# The SHELX-97 Manual

## Contents

# 1. General Introduction to SHELX-97

The first version of **SHELX** was written at the end of the 1960's. The gradual emergence of a relatively portable FORTRAN subset enabled it to be distributed (in compressed form including test data as one box of punched cards) in 1976. **SHELX-76** survived unchanged - the extremely compact globally optimized code proved difficult to modify - until major advances in direct methods theory made an update of the structure solution part necessary (**SHELXS-86**). Rewriting and validating the least-squares refinement part proved more difficult but was finally achieved with **SHELXL-93**. SHELXS-86 and SHELXL-93 were as far as possible upwards compatible with SHELX-76 (for example the format of the reflection data file was unchanged) and are now employed in well over 50% of all small-molecule structure determinations. A commercial version including interactive reciprocal and real space graphics is available in the form of the Siemens **SHELXTL** system.

A further release of **SHELX** in the current millenium was never intended, but the increased (mis)use of the programs by macromolecular crystallographers, and some changes to CIF format, have unfortunately made it necessary to release this new version of the complete package as **SHELX-97**. This also provided an opportunity to update the structure solution algorithms, to fix various bugs, and to improve the documentation.

Various beta-test versions were made available to selected guinea-pigs in 1996; these should be referred to as **SHELX-96**, and the final release (in 1997) as **SHELX-97**.

For the latest news the SHELX homepage at http://shelx.uni-ac.gwdg.de/SHELX/ should be consulted.

### 1.1 Programs

**SHELX-97** contains the following six executable programs:

**SHELXS** - Structure solution by Patterson and direct methods.

**SHELXL** - Structure refinement (**SHELXH** for refinement of very large structures).

**CIFTAB** - Tables for publication via CIF format.

**SHELXA** - Post-absorption corrections (for emergency use only).

**SHELXPRO** - Protein interface to SHELX.

**SHELXWAT** - Automatic water divining for macromolecules.

The structure solution program **SHELXS** now includes more powerful direct methods (Sheldrick, 1990) and the use of the Patterson vector superposition method (Sheldrick et al., 1993) - completely different to the naive Patterson interpretation algorithm used in SHELXS-86 - for the automatic location of heavy atoms. This new Patterson interpretation routine is not only effective for small structures, but is also useful for the location of heavy atom sites from isomorphous or anomalous $\Delta F$ data of macromolecules.

The refinement program **SHELXL** includes many new features to make it easier to use for macromolecules, even at moderate resolution (say better than 2.5Å). It also incoporates a large number of small improvements suggested by small-molecule users of SHELXL-93.

In view of the fact that users were encouraged to adapt the 1993 version of **CIFTAB**, which produces tables from the CIF format files generated by SHELXL, only minor corrections have been made to this program.

An anonymous user has kindly donated the program **SHELXA** that can be used to make an 'absorption correction' by fitting the observed to the calculated intensities (like DIFABS). This is intended for emergency use only, e.g. when it is impossible to apply proper absorption corrections because the world's only crystal has been lost before measurements of crystal faces or azimuthal scans could be made. It would be quite unethical to submit a structure processed in this way for publication, and the anonymous donor does not wish to be cited in this non-existent publication since it would ruin his scientific reputation!

A new feature in SHELX-97 is an interactive interface program **SHELXPRO** that is specific to protein applications; SHELXS and SHELXL are very general and in no way specific to certain types of crystal structure. SHELXPRO handles problems of communication with other widely used protein programs; for example it can convert PDB to SHELX format, adding appropriate restraints etc., and can generate sigma-A maps etc. for map interpretation programs such as O. SHELXPRO also displays the refinement results in the form of Postscript diagrams, and facilitates deposition of the refined structure with the PDB.

**SHELXWAT** is a shell program that calls SHELXL iteratively to locate and refine solvent water atoms in macromolecules.

### 1.2 Distribution

SHELX-97 is provided in the form UNIX and VMS sources, plus precompiled versions for MSDOS, LINUX, AIX and IRIX. The programs are written entirely in a very simple subset of FORTRAN. The UNIX versions are highly portable, but sometimes it will be necessary to replace the routines that return the time, date and CPU time with the alternatives provided. Documentation in WINWORD 6.0, HTML and Postscript form, plus examples and test files, are included in the release. The programs are currently available by ftp and on ZIP diskettes or CDROM.

The programs are available free to academics (there is a small charge for ZIP diskettes and CDROMs) and for a license fee (because it is necessary to cover all the costs of distributing and supporting the programs) to for-profit institutions. The license agreement covers the use of the programs for an unlimited time on an unlimited number of computers at one geographical location.

### 1.3 Support

The author (gsheldr@shelx.uni-ac.gwdg.de) is always interested to receive suggestions and comments, and tries to provide advice on installing and using the programs. Email may be quicker than reading the manual, but all emails asking the questions in Chapter 18 (Frequently Asked Questions) will be ignored! The programs are provided on the understanding that the author is in no way liable for any consequences of errors in the programs or their documentation.

# 2. SHELXL - Structure Refinement

SHELXL is a program for the refinement of crystal structures from diffraction data, and is primarily intended for single crystal X-ray data of small moiety structures, though it can also be used for refinement of macromolecules against data to about 2.5 Å or better. It uses a conventional structure factor summation, so it is much slower (but a little more accurate) than standard FFT-based macromolecular programs. SHELXL is intended to be easy to install and use. It is very general, and is valid for all space groups and types of structure. Polar axis restraints and special position constraints are generated automatically. The program can handle twinning, complex disorder, absolute structure determination, CIF and PDB output, and provides a large variety of restraints and constraints for the control of difficult refinements. An interface program SHELXPRO enables macromolecular refinement results to be displayed in the form of Postscript plots, and generates map and other files for communication with widely used macromolecular programs. An auxiliary program CIFTAB is useful for tabulating the refinement results via the CIF output file for small molecules.

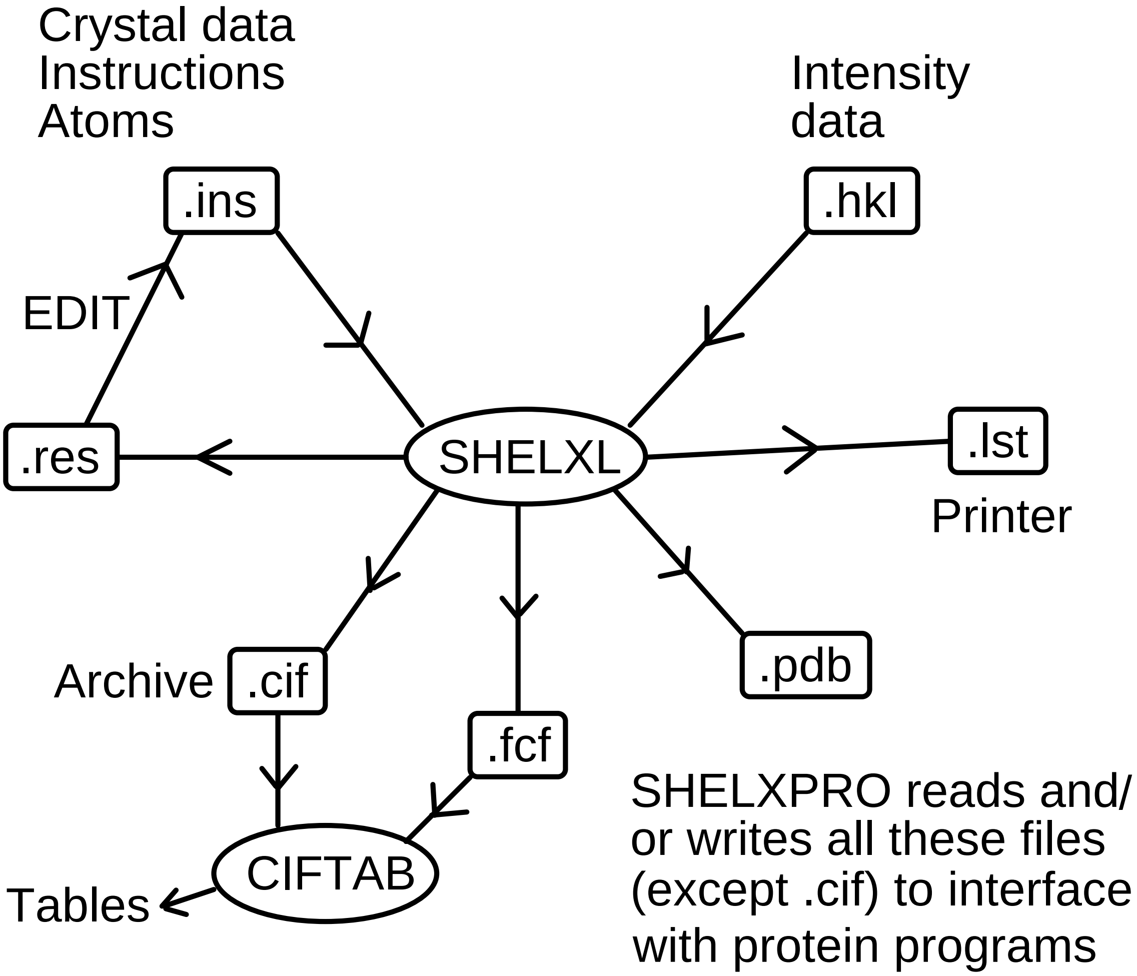
## 2.1 Program organization

To run SHELXL only two input files are required (atoms/instructions and reflection data); since both these files and the output files are pure ASCII text files, it is easy to use the program on a heterogeneous network. The reflection data file (*name.hkl)* contains *h, k, l,* $F^2$ and $\sigma(F^2)$ in standard SHELX format (section 2.3); the program merges equivalents and eliminates systematic absences; the order of the reflections in this file is unimportant. Crystal data, refinement instructions and atom coordinates are all input as the file *name.ins*; further files may be specified as 'include files' in the *.ins* file, e.g. for standard restraints, but this is not essential. Instructions appear in the *.ins* file as four-letter keywords followed by atom names, numbers, etc. in free format; examples are given in the following chapters. There are sensible default values for almost all numerical parameters. SHELXL is normally run on any computer system by means of the command:

```
shelxl name
```

