometry.
EQUI     -  Equi-inclination Weissenberg geometry.
ANTI     -  Anti-equi-inclination Weissenberg geometry.
PRECESSION
CAD4     -  Nonius CAD4 diffractometer, Eulerian angles.
KAPPA    -  Nonius CAD4 in kappa geometry.
ROLLETT  -  Abstract machine, see page 28 , Computing Methods
             in Crystallography.
Y290     -  Hilger-Watts Y290 4-Circle diffractometer.
NONE     -  Default.
```

RADIATION=
   This parameter defines the type of radiation used to collect the data.

```
XRAYS   -  Default value
NEUTRONS
```

CONDITIONS WAVELENGTH= THETA(1)= THETA(2)= CONSTANTS . .
   This directive describes the conditions that were used when the data were collected. CONSTANTS is short for four constants.

CONSTANT(1)= CONSTANT(2)= CONSTANT(3)= CONSTANT(4)=

WAVELENGTH=
    This defines the wavelength of the radiation used to collect the data. If omitted, a default value of 0.71073 is assumed,(Mo k-alpha).

THETA(1)=
    This defines the Bragg angle of the monochromator. If omitted, a default of 6.05 is assumed, indicating that a monochromator was used with Mo radiation

THETA(2)=
    This defines the angle between the plane of the monochromator and the diffracting planes of the crystal. If this parameter is omitted, a default value of 90 is assumed. This value is not used if THETA(1) is zero. Since the angle THETA(2) is fixed, the Lp correction computed using these constants is correct only for experiments where THETA(2) is a constant. This is true for equatorial geometry experiments, but is not true for equipment that uses Weissenberg or precession geometry.

CONSTANT(1)= CONSTANT(2)= CONSTANT(3)= CONSTANT(4)=
    These four parameters are used to input fundamental constants for the diffractometer used to collect the data. How many of the constants, and what values they should have are determined by the equipment and its setting. To determine the values required, consult your local diffractometer expert. The default values for c(1), c(2) and c(3) are the Nonius CAD4 GONCON constants, and c(4) is the theta value for the change from bisecting to fixed chi mode (and has a value of 90 degrees). These constants are important when machine geometry dependent calculations are made - for example, absorption corrections. The defaults in the program were correct for the Nonius CAD4 in the Oxford Chemical Crystallography lab on 13 October 1980.

MATRIX R(1)= R(2)= R(3)= . . . R(9)=
    This directive is used to input the orientation matrix directly. If this directive is input, the directives TWO , THREE , REAL , and RECIPROCAL (detailed below) may not be used. This directive is normally used for diffractometer collected data.

R(1)= R(2)= R(3)= . . . R(9)=
    The elements of the matrix must be input in the order (1,1), (1,2), (1,3), (2,1), etc. The default is a unit diagonal matrix.

TWO H= K= L= THETA= OMEGA= CHI= PHI= KAPPA= PSI=
    This directive is used to input the setting details required to define a diffractometer orientation matrix from two reflections. The details for the two reflections must be input on separate directives, so that this directive must be repeated twice. This directive may only be input when the GEOMETRY parameter on the DIFFRACTION directive is Y290 or CAD4 . If this directive is input, the directives THREE , REAL , RECIPROCAL , and MATRIX may not be used. The reflections should be given in the same order as in the original experiment.

H= K= L=
    These three parameters define the indices of the reflection that is to be used to calculate the orientation matrix.

THETA= OMEGA= CHI= PHI= KAPPA= PSI=
    These parameters define the setting angles for the reflection whose indices are given by H , K and L . There are no default values for THETA , OMEGA and PHI , and one of CHI or KAPPA must be input. The default values for CHI , KAPPA and PSI are zero.

THREE H= K= L= THETA= OMEGA= CHI= PHI= KAPPA= PSI=
    This directive is used to input the setting details required to define a diffractometer orientation matrix from three reflections. The details for the three reflections must be input on separate directives, so that this directive must be repeated three times. This directive may only be input when the GEOMETRY parameter on the DIFFRACTION directive is Y290 or CAD4 . If this directive is input, the directives TWO , REAL , RECIPROCAL , and MATRIX may not be used.

H= K= L= THETA= OMEGA= CHI= PHI= KAPPA= PSI=
    These parameters are defined as for TWO above.

REAL COMPONENTS= H= K= L= ANGLES=
    This directive is used to define the orientation matrix for the Nonius CAD4 diffractometer from the components of the real vector along the phi axis and the setting angles of one reflection. The items COMPONENTS and ANGLES are short for:

    COMPONENT(1)= COMPONENT(2)= COMPONENT(3)=
    and
    THETA= OMEGA= CHI= PHI= KAPPA= PSI=

    If this directive is input, the directives TWO , THREE , RECIPROCAL , and MATRIX may not be used. This directive may only be input when the GEOMETRY parameter on the DIFFRACTION

directive is CAD4 .
COMPONENT(1)= COMPONENT(2)= COMPONENT(3)=
These three parameters provide the components of the real cell vector that is parallel to the phi axis.
H= K= L= THETA= OMEGA= CHI= PHI= KAPPA= PSI=
These parameters are defined as in TWO above

RECIPROCAL COMPONENTS= H= K= L= ANGLES=
This directive is used to define the orientation matrix for the Nonius CAD4 diffractometer from the
components of the reciprocal vector along the phi axis and the setting angles of one reflection. The
items COMPONENTS and ANGLES are short for:

COMPONENT(1)= COMPONENT(2)= COMPONENT(3)=
and
THETA= OMEGA= CHI= PHI= KAPPA= PSI=

If this directive is input, the directives TWO , THREE , REAL , and MATRIX may not be used. This
directive may only be input when the GEOMETRY parameter on the DIFFRACTION directive is
CAD4 .
COMPONENT(1)= COMPONENT(2)= COMPONENT(3)=
These three parameters provide the components of the reciprocal cell vector that is parallel to the phi
axis.
H K L THETA OMEGA CHI PHI KAPPA PSI
These parameters are defined as in TWO above

AXIS H= K= L=
This directive is used to define the axis about which data were collected in Weissenberg geometry.
This directive may only be given when the GEOMETRY parameter on the DIFFRACTION directive is
one of NORMAL , EQUI or ANTI .
H= K= L=
These three parameters define the zone axis [hkl] about which the crystal was rotated during data
collection. If any of these parameters is omitted, a default value of zero is assumed.

4.14 Printing the experimental conditions, LIST 13

\PRINT 13
This prints list 13. There is no command for punching LIST 13.

4.15 Input of the contents of the asymmetric unit - LIST 29

To perform calculations based on elemental properties, such as Sim weighting for Fourier maps (section
#RFOURIER#), connectivity calculations, absorption and density calculations, it is necessary to input the
numbers and properties of the elements in the cell. This information is stored in LIST 29.

```
\LIST 29
READ  NELEMENT=
ELEMENT TYPE= COVALENT= VANDERWAALS= IONIC= NUMBER= MUA= WEIGHT= COLOUR=
END
```

For example:

```
\LIST 29
READ NELEMENT=4
ELEMENT MO NUM=0 .5
ELEMENT S NUM=2
ELEMENT O NUM=3
ELEMENT C NUM=10
END
```

\LIST 29

READ NELEMENT=
NELEMENT
This must be set to the number of atomic species in the asymmetric unit, and consequently the
number of ELEMENT directives that are about to follow this directive. If this directive is omitted, a
default value of one is assumed for NELEMENT.

ELEMENT TYPE= COVALENT= VANDERWAALS= IONIC= NUMBER= MUA= WEIGHT=
Each ELEMENT directive provides the information about that atomic species in the asymmetric unit.
TYPE=

The element TYPE must conform to the TYPE conventions described in the section on atom syntax, #RATOMSYNTAX#. The default value for this parameter is taken from the COMMAND file. When LIST 29 is used for Simm weighting, the TYPE is compared with the TYPEs stored in LIST 3 (section #RLIST03#) to determine the scattering factor of the given species.

COVALENT=
VANDERWAALS=
IONIC=
The radii used during geometry calculations, with a default values set in the COMMAND file. The covalent radius is incremented by 0.1 A for distance contacts, and used for defining restraint targets (see \DISTANCES). The van der Waals radius is incremented by .25A for finding non-bonded contacts, and used for defining energy restraints The ionic radius may be used during geometry calculations.

NUMBER=
This parameter gives the number of atoms of the given type in the asymmetric unit. This number can be fractional, depending on the number of atoms in the cell and whether they occupy special positions, and whether they are disordered.

MUA=
This is the atomic absorption coefficient x10**(-23) /cm as in INT TAB VOL III. Note that in Vol IV the units are x10**(-24). Take care to ensure that the coefficients are appropriate for the wavelength used.

WEIGHT
This is the Atomic weight

COLOUR
This is the colour to be used for each atom in CAMERON. The available colours are:

```
BLACK BLUE     CYAN    GREEN GREY    LBLUE LGREEN LGREY
LRED   MAGENTA ORANGE PINK   PURPLE RED    WHITE  YELLOW
```

4.16 Printing the contents of the asymmetric unit, LIST 29

\PRINT 29
This prints list 29. There is no command for punching LIST 29.

4.17 Input of General Crystallographic Data - LIST 30

This list holds general crystallographic information for later inclusion in the cif file. CRYSTALS contains no COMMAND for editing this list - inputting a new LIST 30 over writes any existing version. However, some CRYSTALS commands update LIST 30 as an analysis proceeds, and there is a SCRIPT which enables some details to be changed.

```
\LIST 30
DATRED      NREFMES= NREFMERG= RMERGE= NREFFRIED= RMERGFRIED= REDUCTION=
CONDITIONS MINSIZE= MEDSIZE= MAXSIZE= NORIENT=
CONTINUE    THORIENTMIN= THORIENTMAX= TEMPERATURE= STANDARDS= DECAY= SCANMODE=
CONTINUE    INTERVAL= COUNT= INSTRUMENT=
REFINEMENT R= RW= NPARAM= SIGMACUT= GOF= DELRHOMIN= DELRHOMAX=
CONTINUE    RMSSHIFT= NREFUSED= FMINFUNC= RESTMINFUN= TOTALMINFUN= COEFFICIENT=
INDEXRANGE HMIN= HMAX= KMIN= KMAX= LMIN= LMAX= THETAMIN= THETAMAX=
ABSORPTION ANALMIN= ANALMAX= THETAMIN= THETAMAX= EMPMIN= EMPMAX=
CONTINUE    DIFABSMIN= DIFABSMAX= ABSTYPE=
GENERAL     DOBS= DCALC= F000= MU= MOLWT= FLACK= ESD= ANALYSE-CUT=
CONTINUE    ANALYSE-NREF= ANALYSE-R= ANALYSE-RW= SOLUTION=
COLOUR
SHAPE
CIFEXTRA    CALC-SIGMA= CALC-NREF= CALC-R= CALC-RW=
CONTINUE    ALL-SIGMA= ALL-NREF= ALL-R= ALL-RW=
END
```

For example

```
\LIST 30
CONDITIONS MINSIZE=.1 MEDSIZE=.3 MAXSIZE=.8 NORIENT=25
CONTINUE    THORIENTMIN=15.0 THORIENTMAX=25.0
CONTINUE    TEMPERATURE=293 STANDARDS=3 DECAY=.05 SCANMODE=2THETA/OMEGA
CONTINUE    INSTRUMENT=MACH3
INDEXRANGE HMIN=-12 HMAX=12 KMIN=-13 KMAX=13 LMIN=-1 LMAX=19
COLOUR RED
SHAPE PRISM
END
```

\LIST 30

DATRED NREFMES= NREFMERG= RMERGE= NREFFRIED= RMERGFRIED= REDUCTION=
  Information about the data reduction process.
  NREFMES=
    The number of reflections actually measured in the diffraction experiment
  NREFMERG=
    Number of unique reflections remaining after merging equivalents applying Friedel's Law
  RMERGE=
    Merging R factor (R int) applying Friedel's Law (as decimal not %)
  NREFFRIED=
    Number of unique reflections remaining after merging equivalents without applying Friedel's Law
  RMERGFRIED=
    Merging R factor (R int) without applying Friedel's Law (as decimal not %)

CONDITIONS MINSIZE= MEDSIZE= MAXSIZE= NORIENT= THORIENTMIN= THORIENTMAX=
CONDITIONS (continued) TEMPERATURE= STANDARDS= DECAY= SCANMODE=
CONDITIONS (continued) INTERVAL= COUNT= INSTRUMENT=
  Information about data collection.
  MINSIZE=
  MEDSIZE=
  MAXSIZE=
    The crystal dimensions, in mm.
  NORIENT=
    Number of orientation checking reflections.
  THORIENTMIN=
    Minimum theta value for orientating reflections.
  THORIENTMAX=
    Maximum theta value for orientating reflections.
  TEMPERATURE=
    Data collection temperature, Kelvin.
  STANDARDS=
    Number of intensity control reflections.
  DECAY=
    Average decay in intensity, %.
  SCANMODE=
    Data collection scan method. Options are

    2THETA/OMEGA (Default)
    OMEGA
    UNKNOWN

INTERVAL=
  Intensity control reflection interval time, minutes. Used if standards are measured at a fixed time interval
  COUNT=
    Intensity control reflection interval count. Used if standards are measured after a fixed number (count) of general reflections.
  INSTRUMENT
    Instrument used for data collection. Known instruments are:

    UNKNOWN (default)
    CAD4
    MACH3
    KAPPACCD
    DIP
    SMART
    IPDS

REFINEMENT R= RW= NPARAM= SIGMACUT= GOF= DELRHOMIN= DELRHOMAX= RMSSHIFT=
REFINE (cont) NREFUSED= FMINFUNC= RESTMINFUNC= TOTALMINFUNC= COEFFICIENT=
  Information about the refinement procedure.
  R=
    Conventional R-factor.
  RW=
    Hamilton weighted R-factor.

    The weighted R-factor stored in LIST 6 (section #RLIST06#) and LIST 30 is that computed during a structure factor calculation. The conventional R-factor is updated by either an SFLS calculation (section #RSFLS#) or a SUMMARY of LIST 6.
  NPARAM=
    Number of parameters refined in last cycle.

SIGMACUT=
The I/sigma(I) threshold used during refinement.
GOF=
GOF, Goodness-of-Fit, S.
DELRHOMIN=
DELRHOMAX=
Minimum and maximum electron density in last difference synthesis.
RMSSHIFT=
R.m.s (shift/e.s.d) in last cycle of refinement.
NREFUSED=
Number of reflections used in last cycle of refinement.
FMINFUNC=
Minimisation function for diffraction observations.
RESTMINFUNC=
Minimisation function for restraints.
TOTALMINFUNC=
Total minimisation function.
COEFFICIENT=
Coefficient for refinement. Alternatives are:

```
F (Default)
F**2
UNKNOWN
```

INDEXRANGE HMIN= HMAX= KMIN= KMAX= LMIN= LMAX= THETAMIN= THETAMAX=
Range of reflection limits during data collection.
HMIN= HMAX= KMIN= KMAX= LMIN= LMAX=
Minimum and maximum values of h,k and l.
THETAMIN= THETAMAX=
Minimum and maximum values of theta.

ABSORPTION ANALMIN= ANALMAX= THETAMIN= THETAMAX= EMPMIN= EMPMAX=
ABSORPTION (continued) DIFABSMIN= DIFABSMAX= ABSTYPE=
Information about absorption corrections.

NOTE the keywords PSIMIN and PSIMAX have been removed. Store values as EMPMIN and EMPMAX
ANALMIN= ANALMAX=
Minimum and maximum analytical corrections
THETAMIN= THETAMAX=
Minimum and maximum theta dependant corrections
EMPMIN= EMPMAX=
Minimum and maximum empirical corrections (usually combination of theta and psi or multi-scan for area detectors).
DIFABSMIN= DIFABSMAX=
Minimum and maximum DIFABS type correction, i.e. based on residue between Fo anf Fc (see section #RDIFABS#). In the cif it is called a refdelf correction.
ABSTYPE=
Type of absorption correction. Alternatives are:

```
NONE (default) EMPIRICAL      GAUSSIAN       SPHERICAL
DIFABS         MULTI-SCAN     ANALYTICAL     CYLINDRICAL
SHELXA         SADABS         NUMERICAL
               SORTAV         INTEGRATION
               PSI-SCAN
```

GENERAL DOBS= DCALC= F000= MU= MOLWT= FLACK= ESD=
GENERAL (continued) ANALYSE-CUT= ANALYSE-NREF=
GENERAL (continue) ANALYSE-R= ANALYSE-RW= SOLUTION=
General information, usually provided by CRYSTALS.
DOBS= DCALC=
Observed density and that calculated by CRYSTALS.
F000=
Sum of scattering factors at theta = zero.
MU=
Absorption coefficient, calculated by CRYSTALS.
MOLWT=
Molecular weight, calculated bt CRYSTALS.
FLACK=

ESD=
   The Flack parameter and its esd, if refined.
ANALYSE-CUT=
ANALYSE-NREF=
ANALYSE-R=
ANALYSE-RW=
   These values are updated when ever \ANALYSE is run, and can be used to record the effect of
   different LIST 28 schemes. **Remenber** that if the LIST 28 conditions are modified to include more
   reflections than were used in the last \SFLS calculation (section #RSFLS#), the values of Fc for the
   additional reflections will be incorrect. A \SFLS calculation sets these to the same values as in
   REFINEMENT above.
SOLUTION=
   The program/procedure used for structure solution

   UNKNOWN (Default)
   SHELXS
   SIR88
   SIR92
   PATTERSON
   SIR97
   DIRDIF

COLOUR
   The crystal colour.
SHAPE
   The crystal shape.
CIFEXTRA
   These are filled in by the \SFLS CALC operation (section #RSFLS#). Structure factors are computed
   for ALL reflections along with R and Rw - LIST 28 is ignored (LIST 28, reflection filtering, see section
   #RLIST28). R and Rw are also computed for reflections above a given threshold.

4.18 Printing the general information, LIST 30

\PRINT 30
   This prints list 30. There is no command for punching LIST 30.

Chapter 5

Reflection Data Input

5.1 Scope of the Reflection Data Input section of the Reference Manual

The areas covered are:

```
Reflection Data
Simple input of F or Fsq data            - \LIST 6
Advanced input of F or Fsq data          - \LIST 6
Reflection Parameter Coefficients
Storage of reflection data
Compressed reflection files
Intensity data                           - \HKLI
Standard Decay Curves                    - \LIST 27
Data Reduction                           - \LP
Systematic absence removal               - \SYSTEMATIC
Sorting data                             - \SORT
Merging equivalent reflections           - \MERGE
Theta-dependent absorption correction    - \THETABS
```

5.2 Reflection Data

Format of reflection data
The reflection data may be embedded into the control data, but it is more normal to hold it in a separate file, the HKLI file. This file may have one of more reflections per line, or a reflection may span several lines. The parameters for each reflection may be in fixed format, *i.e.* right adjusted columns, or be in free-format, with at least a single space separating items.

If fixed-format input is used, the user must supply a FORTRAN format statement. This specifies the width of the input fields, where the decimal points are, and any fields to be skipped. Even though the indices are usually integer values, CRYSTALS read them as floating point numbers. A FORTRAN 'I' format is automatically changed to an 'F' format. Note that if the input figures contain decimal points, these will over-ride values given in the format statement.

```
Examples - ^ represents a space.

FORMAT (3F4.0, 2F8.2)        ^^^1^^12^^^3^^^47.23^^^^9.32
FORMAT (3I4, 2F8.2)          ^^^1^^12^^^3^^^47.23^^^^9.32

FORMAT (3F4.0, 2F8.0)        ^^^1^^12^^^3^123456.^^312.16

FORMAT (3F4.0, 3X,2F8.0)     ^^^1^^12^^^3ABC^123456.^^312.16
```

Termination of reflection data
The reflection data themselves should be terminated with a value less than or equal to -512 for the first value on the final input line.

If the reflections are embedded into the control data, then correct termination is **vital.** Incorrect termination may lead to the program trying to read commands as reflections, producing massive error files. If the reflections are in the HKLI file, most implementations will detect the end-of-file and terminate input.

F or Fsq?
CRYSTALS will accept either F or Fsq observations, signed or unsigned. Either quantity is referred to by the name 'Fo'. If sigma values are given, they must refer directly to the signed input F or Fsq values. reflections are stored as Fo, and standard deviations are transformed or approximated so that Least-Squares refinement can be performed with either F or Fsq independent of input type. Raw intensities, I, can be input with the HKLI command. The reflection input routines (LIST 6 or HKLI) are the only routine able to take the square root of the observation. See the chapter on refinement for a brief discussion of the merits of F and FSQ refinements.

Merged or unmerged data?
CRYSTALS supports two levels of merging (averaging) simultaneously. For Fourier syntheses it is important that all symmetry operations of the Laue Group are applied, including Friedel's Law. For refinement it is permitted to used un-merged data, though in general some merging is performed. For non-centrosymmetric structures containing strong anomalous scatterers Friedel pairs should be kept separate, but other symmetry operations should be applied.

The reflection list with the minimal amount of merging is the principal reflection list, LIST 6 (section #RLIST06#). This can be used to create a full-merged list for Fourier (or other) calculations, LIST 7 (section #RLIST07#). The user can indicate to most commands which use reflections whether to use LIST 6 or LIST 7, but by default all use LIST 6 for backwards compatibility. The experienced or adventurous user can of course use LIST 6 and LIST 7 quite independently for different purposes.

5.3 Simple input of F or Fsq data - LIST 6

LIST 6 will accept reflection data either as F or Fsq. For routine work, a pre-specified set of coefficients

```
h k l Fobs sigma(Fobs)
```

are input and stored for each reflection.

NOTE that 'Fobs' will refer either to F or Fsq, depending on the value of F's.

The input coefficient list may be expanded for non-routine work - see section #RADVANCEDL6# below.

```
\LIST 6
READ F'S=
FORMAT EXPRESSION=
END


\ The OPEN command connects the reflection file
\OPEN HKLI REFLECT.DAT
\LIST 6
READ F'S=FSQ
FORMAT (3F4.0, 2F8.0)
END
\ Close the reflection file
\CLOSE HKLI
```

READ F'S=
F'S=
This parameter is used to indicate whether Fo or Fo**2 type coefficients are being read in, and must take one of the following values :

```
FSQ
FO    - Default
```

The default value of 'FO' indicates that coefficients corresponding to Fo are being read in.

By default, the reflections are assumed to come in fixed format from the HKLI channel, and may be terminated either by the end-of-file, or with -512.

5.4 Creation of LIST 7 from LIST 6 - COPY 6 7

This command creates a LIST 7 as an exact copy of LIST 6 (see #RLIST06#). The LIST 7 can then be merged using Friedel's Law to create a reflection list suitable for Fourier syntheses

```
\COPY INPUT= OUTPUT=
END
```

Example

```
\COPY 6 7
END
```

5.5 Printing LIST 6

The reflections can be output to listing file as follows :
\PRINT 6 mode
Mode controls the type of output.

```
A - Default - The reflections are in compressed
              format, on the scale of Fo.
B -           The reflections are in compressed
              format, on the scale of Fc.
C -           A general print of all the data
              stored for each reflection.
```

See also \REFLECTIONS (section #RREFLECTIONS#), which produces tables for publication.

5.6 Punching LIST 6

LIST 6 can be punched as an ASCII file in several formats.

\PUNCH 6 mode

Mode controls the format of the output.

```
A - Output the reflections in a compressed format - Default.
B - Output the reflections in 'cif' format.
C - Output Fo, Fc, phase information in tabulated format.
E - Output Fo and other input information in tabulated format.
```

LIST 6 is also output by the links to the direct methods programs. In these files, the magnitudes of Fo or Fsq are scaled so that the largest fits the format statement. The SHELX file contains Fsq, the SIR file contains Fo.

5.7 Advanced input of F or Fsq data - LIST 6

LIST 6 will accept reflection data either as F or Fsq. The data may be in free or fixed format. For routine work, a pre-specified set of parameters is stored for each reflection. This may be expanded for non-routine work by **INPUT** and **OUTPUT** coefficients.

```
\LIST 6
READ NCOEFFICIENT= TYPE= F'S= NGROUP= UNIT= CHECK=
INPUT COEFFICIENT(1)= COEFFICIENT(2)= .  .
STORE NCOEFFICIENT= MEDIUM= APPEND=
OUTPUT COEFFICIENT(1)= COEFFICIENT(2)=  .  .
FORMAT EXPRESSION=
MULTIPLIERS VALUE=
MATRIX M11=  M12=  ... M33= TOLER= TWINTOLER=
END


\ The OPEN command connects the reflection file
\OPEN HKLI REFLECT.DAT
\LIST 6
READ NCOEF=5 TYPE=FIXED UNIT=HKLI F'S=FSQ
FORMAT (3F4.0, 2F8.2)
INPUT H K L /FO/ SIGMA(/FO/)
STORE NCOEF=7
OUTPUT INDICES /FO/ SQRTW /FC/ BATCH/PHASE RATIO/JCODE SIGMA(/FO/)
END
\CLOSE HKLI
```

READ NCOEFFICIENT= TYPE= F'S= NGROUP= UNIT= CHECK=

NCOEFFICIENT=

Specifies the number of coefficients to be input per reflection. A list of permitted coefficients is given below. If this directive is omitted, the default is 5.

The default input coefficients are

```
H K L FOBS SIGMA(F)
```

TYPE=

This parameter determines the form of the reflections as they are read in, and must take one of the following values :

```
FIXED      -  Fixed format data
FREE       -  Free format text  -  default value
COMPRESSED-  See 'Compressed Reflection Data' below
COPY       -  LIST 6 is copied from the current input device to the
              output device designated on the STORE directive with
              the number of coefficients given on the OUTPUT and
              COEFFICIENT directives.
```

F'S=

This parameter is used to indicate whether Fo or Fo**2 type coefficients are being read in, and must take one of the following values :

```
FSQ
FO  -  Default value
```

The default value of 'FO' indicates that coefficients corresponding to Fo are being read in.

NGROUP=

This parameter defines the number of reflections per line for fixed format input. (For free format input, the system can work out this information). NGROUP will be less than unity if the reflection

spans several lines.
UNIT=
This parameter defines the source of the reflection data that are to be input.

```
HKLI    -  Default value.
DATAFILE
```

HKLI indicates that the reflection data are in a separate file from the main input data. The local implementation may set up default names for this file, or the \OPEN directive can be used to connect the file to CRYSTALS.

DATAFILE indicates that the reflections follow the directives for '\LIST 6' in the normal data input stream. If this is the case, the directives for \LIST6 **must** be terminated by the directive END, otherwise the reflection lines will be processed as normal directives associated with the \LIST6 command, and generate a very large number of input errors.

By default, the data are assumed to come from the alternative HKLI channel.
CHECK
This parameter determines whether reflections are rejected on input if they have a zero or negative value for Fo.

```
YES
NO -  Default value.
```

By default checking is disabled so that negative reflections are accepted on input.

INPUT COEFFICIENT(1)= COEFFICIENT(2)=. .
This directive defines the coefficients that are to be read in. The number of coefficients is given by the NCOEFFICIENT parameter above, or its default value.
COEFFICIENT(1)= COEFFICIENT(2)=
Defines the coefficients and their input order. The coefficients must be selected from the following list

```
H              K              L              /FO/
SQRTW          FCALC          PHASE          A-PART
B-PART         TBAR           FOT            ELEMENTS
SIGMA(F)       BATCH          INDICES        BATCH/PHASE
SINTH/L**2     FO/FC          JCODE          SERIAL
RATIO          THETA          OMEGA          CHI
PHI            KAPPA          PSI            CORRECTIONS
FACTOR1        FACTOR2        FACTOR3        RATIO/JCODE
```

For the meaning of these coefficients, see section #RREFPARCOEF# - 'Reflection Parameter Coefficients'

NOTE that 'Fobs' will refer either to F or Fsq, depending on the value of F's. Reflections are available during refinement as either signed Fsq or signed Fo independent of the type of input values.

STORE NCOEFFICIENT= MEDIUM= APPEND=
NCOEFFICIENT=
Specifies the number of coefficients to be stored per reflection. A list of permitted coefficients is given above. If this directive is omitted, the default is 9.

The default output coefficients are

```
INDICES /FO/ SQRTW /FC/ BATCH/PHASE RATIO/JCODE SIGMA(/FO/)
CORRECTIONS ELEMENTS
```

MEDIUM
This parameter sets the output reflection storage device. This can be a text file, but more normally it is the database, the '.dsc' file. See section #RSTOREREF# - 'Storage of Reflection Data'.

```
FILE        A named or scratch ASCII serial file
INPUT       A file of the same type as the input reflection source
DISK   -    Default - The current structure database
```

APPEND=
This parameter determines whether the input reflections are to replace or be appended to existing reflections.

```
YES     The input reflections are appended to existing reflections
NO   -  Default - The input reflections replace any existing reflections
```

OUTPUT COEFFICIENT(1)= COEFFICIENT(2)= . .
> This directive defines the coefficients that are to be stored. The number of coefficients is given by the NCOEFFICIENT parameter above, or its default value, and the coefficients selected from the list above.

> If the OUTPUT directive is omitted, as many of the default coefficients as are required by NCOEFFICIENT are used as output coefficients :

> If the OUTPUT directive is omitted and NCOEFFICIENT is greater than 9, it is reset to 9 so that the coefficients above can be used.

FORMAT EXPRESSION=
> This directive allows the user to define a format statement if fixed format input is being used. This directive is only valid if the TYPE parameter on the READ directive is FIXED .
> **EXPRESSION=**
>> This parameter defines the format to be used. Normally this keyword is omitted, so that the directive looks like a FORTRAN format statement, except that there must be at least one space between the 'FORMAT' and the expression, to terminate the directive. Since all the data are read as real numbers, the format expression can only contain F , E , and X field definitions - either find a good Fortran reference book for examples, or ask someone who did crystallography before 1990.

MULTIPLIERS VALUE=
> This directive allows the user to define the multipliers to be applied to the data if they are being read in compressed format. This directive is only valid if the TYPE parameter on the READ directive is COMPRESSED .
> **VALUE=**
>> This parameter, whose default value is unity, is repeated the number of times specified by the NCOEFFICIENT parameter on the READ directive. The order is the same as the INPUT coefficients.

MATRIX M11= M12= ...M33= TOLER= TWINTOLER=
> This directive inputs a matrix to be applied to the reflection indices as they are read in. If any component of the index differs by more than TOLER from an integer, the reflection is rejected. TWINTOLER is a value, in A-2, for overlap of potentially twinned reflections. See the chapter on twinning (#Rtwinning#).
> **Mij=**
>> The 9 elements (by row) of an index transformation matrix. The default is a unit matrix
> **TOLER=**
>> The reflection is rejected if any transformed index differs from an integer by more than TOLER. The default is 0.1.
> **TWINTOLER=**
>> The twin element tag is updated if the generated reciprocal lattice point differs from a base lattice point by less than TWINTOLER reciprocal Angstrom. The default is 0.001, but an ideal value will depend upon the integration method, the mosaicity, and the lengths of the cell edges.

5.8 Reflection Parameter Coefficients

CRYSTALS has a very flexible procedure for storing reflection information, enabling the user to optimise disk space use. The user must indicate to the program what information is available in the input data, and what information is to be stored. Storage space may also be reserved for data yet to be computed.

During data reduction (section #RDATAREDUC#), space is reserved for relevant coefficients. These coefficients (_i{e.g. setting angles) may not be needed during structure analysis, so they are not normally preserved beyond reduction.

Special Reflection storage
> The user might need to arrange special reflection storage under the following conditions:

Refinement using a partial model
> If the user is experiencing difficulties with a small part of an otherwise well behaved large structure, the real and imaginary parts of the structure factors due to the well behaved part can be precomputed and stored and these atoms removed from the atom list (LIST 5). The user then only needs recompute the contributions from the varying fragment. The total Fo, Fc, real and imaginary parts are stored with the keys

> */FO/*   */FC/*   **APART**   **BPART**

Twinned structures
    See chapter #Rtwinning# on handling twinned data.

Recognised reflection coefficients

```
Coefficients recognised are:
H             Reflection index h
K             Reflection index k
L             Reflection index l
INDICES       Packed reflection indices
/FO/          The observed intensity, Fsq or Fo value
/FOT/         The observed intensity, Fsq or Fo value for a twinned crystal
/FC/          The calculated structure factor
SIGMA(/FO/)   Standard deviation of the input observation
SQRTW         Sqrt of weight to be given a reflection during least squares
A-PART        Real part of structure factor
B-PART        Imaginary part of structure factor
PHASE         Phase angle, radians
BATCH         An integer associated with reflections measured in batches
BATCH/PHASE   Packed (compressed into one word) Batch and Phase
SINTH/L**2    (Sintheta/lambda)**2
FO/FC         Fo/Fc
ELEMENTS      Integers corresponding to twin elements
SERIAL        Serial number of reflection
JCODE         reflection quality code. See RC93 manual
RATIO         Ratio Fo**2/sigma(Fo**2)
RATIO/JCODE   Packed ratio and jcode
TBAR          Absorption weighted X-ray path length
THETA         Bragg angle
OMEGA         Setting angle
CHI           Setting angle
PHI           Setting angle
KAPPA         Setting angle
PSI           Setting angle
CORRECTIONS   Composite correction factor for Fo
FACTOR1       Individual correction factor for Fo
FACTOR2       Individual correction factor for Fo
FACTOR3       Individual correction factor for Fo
NOTHING       A spare location for programmers use
```

    If an output coefficient is specified without the corresponding input coefficient, it value is set to zero except for BATCH (default is 1.0) and SINTH/L**2 (value computed from cell parameters). Packed INDICES are restricted to +/- 127, packed RATIO to range 0.0 - 999.0, JCODE to range 0 - 9.

5.9 Storage of reflection data

    Reflections may be stored either in the structure database (the DSC file), or as external binary serial files. The latter is used mainly during data reduction (section #RDATAREDUC#).
    When a change is made to most other data lists, they are either completely overwritten (LIST1, cell parameters), or a new list created in addition to the old list (LIST 5, atom parameters). Because the reflections are special, they are handled differently. A small piece of information (called the LIST 6 Header) is created to hold information about the rest of the reflection list, and new headers are stored each time the main body is updated. The main body of the reflection list is modified in-situ if the only changes are ones which can easily be recomputed (*e.g.* Fc, phase, sqrtw), thus reducing the disk activity. If an error occurs during the updating of the body, the list becomes inaccessible to other processes, and the failing process must be re-run correctly. If the changes involve a change in size of the list, then a new body is created.
    During raw data processing (Data reduction, section #RDATAREDUC#) the size of the reflection list can change a lot (coefficients being added or removed, reflections being merged or rejected). To prevent the .DSC file growing too large, binary serial files are used to hold the body of the reflection list. One is used for input and one for output at each stage, the roles being reversed after each stage. The header is kept in the .DSC file, and keeps track of the bodies. When data reduction is complete, the body must be copied to the .DSC file as follows:

```
\  After data reduction, make a final copy of the reflections
\  and STORE THEM IN THE .DSC FILE:
\LIST 6
READ TYPE=COPY
END
```

5.10 Compressed reflection data

    CRYSTALS can produce files containing reflections is a 'compressed' format. This might be useful for archiving data. The compressed data is headed by the correct information for its reinput.
    The file contains information for h, k, l, /FO/ or /FOT/, RATIO/JCODE and elements. For each KL pair, the K value is given for this group of reflections, then the L value for the group, followed by the H and /FO/ and other values for the first reflection, the H /FO/ and other values for the second reflection, and so on,

finishing with 512, which is the terminator for this KL pair. This pattern is repeated for all the KL pairs, the terminator for the last KL pair being -512, and indicates the end of the reflection list. Take care if you try to edit these files, and note that K and L are the two constant indices for each group, while H changes most rapidly.

5.11 Intensity Data - HKLI

Raw intensity data require more processing than F or Fsq values. The instruction '\HKLI' is related to '\LIST 6', but has different default coefficients and additional directives for geometrical corrections.

```
\HKLI
READ NCOEFFICIENT= TYPE= F'S= NGROUP= UNIT= CHECK=
INPUT COEFFICIENT(1)= COEFFICIENT(2)= .  .
STORE NCOEFFICIENT= MEDIUM= APPEND=
OUTPUT COEFFICIENT(1)= COEFFICIENT(2)=  .  .
FORMAT EXPRESSION=
CORRECTIONS NSCALE NFACTOR
FACTORS COEFFICIENT(1)= COEFFICIENT(2)=  .  .
ABSORPTION PRINT= PHI= THETA= TUBE= PLATE=
PHI NPHIVALUES= NPHICURVES=
PHIVALUES PHI= .........
PHIHKLI H= K= L= I[MAX]=
PHICURVE I= .........
THETA NTHETAVALUES=
THETAVALUES THETA=
THETACURVE CORRECTION= ........
TUBE NOTHING OMEGA= CHI= PHI= KAPPA= MU=A[MAX]=
PLATE NOTHING OMEGA= CHI= PHI= KAPPA= MU=A[MAX]=
END
```

For example

```
\  The OPEN command connects the reflection file:
\OPEN HKLI REFLECT.DAT
\  The HKLI instruction reads the data in:
\HKLI
\  There are 12 items to read:
READ NCOEF=12 FORMAT=FIXED UNIT=HKLI F'S=FSQ CHECK=NO
\  This is what they are:
INPUT H K L /FO/ SIGMA(/FO/) JCODE SERIAL BATCH THETA PHI OMEGA KAPPA
\  And this is their format:
FORMAT (5X,3F4.0,F9.0,F7.0,F4.0,F9.0,F4.0,4F7.2)
\  We only want to store six of them:
STORE NCOEF=6
\  Specifically, these ones:
OUTPUT INDICES /FO/ BATCH RATIO/JCODE SIGMA(/FO/) CORRECTIONS SERIAL
\  Some absorption corrections have been measured:
ABSORPTION PHI=YES  THETA=YES PRINT=NONE
\  Here is the theta dependent absorption curve:
THETA 16
THETAVALUES
CONT 0  5 10 15 20 25 30 35 40 45 50 55 60 65 70 75
THETACURVE
CONT 3.61  3.60  3.58  3.54  3.50  3.44  3.37  3.30
CONT 3.23  3.16  3.09  3.02  2.96  2.91  2.86  2.82
\  And here is one azimuthal absorption curve containing 26 points:
PHI 26  1
PHIVALUES
CONT   6  16  21  26  31  36  41  61  66  76
CONT  81  86  91  96 111 121 131 136 141 146
CONT 151 156 161 166 171 176
\  This is the reflection we used for the scan:
PHIHKLI    -3    -1     0    28392
PHICURVE
CONT    26887   25377   24608   23990   23445   23049
CONT    22867   22801   22782   22937   23104   23368
CONT    23713   24129   25669   26836   27892   28250
CONT    28291   28256   28101   28009   28204   28373
CONT    28392   28203
END
\  All done. Close the hkl file.
\CLOSE HKLI
```

In the following description, for items defined under LIST 6 above only the default value will be given.

\HKLI

READ NCOEFFICIENT= TYPE= F'S= NGROUP= UNIT= CHECK=
This directive is the same as the READ directive in \LIST 6 above, except that the following

parameters have different default values:

```
NCOEFFICIENT= default value is 12
TYPE= default value is FIXED
F'S= default value is FSQ
NGROUP= default value is 1
UNIT= default value is HKLI
```

INPUT COEFFICIENT(1)= COEFFICIENT(2)=. .
> This directive defines the coefficients that are to be read in. The number of coefficients is given by the NCOEFFICIENT parameter above, or its default value.

> The default input coefficients are (i.e. for RC93 output):

```
 H  K  L  /FO/  SIGMA(/FO/)  JCODE  SERIAL  BATCH  THETA  PHI  OMEGA  KAPPA
```

STORE NCOEFFICIENT= MEDIUM= APPEND=
> **NCOEFFICIENT=**
>> The number of coefficients that will appear on the OUTPUT directive. The default is 9.
> **MEDIUM=**
>> The default value is 'FILE'. Since the reflections will be much changed during data reduction (section #RDATAREDUC#), the intermediate storage is usually a scratch serial file.
> **APPEND=**
>> The default value is 'NO'.

OUTPUT COEFFICIENT(1)= COEFFICIENT(2)=. .
> The default coefficients are:

```
 INDICES /FO/ SQRTW /FC/ BATCH/PHASE RATIO/JCODE SIGMA(/FO/)
 CORRECTIONS ELEMENTS
```

> Note that H, K, and L are compressed into one key: 'INDICES'.

FORMAT EXPRESSION=
> This directive is only valid if the TYPE parameter on the READ directive is FIXED.
> **EXPRESSION=**
>> If the diffractometer type indicated in LIST 13 (section #RLIST13#) is CAD4, the default corresponds to RC93 or RC85 output, otherwise an expression must be given.