where *name* defines the first component of the filename for all files which correspond to a particular crystal structure. On some systems, *name* may not be longer than 8 characters. Batch operation will normally require the use of a short batch file containing the above command etc. The executable program must be accessible via the 'PATH' (or equivalent mechanism). No environment variables or extra files are required.

A brief summary of the progress of the structure refinement appears on the console, and a full listing is written to a file *name.lst*, which can be printed or examined with a text editor. After each refinement cycle a file *name.res* is (re)written; it is similar to *name.ins*, but has updated values for all refined parameters. It may be copied or edited to *name.ins* for the next refinement run. The MORE instruction controls the amount of information sent to the *.lst* file; normally the default MORE 1 is suitable, but MORE 3 should be used if extensive diagnostic information is required. The ACTA instruction produces CIF format files for archiving or electronic publication, and the LIST 4 instruction (generated automatically by ACTA) produces a CIF format reflection data file (*name.fcf*). For PDB deposition of macromolecular results, WPDB and LIST 6 should be used. The program SHELXPRO should then be used to complete the PDB file.



Two mechanisms are provided for interaction with a SHELXL job which is already running. The first is used by the MSDOS and some other 'on-line' versions: if the <ctrl-I> key combination is hit, the job terminates almost immediately, but without the loss of output buffers etc. which can happen with <ctrl-C> etc. Usually the <Tab> key may be used as an alternative to <ctrl-I>. If the <Esc> key is hit during least-squares refinement, the program completes the current cycle and then, instead of further refinement cycles, continues with the final structure-factor calculation, tables and Fourier etc. Otherwise <Esc> has no effect. On computer consoles with no <Esc> key, <F11> or <Ctrl-[> usually have the same effect.

The second mechanism requires the user to create the file *name.fin* (the contents of this file are irrelevant); the program tries at regular intervals to delete it, and if it succeeds it takes the same action as after <Esc>. The *name.fin* file is also deleted (if found) at the start of a job in case it has been accidentally left over from a previous job. This approach may be used with batch jobs under most operating systems.

## 2.2 The *.ins* instruction file

All instructions commence with a four (or fewer) character word (which may be an atom name); numbers and other information follow in free format, separated by one or more spaces. Upper and lower case input may be freely mixed; with the exception of the text string input using TITL, the input is converted to upper case for internal use in SHELXL. The TITL, CELL, ZERR, LATT (if required), SYMM (if required), SFAC, DISP (if required) and UNIT instructions must be given in that order; all remaining instructions, atoms, etc. should come between UNIT and the last instruction, which is always HKLF (to read in reflection data).

A number of instructions allow atom names to be referenced; use of such instructions without any atom names means 'all non-hydrogen atoms' (in the current residue, if one has been defined). A list of atom names may also be abbreviated to the first atom, the symbol '>' (separated by spaces), and then the last atom; this means 'all atoms between and including the two named atoms but excluding hydrogens'.

## 2.3 The reflection data file *name.hkl*

The *.hkl* file consists of one line per reflection in FORMAT(3I4,2F8.2,I4) for $h,k,l,F_o^2$, $\sigma(F_o^2)$, and (optionally) a batch number. This file should be terminated by a record with all items zero; individual data sets within the file should NOT be separated from one another - the batch numbers serve to distinguish between groups of reflections for which separate scale factors are to be refined (see the BASF instruction). The reflection order and the batch number order are unimportant. This '*.hkl*' file is read each time the program is run; unlike SHELX-76, there is no facility for intermediate storage of binary data. This enhances computer independence and eliminates several possible sources of confusion. The *.hkl* file is read when the HKLF instruction (which terminates the *.ins* file) is encountered. The HKLF instruction specifies the format of the *.hkl* file, and allows scale factors and a reorientation matrix to be applied. Lorentz, polarization and absorption corrections are assumed to have been applied to the data in the *.hkl* file. Note that there are special extensions to the *.hkl* format for Laue and powder data, as well as for twinned crystals that cannot be handled by a TWIN instruction alone.

In general the *.hkl* file should contain all measured reflections without rejection of systematic absences or merging of equivalents. The systematic absences and $R_{\text{int}}$ for equivalents provide an excellent check on the space group assignment and consistency of the input data. Since complex scattering factors are used throughout by SHELXL, Friedel opposites should normally not be averaged in preparing this file; an exception can be made for macromolecules without significant anomalous scatterers. Note that SHELXS always merges Friedel opposites.

## 2.4 Refinement against $F^2$

SHELXL always refines against $F^2$, even when $F$-values are input. Refinement against ALL $F^2$-values is demonstrably superior to refinement against $F$-values greater than some threshold [say $4\sigma(F)$]. More experimental information is incorporated (suitably weighted) and the chance of getting stuck in a local minimum is reduced. In pseudo-symmetry cases it is very often the weak reflections that can discriminate between alternative potential solutions. It is difficult to refine against ALL $F$-values because of the difficulty of estimating $\sigma(F)$ from $\sigma(F^2)$ when $F^2$ is zero or (as a result of experimental error) negative.

The diffraction experiment measures intensities and their standard deviations, which after the various corrections give $F_o^2$ and $\sigma(F_o^2)$. If your data reduction program only outputs $F_o$ and $\sigma(F_o)$, you should correct your data reduction program, not simply write a routine to square the $F_o$ values ! It is also legal to use HKLF 3 to input $F_o$ and $\sigma(F_o)$ to SHELXL. Note that if an $F_o^2$ value is too large to fit format F8.2, then format F8.0 may be used instead. - the decimal point overrides the FORTRAN format specification.

The use of a threshold for ignoring weak reflections may introduce bias which primarily affects the atomic displacement parameters; it is only justified to speed up the early stages of refinement. In the final refinement ALL DATA should be used except for reflections known to suffer from systematic error (i.e. in the final refinement the OMIT instruction may be used to omit specific reflections - although not without good reason - but not ALL reflections below a given threshold). Anyone planning to ignore this advice should read Hirshfeld & Rabinovich (1973) and Arnberg, Hovmöller & Westman (1979) first. Refinement against $F^2$ also facilitates the treatment of twinned and powder data, and the determination of *absolute structure*.

## 2.5 Initial processing of reflection data

SHELXL automatically rejects systematically absent reflections. The sorting and merging of the reflection data is controlled by the MERG instruction. Usually MERG 2 (the default) will be suitable for small molecules; equivalent reflections are merged and their indices converted to standard symmetry equivalents, but Friedel opposites are not merged in non-centrosymmetric space groups. MERG 4, which merges Friedel opposites and sets $\delta f''$ for all elements to zero, saves time for macromolecules with no significant dispersion effects. Throughout this documentation, $F_o^2$ means the EXPERIMENTAL measurement, which despite the square may possibly be slightly negative if the background is higher than the peak as a result of statistical fluctuations etc. $R_{\mathrm{int}}$ and $R_{\mathrm{sigma}}$ are defined as follows:

$$R_{\mathrm{int}} = \Sigma \mid F_o^2 - F_o^2(\mathrm{mean}) \mid \; / \, \Sigma \, [ \, F_o^2 \, ]$$

where both summations involve all input reflections for which more than one symmetry equivalent is averaged, and:

$$R_{\mathrm{sigma}} = \Sigma \, [ \, \sigma(F_o^2) \, ] \; / \, \Sigma \, [ \, F_o^2 \, ]$$

over all reflections in the merged list. Since these $R$-indices are based on $F^2$, they will tend to be about twice as large as the corresponding indices based on $F$. The 'esd of the mean' (in the table of inconsistent equivalents) is the rms deviation from the mean divided by the square root of ($n$-1), where $n$ equivalents are combined for a given reflection. In estimating the $\sigma(F^2)$ of a merged reflection, the program uses the value obtained by combining the $\sigma(F^2)$ values of the individual contributors, unless the esd of the mean is larger, in which case it is used instead.