```
 e.g. (5X,3F4.0,F9.0,F7.0,F4.0,F9.0,F4.0,4F7.2)
```

> Directives found in HKLI commands, but not in LIST 6 commands are:#

CORRECTIONS NSCALE= NFACTOR=
> **NSCALE=**
>> Set to 1 or 2 to select the first or second scale factor in LIST 27 (see section #RLIST27#).# The default is 2.
> **NFACTOR=**
>> Up to three correction per reflection to be applied to the input observations can be included in the input file. This keyword specifies how many to use.# The default is 0.

FACTORS COEFFICIENT(1)= COEFFICIENT(2)=. .
> The permitted coefficients are FACTOR1, FACTOR2 and FACTOR3. These are applied to the input observation before any other action (including square rooting if requested) is performed.

ABSORPTION PRINT= PHI= THETA= TUBE= PLATE=
> This directive controls approximate absorption corrections to be applied during input. They are only suitable if the diffractometer used is one of those permitted in LIST 13 (section #RLIST13#).
> **PRINT=**
>> Permitted levels are

```
 FULL                 Two lines of information per reflection
 NONE    - Default    No output is produced
 PARTIAL -            Summary for each reflection
```

> **PHI=**

NO   - Default
YES

  If YES, then phi (azimuthal scan) data must follow.
THETA=

NO   - Default
YES

  If YES, then a theta dependent correction curve must follow.
TUBE=

NO   - Default
YES

  If YES, then orientation angles for the tube must follow.
PLATE=

NO   - Default
YES

  If YES, then orientation angles for the plate must follow.

PHI NPHIVALUES= NPHICURVES=
  If phi has been set to 'YES' above, this directive sets up input and computation of azimuthal scan absorption corrections, by the method of North, Phillips and Mathews, Acta Cryst., **A24,** 351 (1968).
  NPHIVALUES=
    Number of sampling points on the phi curve. These need not be equally spaced
  NPHICURVES=
    Number of phi curves that will be entered after this directive.

PHIVALUES PHI= .....
  The 'Nphivalue' phi angles of the points on the absorption curve.

PHIHKLI H= K= L= I[MAX]=
  The h,k,l and Imax values for the following 'Nphicurve' phi profiles, in the same order as the profiles.

PHICURVE I= .....
  The 'Nphivalue' intensity values for the profile at the phi values given on the Phivalues directive. There is a Phicurve corresponding to each PHIHKLI directive.

THETA NTHETAVALUES=
  If theta has been set to 'YES' above this directive sets up the input for and computation of a theta dependent absorption correction. Except when the data has been corrected by a proper analytical correction, a theta dependent correction is **ALWAYS** recommended, since neither a phi scan, multi-scan nor DIFABS (section #RDIFABS#) will make a good theta approximation. See Int Tab, Vol II, p295 and 303 for suitable profiles.
  NTHETAVALUES=
    The number of sampling points on the theta curve.

THETAVALUES THETA= .....
  The Nthetavalues at which the curve is sampled

THETACURVE CORRECTION= ......
  The Nthetavalue values of the correction factor profile.

TUBE NOTHING OMEGA= CHI= PHI= KAPPA= MU A[MAX]
  If TUBE has been set to 'YES' above, this directive sets up the correction for a sample in a tube, or for an acicular crystal steeply inclined to the phi axis. See J. Appl. Cryst, 8. 491, 1975. 'NOTHING' is a place-holder for internal workings.
  OMEGA= CHI= PHI= KAPPA=
    These are the settings needed to bring the tube axis into the equatorial plane and perpendicular to the incident X-ray beam. Only one of Chi and Kappa may be given.
  MU=
    The product of Mu and the thickness of the tube wall.
  A[MAX]
    The maximum permitted correction. Values greater than A[max] generate a warning.

PLATE NOTHING OMEGA= CHI= PHI= KAPPA= MU A[MAX]
    If PLATE has been set to 'YES' above, this directive sets up the correction for an extended plate-like
    sample. See J. Appl. Cryst, 8. 491, 1975. 'NOTHING' is a place-holder for internal workings.
OMEGA= CHI= PHI= KAPPA=
    These are the settings needed to bring the plate normal into the equatorial plane and perpendicular to
    the incident X-ray beam. Only one of Chi and Kappa may be given.
MU=
    The product of Mu and the plate thickness.
A[MAX]
    The maximum permitted correction. Values greater than A[max] generate a warning.

5.12 Intensity Decay Curves \LIST 27

```
\LIST 27
READ NSCALE=
SCALE SCALENUMBER= RAWSCALE= SMOOTHSCALE= SERIAL=
END
```

    If each reflection has been assigned a serial number (or some other incrementing value, such as total
X-ray exposure time) then CRYSTALS can apply a correction which is linked to this value. The corrections,
on the scale of Fsq, are held in LIST 27. Two correction factors can be stored, but only one used. For
example, these can be the actual corrections computed from the decay of the standard reflections, and those
obtained from a 3-point smoothing of the same correction data. The applied scale factor is obtained by
interpolating between those given scale factors with serial numbers above and below the serial number of the
current reflection. If there is a dramatic change in scale (for example due to remeasurement of some very
strong reflections with attenuated X-rays), it is important not to interpolate over this discontinuity. To
achieve this, a dummy scale factor is inserted at this point with scale values the same as the current scales,
but with the same serial number as the first scales after the discontinuity - for example:

```
\LIST 27
READ NSCALE=16
SCALE     1      1.000  1.000     1
SCALE     2      1.066  1.066     4
SCALE     3      1.074  1.053    57
SCALE     4      0.997  1.018    83
SCALE     5      1.003  1.003   564
SCALE     6      0.370  0.370   564
SCALE     7      0.372  0.371   617
END
```

\LIST 27

READ NSCALE=
  NSCALE=
    The number of SCALE directives to follow. There is no default value for this parameter.

SCALE SCALENUMBER= RAWSCALE= SMOOTHSCALE= SERIALNUMBER=
  This directive is repeated once for each scale factor that is to be read in.
  SCALENUMBER=
    This parameter indicates the number of the scale factor, starting from one. There is no default for
    this parameter, which currently is not used.
  RAWSCALE=
    This parameter gives the initial scale factor, computed directly from the intensities of the standard
    reflections.# There is no default.
  SMOOTHSCALE=
    This parameter gives the scale factor after the raw scale factors have been smoothed, so that a
    continuous curve is fitted to all the data.# There is no default.
  SERIALNUMBER=
    This parameter gives the serial number of the first standard reflection contributing to this scale. The
    data reduction programs use the SERIAL to locate the correct scales to use for a given reflection.#
    There is no default.

5.13 Printing the decay curve

\PRINT 27
    This command prints the decay curve. There is no command to punch LIST 27.

5.14 Data Reduction - Lp

   This command causes the Lp correction to be calculated for each reflection.
   The diffraction geometry, wavelength, etc. are taken from LIST 13 (section #RLIST13#). If LIST 13 is input incorrectly, or has to be generated by the system, the message 'illegal diffraction geometry flag' will be output and the job terminated. If the user has forced the storage of Fsq values in \HKLI, it is necessary to indicate this to the Lp correction.

```
\LP
STORE MEDIUM= F'S=
END
```

For example

```
\  Apply an LP correction for the geometry stored
\  in List 13.
\LP
END
```

\LP

  STORE MEDIUM= F'S=
    MEDIUM=
       Determines the output medium.

```
    FILE             A serial file
    INPUT - Default  The same as the input medium
    DISC             The .DSC file.
```

       The default output medium is the same as te input medium - usually a serial file.
    F'S=

```
    FO     -     Default
    FSQ          Indicating that square roots were not taken at
                 input time.
```

5.15 Systematic absence removal - \SYSTEMATIC

```
\SYSTEMATIC INPUTLIST=
STORE MEDIUM= F'S= NEWINDICES=
END
```

For example:

```
\  Remove systematic absences and move each hkl index
\  by symmetry so that they all lie in the same part of
\  the reciprocal lattice:
\SYST
END
```

   This routine uses the symmetry operators in LIST 2 (section #RLIST02#) to identify systematic absences, which are listed and rejected. It can also use the symmetry operators to transform indices to that the reflections fall into a unique part of the reciprocal lattice. The unique set is bounded by the maximum range in 'l', maximum range of 'k' given the 'l' range, and maximum range of 'h', given the 'k,l' range.
   Friedel's Law may be invoked, depending on the flag in LIST 13 (section #RLIST13#). It is important **NOT** to use Friedel's Law for structures which have strong anomalous scatterers, since reflections related by Friedel's law are not equivalent in this case and should not be merged together. Similarly, if orientation dependent corrections are to be made (_i{e.g. DIFABS),original indices should be preserved. Note that in this case, only exactly equivalent reflections will be merged, and care must be taken when computing Fourier maps. See the sections on Fourier maps, #RFOURIER# and DIFABS #RDIFABS#.

\SYSTEMATIC

  SYSTEMATIC INPUTLIST=

    INPUTLIST=
       6 OR 7

  STORE MEDIUM= F'S= NEWINDICES=
    MEDIUM=
       Determines the output medium.

```
FILE              A serial file
INPUT - Default   The same as the input medium
DISC              The .DSC file.
```

The default output medium is the same as the input medium - usually a serial file.

F'S=

```
FO        -       Default
FSQ               Indicating that square roots were not taken at
                  input time.
```

NEWINDICES=
Determines whether new indices are computed.

```
YES  -  Default - Permits transformation of indices.
NO
```

5.16 Sorting of the reflection data - \SORT

```
\SORT INPUTLIST=
STORE MEDIUM=
END
```

For example:

```
\  Sort reflections into order by L, then K, then H:
\SORT
END
```

This routine sorts the data so that the reflections are placed in a predetermined order, in which reflections with the same indices are adjacent in the list. Upon output, the reflections are arranged so that they are in groups of constant L, starting with the group with the smallest L value. Within any L group, the reflections are ordered in groups of constant K, starting with the group with the smallest K value. Within each group of constant K and L, the reflections are arranged with the smallest H value first and the largest last in ascending order.

The method of sorting is a multi-pass tree sort, in which as many reflections as possible are held in memory during each pass. If all the reflections with a given value of L cannot be in memory at the same time, the program will terminate in error.

\SORT

STORE MEDIUM
MEDIUM=
Determines the output medium.

```
FILE              A serial file
INPUT - Default   The same as the input medium
DISC              The .DSC file.
```

The default output medium is the same as the input medium - usually a serial file.

5.17 Merging equivalent reflections - \MERGE

```
\MERGE INPUT= TWINNED=
STORE MEDIUM=
REFLECTIONS NJCODE= LIST= LEVEL= F'S=
JCODE NUMBER= VALUE=
REJECT RATIO= SIGMA=
WEIGHT SCHEME= NPARAMETERS= NCYCLE=
PARAMETERS P .....
END
```

For example:

```
\MERGE
WEIGHT SCHEME=2 NPARAM=6
PARAMETERS .5  3.0  1.0  2.0  .01 .00001
END
```

The merge routine takes a list of reflections and combines groups of adjacent reflections with *exactly* the same indices to produce a single mean structure amplitude.

\SYST (section #RSYSTEMATIC#) and \SORT (section #RSORT#) produce a suitable list, and if either

of them have been omitted, it is extremely likely that the list of reflections produced by the merge process will contain duplicated entries for certain reflections.

It is possible to combine equivalent reflections in several different ways, depending upon how each individual contributor is weighted when the mean is computed. Several different weighting schemes are provided, and these are described in the next section (the WEIGHT directive).

The JCODE key in the list of reflections may be input from some diffractometers (e.g. a CAD4) to indicate that the value may be inaccurate. Reflections which have JCODES that differ from unity are thought to be inaccurate and can be down-weighted or eliminated during the merge process (the JCODE directive). Note that JCODES MUST be positive and less than 10.

Although under normal circumstances LIST 6 (reflections) contains /Fo/ data rather than /Fo/**2 data, the calculations performed during the merge are done on the scale of /Fo/**2. This means that r-values are computed which refer to /Fo/**2, and that reflections can be rejected on the basis of the ratio of /Fo/**2 to its standard deviation. If for some reason the LIST 6 contains /Fo/**2 data rather than the normal /Fo/ data, it is necessary to use the "F's" parameter of the "REFLECTIONS" directive to inform the system of this fact.

During the merge process, the system calculates and then prints a set of merging r-values, which are defined as follows :

```
R = 100*SUM[ Sd(i) ]/SUM[ M(i) ],
            where 'i' runs over all reflections.


Sd(i) = SUM[ <Fsq(i)> - Fsq(j) ],
            summed over 'j' contributors.

and

M(i) = SUM[ <Fsq(i)> ],
            summed 'j' times for 'j' contributors.
```

The sum variable 'i' runs over all the reflections produced by the merge process which have more than one contributor. The sum variable 'j' runs over all the contributors for each reflection produced by the merge process. <Fsq(I)> is the mean value for the reflection 'i' , while Fsq(j) is the observed value of Fsq for the contributor 'j'.

If the crystal is twinned, this will affect the merge. See chapter

If the data is in Batches with different BATCH scale factors, this will affect the merge.

WEIGHTING SCHEMES FOR THE DATA MERGE

At present there are three different weighting schemes available for merging equivalent reflections. These are :

```
    1.      Each reflection is given equal weight (unit weights).
    2.      Weights based on a Gaussian distribution.
    3.      W(i) = 1.0/Sigma(i)**2  for each reflection.
```

Unit and statistical weights (schemes 1 and 3) are more or less equivalent unless some reflections have been remeasured under very different regimes (*e.g.* with an attenuator set, mA turned down, different crystal)

Scheme 2 is designed to discriminate against outliers, *i.e.* reflections lying farther from the mean than might be expected. For this scheme, a weighted mean value of Fsq is determined iteratively, starting from unit weights. At each iteration, the weights are recomputed to discriminate against outliers and the contributing reflections are given a new weight w(i) given by :

```
    w(i) = exp [ (-log(a) * q(i)**2)/(b**2 * e(i)**2) ]
```

```
Where

    q(i)  is the deviation of the particular Fsq(i) from the current average.
    e(i)  is a predicted mean deviation of the reflection 'i' from the
          current mean and is given by a function similar to that used
          in Least Squares :
```

e = c + d*Sig(Fsq) + g*Sig(Fsq)*/Fo/ + h*Sig(Fsq)*Fsq

a,b,c,d,g,h are 6 input parameters provided by the user

'a' and 'b' define the Gaussian distribution.

'a' is the weight to be given to a reflection which has a deviation given by 'q(i) = b*e(i)'.

Suggested values of 'a' and 'b' are 0.5 and 3.0 respectively, so that if for example, 'e(i) = 3*Sig(Fsq)' (d=3, c=g=h=0), a deviation q(i) of 6*Sig*fsq will assign a reflection a weight of 0.5.

'c' Provides the bias necessary to allow for failures in the counting statistics at low count rates.

'd' is a scaling constant.

'g' and 'h' allow for the increased dispersion of strong reflections.

For a conventional diffractometer, suggested values for the parameters are :

    a = .5   b=3.0   c=1.0   d=2.0   g=.01   h=.00001

It is recommended that the Gaussian scheme be used, as it discriminates against zero or widely dispersed intensities very efficiently.

Standard deviations produced by the merge
After the equivalent reflections have been merged two different standard deviations are computed and can be output :

*SIGMA1 = Sqrt (sum [w(i)*q(i)**2] / sum [w(i)])*
*   that is, the weighted r.m.s. deviation.*

*SIGMA2 = Sqrt (sum [w(i)*s(i)**2] / sum [w(i)])*
*   that is, the weighted standard deviation.*

Either of these two standard deviations can be selected as an estimate of Sigma(Fsq), and perhaps be converted to a Least Squares weight. If a reflection is measured very many times, SIGMA1 should be similar to SIGMA2. It is almost always much greater.

\MERGE

MERGE=
  INPUT=
    Either 6 or 7. Default is 6.

  TWINNED=

NO   Treat data as un-twinned
LIST13 Treat data according to list 13
YES   Treat data as twinned

STORE MEDIUM=
  MEDIUM=
    Determines the output medium.

FILE   A serial file
INPUT - Default   The same as the input medium
DISC   The .DSC file.

    The default output medium is the same as the input medium - usually a serial file.

REFLECTIONS NJCODE= LIST= LEVEL= F'S=
  NJCODE=
    Specifies the number of JCODE directives to follow - default zero.
  LIST=
    Determines the amount of information printed during the merge process.

OFF
MEDIUM - Default value
HIGH

    If LIST is 'HIGH' , all Fsq are listed with their contributors and their deviations from the computed mean. The default value of MEDIUM indicates that the merged Fsq are listed with the contributors and their deviations from the computed mean if the r.m.s. deviation exceeds LEVEL*(mean standard deviation). HIGH is equivalent to MEDIUM with LEVEL set at zero.
  LEVEL=
    This parameter specifies the r.m.s. deviation level above which contributors are printed if LIST is equal to MEDIUM . They are printed if sigma1 exceeds level*sigma2. The default value for this parameter is 3.
  F'S=

```
FO        -        Default
FSQ                Indicating that square roots were not taken at
                   input time.
```

JCODE NUMBER= VALUE=
This directive allows reflections whose JCODE key differs from unity to be down-weighted or eliminated from the merge. It is repeated once for each JCODE that is read in.
NUMBER=
The number of the JCODE must be given. There is no default value for this parameter.
VALUE=
This is the absolute weight, associated with the JCODE number, that is given to the reflection. If this parameter is omitted a default value of zero is assumed, indicating that the reflection is to be eliminated and not included in the merge at all.

REJECT RATIO= SIGMA=
This directive causes reflections whose mean intensity is less than product of the ratio and sigma to be eliminated.
RATIO=
The default value for this parameter is -10. Use LIST 28 (section
use of reflections with RATIOs below a suitable threshold.
SIGMA=

```
1
2  -  Default value
```

If sigma is equal to 1 the e.s.d. is the weighted r.m.s. deviation. If sigma is equal to 2 the e.s.d. is the weighted standard deviation.

WEIGHT SCHEME= NPARAMETERS= NCYCLE=
This directive determines the weighting scheme to be used in merging equivalent reflections.
SCHEME=
This parameter determines which of the weighting schemes defined above is to be used in the merging of equivalent reflections, and must take one of the following values:

```
1  -  Default value (unit weights)
2            (modified Gaussian)
3            (statistical)
```

If this parameter is omitted, unit weights are applied (scheme=1).
NPARAMETERS=
This must be set to the number of parameters required to define the weighting scheme, and thus the number of values on the PARAMETERS directive to follow. The default value for this parameter is zero, as schemes 1 and 3 require no parameters.
NCYCLE=
This parameter has a default value of 5 and is the number of cycles of refinement of the weighted mean if scheme 2 is being used in the merge.

PARAMETERS P .....
This directive contains NPARAMETERS values.
P=
For weighting scheme 2, these parameters give the values 'a' to 'h' defined above, and describe the form of the Gaussian distribution.

5.18 Theta-dependent Absorption Correction - \THETABS

```
\THETABS
THETA NTHETAVALUES=
THETAVALUES THETA=
THETACURVE CORRECTION= ........
END
```

For example

```
\THETABS
THETA 16
THETAVALUES
CONT 0   5 10 15 20 25 30 35 40 45 50 55 60 65 70 75
THETACURVE
CONT 3.61  3.60  3.58  3.54  3.50  3.44  3.37  3.30
CONT 3.23  3.16  3.09  3.02  2.96  2.91  2.86  2.82
END
```

   Except when the data has been corrected by a proper analytical correction, a theta dependent correction is **ALWAYS** recommended, since neither a phi scan multi-scan nor DIFABS (section #RDIFABS#) will make a good theta approximation. See Int Tab, Vol II, p295 and 303 for suitable profiles.

   THETA NTHETAVALUES=
      #KNTHETAVALUES=# The number of sampling points on the theta curve.

   THETAVALUES THETA= .....
      The Nthetavalues at which the curve is sampled

   THETACURVE CORRECTION= ......
      The Nthetavalue values of the correction factor profile.

Chapter 6
Atomic And Structural Parameters

6.1 Scope of the atomic and structural parameters Section

The areas covered are:

```
Specifications of atoms and other parameters
Input of atoms and other parameters              - \LIST 5
Modification of lists 5 and 10 on the disc        - \EDIT
Conversion of temperature factors                - \CONVERT
Hydrogen placing                                 - \HYDROGENS
Per-hydrogenation                                - \PERHYDRO
Regularisation of groups in LIST 5               - \REGULARISE
```

6.2 Specifications of atoms and other parameters

There is a consistent syntax thoughout CRYSTALS for refering to atoms and atomic parameters. This was referred to briefly in Chapter 1, and will be defined more fully here.

ATOM SPECIFICATION

There are three different but related ways of specifying an atom or a group of atoms.

TYPE(SERIAL,S,L,TX,TY,TZ)

This specification defines one atom. The various parts of the expression are :

TYPE

The atom type, defined in Chapter 1 in the section on form-factors.

SERIAL

The serial number, in the range 1-9999

Checking of serial numbers

Atoms of the same type are distinguished from one another by having different serial numbers. However, at no stage is a check made to ensure that there is not more than one atom in LIST 5 (atomic parameters) with the same type and serial number. If a routine is searching for an atom with a given type and serial number, the first atom found will always be taken, and any subsequent atoms with the same type and serial number will be ignored.

Serial numbers are considered to be different if they differ from each other by more than 0.0005.

S

’S’ specifies a symmetry operator provided in the unit cell symmetry LIST (LIST 2 - see section #RLIST02#). ’S’ may take any value between ’-NSYM’ and ’+NSYM’, except zero, where ’NSYM’ is the number of symmetry equivalent positions provided in LIST 2. if ’S’ is less than zero, the coordinates of the atom stored in LIST 5 are negated (i.e. inverted through a centre of symmetry at the origin) and then multiplied by the operator specified by the absolute value of ’S’ to generate the new atomic coordinates. ’S’ may be less than zero even if the space group is non-centrosymmetric (*i.e.* introduce a false centre), but must not be greater than ’NSYM’. The default value for ’S’ is ’1’, specifying the first matrix in LIST 2, usually the unit matrix.

L

’L’ specifies the non-primitive lattice translation that is to be added after the coordinates have been modified by the operations given by ’S’. ’L’ must not be greater than the number of allowed non-primitive translations in the space group. The translations provided by ’L’ depend on the lattice type and are given by :

```
 L=    1              2                3                4

P     0,0,0
I     0,0,0         1/2,1/2,1/2
R     0,0,0         1/3,2/3,2/3       2/3,1/3,1/3
F     0,0,0          0,1/2,1/2        1/2, 0 ,1/2      1/2,1/2,0
A     0,0,0          0,1/2,1/2
B     0,0,0         1/2, 0 ,1/2
C     0,0,0         1/2,1/2, 0
```

the default value of ’L’ is ’1’, specifying no non-primitve lattice translation.

TX,TY,TZ

Unit cell translation along the x,y and z directions.

The unit cell translations are added to the coordinates after the ’S’ and ’L’ operations have been performed. The translations may be positive or negative, but must refer to complete unit cell shifts. The default values for ’TX’, ’TY’ and ’TZ’ are all zero, giving no unit cell translations.

The symmetry operations are applied in the order :

```
1.  Centre of symmetry if 'S' negative
2.  Symmetry operator 'S'
3.  Non-primitve lattice translation
4.  Whole unit cell translations 'T(X)', 'T(Y)', 'T(Z)'.
```

```
i.e.

      X'=  [R(s)](+X) + t(s) + L + T(X) + T(Y) + T(Z)
or
      X'=  [R(s)](-X) + t(s) + L + T(X) + T(Y) + T(Z)
```

The format given above is a complete atom definition. For convenience the definition may sometimes be shortened. The obligatory parts are the TYPE and SERIAL. The remaining parameters, S, L, TX, TY, TZ, are optional.

An optional parameter taking its default value may be omitted, though its place must be marked by its associated comma. A series of trailing commas may be omitted.

The following are all equivalent :

```
TYPE(SERIAL,1,1,0,0,0)
TYPE(SERIAL,,,0,0,0)
TYPE(SERIAL,1,,,,0)
TYPE(SERIAL,,,,,)
```

The values of S , L , TX , TY and TZ are exactly those output and used by the distance angles routines under the headings S(I) , L , T(X) , T(Y) and T(Z) respectively. (See the section of the user guide on 'results of refinement').

When the symmetry operators are applied, the actual values of S and L are checked to see that they are reasonable. If the values found are not reasonable, an error message will be output and the job terminated.

In some cases, the symmetry operators are accepted on input, but not used by the routine. The description of the routine will state this.

UNTIL sequences

When a group of atoms lie sequentially in the atom parameter list, there is an abbreviated way to refer to the group.

```
TYPE1(SERIAL1,S,L,TX,TY,TZ)   UNTIL   TYPE2(SERIAL2)
```

This definition specifies all the atoms in the current list starting with the atom TYPE1(SERIAL1) The first atom in the specification must occur before the second atom in the current parameter list, otherwise an error message will be output and the task aborted. If symmetry operators are used, they must be given for the first atom of the sequence, and will be appied to all the atoms in the sequence.

```
Examples


        C(1) until C(6)


Six atoms lying around a centre of symmetry:

        C(1) until C(3) C(1,-1) UNTIL C(3)
```

FIRST AND LAST

These specifications each define one atom. FIRST Refers to the first atom stored in LIST 5 (the model parameters) or LIST 10 (Fourier peaks), and LAST refers to the last atom in the list. If these are used as atom designators, no serial number may be given, but symmetry operators may be. They may be used in until sequences.

```
examples
        LAST
        FIRST(x)
        FIRST(-1) UNTIL C(16)   C(23) UNTIL LAST
```

ALL

This specifies all atoms in the list, can take symmetry operators or parameter names, but cannot be accompanied on the same line by any other atom specifiers.

```
examples
        ALL
        ALL(x)
        ALL(-1)
```

RESIDUE

This specifies all atoms or parameters with the given residue number.

examples
                    RESIDUE(3)
                    RESIDUE(3,X's)

PART

This specifies all atoms or parameters with the given part number.

examples
                    PART(3001)
                    PART(3001,X's)

The part number is constructed from two values, the assembly number and the group number.

*PART NO. = 1000 * ASSEMBLY NO. + GROUP NO.*

The assembly number is normally zero, but a value can be given to all atoms that are involved in a particular disordered area of the structure. E.g. on a disordered methyl all the H atoms could be placed in assembly number 1.

The group number within an assembly groups together those atoms which are simultaneously occupied. E.g. on a disordered methyl, all the H atoms approximately 109 degrees apart would be given the same group number.

The part and group numbers affect the default bonds that are determined by CRYSTALS, and subsequently output in the CIF or summary file. Some bonding rules are applied in the following order of priority:

1. An atom in assembly 0, group 0, will bond to any other nearby atom.
2. Atoms in the same assembly, but with different, non-zero group numbers
   will not bond to each other.
3. Atoms in different assemblies with one zero group number
   will not bond to each other.
4. Atoms in the same assembly and group, but with a negative group number
   will not bond to symmetry related atoms in the same assembly and group.
5. All remaining close contacts will be bonded together.

Rule 3 may be ignored unless you're trying to set up something very special. Rule 4 is useful if you have a group disordered across a symmetry element.

ATOMIC PARAMETER SPECIFICATION

Atomic parameters have a NAME. Some directives permit the use of the parameter name by itself, which implies that parameter for all atoms. The parameter name may be combined with an atom specifier, in which case only the parameter for that atom (or group in an UNTIL sequence) is referenced. Symmetry operators may be used. The normal drop-out rules apply.

Parameter NAMES

The following NAMES are recognised.

X	*Y*	*Z*	*OCC*	*U[ISO]*	*SPARE*
U[11]	*U[22]*	*U[33]*	*U[23]*	*U[13]*	*U[12]*
X'S	*U'S*	*UIJ'S*	*UII'S*		

Examples
  X   *The 'x' coordinate for all atoms*
  C(9,X,Y)   *The 'x' and 'y' coordinates for atom C(9)*
  FIRST(X'S)   *The 'x','y' and 'z' coordinates for the first atom*
  FIRST(U'S) UNTIL C(23)
                   The anisotropic temperature factors for all atoms
                   up to C(23).

Temperature factor definitions
Isotropic temperature factor

The isotropic temperature factor is defined by:

$$T = exp(-8*pi*pi*U[iso]*s**2)$$
                                        where s = sin(theta)/lambda

Anisotropic Temperature Factor

The anisotropic temperature factor (adp) is defined by:

$$T = exp(-2*pi*pi*(h*h*a'*a*u[11]$$
                    $$+k*k*b'*b'*u[22]$$

```
         +l*l*c'*c'*u[33]
      +2.0*k*l*b'*c'*u[23]
      +2.0*h*l*a'*c'*u[13]
      +2.0*h*k*a'*b'*u[12])).
                             where x' are the reciprocal
                             cell parameters and h, k and
                             l are the Miller indices
```

CRYSTALS contains two definitions od Uequiv. Both definitions are acceptable to Acta. The arithmetic mean of the principle axes is often similar to the refined value of Uiso. The geometric mean is more sensitive to long or short axes, and so is more useful in publications. Ugeom is the sphere with the same volume as the ellipsoid.

```
U(arith) = (U1+U2+U3)/3
U(geom)  = (U1*U2*U3)**1/3
                        Where Ui are the principal axes of
                        the orthogonalised tensor.
```

CAUTION

It should be noted that if a set of anisotropic atoms are input with the FLAG key set to anything but 0, then the parameters will be interpreted as Isotropic atoms, or special shapes.

Uequiv

Two expressions are available for the equivalent temperature factor (the geometric or arithmetic mean of the principal components). The Immediate Command 'SET UEQUIV' sets which definition will be used.

```
Ugeom  = (Ui * Uj * Uk)**1/3

Uarith = (Ui + Uj + Uk)/3
                        Where Ui, Uj & Uk are the
                        principal components of U

Ugeom is the radius of the sphere with the same volume as the adp
ellipsoid, and thus gives a good indication of the quality of the
ellipsoid. Uarith is often closer to the value of Uiso, and so is
useful for returning to an isotropic refinement.
```

The Special Shapes

The SPecial Shape keys are

```
type serial occ FLAG x y z u[11]  u[22] u[33] u[23] u[13] u[12] spare
                        U[ISO]                                  spare
                        U[ISO] SIZE                             spare
                        U[ISO] SIZE   DECLINAT AZIMUTH          spare
```

The value of 'FLAG' is used on input of atoms to indicate what kind of patameters will follow, and is used during calculations for the interpretation of the parameters.

FLAG interpretation

The following table shows the interpretation of the FLAG parameter.

```
FLAG  meaning     parameters
'old' types of atoms:

0     Aniso ADP  u[11]  u[22] u[33] u[23] u[13] u[12]
1     Iso ADP     U[ISO]

New 'special' shapes:

2     Sphere      U[ISO] SIZE
3     Line        U[ISO] SIZE   DECLINAT AZIMUTH
4     Ring        U[ISO] SIZE   DECLINAT AZIMUTH
```

The parameters have the following meaning for the new special shapes:

Special U[iso]

U[iso] is related to the 'thickness' of the line, annulus or shell.

Special SIZE

SIZE is the length of the line, or the radius of the annulus or shell.

Special DECLINAT

DECLINAT is the declination angle between the line axis or annulus normal and th z axis of the usual CRYSTALS orthogonal coordinate system, in degrees/100.

Special AZIMUTH

AZIMUTH is the azimuthal angle between the projection of the line axis or annulus normal onto the x - y plane and the x axis of the usual CRYSTALS orthogonal coordinate system, in degrees/100.

If either of these angles is input with a value greater than 5.0, it is assumed that the user has forgotten to divide by 100, which is thus done automatically.

OVERALL PARAMETER SPECIFICATION

Overall parameters are specified simply by their keys. The following overall parameter keys may be given :

```
SCALE       OU[ISO]       DU[ISO]        POLARITY
ENANTIO     EXTPARAM
```

SCALE

This parameter defines the overall scale factor and has a default value of unity. It is the number by which /FC/ must be multiplied to put it onto the scale of /FO/, i.e. /Fo/ = scale*/FC/.

DU[ISO]

This parameter is the dummy overall isotropic temperature factor and has a default value of 0.05.

The dummy overall temperature factor is in no way related to the overall temperature factor, and its use is explained in the input of LIST 12, which comes in the section of the user guide on 'structure factors'.

OU[ISO]

This parameter is the overall isotropic temperature factor and has a default value of 0.05.

POLARITY

This is the Rogers *eta* parameter, and is a multiplier for the imaginary part of the anomalous scattering factor. Setting the value to 1.0 (its default) has the effect of using the imaginary part of the anomalous scattering factor as given. Changing the value to -1.0 has the effect of changing the hand of the model. Setting the value at zero has the effect of removing the contribution of f". However, if contributions from f" are not required, IT IS MORE EFFICIENT to set ANOMALOUS = NO in LIST 23 (structure factor control, see section #RLIST23#). If you need to use f", remember not to apply Friedel's law (LIST 13, section #RLIST13#) during data reduction (section #RDATAREDUC#), and to include anomalous scattering (LIST 3, section #RLIST03# and LIST 23, section #RLIST23#). See D. Rogers, Acta Cryst (1981), A37,734-741. *POLARITY and ENANTIO should not be used simultaniously.*

ENANTIO

This overall parameter is the fractional contribution of F(-h) to the observed structure amplitude, and like the POLARITY parameter is sensitive to the polarity of the structure. It is defined by

```
Fo**2 =(1-x)* F(h)**2 + x*F(-h)**2
```

where x is the ENANTIOpole parameter. A value of 0.0 means the structure stored in LIST 5 is of the correct hand. A value of 1.0 inverts the structure. Its effect on the structure factor is switched on or off by the parameter ENANTIO in LIST 23 (see section #RLIST23#). Computations are more efficient when it is turned off. If the enantiopole is used (or refined) then Friedel's law must not be applied (LIST 13, section #RLIST13#) and anomaloue scattering must be included (LIST 13 and LIST 23). See Howard Flack, Acta Cryst, 1983, A39, 876-881. This parameter is more robust than the POLARITY parameter.

EXTPARAM

This parameter is Larson's extinction parameter , r*, (equation 22 in A.C. Larson, Crystallographic Computing, 1970, 291-294, ed F.R. Ahmed, Munksgaard, Copenhagen , but with V replaced by the cell volume) and has a default value of zero.

Note that many other programs use expression (4), which cannot cope with Neutron data, and gives a value for 'g' which is about 1,000,000 times smaller than 'r*'.

```
g ~= [(e**2/mc**2)**2 . lambda**3/V**2 . Tbar ] . r*
```

Tbar is the absorption weighted mean path length, and is assumed to be stored in LIST 6 (section #RLIST06#) with a key of TBAR . If this key is absent, a default value of 1.0 is used. If extinction is to be included in the model, the mosaic spread should have been set in LIST 13 (section #RLIST13#).

6.3 Input of atoms and other parameters - LIST 5

```
\LIST 5
OVERALL SCALE= DU[ISO]= OU[ISO]= POLARITY= ENANTIO= EXTPARAM=
READ NATOM= NLAYER= NELEMENT= NBATCH=
either ATOM TYPE= SERIAL= OCC= FLAG= X= Y= Z= U[11]= ....U[12]=
or      ATOM TYPE= SERIAL= OCC= FLAG= X= Y= Z= U[ISO]
INDEX P= Q= R= S= ABSOLUTE=
LAYERS SCALE=
ELEMENTS SCALE=
BATCH SCALE=
```

```
\LIST 5
OVERALL SCALE=0.123
READ NATOM=2 NELEMENT=2
ATOM PB 1 FLAG=0 .25 .25 .25 .03 .03 .03 .0 .0 .0
ATOM C 2  X= .23 .13 .67
ELEMENTS 0.8 0.2
END
```

\LIST 5
OVERALL SCALE= DU[ISO]= OU[ISO]= POLARITY= ENANTIO= EXTPARAM=
This directive specifies various parameters that refer to the structure as a whole.
SCALE=
The overall scale factor, default = 1.0
DU[ISO]=
The dummy overall isotropic temperature factor, default = 0.05.
OU[ISO]=
The overall isotropic temperature factor, default = 0.05.
POLARITY=
Rogers *eta* parameter (see above), default = 1.0.
ENANTIO=
Flack enantiopole parameter (see above), default = 0.0.
EXTPARAM=
Larson r* secondary extincion parameter, default = 0.0.
READ NATOM= NLAYER= NELEMENT= NBATCH=
This directive specifies the number of atoms, layer scale factors, element scale factors, and batch scale factors that are to follow.
NATOM=
The number of atom directives to follow, default = 0.
NLAYER=
The number of layer scale factors to follow, default = 0.
NELEMENT=
The number of element scale factors to follow, default = 0.
NBATCH=
The number of batch scale factors to follow, default = 0.
ATOM TYPE= SERIAL= OCC= FLAG= X= Y= Z= U[11]= ..
The parameters for an atom, repeated NATOM times.
TYPE=
The atomic species, an entry for which should exist in LIST 3 (see section #RLIST03#). There is no default value.
SERIAL=
The atoms serial number. There is no default value.
OCC=
This parameter defines the site occupancy **EXCLUDING** special position effects (i.e. is the 'chemical occupancy). The default is 1.0. Special position effects are computed by CRYSTALS and multiplied onto this parameter.
FLAG=
This parameter specifies the type of temperature factor for the atom, and if it is omitted a default value of 1 is assumed. **NOTE** that it **must** be set to 0 for anisotropic atoms.
X= Y= Z=
These parameters specify the atomic coordinates for the atom, for which there are no default values.
U[11]= U[22]= U[33]= U[23]= U[13]= U[12]=
These parameters have different interpretations depending upon the value of FLAG
  If FLAG=0
  These parameters specify the anisotropic temperature factors for the atom and if they are omitted default values of zero are assumed. The order of the cross terms is obtained by dropping 1,2,3 sequentially from [123].
  If FLAG=1
  The first parameter specifies the isotropic temperature factor, which defaults to 0.05.

If FLAG=2,3 or 4, the six parameters represented by u[ij] have the following imterpretation:

```
KEY    shape        parameters


2    Sphere       U[ISO] SIZE
3    Line         U[ISO] SIZE   DECLINAT AZIMUTH
4    Ring         U[ISO] SIZE   DECLINAT AZIMUTH
```

The parameters have the following meaning for the new special shapes:
Special U[iso]
U[iso] is related to the 'thickness' of the line, annulus or shell.
Special SIZE
SIZE is the length of the line, or the radius of the annulus or shell.
Special DECLINAT
DECLINAT is the declination angle between the line axis or annulus normal and th z axis of the usual CRYSTALS orthogonal coordinate system, in degrees/100.
Special AZIMUTH
AZIMUTH is the azimuthal angle between the projection of the line axis or annulus normal onto the x - y plane and the x axis of the usual CRYSTALS orthogonal coordinate system, in degrees/100.
If either of these angles is input with a value greater than 5.0, it is assumed that the user has forgotten to divide by 100, which is thus done automatically.

INDEX P= Q= R= S= ABSOLUTE=
This directive is used to input the constants that define an index for layer scaling. The layer scale index for the reflection with indices HKL is computed from

```
index = (h*p + k*q + l*r + s)
```

and the absolute value is taken if the parameter ABSOLUTE = yes.
P= Q= R=
These parameters have default values of zero.
S=
This parameter has a default value of unity. The zeroth layer must have an index of 1.
ABSOLUTE=

```
NO
YES  -  Default value
```

LAYERS SCALE=
This directive defines the layer scale factors, starting with the scale for an index of 1.
SCALE=
This parameter gives the layer scale, and has a default value of 1. It is repeated NLAYER times.
ELEMENTS SCALE=
This directive defines the scale factors for the elements of a twinned structure. See the chapter on twinned structures.
SCALE=
This parameter gives the element scale factor, and has a default value of 1. It is repeated NELEMENT times - the number of components in the twin.
BATCH SCALE=
This directive defines the batch scale factors.
SCALE=
This parameter gives the batch scale factor, and has a default value of 1. It is repeated NBATCH times. Remember to set appropriate keys in LIST 6

Further examples of parameter input

```
ATOM TYPE=C,SERIAL=4,OCC=1,U[ISO]=0,X=0.027,Y=0.384,Z=0.725,
CONT U[11]=0.075,U[22]=0.048,U[33]=.069
CONT U[23]=-.007,U[13]=.043,U[12]=-.001
ATOM C 5 U[ISO]=0.0 .108,.365,.815,.074
CONT .051 .065 -.015 .048 -.014
ATOM C 2 1 0.05 0.149 0.411 0.651 0 0 0 0 0 0
ATOM C 1 X=0.094,Y=0.343,Z=0.890
ATOM C 3 X=0.050 0.406 0.648
```

6.4 Printing and punching list 5

\PRINT 5
Lists the current LIST 5 to the printer file.
\PUNCH 5 mode

Mode controls the format of the file.

```
    -   Punches the model parameters in CRYSTALS format.
A   -   Punches the model parameters in CRYSTALS format.
B   -   Punches the atomic parameters in XRAY format.
C   -   Punches the atomic parameters in SHELX format.
```

Summary display of LIST 5 - \DISPLAY

```
\DISPLAY LEVEL=
END


\DISP HIGH
END
```

This allows the user to display a summary of the contents of list 5. The output is sent to both monitor and listing channels, so the contents of list 5 can be examined on-line during interactive work. The output produced is more compact than that from PRINT 5, and various levels of detail can be selected. The command required is :-

\DISPLAY   LEVEL=

DISPLAY has one optional parameter.

LEVEL

```
LOW
MEDIUM
HIGH
```

The effects of this parameter are :-

LOW The names of the atoms, overall parameters, and any layer, batch, and element scales in list 5 are displayed.

MEDIUM Each atom in list 5 is displayed with its type, serial, occupancy, isotropic temperature factor (if any), and positional parameters. The values of the overall parameters and of any layer, batch, and element scales are displayed.

HIGH All of the parameters of each atom in list 5 are displayed. The values of the overall parameters, and of any layer, batch, and element scale factors are displayed.

6.5 Editing structural parameters - \EDIT

```
\EDIT INPUTLIST= OUTPUTLIST=
EXECUTE
SAVE
QUIT
MONITOR LEVEL
LIST LEVEL
DELETE  ATOM SPECIFICATIONS   .  .
ATOM TYPE= SERIAL= OCC= FLAG= X= Y= Z= U11= ..
CREATE Z ATOM-SPECIFICATION   ...
SPLIT Z ATOM-SPECIFICATION ...
CENTROID Z ATOM-SPECIFICATION ...
KEEP   Z ATOM-SPECIFICATIONS ...
AFTER   ATOM-SPECIFICATION
MOVE Z ATOM-SPECIFICATION   ...
SELECT ATOM-PARAMETER   OPERATOR   VALUE, . .
SORT TYPE1 TYPE2 ...
SORT KEYWORD
DSORT TYPE1 TYPE2 ...
RENAME ATOM1  ATOM2  (, ATOM1  ATOM2) ...
TYPECHANGE KEYWORD OPERATOR VALUE NEW-ATOM-TYPE
RESET PARAMETER-NAME VALUE ATOM-NAMES
CHANGE  PARAMETER-SPECIFICATION VALUE ...
ADD   VALUE PARAMETERS  ...
SUBTRACT  VALUE  PARAMETERS  ...
MULTIPLY   VALUE   PARAMETERS  ...
DIVIDE   VALUE   PARAMETERS   ...
PERTURB VALUE PARAMETERS ...
SHIFT  V1, V2, V3   ATOM-SPECIFICATION . .
TRANSFORM  R11, R21, R31, . . . R33  ATOM-SPECIFICATION . .
DEORTHOGINAL  ATOM-SPECIFICATION . .
UEQUIV  ATOM-SPECIFICATIONS  .  .
ANISO  ATOM-SPECIFICATIONS  .  .
INSERT IDENTIFIER
SPHERE NEWSERIAL ATOMLIST
RING NEWSERIAL ATOMLIST
LINE NEWSERIAL ATOMLIST
REFORMAT
ROTATE ANGLE POINT VECTOR ATOM-SPECIFICATION
ROTATE ANGLE ATOM VECTOR ATOM-SPECIFICATION
ROTATE ANGLE ATOM1 ATOM2 ATOM-SPECIFICATION
```

```
END


LIST LOW
TYPECHANGE TYPE EQ Q C
SELECT U[ISO] LT 0.1
ADD   0.25 X
RENAME C(1) S(1)
CHANGE  S(1,OCC) UNTIL O(1) .5
KEEP  1 FIRST UNTIL LAST
L L
SPLIT 100 C(45)
DELETE  C(46) UNTIL LAST
RESET OCC 1.0 ALL
```

This is a powerful crystallographic editor for modifying a LIST 5 (the model parameters) or LIST 10 (Fourier peaks). It offers the editing facilities frequently needed for the management of atom parameters, including conditional operations and arithmetic.

EDIT is a semi-interactive command, in that each directive is computed as soon as its input is complete. Since CONTINUE can be used to extend a directive over several lines, completion is indicated be the start of a new directive, or the special directive EXECUTE.

After the terminating END, the resulting list is output to the disc. However if the list has not been changed, a new list will be created only if the list type is being changed (e.g. 10 to 5). The current edited version of the list can be saved at any time to protect against future editing mistakes (the SAVE directive). It is also possible to abandon editing without creating a new list (the QUIT directive).

When used in interactive mode, a new list is created even though errors may have occured during command input unless the QUIT directive is used. In online and batch modes no new list will be created if errors occured during the edit. In this case an error message in generated.

Take care to note that some directives refer to atom or group of atoms, others refer to one or more parameters, and two (CHANGE and SELECT)will refer to either an atom specification or a parameter specification. Although atom definitions can include a series of symmetry operators, the only directives that will use them are those for which the subsequent description explicitly states that the symmetry operators are used. In all other cases, the symmetry information will be read in without any error messages and ignored. Those operations which require a single parameter type as argument (ADD, MULTIPLY etc) will fail if composite parameters ("U'S", etc) are given.

\EDIT INPUTLIST OUTPUTLIST

INPUTLIST

```
5   - Default value, the atomic coordinates
10                  the Fourier peaks search
```

OUTPUTLIST

```
5   - Default value, the atomic coordinates
10                  the Fourier peaks search
```

END
This should be the last directive in the set of modification directives.
EXECUTE
This directive which has no parameters does nothing to the edited list. It is provided to allow the user to see the results of one operation (initiated by the directive whose input is terminated by EXECUTE) before attempting the next.
SAVE
Forces the current atom list to be writen to disk.
QUIT
This directive will cause the edit to be abandoned without the creation of a new list if it is followed by END . If it is followed by any other directive it is ignored.
MONITOR LEVEL
This directive controls the level of monitoring of editing operations. When each operation is performed, the results can be monitored in the monitor channel and in the listing file. Four levels of monitoring are provided. The inital level and the default level used when no value is specified is 'MEDIUM'. The possible values of the parameter 'level' are :-

```
OFF             No monitoring occurs
LOW             Type and serial only are displayed
MEDIUM          Program selects level of display   (default)
HIGH            At least the level represented by
                'MEDIUM' listing is displayed
```

When the program selects a monitor level account is taken of the amount of relevant information for the particular directive. Thus for DELETE only 'type' and 'serial' need be displayed whereas for CHANGE all parameter values are displayed.

LIST LEVEL

This directive produces a list of the current edited list in the monitor output stream and in the listing file. If KEEP has been used, the atoms which will be kept are indicated. The possible values for 'level' are :-

```
OFF             No listing produced
LOW             Type and serial listed
MEDIUM          Type , serial , occ , u[iso] ,
                x , y , z listed
HIGH            All atomic parameters listed
```

DELETE ATOM SPECIFICATIONS . .

All the specified atoms are removed from the current atomic parameter list. Deleted atoms should not be referenced by subsequent directives.

ATOM TYPE SERIAL OCC FLAG X Y Z U11 ..

This directive causes the system to add an atom to the end of the edited list. The format is the same as that used in \LIST 5 (see section #RLIST05#). Values must be provided for 'type' , 'serial' , 'x' , 'y' , and 'z' . Default values are provided for the other parameters as in \LIST 5. Example :

```
ATOM O 1 X = 0.3427 .89004 .09181
```

CREATE Z ATOM-SPECIFICATION ...

This directive applies the symmetry operators given or assumed by default in the atom specification, and creates a set of new atoms from those given. The new atoms are added at the end of the current list. The serial numbers of the new atoms are given by:

```
NEWSERIAL = Z + OLDSERIAL
```

The sequence Z ATOM-SPECIFICATIONS can be repeated. When moving from a centrosymmetric to a non-centrosymmetric space group, for example, atoms formerly related by the centre of symmetry can be generated :

```
CREATE 30 MO(1,-1) UNTIL C(15)

Creates atoms MO(31) until C(45)
```

SPLIT Z ATOM-SPECIFICATION ...

Two new isotropic atoms are added to the end of the atom list for every atom referenced in the atom-specification. These atoms lie at each end of the principal axis of the original atoms anisotropic adp ellipsoid.

The original atoms are not deleted. The sequence Z ATOM-SPECIFICATIONS can be repeated. The new serial numbers are given by

```
NEWSERIAL(1) = Z* OLDSERIAL and
NEWSERIAL(2) = Z* OLDSERIAL +1
```

CENTROID Z ATOM-SPECIFICATION ...

A new atom is created at the centroid of the specified atoms, and with a pseudo adp representing the inertial tensor (ie the 'shape' of the group). The atom TYPE is QC, and its serial Z. The sequence Z ATOM-SPECIFICATIONS can be repeated.

KEEP Z ATOM-SPECIFICATIONS ...

Only the atoms referenced in this directive will be kept in the list, all the others will be lost, even though they can be referenced right up until the final END. The sequence Z ATOM-SPECIFICATIONS can be repeated.

Atoms that are KEPT are moved to the top of the list, and stored in the order in which they are specified on the KEEP directive. Only one KEEP directive may be given. Use CONTINUE if one line isn't long enough for the atom sequence.

The atom specifications may contain symmetry operators, which are used to generate the coordinates of the atoms that are to be retained. 'Z' Is an optional parameter which defines the serial number of the first atom in the specification immediately following it. For each atom thereafter in the current atom specification, the serial number is incremented by one to generate the output serial number. Atoms

whose serial numbers are changed in this way must be referred to in subsequent directives by their new serial numbers. If 'Z' is not given, the atoms retain their old serial numbers.

 If an UNTIL sequence is used after a KEEP directive has been given, it should be used with care, since the order of the new parameter list is different from the input list.

AFTER ATOM-SPECIFICATION ...

This defines the atom in the list after which atoms that are MOVEd should be placed. (See MOVE below). If this directive is omitted, the default option places the first MOVED atom at the head of the list, and successive atoms after it. Once one AFTER directive has been given, atoms are placed behind the given atom in the order in which they are presented on MOVE directives. If no atom specification is given on this directive, subsequent MOVEs will move the atoms to the head of the list.

MOVE Z ATOM-SPECIFICATION ...

This directive moves atoms about in the list and places them in the position defined by the latest AFTER directive. (See the previous directive). This directive does not remove atoms from the list, but simply reorders the list. The sequence Z ATOM-SPECIFICATIONS can be repeated.

 The atom specifications may contain symmetry operators, which are used to generate the coordinates of the atoms that are to be moved. 'Z' is an optional parameter which defines the serial number of the first atom in the specification immediately following it. For each atom thereafter in the current atom specification, the serial number is incremented by one to generate the output serial number. Atoms whose serial numbers are changed in this way must be referred to in subsequent directives by their new serial numbers. If no 'Z' is given, the atoms retain their old serial numbers.

 If an UNTIL sequence is used after one or more MOVE directives have been given, it should be used with care, since the order of the new parameter list is different from the input list.

SELECT ATOM-PARAMETER OPERATOR VALUE, . .

This directive selects and retains atoms with parameters satisfying the specified conditions. Only atoms that satisfy ALL the selection criteria, whether these are in the same or different directives, will be kept. All other atoms will be deleted from the list.

 The operators allowed are :

```
EQ              equal
NE              not equal
GT              greater than
GE              greater than or equal to
LT              less than
LE              less than or equal to
```

Examples of the SELECT directive are :

```
SELECT SERIAL LT 50
SELECT OCC GT 0.5, OCC LT 1.5
SELECT C(1,X) LT 1., C(1,X) GT 0.
SELECT TYPE NE Q
```

This example will only retain atoms with serial numbers less than 50 and occupancies between 0.5 and 1.5. The 'X' parameter of atom c(1) must also lie between 0.0 and 1.0 oterwise it will be rejected, and any atoms of type Q will be deleted.

SORT TYPE1 TYPE2 ...
SORT KEYWORD

This directive has two formats, and is used to sort the atoms stored in LIST 5 into a user-defined order. The default action sorts the atoms on their types and serial numbers. The types are taken in the order found in LIST 5, and atoms of each type are grouped together. In each group the atoms are arranged by ascending serial number. The order of the types of atoms may also be determined by specifying them explicitly on the SORT directive, or by a mixture of these methods.

 In the second format, a keyword corresponding to an atom parameter name (as defined in LIST 5, see section #RLIST05#) is given, and the whole list sorted on increasing value of the specified parameter. Note that sorting on TYPE will give results depending on the 'collating sequence' of the computer. Fortunately, this generally leads to alphabetic sorting.

 SORT sorts the whole list 5, and cancels any existing KEEP directives.

DSORT TYPE1 TYPE2 ...
DSORT KEYWORD

This directive is exactly analogous to SORT, above, except that it sorts into descending order.

RENAME ATOM1 ATOM2 (, ATOM1 ATOM2) ...

This directive requires pairs of atom specifications (optionally separated by a comma). The TYPE and SERIAL of 'atom1' are changed to those of 'atom2'. Atom1 must exist in LIST 5, atom2 must NOT exist in LIST 5. An atom can be renamed repeatedly. If atom1 contains symmetry operators, these are applied to the coordinates of the renamed atom. An atom cannot be renamed to itself in a single step.

TYPECHANGE KEYWORD OPERATOR VALUE NEW-ATOM-TYPE

This directive conditionally changes the TYPES of atoms. If an atomic parameter selected by the keyword (see sort above) satisfies the conditions defined by the 'operator' and 'value' (see SELECT above), then the TYPE of the atom is changed to 'new-atom-type'.

```
TYPECHANGE OCC GT 1.2 O
                    If Occ large, convert to oxygen
TYPECHANGE U[ISO] LE 0.03 N
                    If Uiso small, convert to nitrogen
TYPECHANGE TYPE EQ Q C
                    Convert peaks (type Q) to carbon
```

RESET PARAMETER-NAME VALUE ATOM LIST
This directive assigns the given value to the named parameter for all the atoms in the atom list

```
RESET OCC 1.0 ALL
RESET OCC .5 O(1) O(2) O(3)
RESET U[11] .05 C(27) UNTIL C(50)
```

CHANGE ARG(1) ARG(2) ARG(3) .
There are two possible formats for each 'ARG(i)' on this directive. the first is :

```
PARAMETER(i)   VALUE(i)
```

If ARG(i) is of this form, the specified parameter or parameters are changed to the value VALUE(i) . If PARAMETER(i) defines one or more atomic parameters, then the symmetry operators found or inserted by default are applied to the resulting set of atomic parameters. For overall parameters, no symmetry information can be provided. The VALUE associated with this argument must always be present.
    The second form of ARG(i) on this directive is :

```
ATOM-SPECIFICATION
```

For this form of ARG(i) , the symmetry operators given in the atom specification or assumed by default are applied, but no other atomic parameter is explicitly altered. There is no VALUE associated with ARG(i) in this format.
    The two different types of argument on this directive may be used interchangeably :

```
CHANGE  S(1,OCC) UNTIL O(1) .5
CONT    C(1,-2,1) UNTIL C(12)
CONT    C(13,X) .0179
```

ADD VALUE PARAMETERS ...
SUBTRACT VALUE PARAMETERS ...
MULTIPLY VALUE PARAMETERS ...
DIVIDE VALUE PARAMETERS ...
These directives causes the 'value' to be applied to the parameter. 'PARAMETER(I)' may be an overall parameter, or a single atomic parameter of one or more atoms, as defined above. Any symmetry operators given with this directive will be ignored. Note that the parameter SERIAL is numeric, and so can be arithmetically modified.

PERTURB VALUE PARAMETERS ...
This directive perturbs the specified parameters using a rnadom number generator. The VALUE is the requested rms perturbation, in the natural units of the parameters. The mean deviation applied should be approximately zero, and the rms deviation applied should be approximately that requested.

SHIFT V1, V2, V3   ATOM-SPECIFICATION . .
This directive reads the three numbers of a shift vector, which must be in the same coodrinate system as the atomic parameters, and applies it to the parameters in the atom specification. This directive does not create new atoms, but simply modifies those already present. Any symmetry operators given are applied before the translation.

TRANSFORM R11, R21, R31, . . . R33 ATOM SPECIFICATION . .
This directive reads the nine numbers of a transformation matrix, which must be separated by commas or spaces, and applies the matrix to the atoms given in the atom specification. This directive does not create new atoms, but simply modifies those already present. Any symmetry operators given are applied before the rotation.

DEORTHOGINAL ATOM SPECIFICATION . .
This directive applies the matrix vector saved by a previous MOLAX SAVE directive to the atoms given in the atom specification. THEIR ORIGINAL COORDIANATES x,y,z MUST be in the MOLAX coordinate (Angstrom) system This directive does not create new atoms, but simply modifies those already present. Symmetry operators are not permitted.

UEQUIV ATOM SPECIFICATIONS . .
The specified atoms to be converted so that they have isotropic temperature factors, U(equiv), defined

by the SET UEQUIV command. IT IS NOT simply related to the diagonal elements of U(aniso). If an atom is already isotropic, no action is taken. If this directive is given with no arguments, all the atoms in the current atomic parameter list are converted to isotropic temperature factors. Physically impossible values are not rejected. Symmetry operators are ignored.

ANISO ATOM SPECIFICATIONS . .

This directive causes all the specified atoms to be converted so that they have anisotropic temperature factors. If an atom is already anisotropic, no action is taken, and any symmetry operators given are ignored. If this directive is given with no arguments, all the atoms in the current atomic parameter list are converted to anisotropic temperature factors.

Note that the anisotropic temperature factor produced by this operation is in fact still spherically symmetrical, and that the s.f.l.s. routines automatically ensure that when the temperature factor of an atom is to be refined, it is in the correct form.

INSERT IDENTIFIER=NAME

This directive inserts the value of the named identifier into the parameter 'SPARE' in the atom list, replacing any previous value (except 'RESIDUE' which uses the 'RESIDUE' paramter in the atom list). SPARE is normally used to hold rho after Fourier maps.

Currently available values for NAME are

ELECTRON - This inserts the atomic electron count calculated from the form factor
WEIGHT - This inserts the atomic weight from LIST 29 (see section #RLIST29#).
RESIDUE - This inserts a residue number into the 'RESIDUE' slot of list 5 replacing any previous value, such that all connected atoms have the same residue number and each molecule has a different residue number.
NCONN - This inserts the number of atoms connected to an atom, using the list of bonds.
RELAX - This inserts an ID, based upon the bonding topology and atomic types of the atoms. Atoms at topologically identical positions will be given the same ID. (e.g. the terminal F's in a CF3 group).

SPHERE NEWSERIAL ATOMLIST

This creates a 'shell' shape from the specified atom list. The centre of the shell is at the centre of gravity, the size is the mean distance of the given atoms from the centre, and the occupancy is equal to the sum of the occupancies of the atoms listed. U[iso] is the mean of the U[iso] or Ueqiv of the listed atoms. The atom TYPE is QS, with the given serial number. The original atoms are not deleted, though they should be or their occupancy set to zero. The atom type, QS, should be changed to something appropriate.

RING NEWSERIAL ATOMLIST

This creates an 'annulus' shape from the specified atom list. The centre of the ring is at the centre of gravity, the size is the mean distance of the given atoms from the centre, and the occupancy is equal to the sum of the occupancies of the atoms listed. U[iso] is the mean of the U[iso] or Ueqiv of the listed atoms. The atom TYPE is QR, with the given serial number. The original atoms are not deleted, though they should be or their occupancy set to zero. The atom type, QS, should be changed to something appropriate. The DECLINATION and AZIMUTH are computed from the constituent atoms.

LINE NEWSERIAL ATOMLIST

This creates an 'line' shape from the specified atom list. The centre of the line is at the centre of gravity, the size is twice the mean distance of the given atoms from the centre, and the occupancy is equal to the sum of the occupancies of the atoms listed. U[iso] is the mean of the U[iso] or Ueqiv of the listed atoms. The atom TYPE is QL, with the given serial number. The original atoms are not deleted, though they should be or their occupancy set to zero. The atom type, QS, should be changed to something appropriate. The DECLINATION and AZIMUTH are computed from the constituent atoms.

REFORMAT

This directive converts an old (non-FLAG) version of LIST 5 (see section #RLIST05#) to the new format (extra parameters, old U[iso] slot now used as a flag and u[11] used for u[iso]).

ROTATE

This directive rotates a group of atoms a certain number of degrees around a specified vector. The rotation is carried out in orthogonal space so preserves the geometry of the group.

There are three options available: ROTATE D X Y Z VX VY VZ atom-specification # ROTATE D ATOM1 VX VY VZ atom-specification # ROTATE D ATOM1 ATOMS2 atom-specification # The first rotates the specified atoms, D degrees around the vector VX,VY,VZ keeping point X,Y,Z fixed. (X,Y,Z and VX,VY,VZ are given in crystal fractions).

The second notation uses ATOM1 instead of X,Y,Z to specify the fixed point.

The third notation uses ATOM1 to specify the fixed point and the vector from ATOM1 to ATOM2 to rotate around.

The rotation is D degrees anti-clockwise, when the specified vector is pointing towards you.

1) Rotate the hydrogens of a methyl group by sixty degrees.

```
\EDIT
ROTATE 60 C(1) C(2) H(20) H(21) H(22)
END
```

2) Turn a phenyl ring through 30 degrees around its external connecting bond, c(1) to c(20).

```
\EDIT
ROTATE 30 C(1) C(20) C(21) C(22) C(23) C(24) C(25)
END
```

3) Rotate a residue 90 degrees about the a-direction from its centroid, QC(1) (see also CENTROID and INSERT RESIDUE directives)

```
\EDIT
INSERT RESIDUE
CENTROID 1 RESIDUE(1)
ROTATE 90 QC(1) 1 0 0 RESIDUE(1)
END
```

6.6 Reorganisation of lists 5 and 10 - \REGROUP

```
\REGROUP INPUTLIST= OUTPUTLIST=
SELECT MOVE= KEEP= MONITOR= SEQUENCE= SYMMETRY= TRANSLATION= GROUP=
END
```

```
\REGROUP
SELECT MOVE=1.6,MONITOR=HIGH
END
```

This routine offers a way of re-ordering the atoms in LIST 5 (atomic parameters) or LIST 10 (Fourier peaks), so that related atoms or peaks form a sequential group in the list, and the coordinates put the atoms as close together as possible.

THIS ROUTINE DOES NOT USE LIST 29 (atomic properties) to get bonding distances, but uses a single overall distance.

In this routine, a set of distances is calculated about each atom or peak in the list in turn. For each atom or peak in the list below the current pivot, the minimum contact distance is chosen, and if this is less than a user specified maximum, the atom or peak is moved up the list to a position directly below the pivot. (The MOVE parameter). When more than one atom or peak is moved, their relative order is preserved as they are inserted behind the current pivot atom. As well as reordering the list, the necessary symmetry operators are applied to the positional and thermal parameters to bring the atom or peak into the same part of the unit cell as the current pivot atom. The result of this process is to bring related atoms together in the list, and to place all the atoms in the same part of the unit cell. Setting the GROUP parameter to YES causes the PART to be incremented between isolated parts of the structure.

\REGROUP INPUTLIST= OUTPUTLIST=
INPUTLIST=

```
5   - Default value, the atomic coordinates
10                 the Fourier peaks search
```

OUTPUTLIST=

```
5   - Default value, the atomic coordinates
10                 the Fourier peaks search
```

SELECT MOVE= KEEP= MONITOR= SEQUENCE= SYMMETRY= TRANSLATION= GROUP=
MOVE=
This parameter has a default value of 2.0, and is the distance below which atoms or peaks are considered to be bonded, and are thus moved about the cell and relocated in LIST 5 (atomic parameters).
KEEP=
This is the maximum number of atoms that the final output list can contain. If this parameter is omitted, all the atoms are output. If MOVE is used to move the atoms around, it is unwise to use the KEEP parameter,since some of the original input atoms may find their way to the bottom of the list and be eliminated. (The default value is 1000000).
MONITOR=

```
LOW  - Default value
HIGH
```

If MONITOR is HIGH, then each pivot atom and its associated moved atoms are listed, as well as any deleted atoms. If MONITOR is LOW, the moved atoms are not listed.

SEQUENCE

```
NO   -  Default value
YES
```

If SEQUENCE is YES, the outputlist is resquenced as described above. If SEQUENCE is NO, the serial numbers of the atoms are not changed from the original list.

SYMMETRY=

This parameter controls the use of symmetry information in the calculation of contacts, and can take three values.

```
SPACEGROUP  -  Default value. The full spacegroup symmetry is used in
                            all computations
PATTERSON.     A centre of symmetry in introduced, and the translational
               parts of the symmetry operators are dropped.
NONE.          Only the identity operator is used.
```

TRANSLATION=

This parameter controls the application of cell translations in the calculation of contacts, and can take the values YES or NO

GROUP

```
NO   -  Default value
YES
```

If GROUP is YES, the PART parameter for each atom is set.

6.7 Repositioning of atoms - \COLLECT

This routine changes the atom coordinates so as to form a 'molecule' using the covalent radii given in LIST 29 (atomic properties - see section #RLIST29#). The atom TYPE, SERIAL and order in LIST 5 (atomic parameters - see section #RLIST05#) is not changed.

\COLLECT INPUTLIST= OUTPUTLIST=
INPUTLIST=

```
5  -  Default value, the atomic coordinates
10                the Fourier peaks search
```

OUTPUTLIST=

```
5  -  Default value, the atomic coordinates
10                the Fourier peaks search
```

SELECT MONITOR= TOLERANCE= TYPE= SYMMETRY= TRANSLATION=
MONITOR=

```
LOW   -  Default value
HIGH
```

If MONITOR is HIGH, then each pivot atom and its associated moved atoms are listed, as well as any deleted atoms. If MONITOR is LOW, only deleted atoms are listed.

TOLERANCE=

The tolerance is added to the sum of the co-valent radii taken from LIST 29 (atomic properties - see section #RLIST29#) to give a value used for determining inter-atomic bonds. The default is 0.2 A.

TYPE=

```
ALL
PEAKS
```

If TYPE equals ALL, then the coodinates of all atoms are liable to be modified by the symmetry operators in order to assemble a single fragment. If TYPE equals PEAKS, then only the peaks are moved to bring them as close as possible to existing atoms.

SYMMETRY=

This parameter controls the use of symmetry information in the calculation of contacts, and can take three values.

```
    SPACEGROUP  -  Default value. The full spacegroup symmetry is used in
                        all computations
    PATTERSON.     A centre of symmetry in introduced, and the translational
                   parts of the symmetry operators are dropped.
    NONE.          Only the identity operator is used.
```

TRANSLATION=
This parameter controls the application of cell translations in the calculation of contacts, and can take the values YES or NO

6.8 Conversion of temperature factors - \CONVERT

```
\CONVERT INPUTLIST= OUTPUTLIST= CROSSTERMS=
END


\CONVERT
END
```

This routine will convert the temperature factors of a set of atoms into the correct form when their temperature factor, t, is given by :

```
    T = exp(-B[iso]*S**2)      where s = sin(theta)/lambda.


or for an anisotropic atom :

    T = exp(-(h*h*b[11] + k*k*b[22] + l*l*b[33]
        + k*l*2*b[23] + h*l*2*b[13] + h*k*2*b[12]))
```

The cross terms stored in the original LIST 5 (the model parameters) may either be B[IJ] or 2*B[IJ] . (The correct form of the temperature factor, in terms of u[ii]'s and u[ij]'s, is given in the section on the input of LIST 5). After conversion, the atoms are output to the disc as a new LIST 5. Remember that if U[ISO] is non-zero, (its default at atom input is 0.05) the U[IJ] are ignored and so will not be converted.

\CONVERT INPUTLIST= OUTPUTLIST= CROSSTERMS=
This is the command which initiates the routine to convert the temperature factors.

INPUTLIST=

```
    5   -  Default value, the atomic coordinates
    10                 the Fourier peaks search
```

OUTPUTLIST=

```
    5   -  Default value, the atomic coordinates
    10                 the Fourier peaks search
```

CROSSTERMS=

```
    B[IJ]   -  Default value.
    2B[IJ]
```

6.9 Hydrogen placing - \HYDROGENS

```
\HYDROGENS INPUTLIST= OUTPUTLIST=
DISTANCE  D
SERIAL    N
U[ISO]    U
U[ISO]    NEXT    MULT
AFTER     TYPE(SERIAL)
PHENYL    X R(1) R(2) R(3) R(4) R(5)
H33       X R(1) R(2)
H23       X R(1) R(2)
H13       X R(1) R(2) R(3)
H22       X R(1) R(2)
H12       X R(1) R(2)
H11       X R(1)
HBOND     DONOR ACCEPTOR
END


\HYDROGENS
DISTANCE  1.09
U[ISO]    NEXT    1.2
H33    C(7) C(6) R(5)
H22    C(14) C(15) C(13)
END
```

This routine computes the coordinates of hydrogen atoms bonded to a target atom. The hybridisation of the

 target atom and the identifiers of atoms bonded to it must be given.

\HYDROGENS INPUTLIST= OUTPUTLIST=

   INPUTLIST=

```
5   -  Default value, the atomic coordinates
10                 the Fourier peaks search
```

   OUTPUTLIST=

```
5   -  Default value, the atomic coordinates
10                 the Fourier peaks search
```

DISTANCE D
This sets the central atom-hydrogen atom distance to 'D' angstroms. The default value is 1.0. The current value of 'D' remains in force until another 'DISTANCE' directive is given.

SERIAL N
This sets the serial number of the next hydrogen atom to be added to LIST 5 (atomic parameters) to 'N'. The default value is 1. Subsequent hydrogen atoms will have the serial numbers 'N+1', 'N+2', etc., until the next 'SERIAL' directive is input.

U[ISO]   U
This directive sets the isotropic temperature factor of each hydrogen atom to 'U' angstroms squared, and remains in force until another 'U[ISO]' directive is given. If no values is given for U, the next definition is used.

U[ISO]   NEXT   MULT
This is an alternatine form of the preceding directive. It sets the isotropic temperature factor of each hydrogen atom to 'MULT' times the equivalent temperature factor of the atom it is bonded to. The default value is 1.2. The directive remains in force until another 'U[ISO]' directive is given.

AFTER   TYPE(SERIAL)
The hydrogen atoms generated by the placing routines are inserted in the new LIST 5 (atomic parameters) after the atom 'TYPE(SERIAL)'. This directive must appear immediately after the directive that generated the hydrogen atom coordinates, and applies only to that group of hydrogen atoms. If no 'AFTER' directive is given, the new hydrogen atoms are added at the end of the current LIST 5 (atomic parameters).

PHENYL X R(1) R(2) R(3) R(4) R(5)
This generates the coordinates of the five hydrogen atoms of a phenyl group. The first atom specified must be the atom that bonds the phenyl group to the rest of the structure, and the other atoms must be in the order of connectivity.

H33 X R(1) R(2)
This geneates the hydrogen atoms of a methyl group. The methyl carbon is the first atom specified, and the hydrogen atoms are generated so that one of them is trans with respect to the third atom specified, R(2).

```
H
 \
H-X-R(1)-R(2)
 /
H
```
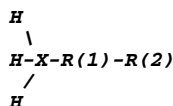
H23 X R(1) R(2)
This generates the coordinates of two hydrogen atoms on an sp3 atom X.

```
H    R(1)
 \ /
  X
 / \
H    R(2)
```
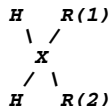
H13 X R(1) R(2) R(3)
This generates the coordinates of one hydrogen atom on an sp3 atom X.

```
    R(1)
   /
H- X-R(2)
   \
    R(3)
```
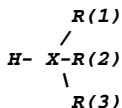
H22 X R(1) R(2)
This generates the coordinates of two hydrogen atoms on an sp2 atom X

```
   H          R(2)
    \         /
      X=R(1)
     /
   H
```
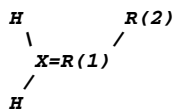
H12 X R(1) R(2)

    This generates the coordinates of one hydrogen atom on an sp2 atom X.

```
     H
      \
       X=R(1)
      /
   R(2)
```
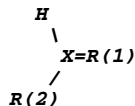
H11 X R

    This generates the coordinates of the single hydrogen atom bonded to an SP hybridised atom.

HBOND X R

    This generates a single H atom 'DISTANCE' angstroms from the donor in the direction of the acceptor. X is the donor, R the acceptor.

```
     X-H....R
```

Place Hydrogen atoms on the following fragment:

```
 C(1)          C(5)
    \          /
       C(2)=C(3)
              \
               C(4)-Br(1)
```
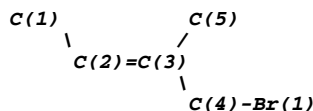
```
 \HYDROGENS
 DISTANCE 0.99
 U[ISO]    0.06
 H33 C(1) C(2) C(3)
 AFTER C(1)
 H12 C(2) C(1) C(3)
 AFTER C(2)
 H23 C(4) Br(1) C(3)
 AFTER C(4)
 H33 C(5) C(3) C(4)
 END
```

6.10 Perhydrogenation - \PERHYDRO

```
 \PERHYDRO INPUTLIST= OUTPUTLIST=
 DISTANCE  D
 SERIAL    N
 U[ISO]    U
 U[ISO]    NEXT    MULT
 ACTION    MODE
 TYPE      C or N
 END
```

```
 \PERHYDRO
 U[ISO] NEXT 1.0
 END
```

This command scans the atomic coordinates for carbon atoms, attempts to assign their hybridisation state (on the basis of bond lengths) and then generates \HYDROGEN commands to create any necessary hydrogen atoms. Existing Hydrogen atoms are not replaced by this routine.

    The generated commands may be processed internally by CRYSTALS without the user needing to see them, or they may be sent to the external files for later use. This is the default mode. If no new hydrogen atoms are generated, no new external files are created.

    The external files are called DELH.DAT and PERH.DAT, with DELH containing an entry for every atom created by PERH. Executing DELH and PERH will delete existing named atoms, and recreate them geometrically.

\PERHYDRO INPUTLIST= OUTPUTLIST=

    INPUTLIST=

```
 5   - Default value, the atomic coordinates
 10                  the Fourier peaks search
```

    OUTPUTLIST

```
    5   -  Default value, the atomic coordinates
    10                    the Fourier peaks search
```

DISTANCE D

This sets the central atom-hydrogen atom distance to 'D' angstroms. The default value is 1.0. The current value of 'D' remains in force until another 'DISTANCE' directive is given.

SERIAL N

This sets the serial number of the next hydrogen atom to be added to LIST 5 to 'N'. The default value is 1. Subsequent hydrogen atoms will have the serial numbers 'N+1', 'N+2', etc., until the next 'SERIAL' directive is input.

U[ISO] U

This directive sets the isotropic temperature factor associated with each hydrogen atom to 'U' angstroms squared. The default value is 0.05. The directive remains in force until another 'U[ISO]' directive is given.

U[ISO] NEXT MULT

This is an alternatine form of the preceding directive. It sets the isotropic temperature factor associated with each hydrogen atom to 'MULT' times the equivalent temperature factor of the atom it is bonded to. The default value is 1.2. The directive remains in force until another 'U[ISO]' directive is given.

ACTION MODE

MODE

```
    NORMAL
    PUNCH
    BOTH      -  Default value.
```

NORMAL causes internal commands to be generated and executed. PUNCH causes output to the PUNCH file only. BOTH forces both actions.

TYPE MODE

MODE

```
    C   -  Default value.
    N
```

C enables the program to place hydrogen atoms on carbon atoms.

P enables the program to place hydrogen atoms on nitrogen.

It is advisable to perform placement on C before N, since the hybridisation states of C are more clearly defined.

6.11 Regularisation of atomic groups - \REGULARISE

```
\REGULARISE      MODE
COMPARE
KEEP
REPLACE
AUGMENT
METHOD NUMBER
GROUP NUMBER
TARGET Atom Specifications
IDEAL  Atom Specifications
RENAME offset number
CAMERON
MAP Atom Specifications
ONTO Atom Specifications
SYSTEM a b c alpha beta gamma
ATOM     x    y    z
CP-RING x
HEXAGON x
OCTAHEDRON x y z
PHENYL
SQP x y z
SQUARE x y
TBP x z
TETRAHEDRON x
END
```

```
\REGULARISE REPLACE
GROUP 6
TARGET C(1) UNTIL C(6)
PHENYL
END
```

This routine calculates a fit between the coordinates of a group of atoms in LIST 5 (atomic parameters) and another group. The calculated fitting matrix may be used to compare the geometry of two groups, or it may be applied to transform the new coordinates which will then replace the existing group in LIST 5 (D. J. Watkin, Act Cryst (1980). A36,975).

In this section, the group of atoms in LIST 5 to whose coordinates the fit is made is referred to as the 'TARGET atoms', and the group to be fitted onto that group is referred to as the 'IDEAL atoms'.

The source of the 'IDEAL atoms' can be the LIST 5, a pre-stored idealised geometry, or values read in from the directives. Those directives that refer to LIST 5 use the usual CRYSTALS formats for atom specifications. Once a transformation has been found, this can be used as the basis for naming one fragment based on the names of another.

Input for REGULARISE

The input to REGULARISE must define the groups to be fitted together, the method used for fitting , and the use to be made of the results. The user must ensure that corresponding atoms are specified in the same positions of the 'TARGET' and 'IDEAL' group definitions, so the program knows which pairs of atoms are to be matched. It is not necessary to have co-ordinates of every atom in the TARGET fragment. The inclusion of atom specifications for which coordinates do not exist in the parameter list indicates that the procedure must generate coordinates for these atoms. This allows the user to give a type and serial to new atoms created by the procedure. Any 'atoms' without coordinates are not included in the fitting process.

The maximum number of atom IDENTIFIERS permitted on an TARGET or IDEAL directive is about 250. Note that an UNTIL sequence only counts as two identifiers. The number of implied atoms permitted is very large.

The 'IDEAL' group may be given in various ways. For calculations on a single structure, it may be extracted from the stored data in the same way as the 'TARGET' group. In this case however, all the atoms must previously exist. Alternatively, explicit co-ordinates may be given in a system defined by the user, or a predefined group may be used. In any case all the positional parameters of the atoms in the 'IDEAL' group will be known before the calculation begins.

Output from REGULARISE

The output from REGULARISE includes the fragment centroids, their sums and differences and the transformation fitting the IDEAL onto the TARGET.

Method of calculation

The centroid of each fragment is moved to the origin. The atomic coordinates are converted to an orthogonal system and rotated to an 'inertial tensor' system (to help condition the L.S. matrix).

The fitting calculation is either constrained to be a pure rotation- inversion, or is a free linear transformation (rotation-diltion). If requested, the pure rotation component of the calculated rotation-dilation matrix is extracted. The calculated matrix is applied to the co-ordinated of the 'IDEAL' group, which is then converted back to crystal fractions, for comparison with the TARGET.

WARNING

The 3 by 3 transformation matrices generated at various stages may well be singular, especially if no rotation is defined about one of the axes. To combat possible problems with matrix inversion, a Moore-Penrose type matrix inverter is used. Even so, the user should be aware that there may be no unique solution to his problem. For example, when a planar fragment is fitted to an almost planar fragment one fit may involve inversion of the non-planar fragment. Inversion can be prevented by using Method 3. Note also that if almost planar groups are being fitted, the dilation factor perpendicular to the plane may be very large, and thus have an undesirable effect if applied to atoms far from the plane.

\REGULARISE   MODE

MODE is an optional parameter.

MODE

```
COMPARE      -       Default value
KEEP
REPLACE
AUGMENT
```

The effects are :-

COMPARE The specified groups are only compared. The translations and rotations necessary to match the groups will be calculated but not applied.

KEEP The specified groups will be compared and the calculated transformations applied. The TARGET atoms are kept, and atoms whose parameters have been calculated will be stored at the end of the new LIST 5.

REPLACE The specified groups will be compared and the calculated transformations applied. The new atoms whose parameters have been calculated will be placed at the end of LIST 5 and the old atoms deleted form the list.

AUGMENT The specified groups will be compared and the calculated transformations applied. The TARGET atoms which actually exist in LIST 5 are retained unaltered. Parameters that have been calculated for dummy atoms (represented by a name only in the TARGET list) will be placed at the end of the new LIST 5.

For REPLACE and KEEP the 'IDEAL' coordinates define the geometry to be preserved, i.e. the model, and the 'TARGET' coordinates specify where, in what orientation and with what atom identifiers the model is to be placed. That is, the TARGET structure is replaced by the IDEAL.

COMPARE
KEEP
REPLACE
AUGMENT
These 4 directives override the option specified by the MODE parameter of the REGULARISE command. The next group calculated will be treated in the specified mode. See the description of MODE for details. There are no parameters

METHOD NUMBER
This directive selects the method for matching the groups by giving its number from the following list:-

```
Number     Method
------     ------
  1        Rotation  component of  rotation-dilation
           matrix applied. ( default )
  2        Rotation-dilation  matrix  calculated and
           applied.
  3        Pure  rotation matrix  calculated  by the
           Kabsch method and applied. This algorithm
           preserves chirality.
```

GROUP NUMBER
This directive specifies the number of atoms in the groups to be matched. It should be the first directive for each group of atoms. The appearance of a second or subsequent GROUP directive in the input initiates the calculation for the previous group.

TARGET Atom Specifications
This directive is used to specify the 'TARGET' group of atoms. The directive will carry a series of atom specifications which will define the positions of the 'TARGET' atoms and the names of any atoms to be created by the routine. Atoms which exist in LIST 5 and atoms to be created can appear in any order in the TARGET group , although the order should be such that corresponding pairs of atoms appear at the same relative positions in the 'TARGET' and 'IDEAL' groups.

IDEAL Atom Specifications
This directive is used to specify a group of 'IDEAL' atoms to be taken from the stored LIST 5. Every atom on this directive must exist.

SYSTEM a b c alpha beta gamma
This directive is will change the co-ordinate system used to interpret any subsequent ATOM directives.
The initial co-ordinate system has orthogonal axes of unit length and is equivalent to :-

SYSTEM 1.0 1.0 1.0 90.0 90.0 90.0

Values must be given for a', b', and c', the angles default to 90.0.

ATOM x y z
This directive allows the cordinates of a single atom to be specified, in fractional co-ordinates in the current co-ordintate system. It must be followed by three decimal numbers which will be the X, Y, and Z coordinates of the atom.

RENAME offset number
This directive can only be used after previous directives have been used to match one group onto another (REGULARISE COMPARE), and enables the use of the MAP and ONTO directives. The MAP list of atoms is transformed by the existing transformation matrix (which may have been computed from only a few specified atoms). Each atom is then compared with the ONTO list, and the TYPE and SERIAL of the MAP atom used to generate a TYPE and SERIAL for the closest ONTO atom.

OFFSET
The serial numbers of the atoms in the group being re-named are related to those of the master group by an increment of 'OFFSET'. The default value is 100

NUMBER
If the number of atoms supplied on the following MAP and ONTO directives does not match NUMBER, a warning is printed.

CAMERON
This matches atoms as in RENUMBER, but only creates CAMERON files with atoms transformed into the common coordinate system.

MAP Atom Specifications
This specifies the atoms whose TYPE and SERIAL are to be propogated into the ONTO atoms. The

atoms can be in any order.

ONTO Atom Specifications

This specifies the atoms to be renamed. The atoms may be in any order and have any TYPE, but there must be EXACTLY as many as on the MAP directive. The atoms can have any TYPE, but must have unique SERIAL numbers.

HEXAGON X

The 'IDEAL' group is a regular hexagon with a side of length 'X'. The default for x is 1.0.

PHENYL

The same as HEXAGON with a fixed side of 1.39.

CP-RING X

The 'IDEAL' group is a regular pentagon with a side of length 'X'. The default for x is 1.4.

SQUARE X Y

The 'IDEAL' group is a rectangle with atoms at (x,0,0) , (0,y,0) , (-x,0,0) , (0,-y,0) . The parameters X and Y specify the size of the group to be used.

OCTAHEDRON X Y Z

The 'IDEAL' group is an octahedron with atoms at (0,0,0) , (-x,0,0) , (0,y,0) , (x,0,0) , (0,-y,0) , (0,0,z) , (0,0,-z). The parameters X, Y and Z specify the size of the octahedron. 'z' defaults to 'y' defaults to 'x' defaults to '1.0'

SQP X Y Z

The 'IDEAL' group is a square pyramid with atoms at (0,0,0) , (x,0,0) , (0,y,0) , (-x,0,0) , (0,-y,0) , (0,0,z). The parameters X, Y and Z specify the size of the octahedron. 'z' defaults to 'y' defaults to 'x' defaults to '1.0'

TBP X Z

The 'IDEAL' group is a trigonal bipyramid with atoms at (0,0,0) , (x,0,0) , (-x/2,0.86603x,0), (-x/2,-0.86603x,0) , (0,0,z) , (0,0,-z) . The parameters X and Z specify the scale in the xy plane and z directions.

TETRAHEDRON X

The 'IDEAL' group is a regular tetrahedron with an atom at the centre. 'x' is the distance in Angstrom from the centre to an apex and defaults to '1.0'

ORIGIN

This directive is not yet implemented.

Uses of \REGULARISE

1 - Extending a fragment to a complete molecule

Three atoms of a phenyl group (C(1), C(2) C((6)) have been located. Fill in the missing atoms from a non-dilated idealised phenyl group.

```
\REGULARISE AUGMENT
GROUP 6
METHOD 1
\ C(3), C(4), and C(5) do not yet exist.
TARGET C(1) C(2) C(3) C(4) C(5) C(6)
PHENYL
END
```

2 - Forcing a regular shape on a group of atoms

A group of atoms is approximately octahedral. Replace them by a (posibly dilated) regular octahedron.

```
\REGULARISE REPLACE
GROUP 7
METHOD 2
TARGET CO(1) N(1) N(2) N(3) N(4) N(5) N(6)
OCTAHEDRON
END
```

3 - Checking for an additional symmetry element

Determine whether the two molecules in an asymmetric unit are related by a symmetry operation not expected for the space group. The matrix relating the molecules and the translation required to make their centroids coincide should display any additional (approximate) symmetry present. Remember that if one molecule is the enantiomer of the other, Method 3 will lead to an unsatisfactory fitting unless one molecule is inverted, (by using the operator -1 in the atom specifications e.g. FIRST(-1) UNTIL C(23). This can be done even if the space group is non-centrosymmetric).