For some refinements of twinned crystals, and for least-squares refinement of batch scale factors, it is necessary to suppress the merging of equivalent reflections with MERG 0.

## 2.6 Least-squares refinement

Small molecules are almost always refined by full-matrix methods (using the L.S. instruction in SHELXL), which give the best convergence per cycle, and allows esd's to be estimated. The CPU time per cycle required for full-matrix refinement is approximately proportional to the number of reflections times the square of the number of parameters; this is prohibitive for all but the smallest macromolecules. In addition the (single precision) matrix inversion suffers from accumulated rounding errors when the number of parameters becomes very large. An excellent alternative for macromolecules is the conjugate-gradient solution of the normal equations, taking into account only those off-diagonal terms that involve restraints. This method was employed by Konnert & Hendrickson (1980) in the program PROLSQ; except for modifications to accelerate the convergence, exactly the same algorithm is used in SHELXL (instruction CGLS). The CGLS refinement can be also usefully employed in the early stages of refinement of medium and large 'small molecules'; it requires more cycles for convergence, but is fast and robust. The major disadvantage of CGLS is that it does not give esds.

For both L.S. and CGLS options, it is possible to block the refinement so that a different combination of parameters is refined each cycle. For example after a large structure has been refined using CGLS (without BLOC), a final job may be run with L.S. 1, DAMP 0 0 and BLOC 1 (or e.g. BLOC N_1 > LAST for a protein) to obtain esds on all geometric parameters; the anisotropic displacement parameters are held fixed, reducing the number of parameters by a factor of three and the cycle time by an order of magnitude.

## 2.7 *R*-indices and weights

One cosmetic disadvantage of refinement against $F^2$ is that *R*-indices based on $F^2$ are larger than (more than double) those based on *F*. For comparison with older refinements based on *F* and an OMIT threshold, a conventional index *R*1 based on observed *F* values larger than $4\sigma(F_o)$ is also printed.

$$wR2 = \{ \Sigma [ w(F_o^2 - F_c^2)^2 ] / \Sigma [ w(F_o^2)^2 ] \}^{1/2}$$

$$R1 = \Sigma \| F_o| - |F_c\| / \Sigma |F_o|$$

The *Goodness of Fit* is always based on $F^2$:

$$\text{GooF} = S = \{ \Sigma [ w(F_o^2 - F_c^2)^2 ] / (n-p) \}^{1/2}$$

where *n* is the number of reflections and *p* is the total number of parameters refined.

The WGHT instruction allows considerable flexibility, but in practice it is a good idea to leave the weights at the default setting (WGHT 0.1) until the refinement is essentially complete, and then to use the scheme recommended by the program. These parameters should give a flat analysis of variance and a GooF close to unity [there was a bug in SHELXL-93 that can occasionally cause the program to abort when trying to estimate the new weighting parameters, though it appeared to happen only with poor quality data or the wrong solution]. If the weights are varied too soon, the convergence may be impaired, because features such as missing atoms are 'weighted down'. For macromolecules it may be advisable to leave the weights at the default settings; and to accept a GooF greater than one as an admission of inadequacies in the model.

When not more than two WGHT parameters are specified, the weighting scheme simplifies to:

$$w = 1 / [\, \sigma^2(F_o^2) + (aP)^2 + bP \,]$$

where $P$ is $[\, 2F_c^2 + \mathrm{Max}(F_o^2,0) \,] / 3$. The use of this combination of $F_o^2$ and $F_c^2$ was shown by Wilson (1976) to reduce statistical bias.

It may be desirable to use a scheme that does not give a flat analysis of variance to emphasize particular features in the refinement, for example by weighting up the high angle data to remove bias caused by bonding electron density (Dunitz & Seiler, 1973).

## 2.8 Fourier syntheses

Fourier syntheses are summarized in the form of peak-lists (which can be edited and re-input for the next refinement job), or as 'lineprinter plots' with an analysis of non-bonded interactions etc. It is recommended that a difference electron density synthesis is performed at the end of each refinement job; it is quick and of considerable diagnostic value. In contrast to SHELX-76, SHELXL finds the asymmetric unit for the Fourier synthesis automatically; the algorithm is valid for all space groups, in conventional settings or otherwise. Before calculating a Fourier synthesis, the Friedel opposites are always merged and a dispersion correction applied; a value of $R1$ is calculated for the merged data (without a threshold). Reflections with $F_c$ small compared to $\sigma(F_o)$ are down-weighted in the Fourier synthesis. The rms density is calculated to give an estimate of the 'noise level' of the map.

## 2.9 The connectivity array

The key to the automatic generation of hydrogen atoms, molecular geometry tables, restraints etc. is the connectivity array. For a non-disordered organic molecule, the connectivity array can be derived automatically using standard atomic radii. A simple notation for disordered groups enables most cases of disorder to be processed with a minimum of user intervention. Each atom is assigned a 'PART' number *n*. The usual value of *n* is 0, but other values are used to label components of a disordered group. Bonds are then generated for atoms that are close enough only when either (a) at least one of them has n=0, or (b) both values of *n* are the same. A single shell of symmetry equivalents is automatically included in the connectivity array; the generation of equivalents (e.g. in a toluene molecule on an inversion center) may be prevented by assigning a negative 'PART' number. If necessary bonds may be added to or deleted from the connectivity array using the BIND or FREE instructions. To generate additional bonds to symmetry equivalent atoms, EQIV is also needed.

## 2.10 Tables

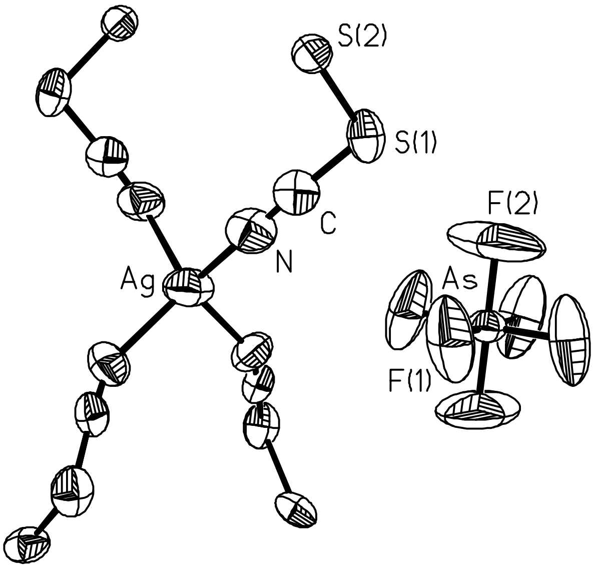
For small structures, bond lengths and angles for the full connectivity array may be tabulated with BOND, and all possible torsion angles with CONF. Although hydrogen atoms are not normally included in the connectivity array, they may be included in the bond lengths and angles tables by BOND $H. Alternatively HTAB produces a convenient way of analysing hydrogen bonds. It is also possible to be selective by naming specific atoms on the BOND and CONF instructions, or by using the RTAB instruction (which was designed with macromolecules in mind). Least-squares planes and distances of (other) atoms from these planes may be generated with MPLA. Symmetry equivalent atoms may be specified on any of these instructions by reference to EQIV symmetry operators. All esds output by SHELXL take the unit-cell esds into account and are calculated using the full covariance matrix. The only exception is the esd in the angle between two least-squares planes, for which an approximate treatment is used. Note that damping the refinement (see above) leads to underestimates of the esds; in difficult cases a final cycle may be performed with DAMP 0 0 (no damping, but no shifts applied) to obtain good esds.

The HTAB instruction has been introduced in SHELXL-97 to analyze the hydrogen bonding in the structure. A search is made over all *hydrogen atoms* to find possible hydrogen bonds. This is a convenient way of finding the symmetry operations necessary for the second form of HTAB instructions (needed to obtain esds and CIF output), and also reveals potential misplaced hydrogens, e.g. because they do not make any hydrogen bonds, or because the automatic placing of hydrogen atoms has assigned the hydrogens of two different O-H or N-H groups to the same hydrogen bond. In the second form of the HTAB instruction, HTAB is followed by the names of the donor atom D and the acceptor atom A; for the latter a symmetry operation may also be specified. The program then finds the most suitable hydrogen atom to form the hydrogen bond D-H•••A, and outputs the geometric data for this hydrogen bond to the *.lst* file and the *.cif* file (if ACTA is present).

# 3. Examples of Small Molecule Refinements with SHELXL

Two test structures supplied with the SHELXL-97 are intended to provide a good illustration of routine small moiety structure refinement. The output discussed here should not differ significantly from that of the test jobs, except that it has been abbreviated and there may be differences in the last decimal place caused by rounding errors.