```
\REGULARISE COMPARE
GROUP 16
TARGET C(101) UNTIL N(102)
IDEAL  C(201) UNTIL N(202)
END
```

4 - Renaming a group of atoms

A second group of atoms is given new TYPES and SERIAL numbers so that the atom names are related to a previously named group.

In the example, the user has identified two sets of four non-coplanar atoms in each group e.g. C(1) with Q(103), C(3) with Q(99) etc. The transformation is then used to map the whole of the first group (C(1) until O(25)) onto the second group (Q(96) until Q(120)). Both of these groups must contain the same number of atoms, but they may be in any order. Atom Q(103) will be renamed to C(101), atom Q(100) to C(107) etc. Once all the atoms have been renamed, the list could be sorted based on the serial numbers.

```
\REGULARISE
GROUP 4
IDEAL C(1) C(3) C(5) C(7)
TARGET Q(103) Q(99) C(116) Q(100)
RENAME 100
MAP C(1) UNTIL O(25)
ONTO Q(96) UNTIL Q(120)
END
\EDIT
SORT SERIAL
END
```

5 - Viewing matched molecules in CAMERON

This does the mapping as RENAME, but doesn't rename the atoms, just outputs CAMERON input files showing the two molecules superimposed. Use as follows:

```
\REGULARISE
group 16
target C(10) until C(26)
ideal  C(60) until C(76)
cameron
map     C(51) until C(99)
onto    C(1) until C(49)
end
```

by choosing Graphics->Special->Cameron (use existing...) from the menu. Then type "obey regular.oby" in Cameron to colour the molecules nicely. The TARGET and IDEAL are used to obtain the mapping. The atoms in MAP and ONTO are just the ones you want to be included. Don't read the atoms back into CRYSTALS when closing CAMERON - they're in orthogonal coordinates.

6.12 Calculation of interatomic bonds - \BONDCALC

```
\BONDCALC
END
```

```
\BONDCALC FORCE
END
```

This routine calculates a list of unique bonds between atoms in LIST 5 including bonds to symmetry related atoms. The bonds are stored in LIST 41.

Method of calculation

The BONDCALC routine uses the atomic positions from LIST 5 (the model parameters, see #RLIST05#) (together with cell (LIST 1, see #RLIST01#) and spacegroup information (LIST 2, see #RLIST02#), the covalent radii from LIST 29 (atomic properties, see #RLIST29#), and any additional bonding information in LIST 40 to calculate a list of bonds. The algorithm and tolerances used depend upon settings in LIST 40.

LIST 41 is only updated by \BONDCALC if there has been a change to LISTS 5 or 40 OR if \BONDCALC FORCE is issued.#

6.13 Bonding information - \LIST 40

```
\LIST 40
DEFAULTS TOLTYPE= TOLERANCE= MAXBONDS= NOSYMMETRY= SIGCHANGE=
READ NELEMENTS= NPAIRS= NMAKE= NBREAK=
ELEMENT TYPE= RADIUS= MAXBONDS=
PAIR TYPE1= TYPE2= MIN= MAX= BONDTYPE=
MAKE TYPE= SERIAL= S= L= TX= TY= TZ=
     TYPE2= SERIAL2= S2= L2= TX2= TY2= TZ2= BONDTYPE=
BREAK TYPE= SERIAL= S= L= TX= TY= TZ=
      TYPE2= SERIAL2= S2= L2= TX2= TY2= TZ2=
END
```

DEFAULTS TOLTYPE= TOLERANCE= MAXBONDS= NOSYMMETRY= SIGCHANGE=
This directive may only appear once. It affects the algorithm used to update LIST 41.

TOLTYPE=
A value of 1 (default) causes \BONDCALC to use as a threshold for bonding, the sum of the covalent radii * the tolerance given. A value of 0 causes \BONDCALC to use the sum of the covalent radii + the tolerance given (in Angstroms), as a threshold.

TOLERANCE=
The tolerance to be used in the \BONDCALC calculation as a threshold for bonds. Exact use depends on the value of the TOLTYPE keyword above.

MAXBONDS=
Specifies the maximum number of bonds that may be formed to an atom. The BONDCALC calculation proceeds through the list of atoms searching for bonds, according to the TOLERANCE criteria. If more than MAXBONDS bonds are found, the best MAXBONDS will be kept. Best bonds are those where the sum of the covalent radii is closest to the actual bond length. (Where a PAIR directive has been used, the best are the closest to the mean of the min and max values on the PAIR directive.) Note well: The calculation proceeds through the list of atoms, so bonds are formed from atoms near the top of the list to those lower down. While atoms lower down will still only form at most MAXBONDS bonds, they are less likely to be the 'best' bonds since they are formed from atoms higher up the list. E.g. You have an H right at the end of the list, and you set MAXBONDS=1 for H (see ELEMENT). If the first atom forms a bond to that H, then no more bonds can be formed to that H even if they are better. If the H were at the top of the list it would get the choice of which bonds to pick. This is fairly unimportant stuff, it is rare that there will be ambiguities over whether something is bonded or not. The default value of MAXBONDS is therefore 15.

NOSYMMETRY=
0 (default) searches for all symmetry related bonds. 1 ignores symmetry, will not find bonds across operators, may speed up bond bond calculation slightly.

SIGCHANGE=
Number of angstroms that any atom in LIST 5 must move during refinement for it to be considered a significant change resulting in a recalculation of bonding.

READ NELEMENTS= NPAIRS= NMAKE= NBREAK=
Specify the number of ELEMENT, PAIR, MAKE and BREAK directives that are to follow.

ELEMENT TYPE= RADIUS= MAXBONDS=
Override the covalent radius in L29 and the MAXBONDS value on the DEFAULTS directive for a specific element.

TYPE=
The element type. E.g. C

RADIUS=
The covalent radius to use for this element.

MAXBONDS=
The maximum number of bonds to this element.

PAIR TYPE1= TYPE2= MIN= MAX= BONDTYPE=
Override the covalent based calculation altogether.

TYPE1=
An element type, e.g. C

TYPE2=
An element type, e.g. O

MIN=
The minimum length of a bond.

MAX=
The maximum length of a bond.

BONDTYPE=
The bondtype to be assigned to this bond. BONDCALC will eventually have a go at bond type assignment, if you are forced to add in extra PAIR commands then there is not much chance that the assignment will be correct so it can be specified here. Use 0 for unknown.
More than one pair of the same elements can be used at once:

```
e.g.
 PAIR C O 1.0 1.2 BONDTYPE=2
 PAIR C O 1.2 1.4 BONDTYPE=1
```

MAKE TYPE= SERIAL= S= L= TX= TY= TZ= TYPE2= SERIAL2= S2= L2= TX2= TY2= TZ2= BO
  Makes a bond between two atoms (possibly symmetry related).
  TYPE= TYPE2=
    An element type, e.g. C.
  SERIAL= SERIAL2=
    The serial number of the atom. (From List 5, atomic parameters).
  S= S2=
    The number of the symmetry matrix used from List 2 (list of space group symmetry operators, see section #RLIST02#) (default, unity = 1). Negative indicates centre of symmetry applied aswell.
  L= L2=
    The number of the non-primitive lattice translation from List 2. (default =1, see section #RLIST02#)
  TX= TY= TZ= TX2= TY2= TZ2=
    Translations from asymmetric unit co-ordinates.
  BONDTYPE=
    The bondtype to be assigned to this bond. BONDCALC will eventually have a go at bond type assignment, if you are forced to add in extra MAKE commands then there is not much chance that the assignment will be correct so it can be specified here. Use 0 for unknown.
BREAK TYPE= SERIAL= S= L= TX= TY= TZ= TYPE2= SERIAL2= S2= L2= TX2= TY2= TZ2=
  As for MAKE, but without the BONDTYPE keyword.

6.14 Bonding information - \BONDING

```
\BONDING ACTION
DEFAULTS TOLTYPE= TOLERANCE= MAXBONDS= NOSYMMETRY= SIGCHANGE=
ELEMENT TYPE= RADIUS= MAXBONDS=
PAIR TYPE1= TYPE2= MIN= MAX= BONDTYPE=
MAKE  atom-specification TO atom-specification bondtype
BREAK atom-specification TO atom-specification
END
```

  THis is a more user-friendly alternative to inputting a LIST 40. Directive syntax is like \LIST 40 with the following exceptions:
  1) ACTION. This can take two values:

```
REPLACE (Default, and replace previous LIST 40 with a new on)
EXTEND  (adds new commands to end of existing LIST 40)
```

  2) The MAKE and BREAK directives look like this:

```
MAKE C(1) TO C(4) 8
BREAK N(1) TO H(14)
```

  Symmetry may be specified in the standard CRYSTALS way, the numbers in parenthesis are serial,S,L,Tx,Ty,Tz (see above) the list may be truncated when the rest are default values: (serial,1,1,0,0,0):

```
MAKE C(1,2,1,0,1,1) TO C(8) 4
```

  3) The READ directive need not be given. This makes it easier to edit text files containing the command as you don't have to remember to alter the values on the READ directive.
  3) The command may be given as \BONDING EXTEND, in which case it takes any directives given and adds them to the existing LIST 40.

6.15 Printing of LIST 40

  LIST 40 may be listed with either
\PRINT 40
  or
\SUMMARY 40
  LIST 40 may be punched with
\PUNCH 40
  which will produce a standard List 40 in CRYSTALS format, or
\PUNCH 40 B
  which will produce a \BONDING command which is easier to edit.

6.16 Creating a null LIST 40

  A null LIST 40, containing no extra information, may be created with

```
\LIST 40
END
```

or

```
\BONDING
END
```

6.17 Printing of LIST 41

LIST 41 may be listed with either

\PRINT 41

or

\SUMMARY 41

Issuing \BONDCALC when there is no LIST 40 will cause a null list 40 to be created.

Chapter 7
Structure Factors And Least Squares

7.1 Scope of the structure factors and least squares section

   This section describes the necessary LISTS and explains how structure factor calculation and least squares refinement can be carried out.

Refinement
LIST 23 - Structure factor calculation control list (#RLIST23#)
SPECIAL - Special positions constraints (#RSPECIAL#)
LIST 12 - Input of refinement directives (#RLIST12#)
LIST 16 - Input of the restraints (#RLIST16#)
LIST 4 - Weighting the reflections (#RLIST04#)
LIST 28 - Reflection restriction list (#RLIST28#)
CHECK - Checking the refinement and restraint directives
SFLS - Structure factor least squares calculations (#RSFLS#)
ANALYSE - Systematic comparisons of Fo and Fc
DIFABS - Least squares absorption correction

Internal workings

LIST 22 - Refinement parameter map (#RLIST22#)
LIST 17 - Input of the special restraints (#RLIST17#)
LIST 11 - The least squares matrix (#RLIST11#)
LIST 24 - The least squares shift list (#RLIST24#)
LIST 26 - Restraints in internal format (#RLIST26#)

7.2 Refinement

   Before a stucture factor-least squares calculation is performed, the following lists must exist in the .DSC file

LIST 1 Cell parameters (section #RLIST01#)
LIST 2 Symmetry inforamation (section #RLIST02#)
LIST 3 Scattering factors (section #RLIST03#)
LIST 4 Weighting scheme
LIST 5 Atomic and other model parameters (section #RLIST05#)
LIST 6 Reflection data (section #RLIST05#)
LIST 12 Refinement definitions (section #RLIST12#)
LIST 16 Restraints (section #RLIST16#)
LIST 17 Special position restraints (section #RLIST17#)
LIST 23 Structure factor control list (section #RLIST23#)
LIST 25 Twin laws, only for twinned refinements (section #RLIST25#)
LIST 28 Reflection control list (section #RLIST28#)

LISTS 12,16 and 17 (constraints #RLIST12#, restraints #RLIST16# and special restraints #RLIST17#) are not required if structure factors are only going to be calculated.

The refinement directives specify which model parameters are to be refined, and the control directives control the terms in the minimisation function.
   During structure factor least squares calculations, the partial derivatives with respect to each of the parameters is calculated for each structure factor and added into the 'normal equations'. This system of equations may be represented in matrix notation as :

```
A.x = b
WHERE :
A        'A' is a symmetric n*n matrix. an element
         'A(i,j)' of 'A'is given by :
          A(i,j) = Sum [ w(k)*Q(i,k)*Q(j,k) ] over k.
n        number of parameters being refined.
k        indicates reflection number 'k'.
w(k)     weight of reflection k.
Q(i,k)   the partial differential of Fc(k) with
         respect to parameter i.
x        a column vector of order n, containing
         the shifts in the parameters.
b        also a column vector, an element
         of which is given by :
          b(i) = Sum [ w(k)*DF(k)*Q(i,k) ] over k.
DF(k)    delta for reflection k, given by :
          DF(k) = [Fo(k) - Fc(k)]
```

As the matrix A is symmetric, only (n(n+1))/2 of its elements need to be calculated and stored, together with a few house keeping items.

In some cases, because of either storage or time considerations, it is impractical to use the full normal matrix A . In this situation, it is necessary to use a 'block diagonal approximation' to the full matrix, in which interactions between parameters which are known not to be highly correlated are ignored. The effect of ignoring such interactions is to leave blank areas of the full matrix, related symmetrically across the diagonal, which do not need to be stored or accumulated. A common (but not very efficient or stable) example of this approach is to place one atom in each of the blocks used to approximate the normal matrix, so that each block is of order either 4 (x, y, z and u[iso]) or 9 (x, y, z and the anisotropic thermal parameters).

One of the main purposes of the refinement directives is to describe the areas of the matrix A that are to be calculated. If the matrix A is approximated by m blocks of order n(1), n(2),.....n(m), The total amount of memory needed to hold the matrix and vector is:

```
Elements = 12 + 4*m + Sum n(i)*(5 +n(i))/2,


                                    i = 1 to m


     Currently (June 2003) elements=8,388,608, giving over 4000
parameters in a single block.
```

The formation of the blocks that are to be used to approximate the normal matrix A is controlled in the refinement directive list by a series of BLOCK directives, each of which contains the coordinates that are to be included in the newly specified block. Further control instructions for the current block may appear on subsequent directives until a new BLOCK directive is found, when the formatio of another block with its associated parameters is started.

Two special directives are provided to allow for the most common cases required, full matrix refinement (a FULL directive) and one atom per block (a DIAGONAL directive). For all these cases only the parameters specified on the control directives and the following directives are refined.

Correlations in Refinement

Highly correleated parameters **MUST** be refined together. Refining them in different cycles or different blocks will lead to an incorrect structure.

As a rough guide, the following groups of parameters are in general highly correlated and should be refined in the same block if possible :

```
1.   Temperature factors, scale factors, the extinction
     parameter, the polarity parameter and the
     enantiopole parameter.
2.   Coordinates of bonded atoms.
3.   Non-orthogonal coordinates of the same atom.
4.   U(11), U(22) and U(33) of the same atom.
```

If it is necessary to split the temperature factors and scale factor into different blocks, their interactions must not be neglected but must be allowed for by using a 'dummy overall isotropic temperature factor'. In this case, the scale factor and the dummy temperature factor must be put into a block of order 2 by themselves, and the program will make the appropriate corrections to all the temperature factors. This dummy temperature factor should not be confused with the 'overall temperature factor' which is a temperature factor that applies to all the atoms and is therefore just a convenience and requires no special treatment.

For further details, Computing Methods in Crystallography, edited by J. S. Rollett, page 50, and Crystalographic Computing, ed Ahmed, 1970, page 174.

Although it is possible to input an overall temperature factor as one of the overall parameters, it is not possible to use it under all circumstances. The structure factor routines always take the temperature factor of an individual atom as the value or values stored for that atom. If the overall temperature factor is to be refined, the system will ensure that the current value of the overall temperature factor is inserted

for the temperature factor of all the atoms. When the new parameters are computed after the solution of the normal equations, this substitution is again made, so that all the atoms have the same overall isotropic temperature factor. However, if the overall temperature factor is not refined, or no refinement is done, the individual temperature factor for each atom will be used, and the overall temperature factor ignored.

CAUTION

It should be noted that if a set of anisotropic atoms are input with no U[ISO] key and U[ISO] data, then the default value of 0.05 will be inserted by the sfls routines. This implies that all such atoms are isotropic, so that the anisotropic temperature factors will be set to zero, and the calculation will proceed for isotropic atoms.

F or Fsq refinement?

Both type of refinement have been available in CRYSTALS since the early 70's. For most data sets, there is little difference between the two correclty weighted refinements. One of the current reasons for choosing Fsq refinement is 'so that -ve observations may be used'. Such a choice is based on the misapprehension that the moduli in /Fo/ are the result of taking the square root of Fsq. In fact, it indicates that the phase cannot be observed experimentally. The experimental value of Fo takes the sign of Fsq and the positive square root. With proper weighting, both refinemets converge to the same minima (Rollett, J.S., McKinlay,T.G. and Haigh, N.P., 1976, Crystallographic Computing Techniques, pp 413-415, ed F.R. Ahmed,Munksgaard; and Prince,E. 1994, Mathematical Techniques in Crystalography and Materials Science, pp 124-125.Springer-Verlag). However, the path to the minima will be different, and there is some evidence that Fsq refinement has less false minima. Using all data, including -ve observations, increases the observation:variable ratio, but it is not evident that a large number of essentially unobserved data will improve the refinement. If the difference between F and Fsq refinement is significant, then the analysis requires care and attention.

Hydrogen Atom Refinement

Several strategies are available for refining hydrogen atoms. Which you use is probably a matter of taste.

Geometric re-placement

The command \HYDROGEN or \PERHYDRO is used to compute geometrically suitable positions for the H atoms. These are **not** refined (either they are left out of LIST 12, or a fixed with the FIX directive). After a few cycles of refinement of the remaining parameters, they are deleted (\EDIT <cr> SELECT TYPE NE H) and new positions computed. This ensures optimal geometry, ensures that Fcalc is optimal, but avoids the cost of including the deviatives in the normal matrix.

Riding hydrogens

As above, the hydrogens are placed geomtrically, but they are included in the formation of the least squares matrix. Their derivatives are added to those of the corresponding carbon, and a composite shift computed for each carbon and its riding hydrogens. This preserves the C-H vector, but can distort C-C-H angles. A cycle of refinement takes almost twice as long as the re-placement method.

Restrained hydrogens

In this method, starting positions are hound for the hydrogen atoms (either from Fourier maps of geometrically), and the hydrogen positions are refined along with other atoms. The C-H distances and C-C-H angles are restrained to acceptable values in LIST 16. This calculation is even slower than the riding model, and would normally only be applied to an atom of special significance (*e.g.* a hydrogen bond H atom).

Free refinement

The hydrogen atom is treated like any other atom. Requires good data, and may be applied to atoms of special interest.

Note that the different methods can be mixed in any way, with some hydrogens placed geometrically, and others refined.

R-Factor and minimisation function definitions

Conventional R-value

This is defined as:

```
R = 100*Sum[//Fo/-/Fc//]/Sum[/Fo/]
```

The summation is over all the reflections accepted by LIST 28. This definition is used for both conventional and F-squared refinement.

Weighted R-value

The Hamilton weighted R-value is defined as :

```
100*Sqrt(Sum[ w(i)*D'(i)*D'(i) ]/SUM[ w(i)*Fo'(i)*Fo'(i) ])


D'  = Fo'-Fc'
Fo' = Fo for normal refinement, Fsq for F-squared refinement.
Fc' = Fc for normal refinement, Fc*Fc for F-squared refinement.
```

Minimisation function
This is defined by :

```
MINFUNC = Sum[ w(i)*D(i)*D(i) ]


D', Fo', Fc' defined above.
```

Residual
The residual and weighted residual are defined by:

```
residual = Sum D'(i)**2
weighted residual = Sum w(i)*D'(i)**2
```

7.3 Special positions

The second major purpose of the refinement directives is to allow for atoms on special positions. For example, the atom at the Wyckoff site H in the space group P6(3)/mmc (no. 194) has coordinates X,2X,Z . In a least squares refinement, the X and Y coordinates of this atom must be set to the same variable, i.e. they become equivalent. The command \SPECIAL (section #RSPECIAL#) can be used to generate the necessary constraints or restraints, and may be invoked automatically before structure factor calculations by setting the appropriate parameters in LIST 23 (structure factor control settings, see section #RLIST23#) The user can do this manually via the refinement directives, LIST 12. The relationship is set up by an EQUIVALENCE directive, which sets all the parameters on the directive to the same least squares parameter. In this example, it is also necessary to alter the contribution of the Y coordinates to the normal matrix by multiplying the derivatives by 2. This facility is provided by the WEIGHT directive, which should not be confus with the weight ascribed to each reflection in the refinement. For a full treatment of atoms on special positions, see Crystallographic Computing, edited by F. R. Ahmed, page 187, or Computing Methods in Crystallography, page 51.

Similar relationships also hold for the anisotropic temperature factors.

The relationships between the variable parameters in a refinement may also be defined by RESTRAINTS. These are held in LIST 17 (see #RLIST17#), and are particularly usefull if a complex matrix has been defined (e.g. using RIDE, LINK, EQUIVALENCE, WEIGHT, BLOCK, GROUP or COMBINE).

7.4 Atomic parameter refinement

Atomic parameters may be specified in three different ways. Firstly, there is an **IMPLICIT** definition, in which parameters for all the atoms are specified simply by giving the appropriate key or keys.

Hydrogen atoms are automatically excluded from implicit definitions.

Secondly, there is an **EXPLICIT** definition, in which the parameters of one atom are specified by giving the atom name followed by the appropriate keys.

Lastly, the parameters for a continuous group of atoms in LIST 5 may be specified by an UNTIL sequence. This type of parameter definition is taken to be implicit.

KEY[1] . . . KEY[K]
parameters defined by the keys KEY[1] . . KEY[K] are included (or excluded) for all the atoms in LIST 5, e.g. X U[ISO] implies that the 'X' and 'U[ISO]' parameters of all the atoms in the current LIST 5 will be used. This is an **implicit** definition, since parameters for all the atoms in LIST 5 are specified simply by giving the appropiate key.

TYPE(SERIAL,KEY[1], . . ,KEY[K])
parameters defined by the keys KEY[1] . . . KEY[K] are included (or excluded) for the atom of type 'TYPE' with the serial number 'SERIAL', e.g. C(21,X,U[ISO]) implies that the 'X' and 'U[ISO]' parameters of atom C(21) will be used. This is an **explicit** definition.

TYPE1(SERIAL1,KEY[1], . ,KEY[K]) UNTIL TYPE2(SERIAL2)
the parameters defined by the keys KEY[1] . . KEY[K] are included (or excluded) for atoms in LIST 5 starting at the atom with type 'TYPE1' and serial 'SERIAL1', and finishing with the atom of type 'TYPE2' and serial 'SERIAL2'. This definition is **implicit,** since the number of atoms included by this definition depends on the number and order of the atoms in LIST 5.

Parameter definitions of all three types may appear on any directive in any desired combination.

```
EXAMPLE
    LIST 5 contains  FE(1) C(1) C(2) C(3) C(4) C(5) C(6) N(1)


    \LIST 12
    BLOCK X'S C(1,U[ISO]) UNTIL C(6) FE(1,U'S)
    END


    This refines x,y,z of all atoms, u[11]...u[12] of iron, and
    u[iso] of the other atoms.
```

The following parameter keys may be given in an atom definition :

```
OCC       X        Y        Z
U[ISO]   SIZE     DECLINAT  AZIMUTH
U[11]    U[22]    U[33]     U[23]     U[13]     U[12]


X'S      Indicating  X,Y,Z
U'S      Indicating  U[11],U[22],U[33],U[23],U[13],U[12]
UII'S    Indicating  U[11],U[22],U[33]
UIJ'S    Indicating  U[23],U[13],U[12]
```

7.5 Overall parameter refinement

Overall parameters, apart from the layer scale factors and the element scale factors, are specified simply by their keys. Such a specification is considered to be an explicit definition. The following overall parameter keys may be given :

```
SCALE       OU[ISO]       DU[ISO]
POLARITY    ENANTIO       EXTPARAM
```

7.6 Scale factor definitions

The OVERALL scale factor is always applied to the structure factor calculation, though it need not necessarily be refined. LAYER and BATCH scale factors are applied only if indicated in LIST 23 (structure factor control settings, see section #RLIST23#), and ELEMENT scales only if the crystal is marked as being twinned in LIST 13. Note that all of these scale factors can be expected to be correlated with each other, and the overall parameters.

The layer scale factors, batch scale factors and the element scale factors may be given in three different ways, all of which are considered to be explicit :

LAYER(M), BATCH(M)   OR   ELEMENT(M)
    this indicates only scale factor 'M' of the specified type. 'M' must be in the correct range, which for 'N' layer scale factors is 0 to 'N-1', and for 'N' element scale factors is 1 to N.
LAYER(P) UNTIL LAYER(Q)   OR   BATCH(P) .....
    this indicates all the scale factors of the specified type from 'P' to 'Q'. 'P' and 'Q' must be in the correct range, as defined for 'M' in the previous section.
LAYER SCALES, BATCH SCALES   OR   ELEMENT SCALES
    this indicates all the scale factors of the given type.

7.7 Structure factor calculation control list - LIST 23

```
\LIST 23
MODIFY ANOM= EXTINCT= LAYERSCALE= BATCHSCALE= PARTIAL= UPDATE= ENANTIO=
MINIMISE NSINGULARITY= F-SQUARED= REFLECTIONS= RESTRAIN=
REFINE  SPECIAL= UPDATE= TOLERANCE=
ALLCYCLES MIN-R= MAX-R= *-WR= *-SUMSQ= *-MINFUNC= U[MIN]=
INTERCYCLE MIN-DR= MAX-DR= *-DWR= *-DSUMSQ= *-DMINFUNC=
END


\LIST 23
MODIFY EXTINCTION=YES, ANOMALOUS=YES
END
```

This LIST controls the structure factor calculation. The default calculation involves the minimum of computation (atomic parameters and overall sale factor). More extensive calculations have to be indicated by entries in this list. The presence of a parameter in the parameter list (LIST 5) does not automatically mean that it will be included in the structure factor calculation.

This list also controls the treatment of atoms on special positions, the use of F or Fsq, and the use of restraints.

The presence of information in the DSC file does not ensure that it will be used by the structure factor routines. Thus, the operations corresponding to RESTRAIN , ANOMALOUS , EXTINCTION , PARTIAL , BATCHSCALES, LAYERSCALES and ENANTIO are not performed unless they are explicitly asked for in a LIST 23.

\LIST 23

MODIFY ANOM= EXTINCT= LAYERSCALE= BATCHSCALE= PARTIAL= UPDATE= ENANTIO=
This directive controls modifications that can be applied to Fo and Fc.
ANOMALOUS=

NO   - Default value
YES

If ANOMALOUS is YES , the imaginary part of the anomalous dispersion correction, input in LIST 3 (see section #RLIST03#, will be included in the s.f.l.s. calculations. For computational efficiency, it is recommended only to use the value YES towards the end of a refinement. Note that the value YES should be used even for centro-symmetric crystals if they contain a heavy atom.
EXTINCTION=

NO   - Default value
YES

If EXTINCTION is YES , the calculated structure factors are modified to allow for the effects of extinction by the method of A. C. Larson. See Atomic and Structural Parameters for the definition.

LAYERSCALES=
BATCHSCALES=
SCALE keys have two alternatives:

NO   - Default value
YES

If either SCALE key is YES , the corresponding scale factors stored in LIST 5 (the model parameters) are applied to the reflection data. If this parameter is omitted, the scale factors are not applied, even if they exist in LIST 5
PARTIAL=

NO   - Default value
YES

If PARTIAL is YES , the fixed partial contributions stored in LIST 6 (section #RLIST06#) are added in during the calculation of Fc and the phase. The partial contributions must already be present in LIST 6, and should have the keys A-PART and B-PART . The atoms which have contributed to the partial terms should be omitted from LIST 5 whenever PARTIAL is YES .
UPDATE=

NO   - Default value
YES

If UPDATE is YES , the contributions of the atoms to A and B are output to LIST 6 with the keys A-PART and B-PART . If UPDATE is NO , its default value, the partial contributions are not changed during structure factor calculations. This requires that LIST 6 contain the keys A-PART and B-PART.
ENANTIO=

NO   - Default value
YES

If ENANTIO is YES, then Fc is computed with

*Fc = SQRT((1-x)*F(h)**2 + x*F(-h)**2)*

Where x is the enantiopole parameter from LIST 5. Once the correct enantiomer has been established, set this parameter back to NO.
MINIMISE= NSINGULARITY= F-SQUARED= REFLECTIONS= RESTRAIN=
This directive controls modifications made to the minimisation function during s.f.l.s.
NSINGULARITY=
The default value is zero. If this parameter is omitted, any singularities discovered during the inversion of the normal matrix will cause the program to terminate after the current cycle of refinement. If NSINGULARITY is greater than zero, it represents the number of singularities allowed before the program will terminate.
F-SQUARED=

NO   - Default value
YES

If F-SQUARED is NO, the traditional minimisation function is:

Minimisation function = Sum[w(Fo - Fc)**2]*

If F-SQUARED is YES , the minimisation function is:

Minimisation function = Sum[w*(Fo**2 - Fc**2)**2]

If F-SQUARED is YES , the weights given by w in the above expression are assumed to be on the correct scale and to refer to Fsq rather than Fo's. Note that refinement can be against Fo or Fsq independent of whether the input was Fo or Fsq.

REFLECTIONS=
REFLECTIONS has two alternatives:

NO
YES   - Default value

If REFLECTIONS is YES, the reflections stored in LIST 6 (and subject to the checks in LIST 28) are used for computing structure factors and the derivatives added into the matrix if required.
  If REFLECTIONS is NO, LIST 6 is not used, whether it is present or not. This setting could be used for refinement against restraints only. See the section DLS, 'Distance Least Squares'.

RESTRAIN=
RESTRAIN has two alternatives:

NO
YES   - Default value

If RESTRAIN is YES, the restraints in LIST 16 (section #RLIST16#) and LIST 17 (section #RLIST17#) are added into the normal equations.

REFINE SPECIAL= UPDATE= TOLERANCE=
This directive controls the refinement of atoms on special positions and the control of floating origins. The default action for atoms is to try to constrain them. However, if an atom is already the subject of a user defined constraint, the symmetry requirements are imposed by restraints. The site occupancy, positional and thermal parameters can be set to satisfy the site symmetry. The site occupancy is indepentant of any chemical or physical partial occupancy by an atom.
  Floating origins are controlled by restraining the center of gravity of the structure along the axis to remain fixed.

SPECIAL=

```
SPECIAL = NO                   No action
        = TEST                 Displays but does not store any restrictions
        = ORIGIN               Tests for and restrains floating origins
        = RESTRAIN             Creates and stores restraints
        = CONSTRAIN (Default)  Attempt to create constraints
```

UPDATE=

```
UPDATE = NO                    Nothing updated
       = OCCUPATION            Site occupancies modified
       = PARAMETERS  (Default) All adjustable parameters modified
```

REWEIGHT=
Not currently used

GROUPS=

```
NO  -  Default value
YES
```

GROUPS is automatically set to YES if LIST 12 contains any GROUP directives. It forces the group derivatives to be recalculated between each cycle or refinement. Not currently used

COMPOSITE=
Not currently used

TOLERANCE=
Atoms within 'TOLERANCE' Angstrom of a symmetry equivalent atom are regarded as being on a special position. The default is 0.6Å. For high symmetry spacegroups with disorder, the value might need reducing if multiplicities are incorrectly calculated.

ALLCYCLES MIN-R= MAX-R= *-WR= *-SUMSQ= *-MINFUNC= U[MIN]=
This directive controls conditions that must be satisfied after each cycle if refinement is to continue. It can be used to detect converged or 'blown-up' refinements. The heading has been abbreviated, the * representing MIN and MAX .

MIN-R=, MAX-R=
The normal R-value must lie between MIN-R and MAX-R, otherwise refinement is terminated after the current cycle. The default values for MIN-R and MAX-R are 0.0 and 100.0 percent.

MIN-WR MAX-WR
>The Hamilton weighted R-VALUE must lie between MIN-WR and MAX-WR, otherwise the refinement is terminated after the current cycle. The default values for MIN-WR and MAX-WR are 0.0 and 100.0 percent respectively.

MIN-SUMSQ=, MAX-SHUMSQ=
>The rms (shift/e.s.d.) fo all parameters in the refinement must lie between MIN-SUMSQ and MAX-SUMSQ, otherwise the refinement is terminated after the current cycle. The sum of the squares of the ratios is defined as :

>   *SUMSQ = SQRT(SIGMA(SHIFT/ESD))/N)*
> *The default values*
of MIN-SUMSQ and MAX-SUMSQ are 0.03 and 10000.0, .

>The minimisation function, on the scale of Fo, must lie between MIN-MINFUNC and MAX-MINFUNC, otherwise the refinement is terminated after the current cycle. The default values of MIN-MINFUNC and MAX-MINFUNC are 0.0 and 1000000000000000.0.

U[MIN]=
>If Uiso or a principal component of the adp of any atom is less than U[MIN] , then a warning is issued and the idp reset to u[min], or the components of the adp reset to MAX(Uii,U[MIN]) or MAX(Uij,0.01U[min]). If this parameter is omitted, a default value of 0.0 is assumed.

INTERCYCLE MIN-DR= MAX-DR= *-DWR= *-DSHIFT/ESD= *-DMINFUNC=
>This directive refers to conditions that must be obeyed before the next cycle of least squares refinement can proceed. (A quantity undergoes a positive change if OLD - NEW is positive, not NEW - OLD). The definitions are similar to ALLCYCLES. The abbreviation '*' represents MIN and MAX .

MIN-DR= MAX-DR=
>Between two cycles of least squares, the change in R-VALUE must lie between MIN-DR and MAX-DR, otherwise the refinement is terminated. The default values are -5.0 and 100.0.

MIN-DWR MAX-DWR
>The default values are -5.0 and 100.0.

MIN-DSUMSQ MAX-DSUMSQ
>The default values are -10. and 10000.0.

MIN-DMINFUNC MAX-DMINFUNC
>The default values are 0.0 and 1000000000000000.0.

7.8 Printing the SLFS control list

\PRINT 23
>This prints LIST 23. There is no command for punching LIST 23.

7.9 Special position constraints - \SPECIAL

>   *\SPECIAL   ACTION= UPDATE= TOLERANCE=#*
> *END*

>   *\SPECIAL*
> *END*

 \SPECIAL can be issued at any time to get information about atoms on special positions. However, normally it is called automatically by setting the SPECIAL keyword in LIST 23 (section #RLIST23#).
 Atoms on special positions may be constrained through LIST 12 (section #RLIST12#), or restrained through LIST 17 (section #RLIST17#). CRYSTALS will attempt to generate the special position conditions when requested via the SPECIAL command, and also update coordinates of atoms on special positions.
 If the RESTRAIN option is chosen, then the special conditions are imposed on the refinement by restraints, which are generated without reference to what is being specified in LIST 12, the refined parameter definition list.
 If the CONSTRAIN option is chosen, then CRYSTALS examines the site restrictions as it processes LIST 12. If an atom on a special position is being refined without any user defined conditions (EQUIVALENCE, RIDE, LINK, COMBINE, GROUP, WEIGHT), and the related coordinates are in the same matrix block, then the internal representation of LIST 12 (LIST 22) is dynamically modified to include the necessary constraints. If the atom is already the object of a constraint, then LIST 12 cannot safely be modified, and the special condition is applied as a restraint. In either case, CRYSTALS warns the user about what is being done.
 The origins of polar space groups are always fixed by restraints, since this produces a better conditioned matrix than one from just fixing atomic coordinates.
 The UPDATE directive controls whether parameters of atoms near special positions will be modified to make them exact. The routine will update just the site occupancies, or the occupancies and the other variable parameters. The crystallographic site occupancy is held temporarily in the key SPARE, leaving the key OCC available for a refinable chemical occupancy. Take care if an atom refines onto (or off) a special position.
 The function SPECIAL is actioned automatically for every round of least squares refinement. Its action is

then determined by values held in LIST 23 (structure factor control, see section #RLIST23#)
\SPECIAL ACTION= UPDATE= TOLERANCE=
    ACTION

```
ACTION = NONE       No action
       = TEST       Displays but does not store any restrictions
       = ORIGIN     Tests for and restrains floating origins
       = RESTRAIN   Creates and store a LIST 17
       = CONSTRAIN  Attempt to create constraints.
       = LIST23     (Default) Takes the action defined in LIST 23
```

    UPDATE

```
UPDATE = NONE        Nothing updated
       = OCCUPATION  Site occupancies modified
       = PARAMETERS  All adjustable parameters modified
       = LIST23      (Default) Takes action defined in LIST 23
```

    TOLERANCE
        TOLERANCE is the maximum separation, in Angstrom, between nominally equivalent sites. The
        default is 0.6A.

7.10 Printing the special position information

Force the atom parameter list (LIST 5) to be updated and send it to the PCH file.

```
\SPECIAL TEST PARAMETER
END
\PUNCH 5  (to get a listing with 5 decimal places)
END
```

7.11 Refinement directives - LIST 12

This list defines the parameters to be refined in the least squares calculation, and specifies relationships
between those parameters.

```
\LIST 12
BLOCK  PARAMETERS ...
FIX  PARAMETERS ...
EQUIVALENCE  PARAMETERS ...
RIDE  ATOM_PARAMETER SPECIFICATIONS ...
LINK PARAMETER_LIST AND PARAMETER_LIST AND PARAMETER_LIST.
COMBINE PARAMETERS_LIST AND PARAMETERS_LIST
GROUP  ATOM SPECIFICATIONS
WEIGHT F1 PARAMETERS F2 PARAMETERS ...
FULL  PARAMETERS
DIAGONAL  PARAMETERS
PLUS  PARAMETERS
END
```


```
\LIST 12
BLOCK SCALE X'S U'S
END
```

\LIST 12
    BLOCK PARAMETERS
        This directive defines the start of a new matrix block. Any parameters that come on this directive and
        any directives until another BLO directive are put into the same matrix block. If only one BLOCK
        directive is given, then the refinement is 'full matrix'.
    FIX PARAMETERS
        The specified parameters are not to be refined.
    EQUIVALENCE PARAMETERS
        Sets the given parameters to a single least squares parameter (see the examples).
    RIDE ATOM_PARAMETER SPECIFICATIONS
        This directive links corresponding parameters for all the atoms specified on the directive. The
        parameters specified for the first atom given on this directive are each assigned to individual least
        squares parameters, and parameters for subsequent atoms are EQUIVALENCED, in the order given, to
        the corresponding least squares parameter. Only explicit atom parameters can be used on this directive.
        Usually, the same parameter keys will be given in the same order for all atoms referenced, though this
        may not be true for high symmetry space groups.
    LINK PARAMETER_LIST AND PARAMETER_LIST (AND PARAMETER_LIST.)
        Links the parameters defined after the AND with those specified in the first parameter list. A least
        squares parameter is assigned to each physical parameter in the first list. Physical parameters specified
        in the second (and subsequent if present) lists are then assigned IN THE ORDER GIVEN to these least
        squares parameters. There must be the same number of parameters in each parameter list. The
        parameter list may contain more than one atom, and is terminated by the 'AND' or the end of the
        directive. Overall and implicit parameters may be given.

COMBINE PARAMETERS_1 AND PARAMETERS_2

Combines the parameters defined before the AND with those defined after. Physical parameters are taken pairwise in the order given from parameter list 1 and 2 and two least-squares parameters defined such that one is the sum and the other is the difference of the physical parameters.

```
x' = x1 + x2
x" = x1 - x2
               where x1 and x2 are physical parameters,
                and x' and x" are least squares parameters.
```

Such a re-parameterisation is useful for dealing with certain sorts of ill- conditioning, such as that due to pseudo-symmetry, of the normal matrix (see Edward Prince, Mathematical Techniques in Crystallography and Material Science, 1982, Springer-Verlag, page 113). NOTE that only one AND can be given.

GROUP ATOM SPECIFICATIONS

The positional coordinates of the atoms given in the ATOM SPECIFICATIONS are refined as a rigid group. Parameter specifications MUST NOT be included. The first atom specified is taken as the pivot atom of the group. All atoms in the group may be the subject of restraints to atoms in other parts of the structure, or in other groups. Use LINK, RIDE or EQUIVALENCE to build a suitable model for the temperature factors.

Because of the linearisation algorithm used, some distortion of the group will occur if there are large parameter shifts. Use REGULARISE to re-form it.

WEIGHT w1 PARAMETERS w2 PARAMETERS . .

Before the contributions of the specified parameters are included in the normal equations, they are multiplied by the number wI . Similarly , when the normal equations are solved, the shifts and e.s.d.'s are multiplied by the same wI. The default value of wI is 1.0. The parameters are multiplied by the value of wI that precedes them (see the examples).

7.12 Obsolete Refinement directives

The following directives may be removed in some future release.

FULL PARAMETERS

The parameters on the directive directive plus any other parameters defined on subsequent directives are to be included in a full matrix refinement. The scale factor is automatically included, while the dummy overall isotropic temperature factor is fixed. This is equivalent to:
    BLOCK SCALE PARAMETERS

PLUS PARAMETERS

The specified parameters are to be refined, and they will be placed in the current block of the normal matrix. This is equivalent to:
    CONTINUE PARAMETERS after the BLOCK directive.

DIAGONAL PARAMETERS

All the specified parameters in the LIST 12 are included in a block diagonal approximation to the full matrix, based on one block for each atom. Both the SCALE FACTOR and the DUMMY OVERALL ISOTROPIC TEMPERATURE FACTOR are automatically included.

7.13 Defining the least squares matrix

Parameters may be referred to either *implicitly,* by just giving the parameter name (in which case that parameter is referenced for all atoms), or *explicitly* by specifying the parameter for an atom or group of atoms. All implicit specifications ignore H atoms.

```
e.g.
    IMPLICIT: x, u's
    EXPLICIT C(1,X), O(1,U'S) UNTIL O(14)
```

A parameter may not be referenced more than once either explicitly or implicitly. A parameter *may* be referenced both implicitly and explicitly, in which case the explicit reference takes precedence.

```
e.g.
    BLOCK x's              (implicit reference)
    FIX Pb(1,y)            (explicit reference)
             This establishes the refinement of z,y,z for all atoms
             except Pb(1), for which only x and z are refined.
```

```
EXAMPLES :
    1. BLOCK SCALE X
       FIX C(1,X)      ALLOWED
    2. BLOCK SCALE  X
       FIX X           NOT ALLOWED
```

The refinement directives are read and stored on the disc. Before the structure factor least squares

routines can use the information in LIST 12 (constraint directives), it is validated aga LIST 5 (the model parameters) and stored symbolically as a LIST 22. This is done automatically by the SFLS routines (section #RSFLS#), but the user can force the verification of LIST 12 by issuing the command \LIST 22.

7.14 Printing of LIST 12

LIST 12 may be listed with either
\PRINT 12
or
\SUMMARY 12
LIST 12 may be punched with
\PUNCH 12

7.15 Creating a null LIST 12

A null LIST 12, containing no refinement directives, may be created with
\CLEAR 12

7.16 Processing of LIST 12

LIST 12 is processed to greate a LIST 22 with
\LIST 22

```
Examples.
1. Full matrix isotropic refinement of a structure without H atoms

     \LIST 12
     BLOCK SCALE X'S U[ISO]
     END


2. Full matrix anisotropic of a structure with C(25) as the last
non-hydrogen, not refining the H atoms.

     \LIST 12
     BLOCK SCALE FIRST(X'S,U'S) UNTIL C(25)
     END


3. Refine all positions, aniso non-H, iso H atoms

     \LIST 12
     BLOCK SCALE X'S
     CONTINUE FIRST(U'S) UNTIL C(25)
     CONTINUE H(1,U[ISO]) UNTIL LAST
     END


4. Ride H(1) positions on C(21) positions, etc. There are 2 H on C(25)

     \LIST 12
     BLOCK SCALE X'S
     CONTINUE FIRST(U'S) UNTIL C(25)
     CONTINUE H(1,U[ISO]) UNTIL LAST
     RIDE C(21,X'S) H(1,X'S)
     RIDE C(22,X'S) H(2,X'S)
     RIDE C(23,X'S) H(3,X'S)
     RIDE C(24,X'S) H(4,X'S)
     RIDE C(25,X'S) H(51,X'S) H(52,X'S)
     END


5. A fragment is distributed over 2 sites. The fragments are
C(100) C(101) O(102) C(103) and C(200) C(201) O(202) C(203)

     \LIST 12
     BLOCK SCALE X'S
     ... ...
     EQUIVALENCE C(100,OCC) UNTIL C(103) C(200,OCC) UNTIL C(203)
     WEIGHT -1 C(200,OCC) UNTIL C(203)
     END
```

7.17 Restraints - LIST 16

This list defines the restraint to be used as supplemental observations.

```
\LIST 16
DISTANCES  VALUE, E.S.D= BOND1, BOND2
DISTANCES  VALUE, E.S.D= MEAN BOND1, BOND2
DISTANCES  VALUE, E.S.D= DIFFERENCE BOND1, BOND2
NONBONDED  VALUE, POWERFACTOR=  BOND1, BOND2
ANGLES     VALUE, E.S.D= ANGLE1, ANGLE2
ANGLES     VALUE, E.S.D= MEAN ANGLE1, ANGLE2
ANGLES     VALUE, E.S.D= DIFFERENCE ANGLE1, ANGLE2
VIBRATIONS VALUE, E.S.D= BOND1, BOND2
U(IJ)'S    VALUE, E.S.D= BOND1, BOND2
PLANAR             E.S.D  FOR 'ATOM SPECIFICATIONS'
LIMIT              E.S.D  FOR 'PARAMETER SPECIFICATIONS'
ORIGIN             E.S.D  FOR 'PARAMETER SPECIFICATIONS'
SUM                E.S.D  FOR 'PARAMETER SPECIFICATIONS'
AVERAGE            E.S.D  FOR 'PARAMETER SPECIFICATIONS'
SAME       BOND-ESD ANGLE-ESD FOR GROUP-1 AND GROUP-2 AND ...
RESTRAIN   VALUE, E.S.D= TEXT
DEFINE NAME = TEXT
COMPILER
EXECUTION
END


\LIST 16
DIST  1.39 , .01 =      C(1) to C(2), C(2) to C(3), C(3) to C(4)
DIST  0.0  , .01 = MEAN C(1) to C(2), C(2) to C(3), C(3) to C(4)
VIBR  0.0  , .01 =      C(1) to C(2), C(2) to C(3), C(3) to C(4)
U(IJ) 0.0  , .02 =      C(1) to C(2), C(2) to C(3), C(3) to C(4)
PLANAR             C(1) until C(6)
SUM                K(1,OCC), K(2,OCC) K(3,OCC)
SUM                ELEMENT SCALES  (twin element scale factors)
LIMIT              U[11] U[22] U[33]
END
```

The restraints that can be applied under this system are of a type originally described by J. Waser, Acta Cryst. 1963, 16, 1091. A good summary of the present facilities and aims is provided by J.S. Rollett in Crystallographic Computing, p170.

In this method of restraints, the user provides a set of physical or chemical restraints that are to be applied to the proposed model. These restraints are usually based upon observations of similar compounds (for example, bond lengths or bond angles) or upon known physical laws (for example, the difference in mean square displacement of two atoms along the bond that joins them). These restraints are not rigidly applied to the model, but each restraint has associated with it an e.s.d., which is used to calculate a weight so that the restraint can then be added into the normal equations. (The e.s.d.'s are provided on an absolute scale, and rescaled by the program onto the same scale as the xray data). In this way, the importance of the restraints, which are treated as extra observations, can be varied with respect to the importance of the X-ray data. If the structure is required to adhere closely to the proposed model, the restraints are given high weights (i.e. small e.s.d.'s) otherwise they can be given smaller weights.

If, at the end of a refinement, the restraints are not compatible with the Xray data, this is shown by a discrepancy between the requested value for the restraint, and that computed from the refine parameters. If this is found, the validity of the restraints that have been imposed should be carefully checked.

In order that the restraint routines should be completely general, each atom that is part of a restraint can be modified by a set of symmetry operators before the restraint is applied. (This is vital for molecules that lie across a symmetry element, as all the atoms that constitute the molecule are not present in LIST 5).

If a structure uses symmetry related atoms to form bonds, the command \DISTANCE with OUTPUT PUNCH=RESTRAIN can be used to set up a proforma restraints list, including symmetry codes. The distances and e.s.ds will have to be edited to the correct target values. Use appropriate values on the SELECT, INCLUDE and EXCLUDE directives for DISTANCE to tailor the generated list.

Note that restraints may be used without diffraction data, see the chapter 'Distance Least Squares' for examples.

NOTE
The restraint directives are read and stored on the disc. Before the structure factor least squares routines can use the information in LIST 16 (restraints), it is validated against LIST 5 (the model parameters) and stored symbolically as a LIST 26 (see #RLIST26#). This is done automatically by the SFLS routines (section #RSFLS#), but the user can force the verification of LIST 16 by issuing the command \CHECK (see later).

7.18 Parameter, atom, bond and angle specifications

Composite parameter specifications are not permitted (*e.g.* U's), atom specifications are as in Chapter 4. Two atoms that are bonded together are defined in the following way :

```
atom1 to atom2,
```

'atom1' and 'atom2' are standard atom specifications as described in chapter 4, and are separated from any other text on the line by at least one space. If there is more than one bond specification on a line, it may be separated from another by either a space or a comma. The 'TO' is mandatory, and is terminated by one or

more spaces.
   The definition of an angle is an extension of the definition of a bond:

```
        atom1 to atom2 to atom3,
```

The angle is defined as the angle subtended at atom2 by atom1 and atom3. The restraints routines apply all the required symmetry if specified in an atom definition, while still conserving the partial derivatives in their correct form.
\LIST 16
   The restraints routines regard all continuation directives as part of the original directive, so that the column of a character on a continuation directive will have had '80*n' added to it, where 'n' is the number of directives between the current continuation directive and the start of the directive. The ',', '=' signs and separator 'MEAN' are mandatory if shown in the definition.
   DISTANCES VALUE, E.S.D. = BOND1, BOND2, ...
      The bonds specified after the '=' sign are restrained to have a length of 'VALUE', with an e.s.d. of 'E.S.D.'.
   DISTANCES VALUE, E.S.D. = MEAN BOND1, BOND2, ...
      Initially the restraints routines calculate the 'MEAN' value of all the bonds specified by the directive. Each of the bonds specified is then restrained to be equal to 'MEAN' + 'VALUE', with an e.s.d. of 'E.S.D.'. The 'DELTA' used in the right hand sides of the normal equations is defined by :

```
        DELTA = MEAN + VALUE - BOND CALCULATED.
```

   DISTANCES VALUE, E.S.D. = DIFFERENCE BOND1, BOND2, . .
      Each of the bonds in this directive is restrained to be equal to 'VALUE' plus the length of each of the bonds that follow it. The computed value of 'DELTA' used in the right hand sides of the normal equations is thus given by :

```
        DELTA = VALUE + BOND(N) - BOND(M)
```

      Where BOND(N) occurs after BOND(M) in the directive.
      Each such restraint is added into the normal equations with an e.s.d. Of E.S.D. . However, as each bond is restrained to each of the bonds that follow it, $(N*(N-1))/2$ separate restraints are generated. Many of these restraints involve the same bond lengths and are thus not independent. To be strictly accurate, a non-diagonal weight matrix should be used with this restraint but such a facility is not available.
      The letters DIFFERENCE are terminated by one or more spaces and may be abbreviated to DIFF.
   NONBONDED VALUE, POWERFACTOR =, BOND1, BOND2, ...
      This restraint is similar to the 'DISTANCE' restraint in that the pairs of atoms defining the bond are restrained to be at the 'VALUE' distance appart. However, the weight to be given to the restraint is computed from the difference between the observed and the requested contact distance using the expression:

```
        weight = 10000*(requested/observed)**(powerfactor*12)
```

      When the observed equals the requested distance, the weight corresponds to an e.s.d. of .01. If the requested is less than the observed, the weight is reduced slowly as a function of the discrepancy. If the requested is greater than the observed, the weight rises rapidly with discrepancy. The function is like the repulsive part of a 6-12 energy expression, having greatest effect on anomalously short contacts. Powerfactors of between 1 and 4 seem to be suitable.

   ANGLES VALUE, E.S.D. = ANGLE1, ANGLE2, ...
      Each of the angles given in the directive is restrained to a value of 'VALUE', with an e.s.d. of 'E.S.D.'. The angles must be in degrees.
   ANGLES VALUE, E.S.D. = MEAN ANGLE1, ANGLE2, ...
      This is the analagous to the MEAN distance restaint, except that the mean value is computed for the specified angles and each of the angles is then restrained to 'MEAN' + 'VALUE', with an e.s.d. of 'E.S.D.'. The 'DELTA' values and the syntax rules are all the same as for the equivalent distance restraint.
   ANGLES VALUE, E.S.D. = DIFFERENCE ANGLE1, ANGLE2, . .
      This restraint is analogous to the DIFFERENCE restraint for bond lengths. Each of the angles in the directive is restrained to be equal to 'VALUE' plus each of the angles after it in the input. Although each such restraint is applied with an e.s.d. of 'E.S.D.', the same reservations about the validity of the weighting scheme exist here as for the equivalent distance restraint.
   VIBRATIONS VALUE, E.S.D. = BOND1, BOND2, ...
      The difference in mean square displacement along the bond direction of the two atoms that form the bond is restrained to be 'VALUE', with an e.s.d. of 'E.S.D.'. In general, 'VALUE' is assumed to be zero, while the e.s.d. reflects the maximum discrepancy in m.s.d. that would be expected for the type of bond being considered. If either or both of the given atoms is isotropic, the program will convert the

m.s.d. into the appropriate form and calculate the derivatives for the isotropic atom correctly.

    Note that the atoms defining a 'bond' need not actually be bonded, but merely serve to define a direction. For really bonded atoms, try an esd of .002; for 1-3 atoms or diagonals of phenyl groups, try .005.

U(IJ)'S VALUE, E.S.D. =, BOND1, BOND2, ...

This is a similarity restraint, and may be used to ensure that the vibration parameters of adjacent atoms are similar, as must be the case even for flexible systems. The esd used must be softer than for a VIBRATION restraint, typically 0.01. In this restraint, the difference between corresponding u(ii) and u(ij) terms is restrained to be 'VALUE', with an e.s.d. of 'E.S.D.'. Each bond that is specified generates therefore six separate restraints, one for each of the anisotropic temperature parameters. If an atom with an isotropic temperature factor is included in this restraint, the specified bond and all six restraints are ignored.

PLANAR E.S.D. FOR 'ATOM SPECIFICATIONS'

This directive instructs the system to compute the mean plane through the atoms given in the atom specifications, and then to restrain each of the atoms to lie in the plane. The 'E.S.D.' with which each atom is restrained to be on the plane is given in angstrom. This parameter is optional and has a default value of 0.01. 'FOR' is optional. 'ATOM SPECIFICATIONS' define the atoms that are on the plane. Each 'ATOM SPECIFICATIONS' may consist of one atom, together with symmetry data, or two atoms separated by 'UNTIL'. One or more specifications must be given.

```
Examples :
 PLANAR C(1,2) UNTIL C(6) C(9) C(10,2,2)
 PLANAR 0.05 C(1) C(2) UNTIL C(6)
 PLANAR 0.05 FOR C(1) C(2) UNTIL C(6)
 PLANAR FOR C(1,2) UNTIL C(6)
```

LIMIT E.S.D. FOR 'PARAMETER SPECIFICATIONS'

This restraint seta a target shift of zero for the specified parameters, with the specified esd, and thus tries to limit the shift in the parameters. Since it modifies the normal matrix, it does not have the same effect as partial shifts (SHIFT,MAXIMUM,and FORCE in SFLS [section #RSFLS#]). In particular, the e.s.d. on the parameter will depend upon the E.S.D. given to this restraint. The default for E.S.D. is .001. Reducing this to about .00001 will have almost the same effect as FIX in LIST 12. Increasing it to 10.0 will cause the restraint to have almost no effect unless the parameter involved is almost singular with respect to some other parameter. Note that this is only a restraint, and if the medel and X-ray data are good, the specified parameters will still shift. This restraint is valuable during the development of a poor starting model.

ORIGIN E.S.D. FOR 'PARAMETER SPECIFICATIONS'

This is used for polar space groups, where the singularity up the polar axis may be removed by holding the electron weighted sum of all the coordinates up that axis constant.

```
Example
 ORIGIN Y
```

SUM E.S.D. FOR 'PARAMETER SPECIFICATIONS'

This restraint holds the sum of the parameters on the directive constant during the refinement. A typical case is where several (more than 2, which are better treated with EQUIVALENCE, in LIST 12) atoms share a site. 'E.S.D.' is the e.s.d. with which the sum of the parameters is held constant. This is an optional parameter and has a default value of 0.0001. 'FOR' is optional. 'PARAMETER SPECIFICATIONS' define the parameters that are to be summed. They may be given as :

```
overall parameters e.g. SCALE,
all atomic parameters of one type e.g. X, Y, U[11],
atomic parameters of one type for a group of atoms
      e.g. NA(1,OCC) UNTIL RB(6),


 Examples :
 SUM 0.0001 NA(1,OCC) UNTIL RB(6)
 SUM LAYER SCALES
```

AVERAGE E.S.D. FOR 'PARAMETER SPECIFICATIONS'

For this directive, the system computes the mean of the given parameters, and then restrains each to have the mean value with an e.s.d. of 'E.S.D.'. The parameters are as for the 'SUM' directive above.

SAME BOND-ESD ANGLE-ESD FOR GRUOP-1 AND GROUP-2 AND ...# The first group on the card is the 'target' all following groups are mapped onto it (in order specified) and the distances angles restrained - using the connectivity of the first group.

The first two arguments are the e.s.d for bond length restraints and the e.s.d for angle restraints. Groups are seperated by the word 'AND'. NOTE the andence of the usual '=' sign. I.E:

```
SAME 0.01, 0.1 FOR RESI(1) AND RESI(2)
  maps all atoms in resi(1) onto all the atoms in resi(2) - note that
  although this shorthand is appealing, the order of resi(1) and resi(2)
  must be identical in List 5, although the residues may interpenetrate.


SAME 0.01 , 0.1
CONT C(17)  C(18)  H(183) H(182) H(181) AND
CONT C(17)  C(18)  H(182) H(181) H(183)
  imposes 3-fold symmetry on a single methyl group.


SAME 0.01 , 0.1
CONT C(17)  C(18)  H(183) H(182) H(181) AND
CONT C(17)  C(19)  H(193) H(192) H(191) AND
CONT C(8)   C(9)   H(93)  H(92)  H(91)  AND
CONT C(8)   C(10)  H(103) H(102) H(101) AND
CONT C(14)  C(15)  H(153) H(152) H(151) AND
CONT C(14)  C(16)  H(163) H(162) H(161)
  restrains six methyl groups to have the same geometry as each
  other. Combining the last two restraints would make all the
  methyls have 3 fold symmetry, and all be the same.
```

Errors are generated if 1) the size of any of the groups on the SAME card is not the same as the first group. 2) the element type in a group does not match the corresponding element type in the first group.

Warnings are printed if there are zero bonds to any of the atoms in the first group.

The comma separating the e.s.d arguments, and the 'FOR' separating the e.s.d.s from the atom specifications are optional. The second e.s.d is optional, the default is 0.1 degrees. The first e.s.d is optional unless you wish to specify the second, the default is 0.01 Angstroms.

List 41 (bonds) is loaded by the restraint generating routine, if it does not exist an error will occur. (By default L41 is kept up to date with the current model.)

7.19 General restraints

The 'general restraint' enables the user to write out a restraining equation explicitly. The system automatically calculates the value of the restraint and then evaluates the partial derivatives for each of the refinable parameters

Thes restraints look like simple fortran statements involving operators and operands.

OPERATORS

The available operators are :

```
(
)
**            must be followed by an operand.
*             must join two operands.
/             must join two operands.
+             must precede an operand.
-             must precede an operand.
```

An operand may be a simple variable or an expression enclosed in parentheses.

The operators above assume their normal FORTRAN meanings, and the combination of operands and operators is the same as in standard FORTRAN, except that all calculations are done in floating point.

ATOMIC COORDINATES

These are specified by a modified form of the atom definition given above. This is :

```
TYPE(SERIAL,S,L,TX,TY,TZ,KEY)
```

KEY Specifies the relevant coordinate of the atom. The KEY is regarded as an obligatory parameter, but for the remaining symmetry parameters, the drop out rules and default settings described under the atom definition may be applied, so that the simplest form of coordinate definition is TYPE(SERIAL,KEY), similar to a LIST 12 definition . The usual parameter keys are recognized.

OVERALL PARAMETERS

The usual overall paameter keys are recognized.

VARIABLES

These are unsubscripted variables specified by up to 8 characters, of which the first must be a letter. Many commonly occurring crystallographic quantities are already prestored by the system, and the user has the ability to declare new constants with a 'DEFINE' directive, which is described below. When a user defines a new variable, he must not use a name that has already been declared by the system. The system variables are:

ARRAY VARIABLES

The system has pre-stored various arrays and variables holding useful crystallographic information,

and users may not define or declare new arrays. The addressing is done in the normal Fortran manner, except that the element required must be specified by numeric arguments, and not variables. Thus A(3,1) is allowed, but A(I,J) is illegal.

```
A(6)        the cell parameters (angles in radians)
CV          real cell volume
AR(6)       reciprocal cell parameters (angles in radians)
RCV         reciprocal cell volume
G(3,3)      real metric tensor
GR(3,3)     reciprocal metric tensor
L(3,3)      real orthogonalization matrix
LR(3,3)     reciprocal orthogonalization matrix
CONV(3)     conversion factor for the 'U(ij)'s' from 'U[iso]'
RIJ(6)      coefficients needed to calculate [sin(theta)/l]**2
ANIS(6)     coefficients needed to calculate the temperature
            factor from the anisotropic temperature factors
SM(3,4,p)   symmetry matrix 'p', where the translational
            part is stored in sm(i,4,p)
SMI(3,4,p)  inverse symmetry operators
NPLT(3,n)   non-primitive lattice translations
PI          3.141......... etc.
TPI         2*Pi
TPIS        2*pi*pi
DTR         conversion of degrees to radians
RTD         conversion of radians to degrees
ZERO        0.0
```

The following functions are also recognized :

```
SIN(ARG)      COS(ARG)      TAN(ARG)      ACOS(ARG)
ASIN(ARG)     ATAN(ARG)     EXP(ARG)      SQRT(ARG)
```

7.20 General restraints

There are two directives.
DEFINE NAME = TEXT
This may be used to set up a user defined variable, which may be referred to later on by 'NAME'. The text comprises a series of variables and numeric constants interspersed with operators. The 'NAME' must not be one of the standard functions or variables, and may be overwritten several times - i.e. its value may be redefined.
RESTRAIN VALUE, E.S.D. = TEXT
The physical or chemical quantity defined by the 'TEXT' is restrained to be 'VALUE', with an e.s.d. of 'E.S.D.'. The text is comprised of operands separated by operators. The system will differentiate the 'TEXT' with respect to each of the refinable coordinates that it contains and add the derivatives to the normal matrix in the usual way.

7.21 Debugging restraints

Debugging commands are available to help with the creation of general restraints
COMPILER
During the formation of LIST 26 (see #RLIST26#), the input directives are listed, together with various internal stacks.
EXECUTION
During the application of the restraints to the normal equations, various stacks are printed and all the calculated derivatives are printed (use with care).

7.22 Printing the contents of LIST 16

The contents of LIST 16 may be listed with:
\PRINT 16
or
\SUMMARY LIST 16
LIST 16 may be punched with:
\PUNCH 16

7.23 Creating a null LIST 16

A null LIST 16, containing no restraints, may be created with
\CLEAR 16

```
restrain a set of distances to 1.5 angstrom with an
e.s.d. of 0.03, note the use of symmetry indicators.

    DISTANCE 1.5 , 0.03 = C(1) TO S(1) , C(1,5) TO S(1,5)
    CONT                  S(1,7,1,-1) TO C(1,7,1,-1)

restrain the first distance above explicitly, by a user defined
restraint
```

```
            RESTRAIN 1.5 , 0.03 = SQRT
            CONT ((C(1,5,X)-S(1,5,X))*(C(1,5,X)-S(1,5,X))*G(1,1)
            CONT +(C(1,5,X)-S(1,5,X))*(C(1,5,Y)-S(1,5,Y))*G(1,2)
            CONT +(C(1,5,X)-S(1,5,X))*(C(1,5,Z)-S(1,5,Z))*G(1,3)
            CONT +(C(1,5,Y)-S(1,5,Y))*(C(1,5,X)-S(1,5,X))*G(2,1)
            CONT +(C(1,5,Y)-S(1,5,Y))*(C(1,5,Y)-S(1,5,Y))*G(2,2)
            CONT +(C(1,5,Y)-S(1,5,Y))*(C(1,5,Z)-S(1,5,Z))*G(2,3)
            CONT +(C(1,5,Z)-S(1,5,Z))*(C(1,5,X)-S(1,5,X))*G(3,1)
            CONT +(C(1,5,Z)-S(1,5,Z))*(C(1,5,Y)-S(1,5,Y))*G(3,2)
            CONT +(C(1,5,Z)-S(1,5,Z))*(C(1,5,Z)-S(1,5,Z))*G(3,3))
```

restrain some distances to their mean

```
            DISTANCE 0.0 , 0.03 = MEAN O(1) TO S(1) O(2) TO S(1)
            CONT                      O(1,2) TO S(1) O(1,7) TO S(1)
```

vibration restraints along a bond

```
            VIBRATION 0.0 , 0.01 = S(1,5) TO O(1,5) S(1,7) TO C(1,7)
            CONT                   S(1) TO O(1) S(1) TO C(1)
```

thermal similarity restraints

```
            U(IJ)  0.0 , 0.01 = S(1,5) TO O(1,5) S(1,7) TO C(1,7)
            CONT                S(1) TO O(1) S(1) TO C(1)
```

user defined restraints to some of the U(IJ)'S. This might cure a npd
atom

```
            RESTRAIN 0.0,0.01=S(1,U[11])-S(1,U[33])
            RESTRAIN 0.0,0.01=S(1,U[12])
            RESTRAIN 0.0,0.01=S(1,U[13])
            RESTRAIN 0.0,0.01=S(1,U[23])
```

7.24 The special restraints - LIST 17

\LIST 17

LIST 17 is generated automatically by the command \SPECIAL (section #RSPECIAL#), and is intended
to take care of floating origins and atoms on special positions. The user may create their own LIST 17, but
this will be over written by SPECIAL unless it this is deactivated.

7.25 Printing and punching LIST 17

LIST 17 may be printed with:
\PRINT 17
    or
\SUMMARY LIST 17

It is punched with:
\PUNCH 17

7.26 Creating a null LIST 17

A null LIST 17, containing no restraints, may be created with
\CLEAR 17

7.27 Checking restraints - CHECK

\CHECK LEVEL= END \CHECK HI END

The target values for the restraints can be checked against the calculated values by issuing the following
command :
\CHECK LEVEL=
    LEVEL=

```
    LOW       Default value
    HIGH
```

This command causes the restraints to be calculated, but not added into the normal equations. The
observed and calculated values are output to the listing file, with a summary on the terminal. If the
LEVEL is LOW, only restraints where the calculated value differs significantly from the target are
printed, otherwise all restraints are printed.

7.28 Weighting schemes for refinement- LIST 4

```
\LIST 4
SCHEME NUMBER= NPARAMETERS= TYPE= WEIGHT= MAXIMUM=
PARAMETERS P=
END
```

```
\LIST 4
SCHEME 14 3
END
```

The weighting of least squares refinement is still very controversial. The matter is discussed at some length by Schwartzenbach *et al* in Statistical Descriptors, and further insights may be gleaned from Numerical Recipies. Weighting the refinement can serve several purposes, and the weighting may need to be changed as the refinement proceeds. The weighting of Fo and Fsq refinements will be different. To a first approximation,

```
w(Fsq) = w(Fo)/2Fo
                    note the problem as Fo approaches 0.0
```

Initially the analyst must choose a scheme which will hasten the rate of convergence, and reduce the risk of the refinement falling into a false minimum. Towards the end of the refinement, once all the parameters have been approximately refined, a different scheme will be necessary to generate reliable parameter s.u.s (e.s.d.s)

My advice (DJW,1996) is to use unit weights for Fo refinement (1./4Fsq for Fsq refinement) until the structure is fully parameterised, and then an empirical scheme for the final refinement. It seems that pure 'statistical' weights are rarely satisfactory. The crucial thing is to look at the analysis of variance (/ANALYSE). The weighted residual (see definition of Fo' etc above) w(Fo'-Fc')**2 should be invariant for any rational ranking of the data. If there are any trends, then either the model is wrong or the estimates of w are wrong. If the model is believed to be full parameterised and substantially correct, the trend in residual can be used to estimate the weights.

7.29 Weighting for refinement against Fo

This set of weighting schemes should be selected when the minimisation function that is to be used during the least squares process is given by :

```
SUM( w*(Fo - Fc)**2 )
```

Where the summation is over all the reflections.

7.30 Weighting for refinement against Fsq

Refinement against Fo or Fsq is also controversial. The controversy is not really concerned with negative observations, since Fo can be given the sign of Io. The real problem is that the error distribution for Fo is not the same as that of Fsq, and is not simply related to it for very weak reflections. However, the argument is academic, since the error estimates for Fsq are not really known.

CRYSTALS provides two different alternatives for the case in which the minimisation function is given by :

```
SUM( w*(Fo**2 - Fc**2)**2 )
```

In the first of these options, the weights are calculated normally for Fo, and then converted so that they apply to Fsq by the operation :

```
w' = w/(4*Fo*Fo)
```

Where w' is the weight for Fsq and w is the weight for Fo. This option is selected by the parameter TYPE = 1/2Fo in the SCHEME directive above. For example, pseudo unit weights are selected by the input :

```
\LIST 4
SCHEME 9 0 1/2Fo
END
```

This option may be used with any of the weighting schemes above.

The second option also uses the weighting scheme types for Fo, except that Fsq is substituted for Fo in the equations. This option is selected by the parameter TYPE = Fo**2 in the SCHEME directive above. This choice would be suitable for the Chebychev weighting schemes.

\LIST 4
  SCHEME= NUMBER= NPARAMETERS= TYPE= WEIGHT= MAXIMUM=
    NUMBER=
      The number of the weighting scheme to be used (see below). The default value is 9 (unit weights).
    NPARAMETERS=
      The number of parameters to be provided for the weighting scheme. The default value is zero,
    TYPE

```
NORMAL
1/2Fo
Fo**2
CHOOSE      -  Default value
```

The value of NORMAL indicates that the weighting scheme type is for refinement against Fo. If TYPE is 1/2Fo or Fo**2 the weighting scheme type is for refinement against Fsq (see above).

If TYPE equals CHOOSE, one of the three previous type is chosen depending on the scheme number and the refinement type set in LIST 23 (structure factor control, see section #RLIST23#).

WEIGHT=

This parameter determines the weight assigned to reflections during the determination of Chebychev coefficients. For each reflection the weight with which it is added into the Chebychev normal equations is given by :

```
W = 1/[1+Fo**WEIGHT].
```

Thus if WEIGHT is equal to zero, all the weights will be the same and equal to 0.5. The default value is 2.0

MAXIMUM=

This parameter is used to set the maximum weight that can be applied, and is usefull for the Dunitz-Seiler scheme (13), and the Chebyshev schemes (10 and 14).

PARAMETERS P=

The parameters that are to be used to compute the weight for a given reflection are specified with this directive.

P=

This directive contains NPARAMETERS values. If this parameter is omitted, default values of zero are assumed for P.

The parameters must always be provided on the scale of Fo, not on the scale of Fc. For example, the agreement analysis programs can work on the scale of Fc, so that constants derived from such output must be put on the scale of Fo by multiplying them by the scale factor in LIST 5 (the model parameters).

7.31 Weights stored in LIST 6

If w is the weight to be applied to a reflection in the least squares refinement, the value to be stored in LIST 6 (section #RLIST06#) is sqrt(w), given the key SQRTW. If weights are computed by some external utiity, then either is should generate sqrt(w), or the values be converted after input to CRYSTALS - see scheme 5 below.

7.32 Weighting schemes

In the equations and explanations below, NP is an abbreviation of NPARAMETERS , the number of parameters required to define the weighting scheme, P(1) is the first such parameter and P(NP) the last parameter.

The available weighting schemes are :

```
1.      sqrt(W) = Fo/P(1), Fo < P(1) OR Fo = P(1)
        sqrt(W) = P(1)/Fo, Fo > P(1)


2.      sqrt(W) = 1       , Fo < P(1) OR Fo = P(1)
        sqrt(W) = P(1)/Fo, Fo > P(1)


3.      sqrt(W) = sqrt(1/(1 + [(Fo - P(2))/P(1)]**2))


4.      sqrt(W) = sqrt(1/[P(1) + Fo + P(2)*Fo**2 + . . + P(NP)*Fo**NP])


        try P(1) = 2*FMIN and P(2) = 2/FMAX,
        Cruickshank, Computing Methods and the Phase
        Problem, Pepinsky et al, 1961, page 45


5.      sqrt(W) = SQRT(data with the key 'SQRTW' in list 6)


6.      sqrt(W) = (data with the key 'SQRTW' in list 6)


7.      sqrt(W) = SQRT(1/(data with the key 'sigma(Fo)' in LIST 6))


8.      sqrt(W) = 1/(DATA WITH THE KEY 'SIGMA(Fo)' IN LIST 6)
                  ** remember that for schemes 7 & 8, LIST 6  **
                  ** must store both weight and sigma.        **


9.      sqrt(W) = 1.0 (Unit weights, default)


10.     sqrt(W) = sqrt(1.0/[A[0]*T[0]'(X) + A[1]*T[1]'(X) . .
                  +A[NP-1]*T[NP-1]'(X)])
                  Chebychev weighting - see below for details


11.      As for 10, but only applying previously determined parameters.


12.     sqrt(W) = sqrt([SIN(THETA)/LAMBDA]**P(1))
                  If  NP  is zero, a value of -1 is assumed for P(1) .


13.     sqrt(W) = sqrt([weight] * exp[8*(p(1)/p(2))*(pi*s)**2])
                  Dunitz Seiler weighting - see below for details


14.     sqrt(W) = sqrt(W' * (1. - (delta(F)/ 6* del(F)est)**2)**2)
            W' = 1.0/[A[0]*T[0]'(X) + A[1]*T[1]'(X) . .
                 +A[NP-1]*T[NP-1]'(X)]
                  Robust-resistant refinement - see below


15.      As for 14, but only applying previously determined parameters.


16.      sqrt(W) = Sheldrick SHELX-97 weights (page 7-31). The P1-P6
                  correspond to Sheldricks parameters a-f, but are not
                  refined automatically. Fo and Fc replace Fosq and
                  Fcsq for Fo refinement.
                  Use 0.1 0 0 0 0 .333 to get Sheldrick defaults.
```

7.33 Dunitz Seiler weighting - scheme 13

S is sin(theta)/lambda, pi is 3.141... Use $p(1) = 1$ and $p(2) = 4$ to simulate p=3, q=9 in Dunitz and Seiler Acta(1973),B29,589. Set MAXIMUM to 100. This scheme may be used for refinement before looking for bonding electrons.

7.34 Chebychev weighting schemes 10, 11

A[i] are the coefficients of a Chebyshev series in t[i]'(x), where x = Fo/Fo(max). (There is an account of CHEBYSHEV series in Computing Methods in Crystallography, edited by J.S. Rollett, p40). For this weighting scheme, the coefficients a[i] are calculated by the program using a least squares procedure which minimizes sum[(Fo - Fc)**4] over all the reflections. The resulting coefficients are stored in a new LIST 4 as weighting scheme type 11 (see below), and then used to calculate the weights for each of the reflections. It is recommended that several different values of NP are used (e.g 3 to 5), so that series of various orders are tested to see which gives the best fit. If negative or very small reciprocal weights are computed (i.e. the computed curve fall close to or crosses the ordinate axis), the parameter MAXIMUM can be used to restrict the maximum weight. For data on 'ordinary' scales, this will require a value of about 100. (This is best seen by computing an agreement analysis once the new weights have been calculated). The parameters P(i) need not be given, because they are to be computed. When the Chebyshev coefficients have been determined, p(1) is overwritten by the value determined for a[1]. (Carruthers and Watkin, Acta Cryst (1979) A35, 698). Scheme 10 generates the parameters needed for a scheme 11.

7.35 Robust-resistant weighting schemes 14, 15

This scheme should only be used towards the end of a refinement, when all of the expected variables have been introduced. It is usefull when there is suspicion of uncorrelated but significant errors in the data, and its effect is similar to scheme 10 in the absence of such errors. The expression for W' is as in Scheme 10 above,

except that X is Fc/Fc(max). This weight is then modified by a function expressing confidence in Fo-Fc. If the observed delta is large compared with the delta estimated from the Chebychev fitting, the reflection is down weighted. It it is more than 6 times the estimate, the weight is set to zero and the reflection flagged as an 'OUTLIER'. This scheme is recommended in preference to Scheme 10, which is kept for old times sake. See E. Prince, Mathematical Techniques in Crystallography, Springer-Verlag, for the background.

7.36 Statistical Weights, 16

This scheme can in principle be introduced at any time, but the parameters P(i) are best optimised near the end of a refinement. Typical values for CAD4 data are: p(1)=.001, p(2)=3.0 and p(4)=1.

7.37 Printing LIST 4

of LIST 4 may be printed with:

\PRINT 4

There is no command for punching LIST 4.

```
Example
     \ Weighting scheme type 10 (Chebyshev) with 3 parameters
     \LIST 4
     SCHEME NUMBER = 10,NPARAM = 3
     END
```

7.38 Weighting the reflections - WEIGHT

If the weighting scheme is changed, new weights are automatically computed for the next structure factor calculation. The computation of weights can be forced at any time with \WEIGHT.

\WEIGHT INPUT=

INPUT

Indicates which reflection list to use.

```
     6       Default
     7       Alternative reflection list
```

7.39 Reflection restriction list - LIST 28

```
\LIST 28
MINIMA COEFFICIENT(1)= COEFFICIENT(2)= ...
MAXIMA COEFFICIENT(1)= COEFFICIENT(2)= ...
READ NSLICES= NOMISSION= NCONDITION=
SLICE P= Q= R= S= T= TYPE=
OMIT H= K= L=
CONDITION P= Q= R= S= T= TYPE
SKIP STEP=
END
```

```
\LIST 28
MINIMA RATIO=3
READ NOMIS=1
OMIT 2 0 0
END
```

LIST 6 (section #RLIST06#) should contain all the reflections, including negative ones. LIST 28 can then be used to dynamically select which ones are to be omitted from a calculation. Several conditions may be specified, and ALL the conditions must be satisfied for a reflection to be used, i.e. the conditions are ANDed together.

It is also possible to specify individual reflections which are to be omitted.

TAKE CARE WHEN CHANGING LIST 28. If the conditions are relaxed, reflections may become acceptable for which Fc and phase have not been recomputed because they were rejected at an earlier stage. Recompute them all.

\LIST 28

MINIMA COEFFICIENT(1)= COEFFICIENT(2)=. .

This defines the coefficients whose minimum values are to be restricted.

COEFFICIENT= VALUE

Each such parameter defines one coefficient and its minimum value. The following are known to the system, BUT REMEMBER, with the exception of (sintheta/lambda)**2, which is computed for each reflection from the cell parameters, only those coefficients specifically stored in the LIST 6 (see section #RLIST06#) will have values.

H	K	L	/FO/
SQRTW	FCALC	PHASE	A-PART
B-PART	TBAR	FOT	ELEMENTS
SIGMA(F)	BATCH	INDICES	BATCH/PHASE
SINTH/L**2	FO/FC	JCODE	SERIAL
RATIO	THETA	OMEGA	CHI

```
        PHI             KAPPA           PSI             CORRECTIONS
        FACTOR1         FACTOR2         FACTOR3         RATIO/JCODE
```

MAXIMA COEFFICIENT(1) COEFFICIENT(2) . .
This defines the coefficients whose maximum values are to be restricted. See MINIMA above.

READ NSLICES= NOMISSION= NCONDITION=
This gives the number of conditional directives to follow.
NSLICES
This specifies the number of SLICE directives, default is zero.
NOMISSIONS
This specifies the number of OMIT directives, default is zero.
NCONDITION
This specifies the number of CONDITION directives, default is zero.
SLICE P= Q= R= S= T= TYPE=
This directive selects reflections to those giving values of $(h*p + k*q + l*r)$ in the range s to t. The number of such directives is specified on the READ directive above. TYPE indicates whether the selected reflections are accepted or rejected. The records are processed in the order given. If a reflection matches the conditions, the specified action is taken and no further slice directives are considered. This enables quite fancy intersections to be specified.
For example, a single layer of reciprocal points, or a set of adjacent layers, oriented in any desired crystallographic direction, can be selected.
P= Q= R= S= T=
These parameters, whose default values are zero, specify selected slices of reciprocal space.
TYPE=

```
    REJECT (default) causes rejection of selected reflections.
    ACCEPT           accepts reflections
```

OMIT H= K= L=
This directive causes the reflection with the indices H, K, and L to be omitted.
H= K= L=
These parameters specify the indices of the reflection to be omitted.
CONDITION P= Q= R= S= T= TYPE=
This directive causes selection of reflections giving values of $(h*p + k*q + l*r + s)$ exact multiples of 't'. TYPE indicates whether the selected reflections are accepted or rejected. The number of such directives is specified on the READ directive above. The records are processed in the order given. If a reflection matches the conditions, the specified action is taken and no further slice directives are considered. This enables quite fancy intersections to be specified.
For example, l odd layers can be rejected by setting 'r' and 's' to 1, 't' to 2.
P= Q= R= S= T=
These parameters, whose default values are zero, specify selected slices of reciprocal space.
TYPE=

```
    REJECT (default) causes rejection of selected reflections.
    ACCEPT           accepts reflections
```

SKIP STEP=
This directive can be used sample the data by skipping through LIST 6, (reflections, section #RLIST06#) and may be usefull to speed up initial refinement.
STEP=
This is the skip step length, and has a default of 1, i.e. all reflections are accepted. A value 3 selects every third reflection.

7.40 Creating a null LIST 28

```
\LIST 28
END
```

Allows all the reflections in LIST 28 to be used.

7.41 Printing the contents of LIST 28

LIST 28 may be listed by the command :
\PRINT 28
There is no command for punching LIST 28.

```
Example 1
\LIST 28
\ Set the minimum ratio I/sigma(i) to 3.0,
\ a maximum Fo to 1000
\ and omit the 0 2 0 reflection
\
MINIMA Ratio=3
MAXIMA Fo=1000
READ NOMIS = 1
OMIT 0 2 0
END


Example 2. To reject h and k simultaneously even:

condit p=1 s=1 t=1 type=accept      \lets ALL with h odd through
condit q=1 s=1 t=1 type=accept      \lets ALL with k odd through
condit s=1 t=1 type=reject          \rejects remaining.