### 3.1 First example (ags4)



The first example (provided as the files *ags4.ins* and *ags4.hkl*) is the final refinement job for the polymeric inorganic structure Ag(NCSSSSCN)$_2$ AsF$_6$, determined by Roesky, Gries, Schimkowiak & Jones (1986). Each ligand bridges two Ag$^+$ ions so each silver is tetrahedrally coordinated by four nitrogen atoms. The silver, arsenic and one of the fluorine atoms lie on special positions. Normally the four unique heavy atoms (from Patterson interpretation using SHELXS) would have been refined isotropically first and the remaining atoms found in a difference synthesis, and possibly an intermediate job would have been performed with the heavy atoms anisotropic and the light atoms isotropic. For test purposes we shall simply input the atomic coordinates which assumes isotropic U's of 0.05 for all atoms. In this job all atoms are to be made anisotropic (ANIS). We shall further assume that a previous job has recommended the weighting scheme used here (WGHT) and shown that one reflection is to be suppressed in the refinement because it is clearly erroneous (OMIT).

The first 9 instructions (TITL...UNIT) are the same for any SHELXS and SHELXL job for this structure and define the cell dimensions, symmetry and contents. The SHELXTL program XPREP can be used to generate these instructions automatically for any space group etc. SHELXL knows the scattering factors for the first 94 neutral atoms in the Periodic Table. Ten least-squares cycles are to be performed, and the ACTA instruction ensures that the CIF files *ags4.cif* and *ags4.fcf* will be written for archiving and publication purposes. ACTA also sets up the calculation of bond lengths and angles (BOND) and a final difference electron density synthesis (FMAP 2) with peak search (PLAN 20). The HKLF 4 instruction terminates the file and initiates the reading of the *ags4.hkl* intensity data file.

It is possible to set up special position constraints on the x,y,z-coordinates, occupation factors, and $U_{ij}$ components by hand. However this is totally unnecessary because the program will do this automatically for any special position in any space group, conventional or otherwise.  Similarly the program recognizes polar space groups ($P\bar{4}$ is non-polar) and applies appropriate restraints (Flack & Schwarzenbach, 1988), so it is no longer necessary to worry about fixing one or more coordinates to prevent the structure drifting along polar axes. It is not necessary to set the overall scale factor using an FVAR instruction for this initial job, because the program will itself estimate a suitable starting value.  Comments may be included in the *.ins* file either as REM instructions or as the rest of a line following '!'; this latter facility has been used to annotate this example.

```
TITL AGS4 in P-4                         ! title of up to 76 characters
CELL 0.71073 8.381 8.381 6.661 90 90 90  ! wavelength and unit-cell
ZERR 1 .002 .002 .001 0 0 0              ! Z (formula-units/cell), cell esd's
LATT -1                              ! non-centrosymmetric primitive lattice
SYMM -X, -Y, Z
SYMM Y, -X, -Z              ! symmetry operators (x,y,z must be left out)
SYMM -Y, X, -Z
SFAC C AG AS F N S          ! define scattering factor numbers
UNIT 4 1 1 6 4 8            ! unit cell contents in same order
L.S. 10                     ! 10 cycles full-matrix least-squares
ACTA                        ! CIF-output, bonds, Fourier, peak search
OMIT -2 3 1                 ! suppress bad reflection
ANIS                        ! convert all (non-H) atoms to anisotropic
WGHT 0.037 0.31             ! weighting scheme
AG  2  .000  .000  .000
AS  3  .500  .500  .000
S1  6  .368  .206  .517     ! atom name, SFAC number, x, y, z (usually
S2  6  .386  .034  .736     ! followed by sof and U(iso) or Uij); the
C   1  .278  .095  .337     ! program automatically generates special
N   5  .211  .030  .214     ! position constraints
F1  4  .596  .325 -.007
F2  4  .500  .500  .246
HKLF 4                      ! read h,k,l,Fo^2,sigma(Fo^2) from 'ags4.hkl'
```

The *.lst* listing file starts with a header followed by an echo of the above *.ins* file. After reading TITL...UNIT the program calculates the cell volume, F(000), absorption coefficient, cell weight and density. If the density is unreasonable, perhaps the unit-cell contents have been given incorrectly. The next items in the *.lst* file are the connectivity table and the symmetry operations used to include a shell of symmetry equivalent atoms (so that all unique bond lengths and angles can be found):

```
Covalent radii and connectivity table for AGS4 in P-4

C    0.770
AG   1.440
AS   1.210
F    0.640
N    0.700
S    1.030

Ag - N N_$3 N_$4 N_$2
As - F2 F2_$6 F1_$7 F1_$6 F1_$5 F1
S1 - C S2
```

```
S2 - S2_$1 S1
C - N S1
N - C Ag
F1 - As
F2 - As

Operators for generating equivalent atoms:

$1   -x+1, -y+1, z
$2   -x, -y, z
$3   y, -x, -z
$4   -y, x, -z
$5   -x+1, -y+1, z
$6   y, -x+1, -z
$7   -y+1, x, -z
```

Note that in addition to symmetry operations generated by the program, one can also define operations with the EQIV instruction and then refer to the corresponding atoms with _$n in the same way. Thus:

```
EQIV $1 1-x, -y, z
CONF S1 S2 S2_$1 S1_$1
```

could have been included in *ags4.ins* to calculate the S-S-S-S torsion angle. If EQIV instructions are used, the program renumbers the other symmetry operators accordingly.

The next part of the output is concerned with the data reduction:

```
 1475  Reflections read, of which     1  rejected

0 =< h =< 10,     -9 =< k =< 10,      0 =< l =<  8,   Max. 2-theta =   55.00

    0  Systematic absence violations

Inconsistent equivalents etc.

 h   k   l       Fo^2    Sigma(Fo^2)  Esd of mean(Fo^2)

 3   4   0      387.25       8.54       47.78

     1  Inconsistent equivalents
   903  Unique reflections, of which      0  suppressed

R(int) = 0.0165     R(sigma) = 0.0202      Friedel opposites not merged
```

Special position constraints are then generated and the statistics from the first least-squares cycle are listed (the output has been compacted to fit the page).  The maximum vector length refers to the number of reflections processed simultaneously in the rate-determining calculations; usually the program utilizes all available memory to make this as large as possible, subject to a maximum of 511. This maximum may be reduced (but not increased) by means of the fourth parameter on the L.S. (or CGLS) instruction; this may be required to prevent unnecessary disk transfers when large structures are refined on virtual memory systems with limited physical memory. The number of parameters refined in the current cycle is followed by the total number of refinable parameters (here both are 55).

```
Special position constraints for Ag
x =  0.0000       y =  0.0000         z =  0.0000         U22 = 1.0 * U11
U23 = 0           U13 = 0             U12 = 0             sof = 0.25000

Special position constraints for As
x =  0.5000       y =  0.5000         z =  0.0000         U22 = 1.0 * U11
U23 = 0           U13 = 0             U12 = 0             sof = 0.25000

Special position constraints for F2
x =  0.5000       y =  0.5000         U23 = 0             U13 = 0
sof = 0.50000


Least-squares cycle 1   Maximum vector length=511  Memory required=1092/82899

wR2 =  0.5042 before cycle   1 for    903 data and   55 /   55 parameters

GooF = S =  8.127;     Restrained GooF =      8.127  for      0 restraints

Weight = 1/[sigma^2(Fo^2)+(0.0370*P)^2+0.31*P] where P=(Max(Fo^2,0)+2*Fc^2)/3

** Shifts scaled down to reduce maximum shift/esd from   17.64  to   15.00 **

  N     value       esd        shift/esd  parameter

  1     2.31065     0.04324     9.042     OSF
  2     0.07314     0.00206    11.250     U11 Ag
 11     0.07309     0.00669     3.453     U33 S1
 47     0.11304     0.01391     4.533     U33 F1

Mean shift/esd =   1.238    Maximum =  11.250 for  OSF

Max. shift = 0.045 A for C      Max. dU = 0.033 for F2
```

Only the largest shift/esd's are printed. More output could have been obtained using 'MORE 2' or 'MORE 3'. The largest correlation matrix elements are printed after the last cycle, in which the mean and maximum shift/esd have been reduced to 0.003 and 0.017 respectively. This is followed by the full table of refined coordinates and $U_{ij}$'s with esd's (too large to include here, but similar to the corresponding table in SHELX-76 except that $U_{eq}$ and its esd are also printed) and by a final structure factor calculation:

```
Final Structure Factor Calculation for  AGS4 in P-4

Total number of l.s. parameters = 55  Maximum vector length = 511
wR2 =  0.0780 before cycle  11 for    903 data and    2 /   55 parameters

GooF = S =     1.063;     Restrained GooF =      1.063  for      0 restraints
Weight = 1/[sigma^2(Fo^2)+(0.0370*P)^2+0.31*P] where P=(Max(Fo^2,0)+2*Fc^2)/3
R1 =  0.0322 for    818 Fo > 4.sigma(Fo)  and  0.0367 for all    903 data
wR2 =  0.0780,  GooF = S =   1.063,  Restrained GooF =    1.063  for all data

Flack x parameter = 0.0224   with esd  0.0260   (expected values are 0
(within 3 esd's) for correct and +1 for inverted absolute structure)
```

There are some important points to note here. The weighted *R*-index based on $F_o^2$ is (for compelling statistical reasons) much higher than the conventional *R*-index based on $F_o$ with a threshold of say $F_o > 4\sigma(F_o)$. For comparison with structures refined against *F* the latter is therefore printed as well (as *R*1). Despite the fact that *wR*2 and not *R*1 is the quantity minimized, *R*1 has the advantage that it is relatively insensitive to the weighting scheme, and so is more difficult to manipulate.