Example 3. To reject all k=0, k=2:

slice q=1 s=0 t=0 type=reject
slice q=1 s=2 t=2 type=reject


Example 4. To reject all k=0, k=2 but keep the l=0 row:

slice r=1 s=0 t=0 type=accept
slice q=1 s=0 t=0 type=reject
slice q=1 s=2 t=2 type=reject


Example 5. To only allow specific zones, the ones wanted are
selected, and then the rest rejected, eg for h=0:

slice p=1 s=0 t=0 type=accept                \ accept the h00 zone
slice p=1 q=1 r=1 s=-500 t=500 type=reject \ reject everything else
```

7.42 Structure Factor Least Squares Calculations - \SFLS

```
\SFLS  INPUT=
CALCULATE LIST= MAP= /Fo/= THRESHHOLD=
SCALE LIST= MAP= /Fo/=
REFINE LIST= MAP= /Fo/= PUNCH= MATRIX= MONITOR= INVERTOR=
SHIFT  KEY= KEY=
MAXIMUM  KEY= KEY=
FORCE  KEY= KEY=
SOLVE MONITOR= MA=P /Fo/= PUNCH= MATRIX=
VECTOR MONITOR= MAP= /Fo/= PUNCH= MATRIX=
END


\SFLS
REFINE
REFINE
END
```

7.43 Definitions

Minimisation funtion for Fsq

```
Minimisation function = Sum[ w*(Fo**2 - Fc**2)**2 ]
```

Minmisation function for Fo

```
Minimisation function = Sum[ w*(Fo - Fc)**2 ]
```

R-factor for Fo

```
R = 100*Sum[//Fo/-/Fc//]/Sum[/Fo/]
```

The summation is over all the reflections accepted by LIST 28. This definition is used for both conventional and F-squared refinement.

R-Factor, Hamilton weighted

```
100*Sqrt(Sum[ w(i)*D'(i)*D'(i) ]/SUM[ w(i)*Fo'(i)*Fo'(i) ])

Fo' = Fo for normal refinement, Fsq for F-squared refinement.
Fc' = Fc for normal refinement, Fc*Fc for F-squared refinement.
D'  = Fo'-Fc'
```

The weighted R-factor stored in LIST 6 (section #RLIST06#) and LIST 30 (section #RLIST30#) is that computed during a structure factor calculation. The conventional R-factor is updated by either an SFLS calculation or a SUMMARY of LIST 6.

Minimisation function

```
Fo' = Fo for normal refinement, Fo*Fo for F-squared refinement.
Fc' = Fc for normal refinement, Fc*Fc for F-squared refinement.
D'  = Fo'-Fc'

MINFUNC = Sum[ w(i)*D(i)*D(i) ]
```

Good references to the theory and practice of structure factor least squares are in the chapters by J. S. Rollett and D. W. J. Cruickshank in Crystallographic Computing, edited by F. R. Ahmed, and chapters 4, 5 and 6 in Computing Methods in Crystallography, edited by J. S. Rollett.

7.44 Unstable refinements

If a refinement 'blows up', i.e. diverges rapidly, the user should seek out the physical cause (wrong space group, pseudo symmetry, incorrect data processing, disorder, twinning etc). If the cause of the divergence is simply that the model is too inaccurate, the divergence can by controlled by limiting the shifts applied in the first few cycles. The modern way to do this is via 'shift limiting restraints' (Marquardt modifier) in LIST 16. An older method was to use partial shift factors. These are set up by directives to the \SFLS command (section #RSFLS#). During the solution of the normal equations, the user may specify that more or less than the whole calculated shift should be applied. Alternatively, the program can be instructed to scale the shifts so that the maximum shift for any parameter group is limited to a given value. (The SHIFT , MAXIMUM and FORCE directives).

7.45 Sorting of LIST 6 for structure factor calculations

During a structure factor least squares calculation, the values for the real and imaginary parts of A and B and their derivatives are computed and stored. These values are then taken and formed into Fc and its derivatives, which are added into the normal matrix. Between reflections, the values for A and B and their derivatives are retained. If the next reflection in LIST 6 (section #RLIST06#) has a set of indices which are equivalent to the last reflection, the same values for the real and imaginary parts of A and B and their derivatives can be used.

This type of situation can arise either when anomalous scatterers are present, implying that F(h,k,l) is not equal to F(-h,-k,-l), or when an extinction parameter is being refined and formally equivalent reflections have different Fo values and mean path lengths. In this sort of case, the time for a structure factor calculation can be significantly reduced if reflections with symmetry related sets of indices are adjacent in LIST 6, when the conserved values of A and B can be used repeatedly.

In a similar way, during a structure factor calculation for a twinned crystal, the contribution and derivatives for each element are stored as they are calculated and then combined to produce /FCT/ when all the contributions have been accumulated. Between reflections this stored information is retained, so that if the next reflection contains contributions from elements with the same indices as the previous reflection, it is unnecessary to re-compute the A and B parts. Obviously, reflections with common contributors must again be adjacent in LIST 6, in which case a structure factor calculation, with or without least squares, takes only slightly longer than the corresponding normal calculation with the same number of observations.

\SFLS

The directives are carried out in the order in which they appear.

The directives REFINE, SCALE, and CALCULATE initiate cycles of S.F.L.S. calculations. If one of the directives SHIFT , MAXIMUM or FORCE is given following REFINE, a scaled shift will be applied to that cycle of refinement.

SFLS INPUT=

INPUT

Indicates which reflection list to use.

```
6       Default
7       Alternative reflection list
```

CALCULATE LIST= MAP= /Fo/= THRESHOLD=

If CALCULATE is included with other commands within a single \SFLS block, it **MUST BE** the last command.

This directive indicates that structure factors should be calculated, but that no refinement of any type should be done. Structure factors are computed for every reflection and used to compute R and Rw for all data. R and Rw are also computed for reflections with I>threshold Sigma(I). The default value for the threshold is 4.

The directives SHIFT , MAXIMUM and FORCE may not be given before the next REFINE directive.

LIST

Controls the listing of reflection information.

```
OFF    -   Default
MEDIUM
HIGH
```

The value OFF indicates that the discrepancy for each reflection (/Fo-Fc/)/Fo is computed and if greater than 3*(overall R factor) from the previous cycle of structure factors, a warning is printed. Only the first 25 such reflections are listed.

If the ENANTIOPOLE parameter is activated in LIST 23 (structure factor control, see section #RLIST23#), sensitive reflections, for which /(F+)-(F-)/ .> .05 *((F+)+(F-)/2), are also listed.

If LIST is MEDIUM , the structure factors are listed as they are computed. The output contains h, k, l, Fo (on the scale of Fc), Fc, the phase and sin(theta)/lambda, the unweighted and weighted delta's. (Fo - Fc or Fo**2 - Fc**2, depending upon the type of refinement being done), and information which is useful when anomalous dispersion effects are present, and contains the real part of Fc (/Fc'/), the imaginary part of Fc (/Fc"/), the computed difference between /F(h,k,l)/**2 and /F(-h,-k,-l)/**2, and the calculated or theoretical Bijvoet ratio (t.b.r.).

When a twinned crystal structure is being refined, LIST = HIGH gives FoT and /FcT/ in place of Fo and Fc, respectively. Also, the contributions of each element to each reflection of a twinned crystal are listed. As well as /FcT/ and the indices, Fc, multiplied by the square root of the element scale factor, and the element number are also printed for each component under the column headed by /FC'/. This option is only obeyed if LIST 13 (section #RLIST13#) indicates that a twinned crystal structure is being refined.

List MEDIUM and HIGH produces one line of output for each reflection.

MAP

Controls printing of the memory map - mainly used by programmers

```
NONE   -   Default value
PART
FULL
```

If MAP is PART , a list of core addresses is printed, together with any unused locations. If MAP is FULL , the addresses and contents of the areas of code claimed by each list as it is brought down are printed on the line printer as the list is loaded. This option produces reams of output and should never be used except by programmers. If MAP is NONE , its default value, a core map is not printed.

/Fo/

Controls the treatment of twinned data.

```
FoT        -   Default value
Scaled-FoT
```

In the refinement of a twinned crystal, if /Fo/ = FoT , its default value, the FoT is output as the data for the key /Fo/ , the corresponding /FcT/ is output for Fc , and the phase is arbitrarily set to zero. If /Fo/ is SCALED-FoT , the data for Fo , Fc and PHASE contain an estimate of the required quantities for the element in whose reference system the nominal indices are given, i.e. estimates of the resolved data are produced.

THRESHOLD

Sets a sigma(I) threshold for computing the restricted Rfactor. The default value is 4.0

SCALE LIST= MAP= /Fo/=

This directive indicates that structure factors should be calculated and the overall scale factor should be refined. The directives SHIFT , MAXIMUM and FORCE may not be given before the next REFINE directive.

LIST

This parameter has the same options as for the CALCULATE directive above.

MAP

This parameter has the same options as for the CALCULATE directive above.

Fo

This parameter has the same options as for the CALCULATE directive above.

REFINE LIST= MAP= /Fo/= PUNCH= MATRIX= MONITOR= INVERTOR=

This directive indicates a complete structure factor least squares calculation.

LIST

This parameter has the same options as for the CALCULATE directive above.

MAP

This parameter has the same options as for the CALCULATE directive above.

Fo

This parameter has the same options as for the CALCULATE directive above.

PUNCH

Controls punching LIST 5 (the model parameters) to the .PCH file

```
NO    -   Default value
YES
```

If PUNCH is YES, LIST 5 is punched after the current cycle of refinement.
MATRIX
Controls re-use of the normal matrix

NEW - Default value
OLD

If MATRIX is NEW, a new matrix is computed for the current cycle.

If MATRIX is OLD , the left hand side of the normal matrix is not accumulated during the cycle of refinement. Instead, the version that already exists is used with the new right hand sides. This option is particularly useful at the end of a refinement of a large structure when the left hand side does not change appreciably from cycle to cycle. It greatly reduces the time for a cycle.
MONITOR
Controls the shift information printed out.

LOW - Default value
MEDIUM

The MEDIUM listing outputs details about all parameters refined, and lists the values, shifts and e.s.d.s of all parameters in LIST 5. The LOW listing outputs information only for those l.s. parameters for which the SHIFT RATIO exceeds 3.0, and/or the SHIFT/ESD exceeds 1.0 . Only those atoms in LIST 5 containing one or more refinable parameters are listed.
INVERTOR
Two matrix inversion methods are provided.

CHOLESKI - Default value
EIGENVALUE

The default Choleski invertor is suitable for most well conditioned problems. The alternative Eigenvalue invertor is suitable for ill-conditioned cases, such as those involving pseudo-symmetry. It is slower than the Choleski method and cannot handle such large normal matrices. The memory required is almost three times that used for each block in the Cholesky method.
AUGFACT
This is a constant to be added to all eigen values during eigenvalue inversion. The default value is 0.0.
FILTER
The inverse of eigen values below this threshold are set to 0.0 rather than 1./value. It filters out latent singularities. The default is 0.0
DISCRIMINATOR
If the ratio of two adjacent eigenvalues in a ranked listing exceeds this value, the inverse of the smaller and all subsequent eigen values is set to 0.0. The default is 100.0
SHIFT KEY = VALUE KEY = VALUE . . . .
This directive sets the shift factor for the specified cycle of refinement. (The shift factor is the amount by which the calculated shift is multiplied before it is applied to generate the new parameters). For each of the parameters given by the KEYS on the directive, the shift factor is changed to the value given by VALUE. The = sign is not optional.

If more than one shift directive (SHIFT , MAXIMUM or FORCE) is given for the same parameter, only the last is obeyed.

The following KEYS are recognized :

GENERAL   This refers to all the variable parameters
OVERALL   This refers to the overall parameters

OCC   U[ISO]   X   Y   Z
U[11]   U[22]   U[33]   U[23]   U[13]   U[12]
SPISO   SPSIZE   LINISO   LINSIZE LINDEC   LINAZI
RINGISO RINGSIZ RINGDEC RINGAZI

This is the default for GENERAL, OVERALL and SPECIAL parameters, with default shift factors 1.0.
MAXIMUM KEY = VALUE KEY = VALUE . . . .
This directive is similar to the directive SHIFT above, except that the maximum shift that is applied for the given parameters cannot be greater than VALUE. The units of VALUE are conventional, WITH x, y, z measured in angstrom, and ADPs in Angstrom sq. If none of the shifts exceend VALUE, then they are applied unmodified. This provides a method of automatically scaling down the applied shifts if the matrix inversion has become unstable. Shift limiting restraints (LIST 16) are a more controlled alternative

The KEYS are the same as in SHIFT above, and this is the default action for Occ, adps and positions,

with maximal values:

```
OCC 1.0
U*  0.05 (Angstrom sq)
X's 1.0  (Angstom)
```

FORCE KEY = VALUE KEY = VALUE . . . .
    This is similar to MAXIMUM above, except that the maximum shift is scaled to VALUE even if it is
    less than VALUE.

    The KEYS are the same as in SHIFT above. There is no default shift.
VECTOR MONITOR MAP Fo PUNCH MATRIX
    This is an obsolete feature, and will be removed at a later date.
        This directive indicates that structure factors are to be calculated and then the shift vector stored in
    LIST 24 (see #RLIST24#) is to be applied. This is used to apply a shift vector calculated from one of
    the eigenvalues of the normal matrix. Although no new matrix is produced by this directive, sufficient
    space must be allocated for the normal matrix, since it is loaded when the new coordinates are
    calculated.
MONITOR
    This parameter has the same options as for the CALCULATE directive above.
MAP
    This parameter has the same options as for the CALCULATE directive above.
Fo
    This parameter has the same options as for the CALCULATE directive above.
PUNCH
    This parameter has the same options as for the REFINE directive above.
MATRIX
    This parameter has the same options as for the REFINE directive above.

7.46 Processing of the refinement directives

    The program expands the CALCULATE, SCALE or REFINE directives into sub-directives. These
sub-directives **MUST NOT** be given by a user:

```
1.   \REFINE
                Compute structure factors and
                derivatives. No refinement is
                actually done.
2.   \SCALE
                Calculate structure factors and
                refine the overall scale factor.
3.   \CALCULATE
                Calculate structure factors.
4.   \RESTRAIN
                Apply the restraints stored in
                the current lists 16 and 17.
5.   \INVERT
                Invert the current normal matrix
                and store a shift list as list 24.
6.   \SOLVE
                Take the current list 5 (the model parameters)
                and apply the shifts given in the current
                list 24.
7.   \NEWSHIFTS
                Allocate space for list 24.
8.   \CYCLENDS
```

e.s.d.s
    Most publication listings require e.s.ds. These are computed from the normal matrix. If LIST 5 (the
    model parameters) has been modified in **ANY WAY** (including simply renaming or ordering atoms)
    since the last refinement cycle, the matrix will be invalid.
        CRYSTALS will warn you that LIST 11, the normal matrix, cannot be loaded. To create a valid
    matrix without changing the parameter values, compute a refinement cycle but set all the shifts to
    zero.

```
\SFLS
REFINE
SHIFT GENERAL = 0.0
END
```

7.47 Analysis of residuals - \ANALYSE

```
\ANALYSE INPUT=
FO INTERVAL= TYPE= SCALE=
THETA INTERVAL=
LIST LEVEL=
LAYERSCALE AXIS= APPLY= ANALYSE=
END


\ANALYSE
LIST HIGH
END
```

ANALYSE provides a comparison between Fo and Fc as a function of the indices, various parity groups, ranges of F and ranges of sin(theta)/lambda. For a well refined structure with suitable weights, <Fo>/<Fc> should be about unity for all ranges, and <wdeltasq> should also be about unity for all ranges. A serious imbalance in Fo/Fc may mean the structure is incomplete, or unsuitable data reduction (section #RDATAREDUC#). A systemstic trend in <wdeltasw> may mean unsuitable weights are being used.

The monitor listing is always just as a funtion of F. The output to the listing file is user controlled.

This routine will also compute approximate layer scale factors for data which has been collected by layers. These can be refined in the least squares to complete a refinement.

\ANALYSE INPUT=
INPUT
Indicates which reflection list to use.

```
6       Default
7       Alternative reflection list
```

FO INTERVAL= TYPE= SCALE=
Controls the analysis as a function of F.
INTERVAL=
The interval between successive ranges of F. Its value should be determined in combination with the parameter TYPE.
TYPE
Controls how F is sampled.

```
sqrt(Fc)  -  Default value
Fc
sqrt(Fo)
Fo
```

If TYPE is sqrt(Fc), (its default value), the interval between successive F ranges is based on 'the square root of Fc' Values of Fc in the interval '0 to INTERVAL**2' will be analysed in the first range, Fc values that lie in the range INTERVAL**2 to 4*INTERVAL**2 in the second and so on). INTERVAL is thus the increment in the square root of Fc between successive ranges and, in this case, has a default value of 1. If TYPE is Fc , the interval between successive Fc ranges is based on the value of Fc. In this case INTERVAL is the increment in Fc and has a default value of 2.5.
SCALE
Controls the scale of the listing

```
Fo  -  Default value
Fc
```

If SCALE is Fo, the reflection information is printed on the scale of Fo. (This is useful as the weighting parameters in LIST 4 (section #RLIST04#) must be provided on the scale of Fo). If SCALE is Fc, the reflection information is printed on the scale of Fc.

THETA INTERVAL=
This directive determines the interval between successive sin(theta)/lambda squared ranges.
INTERVAL=
The default is 0.04.

LAYERSCALE AXIS= APPLY= ANALYSE=
This directive allows the results of layer scaling to be investigated.
AXIS=
Selects the axis for layer scaling

```
NONE  -  Default value
H
K
L
```

The default value of NONE indicates that no layer scaling is to be done. H, K and L indicate the axes up which layer scaling is to be done.
APPLY=

```
NO   - Default value
YES
```

When layer scaling has been completed and the results printed, the calculated scale factors will be applied to the stored Fo data if APPLY is YES . If APPLY is NO , its default setting, then the new scales will not be applied to the data. If AXIS is NONE , then APPLY is ignored.

ANALYSE

```
NO
YES  - Default value
```

If ANALYSE is YES, a second agreement analysis will be performed after the layer scaling so that the results of the new scales can be seen. (This is true whether the new scales are applied or not, i.e. independent of the value of APPLY). In this way the effects of layer scaling can be seen without damaging the data. If ANALYSE is NO , the second agreement analysis is suppressed.

LIST LEVEL=
This directive determines the amount of output produced.

LEVEL=

```
HIGH
LOW   - Default value
```

If LEVEL is LOW the analysis is against Fo and sin theta only.

7.48 Least squares absorption correction - \DIFABS

```
\DIFABS  ACTION= MODE= INPUT=
CORRECTION THETA=
DIFFRACTION GEOMETRY= MODE=
END


\DIFABS ACTION=NEW MODE=FC
END
```

Although this is a least squares fitting technique for an arbitary model, it does not form part of the main refinement module. The DIFABS parameters cannot be refined simultaneously with the atomic parameters.

A low order term Fourier series is used to model an absorption surface for differences between the observed structure factors and those obtained from a structure factor calculation after isotropic least squares refinement. Spherical polar angles are used to define the incident and diffracted beam path directions so that each reflection is characterised by four angles - viz. PHI(p), MU(p), PHI(s), and MU(s). A theta-dependent correction is evaluated to allow for diffracted beams with different path lengths occurring at the same polar angles. A low order term Fourier series is used in Bragg angle THETA, but is highly correlated with the temperature factors, and not normally recommended. This version is general for any 4-circle diffractometer data collection geometry.

The quantity minimised is the sum of the squares of the residuals, Rj, where

```
        Rj = ( //Fc/-/Fo//)wj
```

The weighting function, wj, used is derived from the overall scale factor, the counting statistics standard deviation, and the Lorentz-polarisation factor.

In the original implementation, the correction factor was applied to Fo. This lead to criticism in the literature that the observations were being tampered with. In the current implementaion in CRYSTALS, the correction can be applied to Fo or Fc.

References: N. Walker and D. Stuart, 1983, Acta Cryst., A39, 158 - 166. The code is incorporated with the permission of Dr N. Walker

Implementation#
The correction is evaluated using observed structure factors, /Fo/, corrected for Lorentz-polarisation effects and any decay in intensity standards during data collection, with systematically absent reflections removed. Since equivalent reflections will be measured at different diffractometer settings, the correction should be calculated and applied to the data set without any transformation of the reflection indices, and without symmetry- equivalent or Friedel-pair reflections being averaged. Calculated amplitudes must be obtained from the isotropic refinement of an as-complete a model as practical from the unique (merged) data set. Such a LIST 6 (reflections, section #RLIST06#) will probably be unsuitable for Fourier or difference maps (since these expect a unique segment of data only) unless you then remerge the data. The best maps must be computed with the correction applied to Fo before the data is merged. In addition, the most reliable merging R factor (Rint) must be computed from corrected Fos.

WARNING

To use DIFABS most successfully, you should probably do data-reduction again from scratch, inhibiting the merging of all but exactly equivalent reflections.

In favourable cases, when the observed data is the unique segment plus a small redundant volume (e.g. often the -1 layers at Oxford), you may get away with applying the correction to normally (merged) processed data during structure development.

Once the structure is fully developed (ie all atoms found and partially refined with an extinction correction if necessary), data reduction should be repeated inhibiting all index transformations. New values of Fc must be computed from isotropic atoms (Use UEQUIV in \EDIT to recover equivalent isotropic temperature factors, and then do a few cylces of isotropic refinement) and the DIFABS correction applied to Fc. Anisotropic refinement can be computed to completion (including optimisation of weights) using unmerged data. If you wish to see an absorption correctd Rint and compute a final difference map, the data must be re-merged. Use DIFABS with MODE = TRANSFER to move the correction onto Fo before transforming indices, sorting and merging the data.

\DIFABS ACTION= MODE= INPUT=

INPUT

Indicates which reflection list to use.

6   Default
7   Alternative reflection list

ACTION=

Controls the action on LIST 6 (reflections, section #RLIST06#), and has three values:

TEST   -   Computes the correction, but does not apply it
UPDATE -   Tries to update LIST 6
NEW   -   DEFAULT. Creates new LIST 6

If UPDATE is specified, the stored values of Fo are over written. If NEW is specified, a new LIST 6 is written to disc. The disc will be extended sufficiently to accommodate the new list.

MODE=

Controls the mode of application of the correction, and has three values:

FO   - Applies the correction to Fo
FC   - Applies the correction to Fc
TRANSFER - Applies the inverse of the Fc correction to Fo

CORRECTION THETA=

Controls whether a theta-dependent correction is to be applied. - NOT RECOMMENDED.

THETA=

NO - Default value
YES

DIFFRACTION GEOMETRY= MODE=

Controls the geometry used for data collection to be input.

GEOMETRY=

The type of diffractometer used is specified:

CAD4   - Default value
SYNTEX-P1
SYNTEX-P21
PICKER   - Picker FACS-I
PW1100   - Philips PW1100

MODE=

The mode of data collection is given:

BISECTING   - Default value
PARALLEL
GENERAL

This example assumes that there are no equivalent reflections.
*   \DIFABS*
*   DIFFRACTION GEOMETRY=SYNTEX-P1*
*   END*

This example demonstrates a total re-processing of the data, including
converting atoms to isotropic if they have previously been refined
anisotropically. Note that a theta dependent correction from
International Tables is applied during data reduction
(see also section #RDATAREDUC#). The theta
dependant correction in DIFABS is ill-conditioned and unstable.

```
\ save the contents of the old dsc file
\PURGE NEW
END
\ Connect the reflection file to HKLI
\OPEN HKLI ZNCPD.HKL
\ Use an \HKLI command to apply the tabulated theta correction
\HKLI
READ NCOEF=12 FORMAT=FIXED UNIT=HKLI F'S=FSQ CHECK=NO
INPUT H K L /FO/ SIGMA(/FO/) JCODE SERIAL BATCH THETA PHI OMEGA KAPPA
FORMAT (5X,3F4.0,F9.0,F7.0,F4.0,F9.0,F4.0,4F7.2)
STORE NCOEF=6
OUTPUT INDICES /FO/ BATCH RATIO/JCODE SIGMA(/FO/) CORRECTIONS SERIAL
ABSORPTION PHI=NO   THETA=YES PRINT=NONE
THETA 16
THETAVALUES
CONT 0   5 10 15 20 25 30 35 40 45 50 55 60 65 70 75
THETACURVE
CONT 3.61   3.60   3.58   3.54   3.50   3.44   3.37   3.30
CONT 3.23   3.16   3.09   3.02   2.96   2.91   2.86   2.82
END
\LP
END
\SYSTEMATIC
\ preserve the original indices
STORE NEWINDICES=NO
END
\SORT
END
\ copy from workfile to disk
\LIST 6
READ TYPE=COPY
END
\ unit weights
\LIST 4
END
\WEIGHT
END
\EDIT
UEQUIV FIRST UNTIL LAST
END
\LIST 28
MINIMA RATIO =  3.0
END
\SFLS
SCALE
END
\ assume there are no H atoms
\LIST 12
FULL FIRST(U[ISO]) UNTIL LAST
END
\SFLS
REFINE
REFINE
CALC
END
\LIST 28
\ remove all restrictions to get Fcs.
END
\DIFABS UPDATE FC
END
\ Complete anisotropic refinement, produce publication tables etc
\LIST 12
FULL X'S U'S
END
\LIST 28
MINIMA RATIO=3
END
\SFLS
REFINE
.....
\CIF
END
\ reprocess data so that it can be merged for the final
\ difference map
\DIFABS UPDATE TRANSFER
END
\SYST
END
\MERGE
END
\FOURIER
\ etc etc etc ---
```

7.49 Internal workings

SOme understanding of the internal data management in CRYSTALS may help the user to sort out unexplained failures.

Refinement parameter map - LIST 22

This list contains the refinement directives in internal format and it can only be generated by the computer. After the refinement directives have been read in, they are stored on the disc in binary format ready for processing. Before the structure factor least squares routines can use the information in LIST 12, it is necessary to convert them to a LIST 22.

If the conversion fails, or the input of LIST 5 or LIST 12 is in error, LIST 22 will be marked as an error list, and any job that attempts to reference LIST 22 will terminate in error.

For complex LIST 12s, i.e. those containing EQUIVALENCE, LINK, RIDE, GROUP, WEIGHT or COMBINE, the user is strongly advised to issue \LIST 22 and then \PRINT 22, and look at the LIST 22 generated. The ouput, which is set out like a LIST 5, shows the relationship between the physical and the least squares parameters.

The least squares matrix - LIST 11

The matrix that is produced by the structure factor least squares process is stored on the disc as a LIST 11. This list may be massive, so it is wise to purge the disk regularly with large structures. To recover the maximum space, delete the LIST 11 before purging.

```
 \DISK
 DELETE 11
 END
 \PURGE
 END
```

Printing the contents of LIST 11

LIST 11 is printed by :
\PRINT 11
\PRINT 11 A
Prints those correlation coefficients greater than 0.3.
\PRINT 11 B
Prints the correlation matrix.
\PRINT 11 C
Prints the inverse matrix.

Least squares shift list - LIST 24

When the normal matrix produced by the least squares process has been inverted, a set of shifts is calculated, suitably scaled if necessary, to apply to the atomic parameters. These shifts are output to the disc as a LIST 24, and then applied by the routines that compute the new parameters. List 24 can only be generated in the machine.

Restraints in internal format - LIST 26

This list contains the restraints in internal format. Before the structure factor least squares routines can use the information in LIST 16 and 17, it is necessary to convert it to an internal format held in LIST 26.

If this operation fails, or the input of LIST 12 or LIST 16 goes wrong, LIST 26 will be marked as an error list, and any job that attempts to reference LIST 26 will terminate in error.

Chapter 8
Fourier Routines

8.1 Scope of the Fourier section of the user guide

In this section of the user guide, the lists and commands relating to the Fourier routines are described.

```
Input of the Fourier section limits           -  \LIST 14
Fourier calculations                          -  \FOURIER
Processing of the peaks list                  -   LIST 10
Elimination of duplicated entries in LISTS 5 and 10  -  \PEAKS
Slant fourier calculations                    -  \SLANT
```

8.2 Input of the Fourier section limits - LIST 14

```
\LIST 14
X-AXIS MINIMUM= STEP= MAXIMUM= DIVISION=
Y-AXIS MINIMUM= STEP= MAXIMUM= DIVISION=
Z-AXIS MINIMUM= STEP= MAXIMUM= DIVISION=
X-PAT MINIMUM= STEP= MAXIMUM= DIVISION=
Y-PAT MINIMUM= STEP= MAXIMUM= DIVISION=
Z-PAT MINIMUM= STEP= MAXIMUM= DIVISION=
ORIENTATION DOWN= ACROSS= THROUGH=
SCALEFACTOR VALUE=


\LIST 14
X-AXIS 0.0 0.0 0.5 0.0
Y-AXIS 0.0 0.0 0.9 0.0
Z-AXIS -2 2 32 60
ORIENTATION Z X Y
SCALE VALUE = 10
END
```

The Fourier routines will calculate a map with section edges parallel to any two of the cell axes (a, b or c). The starting and stopping points must be given for each direction (in crystal fractions). The user should choose the asymmetric unit to have one range as small as possible, and the other two approximately equal. Orientate the computation so that the sections are perpendicular to the short range direction. If the command \SPACEGROUP has been used to input the symmetry information, a LIST 14 will have been generated. This will be a valid choice, but may not be optimal.

\LIST 14

X-AXIS MINIMUM= STEP= MAXIMUM= DIVISION=
This directive specifies how the x-axis is to be divided.

MINIMUM=
This parameter gives the initial value along the x-direction. If it is omitted, a default value of 0.0 is assumed for MINIMUM.

STEP=
This parameter, which has a default value of 0.3, gives the step along the x-direction.

MAXIMUM=
This parameter, which has a default value of 1.0, gives the final value along the x-direction.

DIVISION=
If DIVISION is greater than zero, it defines the number of divisions into which the x-axis is to be divided. In this case, the three remaining parameters are expressed in terms of DIVISION and give the first point (MIN), the increment between successive points (STEP) and the final point to be calculated (MAX). If the divisions of the unit cell along the x-axis are given in this way, the user must ensure that sufficient map is calculated for the map scan, by adding one extra point beyond the asymmetric unit at both ends along the x-axis. If this is not done, peaks at the edge of the asymmetric unit may be missed by the peak search.

If DIVISION is equal to zero, which is its default value, the Fourier routines will calculate the number of divisions required along the x-axis. In this case, STEP is the interval between successive points along the axis in angstrom. If this parameter is less than 0.05, a default value of 0.3 angstrom is used. MINIMUM And MAXIMUM define the first and last points to be calculated and are given in fractional coordinates. When the values of MIN and MAX are converted into unit cell divisions, an extra point is added at each end to ensure that the peak search functions correctly.

Y-AXIS MINIMUM= STEP= MAXIMUM= DIVISION=
Similar to X-AXIS above.

Z-AXIS MINIMUM= STEP= MAXIMUM= DIVISION=
   Similar to X-AXIS above.

X-PAT MINIMUM= STEP= MAXIMUM= DIVISION=
   This directive is similar to the X-AXIS directive, but refers to the Patterson asymmetric unit.

Y-PAT MINIMUM= STEP MAXIMUM= DIVISION=
   Similar to X-PAT above.

Z-PAT MINIMUM= STEP= MAXIMUM= DIVISION=
   Similar to X-PAT above.

ORIENTATION DOWN= ACROSS= THROUGH=
   Controls the orientation parameters for the map calculation and printing.
   DOWN=

   X - Default value
   Y
   Z

   The default value X indicates that the x coordinate goes down the printed page.
   ACROSS=
   As DOWN above, but with the default value Y indicating that the y coordinate goes across the page.
   THROUGH=
   As DOWN above, but with the default value Z indicating that the z coordinate changes from section to section.

SCALEFACTOR VALUE=
   VALUE=
   This parameter specifies the value by which the electron density, on the scale of /Fc/, is multiplied before it is printed. If this parameter is omitted, a default value of 10 is assumed.

8.3 Printing the contents of LIST 14

   The contents of LIST 14 can be listed to the line printer by issuing the command :

\PRINT 14
   There is no command available for punching LIST 14.

8.4 Fourier calculations - \FOURIER

```
\FOURIER INPUT=
MAP TYPE= NE= PRINT= SCAN= SCALE= ORIGIN= NMAP= MONITOR=
REFLECTIONS WEIGHT= REJECT= F000= CALC=
LAYOUT NLINE= NCHARACTER= MARGIN= NSPACE= MIN-RHO= MAX-RHO=
PEAKS HEIGHT= NPEAK= REJECT=
TAPES INPUT= OUTPUT=
END
```


```
\FOURIER
MAP TYPE=DIFF
PEAK HEIGHT = 3
END
```

   Before a Fourier is computed, a LIST 14 must have been created or input. The routine will compute a map in any space group, the relevant symmetry being found in LIST 2 (space group information, see section #RLIST02#).
   In the ouput listing, new peaks are labelled, with the following meanings

```
GOOD PEAK - The peak centre was determined by Least-Squares.
POOR PEAK - The peak centre was determined by interpolation.
DUBIUOS PEAK - The peak centre is only a local maximum.
MALFORMED PEAK - The peak centre is extrapolated to be out side
          of the asymmetric unit - usually due to very poor phasing.
```


\FOURIER INPUT=
   INPUT
      Indicates which reflection list to use.

```
6        Default
7        Alternative reflection list
```

MAP TYPE= NE= PRINT= SCAN= SCALE= ORIGIN= NMAP= MONITOR=

TYPE=

```
F-OBS      -  Default value
F-CALC
DIFFERENCE
2F0-FC
OPTIMAL
FO-PATTERSON
FC-PATTERSON
EXTERNAL
```

The map type 'OPTIMAL' implements a suggestion of Peter Main. It is a form of weighted Fo map, with coefficients w*Fo if the reflection is in a centro-symmtric class, otherwise (2*w*Fo)-Fc, where w is the Simm weight. NOTE this is not the same as w(2*Fo-Fc), a Sim weighted 2Fo-Fc map. It has the property that known and unknown atom peak heights are approximately the same, and should be usefull for Fourier refinement.

NE=
This parameter indicates which solution should be used to compute the externally phased map, and has a default value of 1. NE is only used in conjunction with TYPE = EXTERNAL.

PRINT=
Controls the printing pf the map.

```
NO   -  Default value
YES
```

SCAN=
Controls automatic scanningof the map for peaks.

```
NO
YES  -  Default value
```

SCALE=
Controls the scaling of the electron density in the map.

```
NO
AUTOMATIC  -  Default value
YES
```

If SCALE is YES, the program computes a scale factor rather than take one from LIST 14 (Fourier control - section #RLIST14#). The scale factor is computed by summing the modulus of all the contributors to the map, and dividing this total into ORIGIN (see the next parameter). For a Patterson, therefore, the origin is scaled to be ORIGIN, while for other maps a scale factor is computed which guarantees that every number is less than ORIGIN.
   If SCALE is NO, the scale factor is taken from LIST 14 for all types of Fourier maps. If SCALE is AUTOMATIC, there is automatic scaling for an external or Patterson map, while other maps take their scale factors from LIST 14.

ORIGIN=
The default value for this parameter is 999, and is used when the program calculates a scale factor (see SCALE above).

NMAP
Controls negation of the density values, with default NO. Use YES, in which case the density values are negated, when looking for minima. This feature permits location of hydrogen in Neutron maps, and the location of minima (which become maxima) generally. Set the Peak Height positive even when searching for minina, since at the time of the search the minima are inverted. The out put density values have the correct sign. Use \COLLECT 10 5 rather than \PEAKS on negated maps, since PEAKS cannot handle minima.

MONITOR=

```
LOW
MEDIUM  -  Default value
HIGH
```

If MONITOR is MEDIUM the, the peak coordinates are printed as they are found. If HIGH, density at known sites is also printed.

REFLECTIONS WEIGHT= REJECT= F000= CALC=

WEIGHT=

> *SIM*
> *NO   - Default value*
> *LIST-6*

If WEIGHT is NO , its default value, then the map is not weighted.

If WEIGHT is set equal to SIM , then SIM weights are computed. This option requires both LIST 29 (atomic properties, section #RLIST29# and LIST 5. The occupation factors in LIST 5 are used to determine how many atoms of each type are present, and LIST 29 indicates how many should be present. See the notes under 'TYPE', above.

If WEIGHT is LIST-6 , then the map is weighted with the weight stored in LIST 6 (section #RLIST06#).

REJECT=

> *NONE*
> *SMALL   - Default value*
> *QUARTER*
> *HALF*

If REJECT is NONE, all the reflections in LIST 6 which are allowed by LIST 28 are included. In this case, no check is made on the /Fc/ value. For an /Fo/, /Fc/ and difference Fourier, the program expects that there should be an /Fc/ value if the phase is to be defined. Accordingly, reflections where /Fc/ < 0.001 are normally rejected for such Fouriers, and this is the default option of SMALL.

Some users like to omit reflections if Fc is smaller then a fraction of Fo. The options QUARTER and HALF are available.

F000=
The default value for this parameter is zero, and specifies the value of F(000) to be used.

CALC

> *NO   - Default value*
> *YES*

Value YES causes structure factors (i.e. Fc and phase) to be calculated immediately before the map is computed. This option can only be activated if some previous task with the current DSC file has computed phases via a \SFLS command (section #RSFLS#) and left a LIST 33 on the disk (List 33 is the stored representation of the SFLS command, so that the program can rememeber how the last refinement was carried out, see section #RSFLS#)

LAYOUT NLINE= NCHARACTER= MARGIN= NSPACE= MIN-RHO= MAX-RHO=
This directive specifies how the map should be printed, if the value of the PRINT parameter on the MAP directive is YES.
NLINE=
This parameter sets the number of lines per row of map, and has a default value of 2.
NCHARACTER=
This parameter controls the number of characters for each grid point, and has a default value of 4.
MARGIN=
This parameter, whose default value is 4, defines the number of characters per division number down each side of the map.
NSPACE=
This parameter has a default value of 2, and defines the number of spaces between the division number and the grid number down each side of the map. The minimum value for NSPACE is 2.
MIN-RHO=
This parameter has a default value of -1000000, and points less than MIN-RHO are left blank when the map is printed.
MAX-RHO=
This parameter has a default value of 1000000, and points greater than MAX-RHO are left blank when the map is printed.

PEAKS HEIGHT= NPEAK= REJECT=
Controls the search for peaks when the map is searched, i.e. if the value of the SCAN parameter on the MAP directive is YES.
HEIGHT=
This parameter sets the search of the map for all peaks with an electron density greater than HEIGHT. If this parameter is omitted, a default value of 50 is assumed for an external or Patterson map. For all other maps, the map is scanned for peaks greater than 1.5*SCALE, where SCALE is the map scale factor, either taken from LIST 14 (Fourier control - section #RLIST14#) or computed using SCALE = YES above.

NPEAK=

This parameter, whose default value is 0, determines the number of peaks to be retained after they have been ranked by peak height. If NPEAK is zero or negative, the number of peaks saved is computed from

*NPEAK = (Cell volume) / (18 * Space Group multiplicity)*

18 is an average atomic volume.

REJECT=

This parameter, with a default value of 0.01, specifies that peaks within a distance of REJECT angstrom of a peak already ranked on peak height, will be rejected from the list.

TAPES INPUT= OUTPUT=

This directive is used if a map is to be read off magnetic tape, or a computed map is to be written to a magnetic tape. Remember that CRYSTALS will use scratch files unless given named files. To assign a named output file, issue

\OPEN MT1 filename

The tape is unformatted.

```
Record 1: 'INFO  DOWN ACROSS SECTION'
Record 2: 'TRAN'        9 elements of a transformation matrix
Record 3: 'CELL'        Cell parameters, angles in radians
Record 4: 'L14 '        List 14 information
Record 5: 'SIZE'        number of points down, across, and number of sections
Record 6:  number of values,  values for a section
       Record 6 is repeated for every section.
Record n:  number of atoms, number of items per atom
Record n+1: Items for an atom, repeated for all atoms
```

Record 4 contains 6 integers, (No of points down and across the page, number of sections, and the index of these directions, 1 = x). Subsequent records contain a whole section line by line, prefixed by the total number of points in the section.

INPUT=

NO   - Default value
YES

If INPUT is YES, a map will be read in from the 'input magnetic tape', and the resulting map will be the minimum of each point of the calculated and input maps. The input map sections must be on device 'MT2'
    *** THIS FACILITY IS NOT CURRENTLY IMPLEMENTED ***

OUTPUT=

NO   - Default value
YES

If OUTPUT is YES, the map produced is written to the 'output magnetic tape'. You may need to OPEN a permanent file on device 'MT1'.

8.5 Calculation of superposition minimum functions

(Issue 7 - implementation incomplete, 1984)
(Issue 9 - implementation still incomplete, 1993 - no one seems to want it anyway!- use SHELXS if you need to).
(Issue 10 - still no change, 1996)

The Fourier routine provides a way of calculating superposition minimum functions. For each map that is produced, it is possible to specify that another map should be read in from magnetic tape at the same time (the TAPES directive). Each point of the resulting map is taken as the minimum of the newly computed map and that read off the magnetic tape. This output map may be written to a second magnetic tape, also by use of the TAPES directive.

When the input map and the calculated map are superposed, the first point calculated and the first point read off the tape are compared, the second point calculated and the second point input are compared, and so on. This implies that the first point on each map must represent the same point in real space for the output map, and that each map must contain the same number of points. The origin of each map that is to be calculated is altered by changing LIST 14 (Fourier limits - section #RLIST14#). For example, if a 2x, 2y, 2z vector has been identified at 0.36, 0.14 and 0.28, and the 2x, 1/2-2y, 0 vector resulting from a two-fold axis has been found at 0.36, 0.36, 0, then the two LIST 14's for the superposition function might appear as :

```
\LIST 14
X-AXIS 14 4 122 400
Y-AXIS 5 2 59 100
Z-AXIS 12 2 66 100
ORIENT X Y Z
SCALE 10
END


     and


\LIST 14
X-AXIS 14 4 122 400
Y-AXIS 16 2 70 100
Z-AXIS -2 2 52 100
ORIENT X Y Z
SCALE 10
END
```

   For the first map, the origin of real space is at 0.18, 0.07 and 0.14 in vector space. This point is moved so that it is one grid point in along each axial direction, to allow for the map scan. For the second peak, the origin in real space is at 0.18, 0.18 and 0.0. The second LIST 14 places this point one grid point in along each of the axial directions so that the real space origin of the two maps coincides. To convert the coordinates that result from the second map scan to real space coordinates, it is necessary to subtract 0.18 from x and 0.18 from y, since the coordinates are printed in Patterson space for all the maps calculated.

8.6 Processing of the peaks list - LIST 10

\LIST 10
   LIST 10 cannot be input bythe user. When the map scan has been completed, the resulting peaks are output to the disc as a LIST 10. Except for an external or Patterson map, the atoms already in LIST 5 are placed at the beginning of the LIST 10.
      A LIST 10 is usually converted to a LIST 5 by one of the following commands :

```
\EDIT 10 5                    \PEAKS 10 5
\COLLECT 10 5                 \REGROUP  10 5
```

   \PEAKS is the normal choice, since duplicate peaks related by symmetry, or peaks corresponding to known atoms can be eliminated. It is described below; EDIT, COLLECT and REGROUP are in the section on Atomic and Structural Parameters.

8.7 Printing the contents of LIST 10
   The contents of LIST 10 can be listed with:

\PRINT 10
   There is no command available for punching LIST 10 out to a file.

8.8 Elimination of duplicated entries in LISTS 5 and 10 - \PEAKS

```
\PEAKS INPUTLIST= OUTPUTLIST=
SELECT REJECT= KEEP= MONI= SEQ= TYPE= REGROUP= MOVE= SYMM= TRANS=
REFINE DISTANCE= MULTIPLIER=
END


\PEAKS
SELECT REJECT=0.0001
REFINE DISTANCE=.5
END
```

   This routine eliminates atoms or peaks which duplicate other entries in an atomic parameter list. When using this routine, a set of distances is calculated about each atom or peak in turn. Atoms or peaks further down the list than the current pivot are then eliminated if they have a contact distance less than a user specified maximum (the REJECT parameter). Thus, when peaks have been added to a LIST 5, the peaks corresponding to the atoms can be eliminated.

\PEAKS INPUTLIST= OUTPUTLIST=
   INPUTLIST and OUTPUTLIST specify where the atoms are to be taken from, and where they will be put.
   INPUTLIST=

```
5
10  -  Default value
```

   OUTPUTLIST=

```
5    -  Default value
10
```

SELECT REJECT= KEEP= MONI= SEQ= TYPE= REGROUP= MOVE= SYMM= TRANS=
REJECT=
REJECT is the distance above which connected atoms or peaks are assumed to be distinct. If a contact is found which is less than REJECT the second atom or peak of the pair in the list is eliminated, and defaults to 0.5.
KEEP=
This parameter indicates how many entries are to be kept in the output list. The default value of 1000000 is the maximum possible.
MONITOR=

```
LOW
HIGH  -  Default value
```

If MONITOR is given as LOW only the atoms or peaks that are deleted because of the REJECT limit are listed. If MONITOR is HIGH, all the atoms deleted because of both KEEP and REJECT are listed.
SEQUENCE=

```
NO  -  Default value
YES
```

If SEQUENCE is YES, then the program will give sequential serial numbers to the atoms and peaks in the final output list .
TYPE=

```
PEAK   -  Default value
ALL
AVERAGE
```

If TYPE is PEAK, then the program will only delete PEAKS which are within REJECT of an existing atom. It TYPE is ALL, atoms are also deleted.

If TYPE is AVERAGE, coincident atoms or peaks are averaged. The radius for coincidence is taken from the DISTANCE keyword on the REFINE directive. The default radius is .5 Angstrom.
REGROUP=
This parameter has two allowed values :

```
NO  -  Default value
YES
```

If REGROUP is YES, then the program will reorganise LIST 5 so that bonded atoms and peaks are adjacent.
MOVE=
The value of this parameter is the maximum separation for 'bonded' atoms. The default is 2.0 A.
SYMMETRY=
This parameter controls the use of symmetry information in the calculation of contacts, and can take three values.

```
SPACEGROUP   -  Default value. The full spacegroup symmetry is used in
                    all computations
PATTERSON.    A centre of symmetry in introduced, and the translational
              parts of the symmetry operators are dropped.
NONE.         Only the identity operator is used.
```

TRANSLATION=
This parameter controls the application of cell translations in the calculation of contacts, and can take the values YES or NO

REFINE DISTANCE= MULTIPLIER=
Controls action of Fourier refinement.
DISTANCE=
This parameter has a default value of zero, and is the distance below which atoms and peaks are considered to be coincident. The coordinates of an existing atom are replaced by those of a coincident peak. Refinement takes precedence over deletion of peaks.
MULTIPLIER=
This parameter has a default value to give automatic refinement. It is set to 1 for a centric space group and is set to 2 for a non-centric space group. It can be set to 0.0 to preserve original

        coordinates but be given new peak heights.

```
    X(new) = x(atom) + mult(x(peak) - x(atom)).
```

```
\ reject atoms or peaks with contact distances less than 0.7
\ keep 30 entries in the output list
\ list the atoms and peaks rejected because of both 'KEEP'
\ and 'REJECT'
\
\PEAKS 10 5
SELECT REJECT=0.7,KEEP=30,MONITOR=HIGH
END
```

8.9 Slant fourier calculations - \SLANT

```
\SLANT INPUT=
MAP TYPE= MIN-RHO= SCALE= WEIGHT=
SAVED MATRIX=
CENTROID XO= YO= ZO=
MATRIX R(11)= R(12)= R(13)= R(21)=  .  .  . R(33)=
DOWN MINIMUM= NUMBER= STEP=
ACROSS MINIMUM= NUMBER= STEP=
SECTION MINIMUM= NUMBER= STEP=
END
```

   A Slant Fourier is one that is calculated through any general plane of the unit cell. For such a Fourier, the normal Beevers-Lipson expansion of the summation cannot be used, so that it will take many orders of magnitude longer than a conventional one. The algorithm adopted here is as follows :

```
X    A general vector expressed in fractions of the
     unit cell edges (i.e. x/a, y/b and z/c)
XO   The centroid of the required general fourier section,
     also expressed in crystal fractions.
XP   The coordinates of the point 'X' when expressed
     in the coordinate system used to define the
     plane of the general section.
'X' and 'XP' are related by the expression :  XP = R.(X-XO)
R    'R' is the matrix that describes the transformation
     of a set of coordinates in the crystal system to
     a set of coordinates in the required plane.
therefore :  X = S.XP + XO

     'S' is the inverse matrix of 'R'.

The required expression in the fourier is :

    H'.X = H'.S.XP + H'XO

H    H is a vector containing the Miller indices of
     a reflection and H' is the transpose of H.
This may be re-expressed as :

    H'.X = H'.S.DXP + H'.(S.XPS + XO)

DXP  'DXP' represents the increment in going from the
     first point on the section to be calculated.
XPS  'XPS' is the coordinate of the first point on the
     section to be calculated.
     obviously :  XP = XPS + DXP.
```

   When the Fourier is calculated, the term *H'.(S.XPS + XO)* is constant for each section to be calculated. The term *H'.S* , which may be regarded as the transformed indices, is also constant for each reflection, so that a two dimensional recurrence relation may be used to change *DXP* and thus *Cos(2*PI*H.X - ALPHA)'* over the required section for each reflection. (*ALPHA* is the phase angle for the current reflection).
   The input for the slant Fourier thus must include the rotation matrix *R,* the centroid *XO,* and the steps and divisions in the required plane.

\SLANT INPUT=
   This is the command which initiates the slant fourier routines.
      INPUT
         Indicates which reflection list to use.

```
    6       Default
    7       Alternative reflection list
```

MAP TYPE= MIN-RHO= SCALE= WEIGHT=
  TYPE=

    F-OBS
    F-CALC
    DIFFERENCE
    FO-PATTERSON
    FC-PATTERSON

    There is no default value for this parameter
  MIN-RHO=
    This parameter has a default value of zero, and is the value below which all numbers on the map are
    replaced by MIN-RHO.
  SCALE=
    The terms used in the Fourier are put on the same scale as Fc, and then before the map is printed the
    numbers are multiplied by SCALE . (i.e. SCALE is the map scale factor). The default is 10.
  WEIGHT=

    NO   - Default value
    YES

      If WEIGHT = YES, the observed and calculated structure factors are multiplied by the weights in
    LIST 6 (usually SQRT(w)). The user should be aware that this might have a major effect on the scale
    if the map density, and that SCALE may need adjusting.

SAVED MATRIX=
  This directive, which excludes CENTRIOD and MATRIX, uses the matrix and centroid stored in LIST
  20 by a previous GEOMETRY, MOLAX or ANISO command (see section #RLIST20#).
  MATRIX=

    MOLAX
    TLS
    AXES

CENTROID XO= YO= ZO=
  This specifies the slant Fourier map centroid, in crystal fractions, and excludes SAVED.
  XO=
  YO=
  ZO=
    The defaults value for XO,YO,ZO, the coordinates of the centroid, are 0.0.

MATRIX R(11)= R(12)= R(13)= R(21)= . . . R(33)=
  This gives the elements of the rotation matrix R, and excludes SAVED. The trnsformation generally
  used is from crystal fractions to orthogonal Angstroms.
  R(11)= R(12)= R(13=) R(21)= . . . R(33)=
    There are no default values for any of these parameters.

DOWN MINIMUM= NUMBER= STEP=
  This directive defines the printing of the map down the page.
  MINIMUM=
    There is no default value for this parameter, the first point, in Angstrom, down the page of the plane
    to be calculated.
  NUMBER=
    There is no default value for this parameter, the number of points of the plane to be printed down the
    page
  STEP=
    There is no default value for this parameter, the interval in Angstrom between successive points
    down the page.

ACROSS MINIMUM= NUMBER= STEP=
  This directive defines the printing of the map across the page. The parameters have similar meanings to
  those for 'DOWN'.

SECTION MINIMUM= NUMBER= STEP=
  This directive defines the printing of the map sections. The parameters have similar meanings to those
  for 'DOWN'.
    The units of MINIMUM and STEP are based on the coordinate system used to describe the plane,
  with the new 'x' axis going down the page and 'y' across. In general the most convenient axial system
  for the plane is one expressed in Angstrom, so that the initial points and the steps are all expressed in

Angstrom. (The least squares best plane program prints out the centroid in crystal fractions and the rotation matrix from crystal fractions to best plane coordinates in Angstrom, which are the numbers required, and may be saved for use in SLANT by the directive 'SAVE').

```
\ the map will be a difference map
\ we wish to compute the section 0.3 anstrom above the plane
\ numbers less than zero will be printed as zero
\ the molecule lies at a centre of symmetry
\ so that the centroid in crystal fractions is 0, 0, 0
\ the plane coordinates are in angstrom
\ for printing the plane both across and down the page,
\ we will start 4 angstrom from the centroid,
\ and go 4 angstrom the other side of the centroid,
\ making a grid 8 angstrom by 8 angstrom
\
MAP DIFFERENCE 0.3 0
CENTROID 0 0 0
MATRIX 3.4076 10.0498 6.1794
CONT    5.0606  8.287 -9.5483
CONT   -6.9181 11.0121 1.546
DOWN -4 33 0.25
ACROSS -4 33 0.25
END
```

Chapter 9
Analysis Of Results

9.1 Scope of this section of the user guide

```
Analysis of residuals                        ANALYSE
Distance and angles calculations             DISTANCES
Void search                                  VOIDS
Global Geometry (planes,lines & libration)   GEOMETRY
Torsion angles                               TORSION
Publication listing of the atomic parameters PARAMETERS
Publication listing of the reflections       REFLECTIONS
Summary of data lists                        SUMMARY
CIF files                                    CIF
Graphics                                     CAMERON
```

9.2 Analysis of residuals - \ANALYSE

This analyses the residual, Fo-Fc, for systematic trends, which might either indiacate an incomplete model, or an unsatisfactory weighting scheme. It is described in the chapter Structure Factors and Least Squares.

9.3 Distance angles calculations - \DISTANCES

```
\DISTANCES INPUTLIST=
OUTPUT MONITOR=  LIST= PUNCH= HESD=
SELECT ALLDIST= COORD= SORTED= TYPE= RANGE= SYMM= TRANS=
LIMITS DMINIMUM= DMAXIMUM= AMINIMUM= AMAXIMUM=
E.S.D.S COMPUTE= CELL=
INCLUDE atoms
EXCLUDE atoms
ONLY atoms
PIVOT atoms
BONDED atoms
END


\DIST
E.S.D YES
END
```

The distance angles routine is completely general with respect to crystal and lattice symmetry. For distances, the user may either use elemental radii specified in LIST 29 (see section #RLIST29# for input details), or specify minimum and maximum limits, and the program then calculates all possible contacts within these limits. All symmetry operations and unit cell translations are automatically generated. For the angles, LIST 29 or a separate set of distance limits may be used. At a given atom, angles are then calculated between all the atoms which bond to the central atom within the given limits.

The distance-angles routines can calculate the estimated standard deviations of the distances and angles that they produce. These e.s.d.'s are based upon the matrix stored in LIST 11 (see section #RLIST11#), and as many variance and covariance terms as are present are used. (For a full matrix, therefore, the full variance-covariance matrix is used). For this reason, the calculation of e.s.d.'s takes at least ten times as long as a simple distance angles calculation.

When a set of e.s.d.'s are calculated, the variance-covariance matrix for the cell parameters (LIST 31, section #RLIST31#) may also be used.

\DISTANCES INPUTLIST=
INPUTLIST=

```
5   - Default value
10
```

The default is to use the normal atom coordinate list.

OUTPUT MONITOR= LIST= PUNCH= HESD=
MONITOR=
This controls the monitoring information.

```
OFF         - no output
DISTANCES   - only monitors distances. (Default)
ANGLES      - only monitors angles.
ALL         - monitors distances and angles.
```

LIST=
This controls the format of the listing.

```
OFF
LOW   -  Default
HIGH
```

If LIST is LOW , the default, then the listing is in a compressed format, without symmetry information. If LIST is OFF, no output is sent to the listing file unless PUNCH is PUBLISH, when a copy of the publication listing appears in the listing file.

PUNCH=
This controls the output sent to the 'punch' file.
PUBLISH - Produce a listing suitable for publication.
HTML - Produce an HTML format listing
CIF - Produce a listing in CIF format.
H-CIF - Produce a listing of the H-bonds in CIF format.
SCRIPT - Lists bonds in a easily machine readable format.
RESTRAIN - Produce a proforma LIST 16 (restraints - #RLIST16#). Use the RA LIMIT, TYPE INCLUDE and EXCLUDE parameters to restrict the restraints produced
DELU - Proforma LIST 16 for delta U restraints
SIMU - Proforma LIST 16 for U-similarity restraints
NONBONDED - Proforma LIST 16 with anti-bumping restraints
H-RESTRAIN - Produces a list of H-C,N and O distance and angle restraints in the PUNCH file, and a list of the referenced H atoms in the SCRIPTQUEUE file.
H-CIF - Puts hydrogen bond donor and acceptors into the cif file.

If hydrogen atom restraints are being generated, the following target values are used:

```
No H    No #H  U mult    dist
C-H
>4             1.5       .96 disorder
1       1      1.2       .93 C C-H (acetylene)
1       2      1.2       .93 C-C(H)-C
1       3      1.2       .98 (C)3-C-H
2       1      1.2       .93 C=C-H(2)
2       2      1.2       .97 (C)2-C-(H)2
3       1      1.5       .96 C-C-(H)3
N-H
>4             1.5       .89 NH4 or disorder
1       1      1.2       .86 N-N/H
1       2      1.2       .86 (C)2-N-H
1       3      1.2       .89 (C)3-N-H
2       1      1.2       .86 C-N-(H)2
2       2      1.2       .89 (C)2-N-(H)2
3       1      1.2       .89 C-H-(H)3
O-H
1       1      1.5       .82 O-H


Dist      esd = 0.02
Vib       esd = 0.002
Angle     esd = 2.0
```

HESD=
This controls the output of ESDs to the CIF file.

```
ALL       - (Default) Output all bond length and angle standard
            uncertainties (if requested) to the CIF (if requested),
            including those of bonds to fixed atoms (i.e. to atoms on
            special positions, or to atoms that are not refined).
NONFIXED - Exclude standard uncertainties of bond distances and angles
           to Hydrogen atoms that have not been refined. (as
           required by Acta's notes for authors).
```

SELECT ALLDIST= COORD= SORTED= TYPE= RANGE= SYMMETRY= TRANS= ALLDISTANCES=

```
NO   -  Default value
YES
```

If ALLDISTANCES is NO, the distances calculated about each atom will only be those to atoms

that occur after the central atom in LIST 5. (i.e. each distance is only printed once).

If ALLDISTANCES is YES , then the distances from each atom to all the other atoms are calculated for all the atoms. (In this case, each distance will appear twice in the list).

COORDINATES=

```
NO    -  Default value
YES
```

If COORDINATES is YES, the transformed coordinates of each atom in a distance calculation are printed. If COORDINATES is NO, the transformed coordinates are not printed.

SORTED=

```
NO    -  Default value
YES
```

If SORTED is NO, the distances from the central atom are in the order in which the other atoms occur in LIST 5. If SORTED is YES , the distances are printed in order of increasing magnitude.

TYPE=

This parameter indicates the type of distances which will be calculated.

```
ALL    -  Default value
INTRA
INTER
```

If TYPE is ALL, then all distances are printed; if TYPE is INTRA then only intramolecular distances are printed, and if TYPE is INTER then the intermolecular distances are printed (Note that the whole asymmetric unit is regarded as a 'molecule'.

RANGE=

This parameter defines how the range is to be selected. Except when RANGE = LIMITS (when the lowest acceptable distance is user-specified) contacts of zero angstrom are suppressed.

```
COVALENT       Use 'covalent' radii from LIST 29.
VANDERWAALS.   Use 'VanderWaals' radii from LIST 29, but angles are
               suppressed.
IONIC.         Use 'ionic' radii from LIST 29.
LIMITS.        Use specified or default ranges set by the LIMIT directive.
```

SYMMETRY=

This parameter controls the use of symmetry information in the calculation of contacts, and can take three values.

```
SPACEGROUP   -  Default value. The full spacegroup symmetry is used in
                             all computations
PATTERSON.      A centre of symmetry in introduced, and the translational
                parts of the symmetry operators are dropped.
NONE.           Only the identity operator is used.
```

TRANSLATION=

This parameter controls the application of cell translations in the calculation of contacts, and can take the values YES or NO

LIMITS DMINIMUM= DMAXIMUM= AMINIMUM= AMAXIMUM=

This directive specifies the limits for the distance angles calculations, and may only be given if RANGE = LIMITS has been specified on a preceding SELECT directive.

DMINIMUM

This defines the distance below which distances are not calculated or printed. The default is zero.

DMAXIMUM

This parameter defines the maximum distance above which distances are not calculated or printed. Use \COMMANDS DISTANCES to find the default value for DMAXIMUM. All the distances that are to be calculated and printed must lie between DMINIMUM and DMAXIMUM.

AMINIMUM

For a given central atom, other atoms which make contacts that are less than AMINIMUM will not be considered when the angles at the central atom are computed. The default is zero.

AMAXIMUM

For a given central atom, other atoms which make contacts that are greater than AMAXIMUM will not be considered when angles at the central atom are computed. The default value for AMAXIMUM is set in the COMMAND file. AMAXIMUM And AMINIMUM define a shell about each pivot atom outside of which angles are not computed.

E.S.D.S COMPUTE= CELL=
  This directive determines whether estimated standard deviations of the distances and angles are calculated.
  COMPUTE

NO   - Default value
YES

  If this parameter is NO, standard deviations are not computed. Note that if e.s.d.'s are to be calculated, i.e. COMPUTE is set equal to YES , then a suitable least squares matrix (LIST 11, see section #RLIST11#) must be available.
  CELL=

NO   - Default value
YES

  If this parameter is NO, the variance-covariance matrix for the cell parameters is not included when the e.s.d.'s are calculated.

INCLUDE atoms
  This directive determines which atoms are included as pivot atoms in the calculation. The arguments may be either a type of atom , or an atom specification of the 'type(serial)' or 'type(serial) UNTIL type(serial)' kind described elsewhere in the manual. Only INCLUDEd atoms are used as pivots, but distances and angles are computed to all other atoms in the current LIST 5 within the ranges specified on the SELECT directive.

ONLY atoms
  Similar to INCLUDE, except that specified atoms may be pivot or bonded. The arguments may be either a type of atom , or an atom specification of the 'type(serial)' or 'type(serial) UNTIL type(serial)' kind described elsewhere in the manual. Distances and angles are computed only to specified atoms in the current LIST 5 within the ranges specified on the SELECT directive.

PIVOT atoms
  Similar to INCLUDE, except that atoms excluded with an EXCLUDE directive can still be used to bond to. The arguments may be either a type of atom , or an atom specification of the 'type(serial)' or 'type(serial) UNTIL type(serial)' kind described elsewhere in the manual. Distances and angles are computed only to specified atoms in the current LIST 5 within the ranges specified on the SELECT directive.

BONDED atoms
  Similar to INCLUDE, except that non-included atoms can still be used as pivots. The arguments may be either a type of atom , or an atom specification of the 'type(serial)' or 'type(serial) UNTIL type(serial)' kind described elsewhere in the manual. Distances and angles are computed only to specified atoms in the current LIST 5 within the ranges specified on the SELECT directive.

EXCLUDE atoms
  This directive determines which atoms are excluded as pivots in the calculation. The arguments may be either a type of atom , or an atom specification of the 'type(serial)' or 'type(serial) UNTIL type(serial)' kind described elsewhere in the manual. If EXCLUDE directives alone are used, all atoms except those EXCLUDEd either explicitly or by type, are used as pivot atoms in the calculation. However, if both INCLUDE and EXCLUDE are used, the only atoms used in the calculation will be those INCLUDEd and not EXCLUDEd.

9.4 Distance-angles symmetry operations

  Accompanying each atom in a distance or angle calculation with LIST equal to HIGH are the symmetry operators that are necessary to bring the atom into the correct position in the cell to make a contact with the central atom. These symmetry operations are divided into six parts, which are indicated by five flags. These are explained in the section on Atomic and Structural Parameters.

```
\
\ distances from 0 to 2.5
\ angles from 0 to 2.0
\ the e.s.d.'s of the distances and angles are calculated
\ distances from each atom to all other atoms are printed
\ transformed coordinates are printed
\ the distances are sorted in order of increasing magnitude
\
\DISTANCES
SELECT ALL=YES,COORD=YES,SORT=YES,RANGE=LIMITS
LIMITS DMAX=2.5, AMAX=2.0
E.S.D. YES
END



\DIST
EXCLUDE ALL
ONLY C(1) C(3) C(4)
END
```

9.5 Void Location - \VOIDS

```
\VOIDS INPUTLIST=
DISTANCE
TOLERANCE
CONTACTS
RESOLUTION
END

\VOIDS
DISTANCE 2.2
END
```

This utility searches for the asymmetric unit for points which lie outside the known atoms. The 'radii' of the known atoms is independent of type, and in an input value. A pseudo atom in inserted at every point on a search grid outside the known atoms. The pseudo atoms are given a 'TYPE' dependant upon the number of neighbouring pseudo atoms. Atoms of type R are at the core of large voids, type L are intermediate, and M at the surface.

\VOIDS INPUTLIST=
INPUTLIST=

```
5   - Default value
10
```

The default is to use the normal atom coordinate list.

DISTANCE value
This sets the radii of the known atoms, default 2.5A.

RESOLUTION value
This sets the sampling interval for the search grid, default 0.8 A.

CONTACT value1 value2
This sets the number of pseudo-atom contacts required for the core and intermediate pseudo atoms. The defaults are 27 (R type atoms), 15 (L type atoms). All other atoms are of type M.
   \COLLECT and \REGROUP can be used to re-group the pseudo-atoms, and the augmented structure can be viewed in CAMERON.

9.6 TLS analysis, best planes and lines - \GEOMETRY

```
\GEOMETRY INPUTLIST=
 ATOMS  W(1)  SPECIFICATION(1)  W(2)  SPECIFICATION(2) .
 PLANE
 LINE
 AXES
 TLS
 EXECUTE
 EVALUATE  ATOM SPECIFICATIONS . . . .
 REPLACE ATOM SPECIFICATIONS . . .
 PUNCH
 SAVE
 DIHEDRAL  NP(1)  AND  NP(2)
 QUIT
 CENTRE   X=, Y=, Z=
 REJECT   NV=
 LIMITS    VALUE=   RATIO=
 MODL L(11), L(22) L(33) L(23) L(13) L(12)
 MODT T(11), T(22) T(33) T(23) T(13) T(12)
 ZEROS
 DISTANCES  DL=   AL=
 ANGLES  AL=
 PLOT
 END


\GEOMETRY
ATOMS FIRST UNTIL LAST
PLANE
EXECUTE
SAVE
ATOMS FIRST UNTIL LAST
TLS
EXECUTE
ANGLE 1 AND 2
EXECUTE
DISTANCES
END
```

GEOMETRY is used for computing the following global derived parameters:

```
    Centroid (centre of gravity)
    Inertial Tensor
    Best Plane
    Best Line
    Shape Indices
    Principal Axes of adps
    Librational and Translational Thermal Tensors
    Dihedral Angles
```

It replaces the old \MOLAX, \AXES and \ANISO commands

PLANE & LINE are used for computing the principal axes of inertia through groups of atoms using the routines described in Computing Methods in Crystallography, edited by J. S. Rollett, Pergamon Press, 1965, p67-68.

The best plane for a series of N atoms whose positions have varying reliability, such that they can be assigned weights, w(1), w(2), . . . w(n), is defined as that for which the sum of the squares of the distances (in angstroms) of the atoms from the plane, multiplied by the weights, w(i), of the atomic positions, is a minimum. Note that the normal to the 'worst plane' is the 'best line', and if masses are used for weights, then the calculation gives the principal inertial axes.

The atomic positions are taken from LIST 5, possibly modified by symmetry information, to compute inertial axes & deviations of atoms from the planes or lines.

Each time a line or plane is computed, the direction cosines of the relevent axis are stored as AXIS number 'n'. The dihedral angles between these axes can be computed. Three geometry indices are also computed. The geometry is best described by the index closest to unity. (Mingos,D.P.M & Rohl,A.L., J.Chwm.Soc. Dalton Trans (1991) pp 3419 - 3425)

TLS. This routine calculates the overall rigid-body motion tensors T, L, S (Shoemaker and Trueblood, Acta Cryst. B24, 63, 1968) by a least-squares fit to the individual anisotropic temperature factor components, together with librational corrections to bond lengths and angles.

Shoemaker and Trueblood's conventions and reductions are followed throughout; in particular, the trace of S, which is indeterminant, is set to zero. The program therefore determines 20 overall tensor components - the upper triangles of T and L together with the whole of S apart from S(33). (See also: Johnson in Crystallographic Computing, ed R.Ahmed, Munksgaard, 1970, pp 207-219)

Even when the trace-of-S singularity has been removed, however, the nature of the rigid body problem is such that ill-conditioned and singular normal matrices are much more common than in structure refinement and the program therefore proceeds via the eigenvalues and eigenvectors of the normal matrix. In most cases the largest and smallest eigenvalues are output for inspection, but if the ratio of these quantities is less than the LIMITing RATIO, a full eigenvalue/vector listing is produced. Further, if any eigenvalue is itself less

than the LIMITing VALUE, the corresponding parameter combination is set to zero, thus removing the near-singularity. These actions can be modified by the use of the LIMIT and REJECT directives described below. If the TLS calcuation cannot be stabilised by means of these filters, the user can modify either T, L or S directly before applying the REPLACE or PUNCH commands. Though here is some danger in this, especially if the supposed rigid group is infact flexible, it may be preferable to using a model yielding negative vibrational or librational amplitudes.

The direction cosines of the principal axis of L are stored for use in inter-axis angle comutations.

Immediate execution of a directive can be forced by issuing an EXECUTE directive.

\GEOMETRY INPUTLIST=
INPUTLIST=

5   - Default value
10

ATOMS W(1) SPECIFICATION(1) W(2) SPECIFICATION(2) .
This specifies atoms to be used in the calculation of the best plane. W(1) Is the weight assigned to the atoms contained in the first atom specification, W(2) is the weight assigned to the second group of atoms, and so on. If W(1) is omitted, a default value of 1 is used, but any other W(I) term applies to all the atoms following it, until another W is found or the end of the directive is encountered. At least one ATOM directive must precede each PLANE, LINE, TLS, AXES or PLOT directive. An ATOM directive will over-rule an immediately preceding ATOM directive. If an input line is not long enough for the full atom list, use CONTINUE.

PLANE
This directive, (or LINE, TLS, AXES, PLOT) must follow immediately after an ATOM directive and causes the calculation of a least squares best plane.

LINE
This directive, (or PLANE, TLS, AXES, PLOT) must follow immediately after an ATOM directive and causes the calculation of a least squares best line.

AXES
This directive (like \AXES) computes the principal axis lengths and directions for the atoms specified on a preceding ATOM directive.

TLS
This causes the TLS calculation to be initiated. It MUST have been preceded by an ATOM directive.

EXECUTE
This forces the execution of preceding directives.

EVALUATE ATOM SPECIFICATIONS . . . .
If present, this directive must appear after a PLANE, LINE, TLS or PLOT directive, and causes the co-ordinates or adps of the atoms specified to be calculated and printed with respect to the current axial system.

REPLACE ATOM SPECIFICATIONS . . .
if present, this directive must appear after a PLANE, LINE, TLS or PLOT directive, and causes the co-ordinates or adps of the atoms specified to be modified so that they conform to the most recent geometry calculation. The LIST 5 in core is immediately updated, so that the new coordinates will be used for any subsequent computation. A LIST 5 is only written to the disc on a satisfactory exit from GEOMETRY.

PUNCH
This directive causes the orthogonal coordinates of the atoms of any plane or line computed or EVALUATED in the current task to be output to the 'punch' file. For a TLS calculation, it causes a restraint list to be output to TLSREST.DAT

SAVE
This directive is optional.
If it follows a PLANE, LINE or TLS directive, it causes the latest rotation matrix and CENTRE to be stored in the appropriate position in LIST 20.
If it follows an AXES directive, the direction cosines and centre if the ellipse FOR THE LAST ATOM are stored in LIST 20.
A LIST 20 is only written to the disc on a satisfactory exit from ANISO.

DIHEDRAL NP(1) AND NP(2)
   If present, thus directive must follow at least two PLANE, LINE or TLS computations. It causes the program to calculate the angle between the axes with serial numbers NP(1) and NP(2) . The AND must be present.

QUIT
   This directive abandons the calculation without modifying the disc LISTs.

CENTRE   X=, Y=, Z=
   This directive specifies the centre of libration, in crystal fractions, to be used in the original derivation of the overall motion tensors. The program derives and uses a unique origin at a later stage in the calculations. This directive is optional, the default centre being (0,0,0). If a centre of (0,0,0) is given or set by default, the program computes and uses the mean position of the given atoms, INCLUDING any which are isotropic, even though these are not used to compute TLS. The stored CENTRE is updated during TLS, and a second TLS computation may be performed using this new value as CENTRE. This may help stabilise certain forms of ill-conditioning.

REJECT NV=
   Overrides normal action and sets the parameter combination corresponding to eigenvector number nv to zero. Eigenvectors are numbered in ascending order of their eigenvalues, so that nv is in the range 1 to 20 inclusive and will usually have been obtained from a full eigenvalue/vector listing produced in a previous run.

LIMITS VALUE= RATIO=
   If an eigenvalue is less than VALUE or its size is less than RATIO * (the next bigger), it is eliminated from the analysis. VALUE is currently .000001 and RATIO .01 .

MODL L(11), L(22) L(33) L(23) L(13) L(12)
   This directive enables the user to change the values of the L tensor before EVALUATING or REPLACING the Uij. The L tensor changed is that with respect to the inertial axes and the input centre of libration. It does not depend upon S. All six values must be given.

MODT T(11), T(22) T(33) T(23) T(13) T(12)
   This directive enables the user to change the values of the T tensor before EVALUATING or REPLACING the Uij. The T tensor changed is that with respect to the inertial axes and the input centre of libration, NOT the final tensor, since this involves an interaction with S and L. All six values must be given.

ZEROS
   #PThis directive enables the user to set the S tensor to zero before EVALUATING or REPLACING the Uij. It decouples T from L.

DISTANCES DL= AL=
   This directive calculates all interatomic distances less than DL angstroms with librational corrections. If this directive is omitted, no distances are calculated; if DL is absent, a default value of 1.8 is inserted. If AL is present, angles between atoms separated by less than AL angstroms are computed.

ANGLES AL=
   This directive calculates angles between all bonds less than AL angstroms. If this directive is omitted, no angles are calculated; if AL is absent, a default value of 1.8 is inserted.
       *********************** WARNING *************************

       The directive DISTANCE may only be followed by ATOM, EXECUTE, or END.

PLOT
   This obsolete directive produces a join-the-dots diagram on the monitor or printer. It (or PLANE, LINE, TLS, AXES) must follow immediately after an ATOM directive and causes the calculation of inertial axes. Details of the computation are suppressed on the Monitor, but a line drawing projected onto the best plane is produced. MOLAX Can thus be used as a means of displaying some or all of the atoms in a structure.

```
\
\ these instructions compute a plane
\ involving n(1),n(2),n(3) and c(1), and
\ prints the co-ordinates of all the atoms with
\ respect to this plane.  The positions of the
```

```
\ nitrogen atoms have double weight
\
\GEOMETRY
ATOMS 2 N(1) UNTIL N(3)  1 C(1) C(2)
PLANE
EVALUATE ALL
\
\ these instructions calculate another plane,
\ printing only the co-ordinates of c(5) with respect to
\ the second plane.  The angle between the two planes
\ is then calculated
\
ATOMS C(1) S(1) N(1)
PLANE
EVALUATE C(5)
DIHEDRAL 1 AND 2
END
```

9.7 Torsion angles - \TORSION

```
\TORSION INPUTLIST=
ATOMS  SPECIFICATIONS
PUBLICATION  PRINT=
END
```

```
\TORSION
ATOM C(1) C(2) C(3) C(4)
END
```

   The routines described in this section calculate torsion angles which are defined as follows. The torsion angle about the bond j-k is the angle the bond k-l is rotated from the ijk plane. It is positive when, on looking from ij to kl, the rotation is clockwise.
   The program uses atomic positions taken from LIST 5. These can be modified by the space group symmetry operators stored in LIST 2 (space group information, see section #RLIST02#)

\TORSION INPUTLIST=
   INPUTLIST

```
5   - Default value
10
```

   ATOMS SPECIFICATIONS
      This directive specifies atoms that are to be used in the calculation of the torsion angle. More than one ATOMS directive can be given. Each directive must define at least four atoms, the torsion angle being computed with respect to the first three atoms and each of the subsequent ones.

   PUBLICATION PRINT=
      The parameter PRINT controls the publication listing, which is sent to the file open on the CRYSTALS PUNCH unit.

```
NO   - DEFAULT.  There is no publication listing
YES    There is a publication listing sent to the PUNCH file
CIF    The listing is in CIF format
```

```
Example.
\ the torsion angle about C(3)-C(4) is calculated
\ two torsion angles about C(4)-C(5) are calculated
\
\TORSION
ATOMS N(2) C(3) C(4) C(5)
ATOMS C(3) C(4) C(5) C(6) O(1)
END
```

9.8 Publication listing of the atomic parameters - \PARAMETERS

```
\PARAMETERS
LAYOUT INSET= ATOM= DOUBLE= CHOOSE= FLOAT= NCHAR= NLINE= LISTAXES= ESD=
COORDINATES NCHAR= NDECIMAL= SELECT= TYPE= DISPLAY= PRINT= PUNCH=
U'S NCHAR= NDEC= SELECT= TYPE= DISPLAY= PRINT= PUNCH=
END
```

```
\PARAMETERS
LAYOUT ATOM-NAME=6,DOUBLE=YES
END
```

   This routine sends the atomic parameters to the PUNCH file in a suitable format for publication or binding into a thesis. As well as the current atomic parameters in LIST 5, the estimated standard deviations derived from the least squares normal matrix are also printed. **THIS ROUTINE WILL NOT WORK** if LIST 5 is modified *in any way* since the last round of refinement. If any changes, including renaming, are made, a further round of refinement must be done. If you wish to preserve parameter values, and create a valid matrix without changing the parameter values, compute a refinement cycle but set all the shifts to zero.

```
\SFLS
REFINE
SHIFT GENERAL = 0.0
END
```

The output is in two halves, the first containing the positional coordinates and any isotropic temperature factors, and the second containing all the anisotropic temperature parameters.
   For the first part, a page is split into 6 separate fields. The first field is blank, and is an offset so that the information is centred on the page. The remaining fields contain the atom type and serial number, the three positional parameters, and a temperature factor. This will be the value of U(iso) with its e.s.d for isotropic atoms, otherwise U(equiv), without an e.s.d, for anisotropic atoms. U(equiv) **is not** simply related to the diagonal elements of U(aniso), and may be computed as either the arithmetic or geometric mean of the principal axes of the ellipsoid. See \SET UEQUIV in the chapter on IMMEDIATE commands. The width of each type of field may be altered by the user, using respectively the INSET , ATOM-NAME , and NCHARACTER parameters. The default length of a page of this type of output is that required for A4 paper.
   The second part contains the anisotropic temperature factors, and each page is split into eight fields. As for the atomic coordinates, the first field is blank and represents an offset. The second field contains the atom type and serial number, and the remaining six fields contain the components of the anisotropic temperature factors. The width of each type of field may be adjusted by the user, using respectively the INSET , ATOM-NAME and NCHARACTER parameters. If a different value for INSET or ATOM-NAME is required in the first and second parts of the output, the job must be run twice. Depending upon the width across the page, the second part of the output occupies one sheet of A4 paper either across the page or down the page.
   For both types of output, the user can select double spacing down the page with the DOUBLE parameter. Similarly for each of the numeric fields, the user can choose the number of decimal places to be printed (the NDECIMAL parameter), and whether the numbers are printed as integers or in floating point with a decimal point. (The FLOATING parameter). The e.s.d.'s are printed to the same accuracy as the atomic parameters, so that if the chosen field is too small and an e.s.d. appears to be zero, it will be omitted in exactly the same way as for a parameter that has not been refined. A parameter printed with 4 decimal places might thus appear as :

```
0.0123(4)
OR
   123(4)
```

Depending upon the format. In either case, the numbers are right justified in their field.
   As an alternative to the user selecting the number of decimal places that should be printed, it is possible to get the program to choose the number of decimal places required for each parameter automatically. (The CHOOSE parameter). If the parameters are to be printed in floating point, the number of decimal places is chosen so that the e.s.d. Can be represented as a one digit number in the last decimal place. For numbers that are to be printed as integers, the field used is never less than that given by the NDECIMAL parameter. If the required field is larger than that defined by these s, a decimal point is inserted and the required number of extra digits is output. For example, if the number of decimal places required is four, but the e.s.d. is too small, it would appear as :

```
0.12345(6)
OR
 1234.5(6)
```

Depending upon whether floating point or integer output was required. For either type, if the parameter has not been refined, the number of decimal places is that given by the NDECIMAL instruction.
   Since this routine prints the e.s.d.'s, it is vital that the least squares matrix (LIST 11, see section #RLIST11#) belongs to the current LIST 5 (the model parameters). If LIST 5 has been modified in any way since the last Least Squares, this routine will abort.
   When anisotropic atoms are present in LIST 5, U[EQUIV] is calculated according to the current setting of \SET UEQUIV.

\PARAMETERS

This command initiates the routines for printing of the atomic parameters in a suitable format for publication.

LAYOUT= INSET= ATOM= DOUBLE= CHOOSE= FLOAT= NCHAR= NLINE= LISTAXES= ESD=
This directive defines how the atomic parameters, both positional and thermal, are to be laid out on the page.

INSET
This parameter sets the number of blank spaces on each line before the atom type and serial number. If this parameter is omitted a default value of 1 is assumed.

ATOM-NAME
This parameter sets the width of the field that contains the atom type and serial number. The characters are left justified in the field, and the format is as follows :

TYPE(SERIAL)

The serial number is printed as an integer, and the unoccupied spaces are filled with blanks. If this parameter is omitted, a default value of 6 is assumed.

DOUBLE
This parameter has two possible values :

NO - DEFAULT VALUE
YES

If DOUBLE is YES each line of parameters is double spaced. The default option if this parameter is omitted is single spacing, with no interleaving blank lines.

CHOOSE
This parameter has two possible values :

NO
YES - DEFAULT VALUE

If CHOOSE is YES the program chooses the number of decimal places that need to be printed for each parameter, depending upon its e.s.d.. The format of the output depends upon whether a decimal point is being used, as explained above.

FLOATING
This parameter has two possible values :

YES - DEFAULT VALUE
NO

If FLOATING is NO , the parameters are printed as integers, with an accuracy given either by the NDECIMAL parameters to the directives COORDINATES and "U'S, or by the 'CHOOSE' parameter. parameter.

NCHARACTER
This parameter indicates the total number of printing positions on the output device. If this parameter is omitted, a default value of 118 is assumed.

NLINE
This parameter indicates the total number of lines on the on the output media. Set a very lartge value (1000) to get continuous output.

LISTAXES
This parameter can have two values

YES
NO - DEFAULT VALUE

If the value is YES the principal axes of the temperature factors are printed.

ESDS
This parameter can take 3 values

NO
YES - DEFAULT VALUE
EXCLRH

EXCLRH inhibits printing the e.s.ds for riding hydrogen atoms

COORDINATES NCHAR= NDECIMAL= SELECT= TYPE= DISPLAY= PRINT= PUNCH=

This directive defines how the positional coordinates are to be set out on the page.

NCHARACTER

This parameter sets the width of the field that contains the positional coordinates. The characters are right justified in the field, and if this parameter is omitted, a default value of 14 is assumed.

NDECIMAL

This parameter sets the number of decimal places to be printed for the positional parameters. It may be partially or completely overriden by the CHOOSE parameter, depending upon the format of the output. If this parameter is omitted, a default value of 4 is assumed.

SELECT

This parameter selects the kinds of data to be printed, and can have five values.

ALL - Default. All atoms are printed.
NONE - No atoms are printed.
ONLY - Only atoms with TYPEs given on a TYPE directive are printed.
EXCLUDE - Atoms with TYPEs given on a TYPE directive are not printed.
SEPARATE- Atoms with TYPEs given on a TYPE directive are printed separately

TYPE

Used in conjunction with SELECT to determine which atom types to INCLUDE,EXCLUDE or SEPARATE. TYPE is ignored if SELECT is ALL or NONE. Its default value is 'H'.

DISPLAY

This parameter has two possible values

NO   No output is displayed on the terminal.
YES Output is displayed on the terminal.

PRINT

This parameter has two possible values

NO   No output is sent to the listing file
YES   Output is sent to the listing file

PUNCH

This parameter has three possible values

NO   No output is sent to the punch file
YES   Output is sent to the punch file
CIF   Output is in CIF format

U'S NCHAR= NDEC= SELECT= TYPE= DISPLAY= PRINT= PUNCH=

This directive defines how the thermal parameter are to be set out on the page.

NCHARACTER

This parameter sets the width of the field that contains the thermal parameters. The characters are right justified in the field, and if this parameter is omitted, a default value of 11 is assumed.

NDECIMAL

This parameter sets the number of decimal places to be printed for the thermal parameters. If this parameter is omitted, a default value of 4 is assumed.

SELECT

This parameter selects the kinds of data to be printed, and can have five values.

ALL - Default. All atoms are printed.
NONE - No atoms are printed.
ONLY - Only atoms with TYPEs given on a TYPE directive are printed.
EXCLUDE - Atoms with TYPEs given on a TYPE directive are not printed.
SEPARATE- Atoms with TYPEs given on a TYPE directive are printed separately

TYPE

Used in conjunction with SELECT to determine which atom types to INCLUDE,EXCLUDE or SEPARATE. TYPE is ignored if SELECT is ALL or NONE. Its default value is 'H'.

MONITOR

This parameter has two possible values

OFF   No output is displayed on the terminal.
HIGH Output is displayed on the terminal.

PRINT

This parameter has two possible values

```
NO        No output is sent to the listing file
YES       Output is sent to the listing file
```

PUNCH

This parameter has three possible values

```
NO        No output is sent to the punch file
YES       Output is sent to the punch file
CIF       Output is in CIF format
```

9.9 Publication listing of reflection data - \REFLECTIONS

```
\REFLECTIONS INPUT=
LAYOUT NCOLUMNS= NLINES= INSET= NSPACE= SCALE= NCHARACTER=
OUTPUT PRINT= PUNCH= LIST28=
END
```

This routine prints the reflection data in LIST 6 (section #RLIST06#) in a suitable format for publication or binding into a thesis. The information printed falls into one or more columns, each of which contains h, k, l, /Fo/, /Fc/, and the phase angle in degrees. Each column is 18 characters wide. Although the user has no control over the contents of each column, it is possible to vary the number of blank spaces at the start of each line, the number of columns across the page, the number of spaces between successive columns, and the number of lines per page. (The INSET , NCOLUMNS , NSPACE and NLINES parameters, respectively). /Fo/ and /Fc/ are both put on the same scale of /Fc/, using the scale factor in LIST 5, and both these two numbers may be modified by a scaling constant before they are printed. (The SCALE parameter). However, all the values of both /Fo/ and /Fc/ must be less than 10000 when they are printed.

LIST 28 is used for checking whether or not to print a reflection. Remember that if LIST 28 was used to reject some reflections when structure factors were last calculated, removing these restrictions before printing LIST 6 will mean that some reflections will have incorrect values of Fc and phase.

\REFLECTIONS INPUT=

INPUT

Indicates which reflection list to use.