Since the structure is non-centrosymmetric, the program has automatically estimated the Flack absolute structure parameter x in the final structure factor summation. In this example x is within one esd of zero, and its esd is also relatively small. This provides strong evidence that the absolute structure has been assigned correctly, so that no further action is required. The program would have printed a warning here if it would have been necessary to 'invert' the structure or to refine it as a racemic twin.
.
This is followed by a list of principal mean square displacements U for all anisotropic atoms. It will be seen that none of the smallest components (in the third column) are in danger of going negative [which would make the atom 'non positive definite' (NPD)] but that the motion of the two unique fluorine atoms is highly anisotropic (not unusual for an $AsF_6$ anion). The program suggests that the fluorine motion is so extended in one direction that it would be possible to represent each of the two fluorine atoms as disordered over two sites, for which x, y and z coordinates are given; this may safely be ignored here (although there may well be some truth in it). The two suggested new positions for each 'split' atom are placed equidistant from the current position along the direction (and reverse direction) corresponding to the largest eigenvalue of the anisotropic displacement tensor.

This list is followed by the analysis of variance (reproduced here in squashed form), recommended weighting scheme (to give a flat analysis of variance in terms of $F_c^2$), and a list of the most disagreeable reflections. For a discussion of the analysis of variance see the second example.

```
Principal mean square atomic displacements U

    0.1067    0.1067    0.0561    Ag
    0.0577    0.0577    0.0386    As
    0.1038    0.0659    0.0440    S1
    0.0986    0.0515    0.0391    S2
    0.0779    0.0729    0.0391    C
    0.1004    0.0852    0.0474    N
    0.3029    0.0954    0.0473    F1
   may be split into  0.5965  0.3173  0.0288  and 0.5946  0.3324 -0.0369
    0.4778    0.1671    0.0457    F2
   may be split into  0.5320  0.5089  0.2462  and 0.4680  0.4911  0.2462


Analysis of variance for reflections employed in refinement
K = Mean[Fo^2] / Mean[Fc^2]  for group

Fc/Fc(max)     0.000 0.026 0.039 0.051 0.063 0.082 0.103 0.147 0.202 0.306 1.0
Number in group    94.   89.   90.   91.   89.   91.   89.   91.   88.   91.
GooF         1.096 1.101 0.997 1.078 1.187 1.069 1.173 0.922 1.019 0.966
K           1.560 1.053 1.010 1.004 1.007 1.021 1.026 1.002 0.997 0.984
```

```
Resolution(A)  0.77  0.81  0.85  0.90  0.95  1.02  1.10  1.22  1.40  1.74  inf
Number in group    97.   84.   92.   91.   89.   90.   89.   90.   93.   88.
GooF         1.067 0.959 0.935 0.895 1.035 1.040 1.115 1.149 1.161 1.228
K            1.047 1.010 1.009 0.991 1.004 0.996 0.989 1.012 0.997 0.982
R1           0.166 0.100 0.069 0.059 0.051 0.036 0.033 0.027 0.020 0.020

Recommended weighting scheme:  WGHT   0.0314 0.3674


Most Disagreeable Reflections (* if suppressed or used for Rfree)

    h   k   l       Fo^2      Fc^2    Delta(F^2)/esd  Fc/F(max)  Resolution(A)
    4   4   4       18.32    33.30      3.62          0.062       1.11
   -4   1   3       15.79     4.17      3.50          0.022       1.50
    0   2   2       41.60    57.32      3.26          0.082       2.61      etc.
```

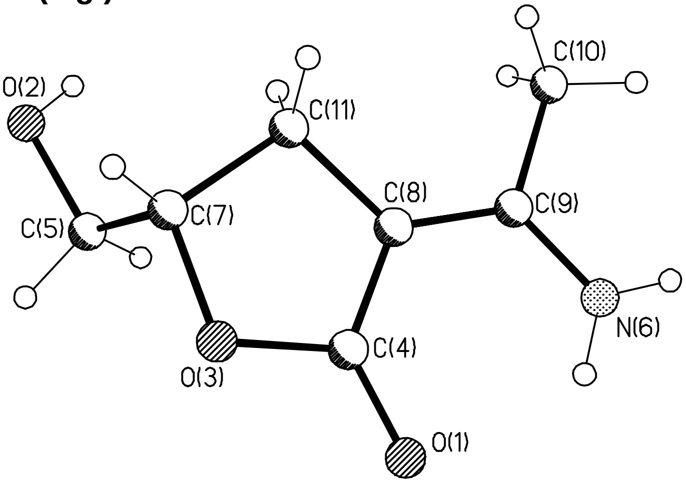
After the table of bond lengths and angles (BOND was implied by the ACTA instruction), the data are merged (again) for the Fourier calculation after correcting for dispersion (because the electron density is real). In contrast to the initial data reduction, Friedel's law is assumed here; the aim is to set up a unique reflection list so that the (difference) electron density can be calculated on an absolute scale.

The algorithm for generating the 'asymmetric unit' for the Fourier calculations is general for all space groups, in conventional settings or otherwise. The rms electron density (averaged over all grid points) is printed as well as the maximum and minimum values so that the significance of the latter can be assessed. Since PLAN 20 was assumed, only a peak list is printed (and written to the *.res* file), followed by a list of shortest distances between peaks (not shown below); PLAN -20 would have produced a more detailed analysis with 'printer plots' of the structure. The last 40 peaks and some of the interatomic distances have been deleted here to save space. In this table, 'distances to nearest atoms' takes symmetry equivalents into account.

```
Bond lengths and angles        [severely squashed to fit page!]

Ag - Distance     Angles
N    2.2788(0.0058)
N_$2 2.2788(0.0058) 113.08(0.15)
N_$4 2.2788(0.0058) 113.08(0.15) 102.47(0.29)
N_$3 2.2788(0.0058) 102.47(0.29) 113.08(0.15) 113.08(0.15)
      Ag -          N            N_$3         N_$4

As - Distance     Angles
F2    1.6399(0.007)
F2_$6 1.6399(0.007)180.00(0.00)
F1_$7 1.6724(0.0037) 89.08(0.41) 90.92(0.41)
F1_$6 1.6724(0.0037) 89.08(0.41) 90.92(0.41)178.15(0.82)
F1_$5 1.6724(0.0037) 90.92(0.41) 89.08(0.41) 90.01(0.01) 90.01(0.01)
F1    1.6724(0.0037) 90.92(0.41) 89.08(0.41) 90.01(0.01) 90.01(0.01)178.15(0.82)
       As -         F2           F2_$6       F1_$7       F1_$6       F1_$5

S1 - Distance     Angles
C     1.6819(0.0069)
S2    2.0633(0.0025)  98.61(0.20)
         S1 -        C
```

```
S2 - Distance  Angles
S2_$1 2.0114(0.0028)
S1    2.0633(0.0025) 105.37(0.07)
         S2 -        S2_$1

C -  Distance  Angles
N   1.1472(0.0074)
S1  1.6819(0.0069) 175.67(0.49)
       C -         N

N -  Distance  Angles
C   1.1472(0.0074)
Ag  2.2788(0.0058) 152.38(0.45)
       N -         C
F1 - Distance  Angles
As  1.6724(0.0037)
       F1 -

F2 - Distance  Angles
As  1.6399(0.0075)
       F2 -


FMAP and GRID set by program

FMAP 2   3  18
GRID  -3.333  -2  -1     3.333   2   1

R1 = 0.0370 for 590 unique reflections after merging for Fourier


Electron density synthesis with coefficients Fo-Fc

Highest peak   0.32  at  0.0000  0.0000  0.5000  [2.60 A from N]
Deepest hole  -0.36  at  0.5000  0.5000  0.1863  [0.40 A from F2]
Mean = 0.00, Rms deviation from mean = 0.07 e/A^3 Highest memory used 1133/13851


Fourier peaks appended to .res file

         x       y       z       sof      U   Peak  Dist to nearest atoms
Q1  1  0.0000  0.0000  0.5000  0.25000  0.05  0.32  2.60 N  2.69 C  3.33 AG
Q2  1  0.5690  0.3728  0.1623  1.00000  0.05  0.27  1.20 F1 1.34 F2 1.62 AS
Q3  1  0.5685  0.3851 -0.1621  1.00000  0.05  0.24  1.19 F1 1.25 F2 1.56 AS
Q4  1  0.4075  0.4717  0.2378  1.00000  0.05  0.23  0.81 F2 1.78 AS 1.79 F1
Q5  1  0.5848  0.2667  0.0312  1.00000  0.05  0.23  0.55 F1 2.09 AS 2.47 F1
Q6  1  0.5495  0.3425 -0.1122  1.00000  0.05  0.21  0.83 F1 1.57 AS 1.65 F2
Q7  1  0.2617 -0.1441  0.1446  1.00000  0.05  0.20  1.59 N  2.17 F1 2.40 C
Q8  1  0.7221  0.1898  0.0030  1.00000  0.05  0.20  1.55 F1 2.39 N  2.54 N
Q9  1  0.1997  0.0293  0.1024  1.00000  0.05  0.19  0.75 N  1.79 C  1.82 AG
Q10 1  0.4606 -0.0113  0.8165  1.00000  0.05  0.19  0.91 S2 1.41 S2 2.82 S1
```

### 3.2 Second example (sigi)



In the second example (provided as the files *sigi.ins* and *sigi.hkl*) a small organic structure is refined in the space group $P\overline{1}$. Only the features that are different from the ags4 refinement will be discussed in detail. The structure consists of a five-membered lactone [-C7-C11-C8-C4(O1)-O3-] with a $-CH_2-OH$ group [-C5-O2] attached to C7 and a $=C(CH_3)(NH_2)$ unit [=C9(C10)N6] double-bonded to C8.

Of particular interest here is the placing and refinement of the 11 hydrogen atoms via HFIX instructions. The two $-CH_2-$ groups (C5 and C11) and one tertiary CH (C7) can be placed geometrically by standard methods; the algorithms have been improved relative to those used in SHELX-76, and the hydrogen atoms are now idealized before each refinement cycle (and after the last). Since N6 is attached to a conjugated system, it is reasonable to assume that the $-NH_2$ group is coplanar with the C8=C9(C10)-N6 unit, which enables these two hydrogens to be placed as ethylenic hydrogens, requiring HFIX (or AFIX) 9n; the program takes into account that they are bonded to nitrogen in setting the default bond lengths. All these hydrogens are to be refined using a 'riding model' (HFIX or AFIX m3) for x, y and z.

The -OH and $-CH_3$ groups are trickier, in the latter case because C9 is $sp^2$-hybridized, so the potential barrier to rotation is low and there is no fully staggered conformation available as the obvious choice. Since the data are reasonable, the initial torsion angles for these two groups can be found by means of difference electron density syntheses calculated around the circles which represent the loci of all possible hydrogen atom positions. The torsion angles are then refined during the least-squares refinement. Note that in subsequent cycles (and jobs) these groups will be re-idealized geometrically with retention of the current torsion angle; the circular Fourier calculation is performed only once. Two 'free variables' (2 and 3 yes, they still exist!) have been assigned to refine common isotropic displacement parameters for the 'rigid' and 'rotating' hydrogens respectively. If these had not been specified, the default action would have been to hold the hydrogen U values at 1.2 times the equivalent isotropic U of the atoms to which they are attached (1.5 for the -OH and methyl groups).

The *sigi.ins* file (which is provided as a test job) is as follows. Note that for instructions with both numerical parameters and atom names such as HFIX and MPLA, it does not matter whether numbers or atoms come first, but the order of the numerical parameters themselves (and in some cases the order of the atoms) is important.

```
TITL SIGI in P-1
CELL 0.71073 6.652 7.758 8.147 73.09 75.99 68.40
ZERR 2 .002 .002 .002 .03 .03 .03
SFAC C H N O
UNIT 14 22 2 6        ! no LATT and SYMM needed for space group P-1

L.S. 4
EXTI 0.001            ! refine an isotropic extinction parameter
WGHT .060 0.15        ! (suggested by program in last job);  WGHT
OMIT 2 8 0            ! and OMIT are also based on previous output
BOND $H               ! include H in bond lengths / angles table
CONF                  ! all torsion angles except involving hydrogen
HTAB                  ! analyse all hydrogen bonds
FMAP 2                ! Fo-Fc Fourier
PLAN -20              ! printer plots and full analysis of peak list

HFIX 147 31 O2        ! initial location of -OH and -CH3 hydrogens from
HFIX 137 31 C10       ! circular Fourier, then refine torsion, U(H)=fv(3)

HFIX 93 21 N6         ! -NH2 in plane, xyz ride on N, U(H)=fv(2)
HFIX 23 21 C5 C11     ! two -CH2- groups, xyz ride on C, U(H)=fv(2)
HFIX 13 21 C7         ! tertiary CH, xyz ride on C, U(H)=fv(2)

EQIV $1 X-1, Y, Z     ! define symmetry operations for H-bonds
EQIV $2 X+1, Y, Z-1
HTAB N6 O1            ! outputs H-bonds D-H...A with esds
HTAB O2 O1_$1         ! _$1 and _$2 refer to symmetry equivalents
HTAB N6 O2_$2
                                    ! l.s. planes through 5-ring and through
MPLA 5 C7 C11 C8 C4 O3 O1 N6 C9 C10 ! CNC=CCC moiety, then find deviations
MPLA 6 C10 N6 C9 C8 C11 C4 O1 O3 C7 ! of last 4 and 3 named atoms resp. too

FVAR 1 .06 .07                      ! overall scale and free variables for U(H)

REM name sfac# x y z sof(+10 to fix it) U11 U22 U33 U23 U13 U12 follow

O1  4   0.30280   0.17175   0.68006   11.00000   0.02309   0.04802 =
    0.02540  -0.00301  -0.00597  -0.01547
O2  4  -0.56871   0.23631   0.96089   11.00000   0.02632   0.04923 =
    0.02191  -0.00958  0.00050  -0.02065
O3  4  -0.02274   0.28312 0.83591 11.00000 0.02678 0.04990 =
    0.01752  -0.00941  -0.00047  -0.02109
C4  1   0.10358   0.23458 0.68664 11.00000 0.02228 0.02952 =
    0.01954  -0.00265  -0.00173  -0.01474
C5  1  -0.33881   0.18268 0.94464 11.00000 0.02618 0.03480 =
    0.01926  -0.00311  -0.00414  -0.01624
N6  3  0.26405    0.17085 0.33925 11.00000 0.03003 0.04232 =
    0.02620  -0.01312  0.00048  -0.01086
C7  1  -0.25299   0.33872 0.82228 11.00000 0.02437 0.03111 =
    0.01918  -0.00828  -0.00051  -0.01299
C8  1  -0.03073   0.27219 0.55976 11.00000 0.02166 0.02647 =
    0.01918  -0.00365  -0.00321  -0.01184
C9  1  0.05119    0.24371 0.39501 11.00000 0.02616 0.02399 =
    0.02250  -0.00536  -0.00311  -0.01185
C10 1  -0.10011   0.29447 0.26687 11.00000 0.03877 0.04903 =
    0.02076  -0.01022  -0.00611  -0.01800
C11 1  -0.26553   0.36133 0.63125 11.00000 0.02313 0.03520 =
    0.01862  -0.00372  -0.00330  -0.01185

HKLF 4   ! read intensity data from 'sigi.hkl'; terminates '.ins' file
END
```

The data reduction reports 1904 reflections read (one of which was rejected by OMIT) with indices $-7 \le h \le 7$, $-8 \le k \le 9$ and $-9 \le l \le 9$. Note that these are the limiting index values; in fact only about 1.5 times the unique volume of reciprocal space was measured. The maximum $2\theta$ was 50.00, and there were no systematic absence violations, 34 (not seriously) inconsistent equivalents, and 1296 unique data.  R(int) was 0.0196 and R(sigma) 0.0151.