```
6         Default
7         Alternative reflection list
```

LAYOUT NCOLUMNS= NLINES= INSET= NSPACE= SCALE= NCHARACTER=

This directive defines how the reflection data is to be printed.

NCOLUMNS=

This parameter indicates the number of columns of reflection data to be printed across the page. If this parameter is omitted, a default value of 3 is assumed.

NLINES=

This parameter indicates how many lines should be on each page of output. If this parameter is omitted a default value of 52 is assumed.

INSET=

This parameter indicates how many blank spaces should be inset at the beginning of each line. If this parameter is omitted a default value of 30 is assumed.

NSPACE=

This parameter indicates the number of spaces separating successive columns across the page. If this parameter is omitted a default value of 3 is assumed, which means that each column occupies 21 characters across the page.

SCALE=

This parameter indicates the scaling constant by which /Fo/ and /Fc/ should be multiplied before they are printed, after they have been put on the same scale (the scale of /Fc/). If this parameter is omitted, a default value of 10 is assumed.

NCHARACTER=

This parameter indicates the total number of printing positions on the output device. If this parameter is omitted, a default value of 120 is assumed.

OUTPUT PRINT PUNCH LIST28

This directive defines where the reflection data is to be printed.

PRINT=

This has two allowed values :-

```
NO        No output is sent to the listing file
YES       Output is sent to the listing file
```

PUNCH=

This has two allowed values :-

```
NO        No output is sent to the punch file
YES       Output is sent to the punch file
```

9.10 Summary of data lists - \SUMMARY

```
\SUMMARY OF= TYPE= LEVEL=
```

```
\SUMMARY LIST 5 HIGH
END
\SUMMARY EVERYTHING
END
```

This command produces a summary on the terminal of the contents of a list. Use \PRINT if you need full details.

\SUMMARY OF= TYPE= LEVEL=
OF
LIST Default, also requires TYPE to be set EVERYTHING

The value EVERYTHING generates a summary of all LISTS.
TYPE
This parameter requires a list type number if the previous parameter was 'LIST"
LEVEL
Some lists may be listed in more or less detail.
OFF LOW MEDIUM Default HIGH

\SUMMARY 6
Unlike all the other SUMMARIES, which only generate readable output, SUMMARY 6 computes the conventional R on the basis of the current Fo, Fc and LIST 28 (section #RLIST28), and updates the value stored in LIST 6 (section #RLIST06#) and LIST30 (section #RLIST30#). The weighted R is not affected.

9.11 CIF lists - \CIF

Data can be produced in CIF format for direct deposition at CCDC or submission to journals. The required information is taken from several lists, including LIST 30 (see section #RLIST30#). F000, Mu etc are also computed and inserted in LIST 30.

\CIF
There are no qualifiers.
See \PARAMETERS and \REFLECTIONS for the CIF printing of parameters and reflections .

CheckCIF
CheckCIF and other validators are continuously updated to meet the changing needs of the community. It is unlikely that a CRYSTALS .cif will pass all checks first time, and edits may be necessary to accommodate special situations. Some of these have been foreseen, and the .cif contains possible alternative texts as 'comments'. These can be found by searching for the text 'choose'.
References
The SCRIPT directory contains two text files that contain information copied into the cif file. The user may edit them.
Refcif.dat
This file is copied in its entirety to the head of the cif file. If it is edited, care must be taken to follow the rules about text delimiters.
Reftab
This is a loosely formatted file containing the references to be transcribed into the cif.
Every reference is composed of 2 parts - a short text used as a data item in the cif, and the full reference. The two parts must be kept together, be separated from each other by a blank line, and be separated from any other item by a blank line.

```
n  a four-digit number giving the number of references to follow.
   Other text on the line is ignored.
Next items repeated 'n' times:
   m  a three digit number preceded by a 'hash' symbol used as an
      identifier for the reference. The numbers must be unique, not
      necessarily in any order, with the largest one equal to 'n'

   The full reference. References are put in the file in alphabetic
   order.
```

Items 001 to 004 are associated with the keywords 'unknown' for the corresponding items in LIST

30 (see section #RLIST30#), and thus enable the user to insert their own references. Don't forget to move them to their correct alphabetical place.

e.s.d.s

The esds output in CIF files try to follow the 'Rule of 19', as requested by Acta Cryst. Syd Hall, former Editor for Acta C, summarised the rule as follows:

'This method of handling the su (esd) values has been in force with Acta since about 1984 apparently. In my time it came up for discussion about two years ago (1996) and after much to-ing and fro-ing it was readopted as the preferred level of precision for su's.

What it means is as follows....

```
(1) if one adopts esd values to one digit precision (rule of 9) the values

    5.548(1)      1.453(2)  3.921(3)  1.2287(8)  are acceptable.

(2) if one permits two digits precision with a limit of 19 (rule of 19)...

    5.5483(9)     1.4532(16) 3.921(3)  1.2287(8)  are acceptable.

(3) if one permits two digits precision with a limit of 29 (rule of 29)...

    5.5483(9)     1.4532(16) 3.9214(28) 1.2287(8)  are acceptable.
```

The object of this approach is to provide a more consistent distribution of precision across all values. These particular matters are not really my responsibility but we try to conform to recommendation of the nomenclature people. This is one such occasion.'

9.12 Graphics - CAMERON

CAMERON

The graphics module CAMERON is part of the graphical user interface, and can only be started from the GUI. Like CRYSTALS, a sub-set of the possible commands are packaged up into menus, but the advanced full potential is still available from the command line. There is a separate guide for CAMERON
On exit from CAMERON the current image of the structure is padded back to CRYSTALS in the file CAMERON.L5. This contains **all** and **only** the atoms last displayed by CAMERON. Be careful - it could be a packing diagram!

Chapter 10
Twinned Crystals

10.1 Twinning - introduction

The terminology in articles on twinning is complicated and sometimes contradictory, with the same term being used in different contexts by different authors. We shall use the following terms, based upon observations made from the **whole** reciprocal lattice.

It is assumed that sufficient reflections are measured to give a complete coverage of the asymmetric part of the r.l. for at least one (called the major) component of the twinned crystal.

TLQS twins

Some, but possibly not all, of the reflections from the major component contain contributions from other twin components. Overlap is controlled by accidental relationships between cell parameters. If the relationship is very exact, so that all reflections are overlapped, the sample is a **pseudo TLS** twin.

TLS twins

Every reflection from the major component contains a constant fractional contribution from other components. The overlap is controlled by the crystal class rather than accidental relationships between cell parameters.

TLS twins - Class I

Except for the effect of anomalous dispersion, the Laue symmetry of the diffracion pattern is the same as that of an un-twinned crystal.

TLS twins - Class II

The Laue symmetry of the diffracion pattern is **not** the same as that of an un-twinned crystal.

10.2 Twinning Problems

The analysis of twinned structures is complicated by several issues.

1. Identification that the crystal is indeed twinned.

Twinning - Initial clues

These may include may include:

a. Evident interpenetrating reciprocal lattices.

b. Split reflections, with a varying intensity ratio.

c. Systematic absences not conforming to any space group.

d. The ratio of intensties of equivalent reflections from different samples is not constant.

Other clues are:

a. Failure to solve the structure from apparently good data.

b. Irreducible R factor from seemingly good quality data.

c. Inexplicable strong residual peaks in the difference density map.

Twinning - Data collection and processing

a.

There is usually no difficulty in collecting data for TLS twins. For TLQS twins, each observation needs to be tagged to indicate which twin components (elements) contribute to the observation. This may be simply computed from the indices if the different lattices have a more-or-less exact relationship between them, of may need to be assigned more carefully if the twin obliquity causes only partial overlapping of some reflections. For doublet spots, it is important that either the whole doublet is integrated (tag '12'), or the principal component is separated out (tag '1').

b.

There may be serious difficulties in determining the space group. Trial and error may be the only procedure available.

c.

The space group used for data reduction (section #RDATAREDUC#) and merging may not be that of the

major component. A Space group showing the symmetry of the twinned diffraction data should be used initially. The correct space group should be used once data reduction is complete.

Twinning - Structure solution

In general, structure solution is the major difficulty in working with twinned crystals.

a.

For TLQS structures, if a substantial number of reflections are from the major component only, the structure may solve by traditional methods.

b.

For Class I TLS structures, structure solution is usually straight forward, the components of the twin differing only by the effects of anomalous scattering. Such twins (merohedral twins, or twinning by inversion) can be processed without further reference to this part of the manual. All that needs to be refined is the Flack enantiopole parameter. See the main chapter on refinement.

c.

For Class II TLS structures, if the twin ratio is far from 50:50, the structure may solve by traditional methods.

Twinning - Structure Refinement

If the space group, trial structure, twin law and reflection components are known, this is straight forward. The sum of the twin fractions must be 1.0

Twin Data stored by CRYSTALS

For a twinned crystal the following equation holds.

    Fsq(obs) = v1.Fsq(1) + v2.Fsq(2) ....

and similarly for F(calc). The $v(i)$ are the volume fractions of the components contributing to the observation. A Fourier synthesis using /Fobs/ as coefficient is meaningless, since the phase alpha(calc) will belong to only one of the components. The terms needed for Fourier and other calculations are Fcalc(1), alpha(1) Fobs.vol-fract(1), i.e. only that contribution to Fo due to the principal element.

For a twin with two components, each observation may contain a contribution from each component, or from both. The reflections have to be 'tagged' to indicate which components are contributing, the ELEMENT coefficient in LIST 6 (section #RLIST06#)

For a TLS twin, every observation contains a contribution from both components (though if it is a systematic absence for one component, the contribution will be zero). Since the tagging is the same for every reflection, it can be inserted automatically by CRYSTALS

For a TLQS twin, some observations will contain a contribution from the principal component, and some from both components, giving ELEMENT tags of '1' and '12' respectively. If additional observations have been made based on the reciprocal lattice of component 2, and are indexed with respect to lattice 2, they are given the tag '2'. If any of these also contain a contribution from component 1, the tag will be '21'.

Example 1. An orthorhombic space group with a~b, twinned by interchange of 'a' and 'b'. If 'a' is very similar to 'b', every observation 'hkl' will overlap with twin component 'khl', and the ELEMENT tag will be '12', the default. If a systematic absence from element 1 falls on element 2, the reflection should not be eliminated during data reduction, and will have the tag '12', even though the contrinbution from 1 is zero.

Example 2. A monoclinic crystal with 2cCos(beta)/a about 1/3. Twinning by a 2 fold rotation about 'a' gives a twin law

        1   0   0
        0   -1   0
     -1/3   0   -1

Overlap of reflections from both components will only occur when 'h' = 3n, giving the ELEMENT tag '12'. If the lattice is only sampled at r.l. points corresponding to the principal indexing, reflections with 'h' # 3n will have the tag '1'.

Twinning - LISTS affected

    LIST 5 - Parameters: the number of twin elements and their values must be set.
    LIST 6 - Reflections: the observed twinned data must be stored as /FOT/, and th
              twin element tags be set.
    LIST 12 - Constraint matrix: the twin elements must be refined, and possibly con
    LIST 13 - Experimental info: the key CRYSTAL TWIN=YES must be set.
    LIST 16 - Restraints: the twin elements may be restrained.
    LIST 25 - This contains the twin laws themselves.

Twin List 5

The number of twin elements and their values must be given. Currently, the number of elements and their starting values cannot be input in \EDIT (though values can be changed later). Punch LIST 5, edit it, and re-input it, or use the SCRIPT EDLIST5.

```
\LIST 5
READ NATOM=  NELEMENT=
ELEMENT value(1) value(2) ...
ATOM ..........
......
END
```

Twin List 6

For TLQS twins, the element tags (section #RLIST06#) really depend upon exact experimental conditions, and should be computed by the data collection software. If a reflection is entered without a twin element tag (eg a SHELX HKL 4 file), CRYSTALS tries to compute the tag from the twin laws as follows:

```
h        the index with respect to LIST 1 (cell) and LIST 2 (space group)
         (this is the index in LIST 6)
T        The twin law matrix.
n        the nominal index for the twinned reflection.
         n = T.h
d        the difference between an exact lattice point and the
         generated point.
         n-nint(n)
s        The length**2 of the difference vector, in A-2.
```

If 's' is less than the TWINTOLERANCE given on the LIST 6 MATRIX directive, the twinned reflection is regarded as falling upon a primary element reflection, and the element tag is updated to indicate this. This method is only an approximation, but may help to make otherwise useless data useable. LIST 13 (section #RLIST13#) will be automatically updated to indicate that twinned data are being refined.

a)Analysis was started as untwinned, and the user wishes to convert to a twinned refinement
The twin laws must be entered and CRYSTALS instructed to convert the reflection list to a twinned list.

```
\LIST 25
READ NELEMENT=2
MATRIX 1 0 0  0 1 0  0 0 1
MATRIX 0 1 0  1 0 0  0 0 1
END
\LIST 6
READ TYPE=TWIN
MATRIX TWINTOL=.001
END
```

b)Crystal identified as twinned, and data reduction, sorting and merging done outside of CRYSTALS

If the reflection data has been preprocessed so that it is a full, unique, set for the corret space group, then the correct space group should be entered, and the reflections input as FOT directly. This tells CRYSTALS that the data is twinned.

```
\LIST 25
READ NELEMENT=2
MATRIX 1 0 0  0 1 0  0 0 1
MATRIX 0 1 0  1 0 0  0 0 1
END
\OPEN HKLI TWINREF.HKL
\LIST 6
READ    F'S=FSQ  NCOEF = 5  TYPE = FIXED CHECK = NO
INPUT H K L /FOT/ SIGMA(/FO/)
FORMAT (3F4.0,2F8.0)
STORE NCOEF=9
OUTPUT   INDICES /FO/ SIGMA(/FO/) /FOT/ /FC/ SQRTW ELEMENT
CONTINUE RATIO/JCODE CORRECT
MATRIX TWINTOL=.001
END
```

c)Data reduction, sorting and merging to be done in CRYSTALS

During initial data reduction (section #RDATAREDUC#) the crystal must be given as **untwinned** in

LIST 13 (section #RLIST13#), and the 'space group' should be that of the Laue Class of the intensity data, so that the symmetry of the data is preserved. In general, systematic absences should be preserved, unless centring of the cell matches for all twin components. Twin elelemt tags may be provided by an external program, or computed by CRYSTALS.

  If there are special ELEMENT tags, use something like the following:

```
\OPEN HKLI twin.hkl
\LIST 6
READ   F'S=FSQ  NCOEF = 6   TYPE = FIXED CHECK = NO
INPUT H K L /FO/ SIGMA(/FO/) ELEMENTS
FORMAT (3F4.0, 2F8.0, F3.0)
STORE NCOEF=7
OUTPUT INDICES /FO/ SIGMA(/FO/) ELEMENTS RATIO/JCODE CORRECTIONS SERIAL
END
```

After initial processing, LIST 13 (section #RLIST13#) should be changed to twinned, the correct space group entered, and the value of the observed structure factor stored as FOT, the Total or Twinned structure factor. This is done by a special call to the LIST 6 instruction (which also sets the TWIN flag in LIST 13).

```
\LIST 6
READ TYPE=TWIN
MATRIX TWINTOL=.001
END
```

TWIN LIST 13

The keyword TWINNED must be set to YES for structure factor calculations. Because different components of a twin will probably have different extinction corrections, refinement of extinction is deprocated for twins. CRYSTALS prints a warning, then lets you continue at your own risk. The special use on the LIST 6 command (above) will update LIST 13 automatically.

```
\LIST 13
....
CRYSTALS FRIEDEL=NO   TWIN=YES   EXTINCTION=NO
```

Twin List 12

If all the element scale factors are refined simultaneously with the overall scale factor, the calculation will be singular. In general, the sum of the element scale factors is held at unity. For only two twin componenets, this can be done in LIST 12 as a constraint. For more, it can be done in LIST 16 as a restraint. The sum of the elements in input to LIST 5 should be unity.

```
\LIST 12
FULL ........
EQUIVALENCE ELEMENT(1) ELEMENT(2)
WEIGHT -1 ELEMENT(2)
END
```

Twin List 16

The sum of the element scale factors can be restrained to unity in LIST 16. In this case, they must all be freely refined in LIST 12.

```
\LIST12
FULL ........
CONTINUE ELEMENT SCALES
END
\LIST 16
SUM .0001 ELEMENT SCALES
END
```

10.3 SORTING TWINNED STRUCTURE DATA - \REORDER

For a twinned structure, after the data have been merged, it is advisable to re-sort the reflections, placing observations that contain contributions from elements with the same indices adjacent in the new LIST 6.
\REORDER
This directive initiates the re-sorting of reflections for a twinned structure. It is IMPERATIVE that the previous command has put the reflections on the disc. This is automatic if input is via a \LIST 6 command (section #RLIST06#) or you can use the \LIST 6 READ TYPE=TWIN command.
STORE MEDIUM=
This directive determines the output medium of the new LIST 6.
MEDIUM
This parameter selects the output medium of the new LIST 6. The allowed values for this parameter are :

```
M/T
DISC   -  DEFAULT VALUE.
```

The default output medium is usually to disk.

```
/REORDER
END
```

Twins - backward compatability

```
Note that the key /FOT/ can be given in the initial data reduction if the
crystal is also marked as twinned in LIST 13 (section #RLIST13#), and
the observed intensity
input as /FOT/. This is preserved for backwards compatibility.
```

Twins - Worked Example

The data were provided by Simon Parsons, for a TLQS twin, where the bulk of the data is from only one component. For reciprocal lattice layers with h=3n, there is overlap from the second twin component. The 'elelent keys' are thus '12' for reflections with h=3n, otherwise '1'.
Sections of reflection file 'example.hkl'

```
 -6    0    0     2.16    1.08   12
 -6    0   -1    -0.47    0.93   12
 -6    0   -2    24.98    1.63   12
......
 -6   -2    0     1.64    0.95   12
 -6   -2   -1     8.40    1.06   12
 -6   -2   -2     3.33    1.18   12
 -5    5    1    10.61    1.22    1
 -5    5    2     0.75    0.96    1
........
 -4    0    3    -0.45    0.63    1
 -4    0    4     4.73    0.82    1
 -4    0    5    -0.78    0.71    1
 -4    0    6    48.40    1.69    1
 -4    0    7     0.12    0.68    1
 -4    0    8    -0.35    0.83    1
 -3   -7    0     7.68    1.24   12
 -3   -7   -1    13.11    1.45   12
 -3   -7   -2    13.89    1.36   12
.......
```

The data can be processed in the true space group. LIST 6 (reflection) input includes the 'element keys'. After data reduction, the data is stored as 'TWINNED' by the call to LIST 6 which saves the data in the .DSC file.

```
\  Input the cell parameters
\LIST 1
REAL 7.2847 9.74 15.231 90 94.386 90
END


\  Input the space group
\SPACEGROUP
SYMBOL p 21/n
END


\  Input the experimental data
\list 13
crystal  friedel = no twinned=no
cond wave=1.5418
end


\  Input the twin laws, including the identity matrix
\  which corresponds to the first component of the
\  twin, i.e. the one it was indexed on.
\list 25
read nele=2
matrix 1 0 0 0 1 0 0 0 1
matrix 1 0 0 0 -1 0 -.33333 0 -1
end


\  Input scattering factors (list 3) and cell contents
\  (list 29) using the composition command:
\COMPOSITION
CONTENTS c  48  h  44  s  4  o  4  n  4
SCATT   CRSCP:SCATT
PROPER CRSCP:PROPERTIES
END


\  Specify how the SFLS calculations should be done:
\LIST 23
MINIMISE  F-SQ=no
modify  anomalous=yes
```

```
END


\  Input a whole model: scale parameter, twin element scales and
\  the atom parameters.
\list 5
read natom = 5 nelem=2
overal scale=.2
elem .5 .5
atom s    1          1.0000    0.0398    0.9390    0.3740    0.3888
atom n    2          1.0000    0.0617    0.6708    0.1939    0.3428
atom o    3          1.0000    0.0460    0.6967    0.4265    0.5265
atom c    4          1.0000    0.0416    0.9097    0.0426    0.2936
atom c    5          1.0000    0.0317    0.7467    0.2938    0.3989
end


\  Open a file on the device called 'HKLI'
\CLOSE HKLI
\OPEN HKLI example.hkl


\  Read data from that device into LIST 6 in the
\  specified format and leave space for the specified
\  keys.
\list 6
READ   F'S=FSQ  NCOEF = 6  TYPE = FIXED CHECK = NO
INPUT H K L /FO/ SIGMA(/FO/) ELEMENT
FORMAT (3F4.0, 2F8.0,f4.0)
STORE NCOEF=7
OUTPUT INDICES /FO/ SIGMA(/FO/) RATIO/JCODE CORRECTIONS SERIAL ELEMENT
END


\  Remove systematic absences and move hkl indices by symmetry so that
\  they fall into a unique volume of reciprocal space:
\SYST
\  Sort the reflections:
\SORT
\  Merge adjacent reflections with the same indices:
\MERGE
END


\  Store the reflections and at the same time, guess the element
\  key using the twin laws in L25 to predict if overlap is likely.
\List 6
read type=twin
end


\  Compute the scale factor
\SFLS
SCALE
END


\  Set up the matrix of constraint (aka the refinement
\  directives):
\LIST 12
FULL FIRST(X'S, U[ISO]) UNTIL C(15)
equivalence element(1) element(2)
weight -1 element(2)
END


\  Carry out one cycle of least squares refinement:
\SFLS
REF
END
```

Twinning - Mathematical aspects

In a twinned crystal, two or more separate components or ELEMENTS contribute to the diffraction pattern, and the observed intensities may contain contributions from any one of the possible twin component In addition, the amount of each twin component present in a specified unit of volume is not restricted, and in general will vary between different samples of the same material.

The expression for an observed intensity in such a case is given by :

```
It = v1*I1 + v2*I2 + . . + vn*In
```

Where *It* is the total observed intensity to which *N* components contribute, *Ii* is the intensity of component *i* , and *vi* is the amount of component *i* present in a given volume. The *vi* are known as the 'component scale factors', and are conventionally taken to be the amount of the given component present in a unit volume of the crystal, so that :

```
SUM(vi) = 1       over all the components.
```

When a set of reflection data is handled for a twinned crystal, it is thus necessary to know which of the possible components contribute to the current reflection, and to be able to generate the indices of each of the components from a set of indices given in a standard reference system. If the indices of an component in its own reference system are given by the vector Hc and those in the standard system by H , the necessary interconversion is given by :

`Hc = R.H`

R is a rotation matrix that describes the transformation of the indices. (The generation of the various sets of indices can be thought of as a rotation centred on the origin). The indices Hc are of necessity integers, but the components of H may in general take any value.

The interconversion of atomic coordinates between the various reference systems in a twinned crystal can also be expressed in terms of R :

`Hc[T].Xc = H[T].X        for any component.`

Where X is the coordinate vector for any atom in the standard reference system, Xc is the coordinate vector for the same atom in the reference system for one of the components and H[T] indicates H transposed. The above expression may be rewritten as :

`H[T].R[T].Q.X = H[T].X`

Where Q is the matrix that converts the atomic coordinates. Therefore :

`R[T].Q = I`

Where I is the unit matrix. The matrix Q is thus given by:

`Q = R[TI]`

Where R[TI] indicates R transposed and inverted. The coordinates therefore transform as :

`Xc = R[TI].X`

Before any reflections can be processed, the matrices R must be provided. These are given in LIST 25, which must contain one matrix for each possible component. (If the standard system is chosen as that of component 1, for example, the first R matrix will be the unit matrix, which must be given as it is not assumed).

Chapter 11
Matrix Calculations

11.1 The Basic Matrix Calculator \MATRIX

```
\MATRIX
A
B
MM
TM
MT
TT
TRANS
INV
EIG
ACC2A
ACC2B
EXECUTE
END

\MATRIX
A 1 2 3 4 5 6 7 8 9
B 4 5 6 7 8 9 10 11 12
MM
END
```

   CRYSTALS contains a simple calculator for processing 3x3 matrices. Two matrices, A and B, can be input and operated on. The result is output to the screen and left in an accumulator. It can be transfered to A or B for further operations. There is currently no interface to any stored crystallographic information.

\MATRIX

A
   Followed by the nine values for the matrix A by rows

B
   Followed by the nine values for the matrix B by rows

MM
   Accumulator = AxB

TM
   Accumulator = A'xB

MT
   Accumulator = AxB'

TT
   Accumulator = A'xB'

TRANS
   Accumulator = A'

INV
   Accumulator = A-1

EIG
   Accumulator = Eigenvectors of A

ACC2A
   Matrix A = Accumulator

ACC2B
   Matrix B = Accumulator

EXECUTE
   Forces execution of the last directive

Chapter 12

Obsolete Commands

12.1 Obsolete Commands

The following Commands were available in earlier versions of CRYSTALS. They are retained for compatibility reasons, but have been suppressed or superceeded by new commands.

```
Least squares best planes                      MOLAX
Thermal displacement parameter analysis        ANISO
Principal atomic displacement directions       AXES
Structure factors for a group of trial models  TRIAL
```

12.2 Least squares best planes and lines - \MOLAX

```
\MOLAX INPUTLIST=
EXECUTE
PUNCH
ATOMS  W(1)  SPECIFICATION(1)  W(2)  SPECIFICATION(2) .
PLOT
PLANE
LINE
ANGLE  NP(1)  AND  NP(2)
EVALUATE  ATOM SPECIFICATIONS . . . .
REPLACE ATOM SPECIFICATIONS . . .
SAVE
QUIT
END


\MOLAX
ATOM FIRST UNTIL LAST
PLANE
SAVE
END
```

MOLAX is used for computing the principal axes of inertia through groups of atoms using the routines described in Computing Methods in Crystallography, edited by J. S. Rollett, Pergamon Press, 1965, p67-68. It can be used to compute best lines and planes, and produce simple line printer plots of the atoms.

The best plane for a series of N atoms whose positions have varying reliability, such that they can be assigned weights, $w(1)$, $w(2)$, . . . $w(n)$, is defined as that for which the sum of the squares of the distances (in angstroms) of the atoms from the plane, multiplied by the weights, $w(i)$, of the atomic positions, is a minimum. Note that the normal to the 'worst plane' is the 'best line', and if masses are used for weights, then the calculation gives the principal inertial axes.

The atomic positions are taken from LIST 5, possibly modified by symmetry information, to compute inertial axes, deviations of atoms from the planes or lines, and the angles between normals to these planes or axes. Shape indices (Mingos M.P. and Rohl A.L. J Chem Soc Dalton Trans (1991) 3419) are computed.

Each time a line or plane is computed, the direction cosines of the relevent axis are stored as AXIS number 'n'. The angles between these axes can be computed. Three geometry indices are also computed. The geometry is best described by the index closest to unity. (Mingos,D.P.M & Rohl,A.L., J.Chwm.Soc. Dalton Trans (1991) pp 3419 - 3425)

Immediate execution of a directive can be forced by issuing an EXECUTE directive.

\MOLAX INPUTLIST=
   INPUTLIST=

```
5   - Default value
10
```

EXECUTE
This forces the execution of preceding directives.

PUNCH
This directive causes the orthogonal coordinates of the atoms of any plane or line computed in following tasks to be output to the 'punch' file.

ATOMS W(1) SPECIFICATION(1) W(2) SPECIFICATION(2).
This specifies atoms to be used in the calculation of the best plane. W(1) Is the weight assigned to the atoms contained in the first atom specification, W(2) is the weight assigned to the second group of

atoms, and so on. If W(1) is omitted, a default value of 1 is used, but any other W(I) term applies to all the atoms following it, until another W is found or the end of the directive is encountered. At least one ATOM directive must precede each PLANE or PLOT directive. An ATOM directive will over-rule an immediately preceding ATOM directive. If an input line is not long enough for the full atom list, use CONTINUE.

PUNCH

This directive causes the orthogonal coordinates of the atoms of any plane or line computed or EVALUATED in the current task to be output to the 'punch' file.

PLOT

This directive, (or PLANE or LINE) must follow immediately after an ATOM directive and causes the calculation of inertial axes. Details of the computation are suppressed on the Monitor, but a line drawing projected onto the best plane is produced. MOLAX Can thus be used as a means of displaying some or all of the atoms in a structure.

PLANE

This directive, (or LINE or PLOT) must follow immediately after an ATOM directive and causes the calculation of a least squares best plane.

LINE

This directive, (or PLANE or PLOT) must follow immediately after an ATOM directive and causes the calculation of a least squares best line.

ANGLE NP(1) AND NP(2)

If present, thus directive must follow at least two ATOMS/PLANE (ATOMS/LINE, ATOMS/PLOT) directive sequences. It causes the program to calculate the angle between the axes with serial numbers NP(1) and NP(2) . The AND must be present.

EVALUATE ATOM SPECIFICATIONS . . . .

If present, this directive must appear after a PLANE, LINE or PLOT directive, and causes the co-ordinates of the atoms specified to be calculated and printed with respect to the least squares axial system.

REPLACE ATOM SPECIFICATIONS . . .

if present, this directive must appear after a PLANE, LINE or PLOT directive, and causes the co-ordinates of the atoms specified to be modified so that they lie on the previously defined plane. The LIST 5 in core is immediately updated, so that the new coordinates will be used for any subsequent computation. A LIST 5 is only written to the disc on a satisfactory exit from MOLAX.

SAVE

This directive causes the latest plane defining matrix and vector to be stored in LIST 20. A LIST 20 is only written to the disc on a satisfactory exit from MOLAX.

QUIT

This directive abandons the calculation without modifying the disc LISTs.

```
\
\ these instructions define a plane
\ involving n(1),n(2),c(1),c(2) and n(3) and
\ prints the co-ordinates of all the atoms with
\ respect to this plane.  the positions of the
\ nitrogen atoms have double weight
\
\MOLAX
ATOMS 2 N(1) UNTIL N(3)  1 C(1) C(2)
PLANE
EVALUATE ALL
\
\ this set of directives also calculates another plane,
\ printing only the co-ordinates of c(5) with respect to
\ the second plane.  the angle between the two planes
\ is then calculated
\
ATOMS C(1) S(1) N(1)
PLANE
EVALUATE C(5)
ANGLE 1 AND 2
END
```

12.3 Thermal displacement parameter analysis - \ANISO

```
\ANISO  INPUTLIST
EXECUTE
ATOMS    WEIGHT ATOM SPECIFICATIONS
CENTRE   X=, Y=, Z=
REJECT   NV=
LIMITS   VALUE=   RATIO=
TLS
EVALUATE ATOM SPECIFICATIONS
REPLACE ATOM SPECIFICATIONS . . .
SAVE
QUIT
AXES
DISTANCES  DL=   AL=
ANGLES  AL=
END


\ANISO
ATOM C(1) UNTIL C(6)
TLS
SAVE
END
```

   This routine calculates the overall rigid-body motion tensors T, L, S (Shoemaker and Trueblood, Acta Cryst. B24, 63, 1968) by a least-squares fit to the individual anisotropic temperature factor components, together with librational corrections to bond lengths and angles.
   Shoemaker and Trueblood's conventions and reductions are followed throughout; in particular, the trace of S, which is indeterminant, is set to zero. The program therefore determines 20 overall tensor components - the upper triangles of T and L together with the whole of S apart from S(33).
   Even when the trace-of-S singularity has been removed, however, the nature of the rigid body problem is such that ill-conditioned and singular normal matrices are much more common than in structure refinement and the program therefore proceeds via the eigenvalues and eigenvectors of the normal matrix. In most cases the largest and smallest eigenvalues are output for inspection, but if the ratio of these quantities is less than the LIMITing RATIO, a full eigenvalue/vector listing is produced. Further, if any eigenvalue is itself less than the LIMITing VALUE, the corresponding parameter combination is set to zero, thus removing the near-singularity. These actions can be modified by the use of the LIMIT and REJECT directives described below.

\ANISO INPUTLIST
   INPUTLIST

```
5    -  Default value
10
```

EXECUTE
   This causes immediate execution of the previous directive, otherwise directives are executed on input of a new directive (or END).

ATOMS   WEIGHT ATOM SPECIFICATIONS
   This parameter specifies the set of atoms to be used for the following calculation.
      WEIGHT. The default weight of 1.0 is used for all atoms except those following a WEIGHT value. Any decimal number on the ATOM directive is taken as a weight and applied to any following atoms. A subsequent atom directive over rules all previous atom directives. If the full atom specification cannot be got on one directive, use CONTINUE. The atom specifications are in the usual form with symmetry operators and UNTIL sequences permitted. An ATOM directive resets the CENTRE its default value, 0,0,0.

CENTRE   X=, Y=, Z=
   This directive specifies the centre of libration, in crystal fractions, to be used in the original derivation of the overall motion tensors. The program derives and uses a unique origin at a later stage in the calculations. This directive is optional, the default centre being (0,0,0). If a centre of (0,0,0) is given or set by default, the program computes and uses the mean position of the given atoms, INCLUDING any which are isotropic, even though these are not used to compute TLS. The stored CENTRE is updated during TLS, and a second TLS computation may be performed using this new value as CENTRE. This may help stabilise certain forms of ill-conditioning.

REJECT NV=
   Overrides normal action and sets the parameter combination corresponding to eigenvector number nv to zero. Eigenvectors are numbered in ascending order of their eigenvalues, so that nv is in the range 1 to 20 inclusive and will usually have been obtained from a full eigenvalue/vector listing produced in a

previous run.

LIMITS VALUE= RATIO=
> If an eigenvalue is less than VALUE or its size is less than RATIO * (the next bigger), it is eliminated from the analysis. VALUE is currently .000001 and RATIO .01 .

TLS
> This causes the TLS calculation to be initiated. It MUST have been preceded by an ATOM directive.

EVALUATE ATOM SPECIFICATIONS
> This may be used after a successfull TLS calculation to list Ucalcs for the specified atoms. The atom list is not modified.

REPLACE ATOM SPECIFICATIONS . . .
> If present, his directive must appear after a TLS directive, and causes the co-ordinates of the atoms specified to be modified so that they have U's defined by the current T, L, and S matrices. The LIST 5 in core is immediately updated, so that the new coordinates will be used for any subsequent computation if a new ATOM directive is issued. The updated LIST 5 is only written to the disc on a satisfactory exit from ANISO.

SAVE
> This directive is optional. If it follows a TLS directive, it causes the latest L matrix and CENTRE to be stored in LIST 20. If it follows an AXES directive, the direction cosines and centre if the ellipse FOR THE LAST ATOM are stored in LIST 20. A LIST 20 is only written to the disc on a satisfactory exit from ANISO.

QUIT
> This directive abandons the calculation without modifying the disc LISTs.

AXES
> This directive (like \AXES) computes the principal axis lengths and directions for the atoms specified on a preceding ATOM directive.

DISTANCES DL= AL=
> This directive calculates all interatomic distances less than DL angstroms with librational corrections. If this directive is omitted, no distances are calculated; if DL is absent, a default value of 1.8 is inserted. If AL is present, angles between atoms separated by less than AL angstroms are computed.

ANGLES AL=
> This directive calculates angles between all bonds less than AL angstroms. If this directive is omitted, no angles are calculated; if AL is absent, a default value of 1.8 is inserted.
> ********************* WARNING ************************

> The directive DISTANCE may only be followed by ATOM, EXECUTE, or END.

```
\ANISO
ATOMS O(12) UNTIL LAST
AXES
TLS
DISTANCES
END
```

12.4 Principal atomic displacement directions - \AXES

```
\AXES INPUTLIST=
END
```

```
\AXES
 END
```

This routine calculates the magnitudes and directions of the principal axes of the atomic dispacement ellipsoid of an anisotropic atom. Atoms which are isotropic are ignored. Atoms with a negative principal axis generate a warning. The output gives the mean square displacement in angstroms squared along each of the principal axes, together with the direction cosines with respect to the orthogonalized axes and with respect to the real cell axes.

This routine can also be called from \ANISO to get the axes of specified atoms only.

\AXES INPUTLIST=

This command initiates the routine for calculating the principal atomic vibration directions, and requires no other directives.

INPUTLIST=

```
5   - Default value
10
```

The default value is 5.

12.5 Structure factors for a group of trial models - \TRIAL

This procedure is currently unsupported. It is kept in the code because it offers an opportunity for a new programmer to experiment with improved 'COST' functions.

At some stage during a structure determination, the orientation of a group of atoms may be known, but not their position in the unit cell. The routine described in this section provides a rapid method of calculating structure factors for a group of atoms at a series of points that fall on a grid in the unit cell. The algorithm used is similar to that employed in the slant fourier, (see the section of the user guide on 'Fourier routines') and is as follows :

The A part of the structure factor for the reflection with indices given by the vector H may be written as :

```
A(H) = SUM[ G.SUM[ COS2PI(H'.S.X + H'.T) ] ]
```

With a similar expression for the B part. (G Is the required form factor, modified by the temperature factor expression). Conventionally, the inner sum runs over the various symmetry operators that define the space group, and the outer sum runs over the number of atoms in the asymmetric unit. However, if the summation order is changed, it is possible to accumulate sums for all the atoms for each symmetry position :

```
P(H,S) = SUM[ G.COS2PI(H'.S.X + H'.T) ]
```

With a similar expression for Q(H,S) for the B part. It is now possible to use a recurrence relationship for P and Q to give :

```
      P(H,S,2) = P(H,S,1)*2*COS2PI(H'.S.DX) - P(H,S,0)
and
      Q(H,S,2) = Q(H,S,1)*2*COS2PI(H'.S.DX) - Q(H,S,0)
```

P(H,S,0) Is the original value of P for the symmetry position S for the reflection given by H . P(H,S,1) Is the corresponding value of P after a vector DX has been added to each set of coordinates, and P(H,S,2) is the corresponding term after a vector 2*DX has been added. Similar relationships hold for the Q terms. After the initial eight cosine and sine terms have been calculated, it is possible to calculate structure factors very rapidly as the group of atoms is moved about the unit cell, using the relationships given above.

Apart from an array to hold each section through the unit cell, it is necessary to store the eight cosine and sine terms, together with the three step vector cosines, for each reflection for each symmetry position. Because this imposes certain storage limitations, it is necessary to restrict the number of reflections that are used. In practice it is only the large reflections that must agree, and so the user is required to input a minimum Fo value, below which reflections are not used. The function that is displayed for each grid point is given by :

```
SCALE*SUM[ Fo*Fc ]
```

Accordingly, the largest value printed represents the most likely solution. The SCALE term may be calculated by the program to give numbers in a reasonable range, or input by the user. The time for each calculation is proportional to the number of reflections used, the number of symmetry operators in LIST 2, and the number of grid points calculated. (A calculation in a non-centro space group takes twice as long as a calculation in the corresponding centro space group). The atoms to be moved around are taken directly from LIST 5.

\TRIAL
This is the command which initiates the routine to calculate structure factors for a group of trial models.

MAP Fo-MIN SCALE MIN-RHO
This directive determines which reflections will be used in the calculations and how the map will be printed.

Fo-MIN
This parameter is the minimum value of Fo that a reflection must have if it is to be used (this number must be on the scale of Fo). If this parameter is omitted, a value of zero is assumed.

SCALE
If SCALE is equal to zero, its default value, the program will choose a scale factor that places all the numbers on a reasonable scale for printing. If this parameter is greater than zero, the sum of Fo*Fc is multiplied by SCALE before it is printed. (The scale factor computed by the program is dependent upon the origin chosen for the group of atoms, so that successive maps with different origins will be on different scales, unless this parameter is specified for all the maps after the first).

MIN-RHO

This parameter is a cut-off value, such that all numbers less than MIN-RHO are printed as zero. If this parameter is absent, a default value of zero is assumed, which means that all the points are printed.

DISPLACEMENT DELTA-X DELTA-Y DELTA-Z

This directive defines a vector which is added to each set of coordinates in LIST 5 before the structure factor calculation starts. DELTA-X , DELTA-Y And DELTA-Z thus correspond to an initial origin shift for the group in LIST 5.

DELTA-X

The shift along the x-direction.

DELTA-Y

The shift along the y-direction.

DELTA-Z

The shift along the z-direction.

The default values for these parameters are zero, indicating no initial origin shift before the structure factor calculation.

DOWN NUMBER X-COMPONENT Y-COMPONENT Z-COMPONENT

This directive specifies the printing down the page.

NUMBER

The number of points to be printed down the page, for which there is no default value.

X-COMPONENT Y-COMPONENT Z-COMPONENT

There are no default values for these parameters, which specify the fractional coordinate shift vector. The vector moves the group so that :

```
X1 = X0 + X-COMPONENT
Y1 = Y0 + Y-COMPONENT
Z1 = Z0 + Z-COMPONENT
```

Where 1 and 0 define successive points down the page.

ACROSS NUMBER X-COMPONENT Y-COMPONENT Z-COMPONENT

These are the corresponding values across the page.

NUMBER

The number of points to be printed across the page, for which there is no default value.

X-COMPONENT Y-COMPONENT Z-COMPONENT

There are no default values for these parameters, which specify the fractional coordinate shift vector.

THROUGH NUMBER X-COMPONENT Y-COMPONENT Z-COMPONENT

These are the values that define the change from section to section.

NUMBER

The number of sections to be printed, for which there is no default value.

X-COMPONENT Y-COMPONENT Z-COMPONENT

There are no default values for these parameters, which specify the fractional coordinate shift vector. These shift vectors allow any change of position for the group to be plotted out.

```
\TITLE MOVE 2 SULPHURS AROUND
\LIST 5
READ NATOM=2
ATOM S 1 X=0.00 0.15 0.37
ATOM S 2 X=0.13 0.05 0.24
\ call '\trial' with a min. fO of 250
\TRIAL
MAP Fo-MIN=250
\ initial origin shift
DISPLACEMENT 0 0 -0.3
\ plot half of y down the page
DOWN 26 0 0.02 0
\ plot half of x across the page
ACROSS 26 0.02 0 0
\ plot half of z up the page negatively
THROUGH 51 0 0 -0.01
\FINISH
```

Index

\set watch number 7
\sfls 97
\slant input= 113
\sort 43
\spacegroup 20
\spawn   'shell command' 5
\special action= update= tolerance= 82
\summary 12 84
\summary 40 72
\summary 41 73
\summary 6 129
\summary list 16 89
\summary list 17 90
\summary of= type= level= 15,129
\systematic 42
\title ..... a title to be printed 5
\torsion inputlist= 124
\trial 143
\type 'filename' 5
\use source 5
\voids inputlist= 120
\weight input= 94

OBSOLETE COMMANDS

Table of Contents