The program uses different default distances to hydrogen for different bonding situations; these may be overridden by the user if desired. These defaults depend on the temperature (set using TEMP) in order to allow for librational effects. The list of default X-H distances is followed by the (squashed) circular difference electron density syntheses to determine the C-OH and C-$CH_3$ initial torsion angles:

```
Default effective X-H distances for T =   20.0 C

AFIX m =    1    2    3     4   4[N]  3[N]  15[B]  8[O]   9   9[N]   16
d(X-H) =  0.98 0.97 0.96  0.93  0.86  0.89  1.10  0.82  0.93  0.86  0.93

Difference electron density (eA^-3x100) at 15 degree intervals for AFIX 147 group
attached to O2. The center of the range is eclipsed (cis) to C7 and rotation is
clockwise looking down C5 to O2
  2 -2 -6 -9 -8 -5 -1  0  0  0  1  0 -2 -2  0  9 23 39 48 42 29 16  9  5

Difference electron density (eA^-3x100) at 15 degree intervals for AFIX 137 group
attached to C10. The center of the range is eclipsed (cis) to N6 and rotation is
clockwise looking down C9 to C10
  50 47 39 28 19 15 20 30 38 41 39 37 34 29 25 27 33 35 29 19 12 15 29 43

After local symmetry averaging:   40  41  36  28  21  20  24  33
```

It will be seen that the hydroxyl hydrogen is very clearly defined, but that the methyl group is rotating fairly freely (low potential barrier). After three-fold averaging, however, there is a single difference electron density maximum. The (squashed) least-squares refinement output follows:

```
Least-squares cycle 1   Maximum vector length=511  Memory required=1836/136080

wR2 =  0.1130 before cycle   1 for   1296 data and  105 /  105 parameters

GooF = S =     1.140;     Restrained GooF =      1.140  for      0 restraints

Weight = 1/[sigma^2(Fo^2)+(0.0600*P)^2+0.15*P] where P=(Max(Fo^2,0)+2*Fc^2)/3

   N     value      esd      shift/esd  parameter

   1    0.97891    0.00384   -10.702     OSF
   2    0.04044    0.00261    -7.494    FVAR  2
   3    0.07317    0.00394     0.805    FVAR  3
   4    0.01781    0.00946     1.777    EXTI

Mean shift/esd =   0.747    Maximum = -10.702 for FVAR  2

Max. shift = 0.028 A for H10A     Max. dU =-0.020 for H5A

  .......... etc (cycles 2 and 3 omitted) .........

Least-squares cycle 4  Maximum vector length = 511 Memory required =1836/136080

wR2 =  0.1035 before cycle   4 for   1296 data and  105 /  105 parameters

GooF = S =     1.016;     Restrained GooF =      1.016  for      0 restraints

Weight = 1/[sigma^2(Fo^2)+(0.0600*P)^2+0.15*P] where P=(Max(Fo^2,0)+2*Fc^2)/3

   N     value       esd     shift/esd  parameter
   1    0.97902    0.00358    -0.003     OSF
   2    0.03605    0.00176     0.012    FVAR  2
   3    0.07345    0.00376    -0.031    FVAR  3
   4    0.02502    0.01081    -0.010    EXTI

Mean shift/esd =   0.008    Maximum =  -0.244 for tors H10A

Max. shift = 0.004 A for H10A       Max. dU = 0.000 for H2


Largest correlation matrix elements

 0.509 U12 O2 / U22 O2    0.507 U12 O3 / U11 O3
 0.509 U12 O2 / U11 O2    0.500 U12 O3 / U22 O3


Idealized hydrogen atom generation before cycle   5

Name     x       y       z    AFIX  d(X-H)  shift  Bonded   Conformation
                                                    to     determined by
H2   -0.6017  0.2095  0.8832  147   0.820   0.000   O2        C5  H2
H5A  -0.2721  0.0676  0.9001   23   0.970   0.000   C5        O2  C7
H5B  -0.2964  0.1554  1.0576   23   0.970   0.000   C5        O2  C7
H6A   0.3572  0.1389  0.4085   93   0.860   0.000   N6        C9  C8
H6B   0.3073  0.1559  0.2347   93   0.860   0.000   N6        C9  C8
H7   -0.3331  0.4598  0.8575   13   0.980   0.000   C7        O3  C5  C11
H10A -0.0176  0.2947  0.1525  137   0.960   0.000   C10       C9  H10A
H10B -0.2042  0.4192  0.2692  137   0.960   0.000   C10       C9  H10A
H10C -0.1764  0.2036  0.2964  137   0.960   0.000   C10       C9  H10A
H11A -0.3575  0.2948  0.6198   23   0.970   0.000   C11       C8  C7
H11B -0.3198  0.4943  0.5737   23   0.970   0.000   C11       C8  C7
```

The final structure factor calculation, analysis of variance etc. produces the following edited output:

```
Final Structure Factor Calculation for  SIGI in P-1
Total number of l.s. parameters = 105    Maximum vector length =  511
```

```
wR2 =  0.1035 before cycle 5 for 1296 data and    0 /  105 parameters

GooF = S =   1.016;     Restrained GooF =      1.016  for    0 restraints

Weight = 1/[sigma^2(Fo^2)+(0.0600*P)^2+0.15*P] where P=(Max(Fo^2,0)+2*Fc^2)/3
R1 = 0.0364 for   1189 Fo > 4.sigma(Fo)  and  0.0397 for all   1296 data
wR2 =  0.1035,  GooF = S = 1.016,  Restrained GooF =    1.016  for all data

Occupancy sum of asymmetric unit = 11.00 for non-hydrogen and 11.00 for
hydrogen atoms.


Principal mean square atomic displacements U

   0.0504   0.0254   0.0188   O1
   0.0492   0.0229   0.0189   O2
   0.0513   0.0194   0.0165   O3
   0.0326   0.0208   0.0159   C4
   0.0376   0.0204   0.0190   C5
   0.0439   0.0319   0.0214   N6
   0.0329   0.0201   0.0185   C7
   0.0276   0.0190   0.0181   C8
   0.0289   0.0220   0.0191   C9
   0.0493   0.0352   0.0181   C10
   0.0353   0.0215   0.0183   C11

Analysis of variance for reflections employed in refinement
K = Mean[Fo^2] / Mean[Fc^2]  for group

Fc/Fc(max)    0.000 0.009 0.017 0.027 0.038 0.049 0.065 0.084 0.110 0.156 1.0

Number in group   135.  125.  131.  139.  119.  132.  131.  128.  131.  126.

           GooF  1.034 1.000 1.085 1.046 1.093 0.999 0.937 0.995 1.027 0.931

           K     1.567 1.127 0.964 1.023 1.008 0.992 0.997 0.998 1.008 1.010


Resolution(A)  0.84  0.88  0.90  0.95  0.99  1.06 1.14  1.25  1.44  1.79  inf

Number in group   136.  127.  128.  128.  136.  124.  128.  130.  130.  129.

           GooF  0.978 0.881 0.854 0.850 0.850 0.921 0.874 1.088 1.242 1.434

           K     1.024 1.013 1.017 0.990 0.991 0.989 1.013 0.995 1.037 1.004

           R1    0.061 0.049 0.050 0.046 0.034 0.034 0.031 0.039 0.038 0.037

Recommended weighting scheme:  WGHT   0.0545   0.1549
```

The analysis of variance should be examined carefully for indications of systematic errors. If the *Goodness of Fit* (GooF) is significantly higher than unity and the scale factor K is appreciably lower than unity in the extreme right columns in terms of both *F* and resolution, then an extinction parameter should be refined (the program prints a warning in such a case). This does not show here because an extinction parameter is already being refined. The scale factor is a little high for the weakest reflections in this example; this may well be a statistical artifact and may be ignored (selecting the groups on $F_c$ will tend to make $F_o^2$ greater than $F_c^2$ for this range). The increase in the GooF at low resolution (the 1.79 to infinity range) is caused in part by systematic errors in the model such as the use of scattering factors based on spherical atoms which ignore bonding effects, and is normal for purely light-atom structures (this interpretation is confirmed by the fact that difference electron density peaks are found in the middle of bonds). In extreme cases the lowest or highest resolution ranges can be conveniently suppressed by means of the SHEL instruction; this is normal practice in macromolecular refinements, but refining a diffuse solvent model with SWAT may be better, inadequate solvent modeling for macromolecules produces similar symptoms to lack of extinction refinement for small molecules.

The weighting scheme suggested by the program is designed to produce a flat analysis of variance in terms of $F_c$, but makes no attempt to fit the resolution dependence of the GooF. It is also written to the end of the *.res* file, so that it is easy to update it before the next job. In the early stages of refinement it is better to retain the default scheme of WGHT 0.1; the updated parameters should not be incorporated in the next *.ins* file until all atoms have been found and at least the heavier atoms refined anisotropically.

The list of most disagreeable reflections and tables of bond lengths and angles (BOND $H - omitted here) and torsion angles (CONF) are followed by the HTAB (hydrogen bonds) and MPLA (least-squares planes) tables:

```
 Selected torsion angles

  -175.08 ( 0.12)  C7 - O3 - C4 - O1
     5.73 ( 0.15)  C7 - O3 - C4 - C8
   109.69 ( 0.12)  C4 - O3 - C7 - C5
   -11.65 ( 0.15)  C4 - O3 - C7 - C11
   171.12 ( 0.10)  O2 - C5 - C7 - O3
   -72.04 ( 0.15)  O2 - C5 - C7 - C11
    -1.46 ( 0.24)  O1 - C4 - C8 - C9
   177.61 ( 0.12)  O3 - C4 - C8 - C9
  -176.27 ( 0.14)  O1 - C4 - C8 - C11
     2.80 ( 0.16)  O3 - C4 - C8 - C11
     3.08 ( 0.22)  C4 - C8 - C9 - N6
   176.93 ( 0.13)  C11 - C8 - C9 - N6
  -177.23 ( 0.13)  C4 - C8 - C9 - C10
    -3.39 ( 0.22)  C11 - C8 - C9 - C10
   176.05 ( 0.13)  C9 - C8 - C11 - C7
    -9.39 ( 0.14)  C4 - C8 - C11 - C7
    12.37 ( 0.14)  O3 - C7 - C11 - C8
  -104.74 ( 0.13)  C5 - C7 - C11 - C8


 Specified hydrogen bonds (with esds except fixed and riding H)

  D-H          H...A        D...A        <(DHA)
  0.86         2.23         2.8486(18)   129.3        N6-H6A...O1
  0.82         2.04         2.8578(16)   174.0        O2-H2...O1_$1
  0.86         2.17         2.9741(19)   155.1        N6-H6B...O2_$2


Least-squares planes (x,y,z in crystal coordinates) and deviations from them
```

```
(* indicates atom used to define plane)

 2.3443 (0.0044) x + 7.4105 (0.0042) y - 0.0155 (0.0053) z = 1.9777 (0.0044)

*   -0.0743 (0.0008)  C7
*    0.0684 (0.0008)  C11
*   -0.0418 (0.0009)  C8
*   -0.0062 (0.0008)  C4
*    0.0538 (0.0008)  O3
    -0.0061 (0.0020)  O1
    -0.0980 (0.0028)  N6
    -0.0562 (0.0023)  C9
    -0.0314 (0.0030)  C10

Rms deviation of fitted atoms =   0.0546


 2.5438 (0.0040) x + 7.3488 (0.0040) y - 0.1657 (0.0042) z = 1.8626 (0.0026)

Angle to previous plane (with approximate esd) =  2.45 ( 0.07 )

*    0.0054 (0.0008)  C10
*    0.0082 (0.0008)  N6
*   -0.0052 (0.0012)  C9
*   -0.0337 (0.0012)  C8
*    0.0135 (0.0008)  C11
*    0.0118 (0.0009)  C4
     0.0568 (0.0019)  O1
     0.0214 (0.0018)  O3
    -0.1542 (0.0020)  C7

Rms deviation of fitted atoms =   0.0162


Hydrogen bonds with  H..A < r(A) + 2.000 Angstroms  and  <DHA > 110 deg.

D-H           d(D-H)   d(H..A)   <DHA    d(D..A)   A
O2-H2          0.820    2.041   174.05    2.858    O1 [ x-1, y, z ]
N6-H6A         0.860    2.225   129.29    2.849    O1
N6-H6B         0.860    2.172   155.06    2.974    O2 [ x+1, y, z-1 ]
```

All esds printed by the program are calculated rigorously from the full covariance matrix, except for the esd in the angle between two least-squares planes, which involves some approximations. The contributions to the esds in bond lengths, angles and torsion angles also take the errors in the unit-cell parameters (as input on the ZERR instruction) rigorously into account; an approximate treatment is used to obtain the (rather small) contributions of the cell errors to the esds involving least-squares planes.

There follows the difference electron density synthesis and line printer 'plot' of the structure and peaks. The highest and lowest features are 0.27 and -0.17 eA$^{-3}$ respectively, and the rms difference electron density is 0.04. These values confirm that the treatment of the hydrogen atoms was adequate, and are indeed typical for routine structure analysis of small organic molecules. This output is too voluminous to give here, and indeed users of the Siemens SHELXTL molecular graphics program XP will almost always suppress it by use of the default option of a positive number on the PLAN instruction, and employ interactive graphics instead for analysis of the peak list.

# 4. Constraints and Hydrogen Atoms

## 4.1 *Constraints* versus *restraints*

In crystal structure refinement, there is an important distinction between a *constraint* and a *restraint*. A constraint is an exact mathematical condition that enables one or more least-squares variables to be expressed exactly in terms of other variables or constants, and hence eliminated. An example is the fixing of the x, y and z coordinates of an atom on an inversion center. A *restraint* takes the form of additional information that is not exact but is subject to a probability distribution; for example two chemically but not crystallographically equivalent bonds could be restrained to be approximately equal. A restraint is treated as an extra experimental observation, with an appropriate esd that determines its weight relative to the X-ray data. An excellent account of the use of constraints and restraints to control the refinement of difficult structures has been given by Watkin (1994).

Often there is a choice between constraints and restraints. For example, in a triphenylphosphine complex of a heavy element, the light atoms will be less well determined from the X-ray data than the heavy atoms. In SHELX-76 a rigid group *constraint* was often applied to the phenyl groups in such cases: the phenyl groups were treated as rigid hexagons with C-C bond lengths of 1.39 Å. This introduces a slight bias (e.g. in the P-C bond length), because the *ipso*-angle should be a little smaller than 120º. In SHELXL such rigid group constraints may still be used, but it is more realistic to apply FLAT and SADI (or SAME) *restraints* so that the phenyl groups are planar and have mm2 ($C_{2v}$) symmetry, subject to suitable esds. In addition, the phenyl groups may be restrained to have similar geometries to one another.

## 4.2 Free variables, occupancy and isotropic U-constraints

SHELXL employs the concept of *free variables* exactly as in SHELX-76. A free variable is a refinable parameter that can be used to impose a variety of additional linear constraints, e.g. to atomic coordinates, occupancies or displacement parameters. Starting values for all free variables are supplied on the FVAR instruction. Since the first FVAR parameter is the (*F*-relative) overall scale factor, there is no free variable 1. If an atom parameter is given a value greater than 15 or less than -15, it is interpreted as a reference to a free variable. A positive value (10*k*+*p*) is decoded as *p* times free variable number *k* [*fv*(*k*)], and a negative value (i.e. *k* and *p* both negative) is decoded as *p* times [*fv*(–*k*)–1]. This appears more complicated than it is in practice: for example to assign a common occupancy parameter to describe a two component disorder, the occupancies of all atoms of one component can be replaced by 21, and the occupancies of all atoms of the second component by -21, where the starting value for the occupancy is the second FVAR parameter. A further disorder, not correlated with the first, would then use free variable number 3 and codes 31 and -31 etc. If there are more than two components of a disordered atom or group, it is necessary to apply a restraint (SUMP) to the free variables used to represent the occupancies.

Free variables may be used to constrain the isotropic U-values of chemically similar hydrogen atoms to be the same; for example one could use the fourth FVAR parameter and code 41 for all methyl hydrogens (which tend to have larger U-values), and the fifth FVAR parameter and code 51 for the rest. An alternative way to constrain hydrogen isotropic displacement parameters is to replace the U-value on the atom instruction by a code *q* between -0.5 and -5; the U-value is then calculated as |*q*| times the (equivalent) isotropic U of the last atom not treated in this way (usually the carbon or other atom on which the hydrogen rides). Typical q values are -1.5 for methyl and hydroxyl hydrogens and -1.2 for others.

### 4.3 Special position constraints

Constraints for the coordinates and anisotropic displacement parameters for atoms on special positions are generated automatically by the program for ALL special positions in ALL space groups, in conventional settings or otherwise. For upwards compatibility with SHELX-76, free variables may still be used for this, but it is better to leave it to the program. If the occupancy is not input, the program will fix it at the appropriate value for a special position. If the user applies (correct or incorrect) special position constraints using free variables etc., the program assumes this has been done with intent and reports but does not apply the correct constraints; accidental application of wrong special position constraints is one of the easiest ways to cause a refinement to 'blow up' !

### 4.4 Atoms on the same site

For two or more atoms sharing the same site, the xyz and $U_{ij}$ parameters may be equated using the EXYZ and EADP constraints respectively (or by using 'free variables'). The occupation factors may be expressed in terms of a 'free variable' so that their sum is constrained to be constant (e.g. 1.0). If more than two different chemical species share a site, a *linear free variable restraint* (SUMP) is required to restrain the sum of occupation factors.

### 4.5 Rigid group and riding model constraints; fitting of standard fragments

The generation of idealized coordinates and geometrical constraints in the refinement are defined in SHELXL by the two-part AFIX code number (*mn*). This notation is perhaps a little too concise, but has been retained for upwards compatibility with SHELX-76, although several of the options are new. The last digit, *n*, describes the constraints to be used in the refinement, and the one or two-digit component *m* defines the starting geometry. For example AFIX 95 followed by five carbon atoms (possibly with intervening hydrogens) and then AFIX 0 means that a regular pentagon (*n*=5) should be fitted (to at least three atoms with non-zero coordinates), and then refined as a rigid group with variable overall scale (*m*=9). This could be used to refine a cyclopentadienyl ligand. Similarly AFIX 106 would be used for an idealized pentamethyl-cyclopentadienyl ligand refined as a rigid group with fixed interatomic distances. Note that riding (or restrained) hydrogens may be included in such rigid groups, and are ignored when fitting the idealized group (in contrast to SHELX-76).

A rigid group involves 6 refinable parameters: three rotation angles and three coordinates. The first atom in the group is the pivot atom about which the other atoms rotate; this is useful when it is necessary to fix its coordinates (by adding 10 in the usual way). In a variable metric rigid group (m=9) a seventh parameter is refined; this is a scale factor that multiplies all distances within the group. Any of the atoms in a rigid group may be subject to restraints, e.g.

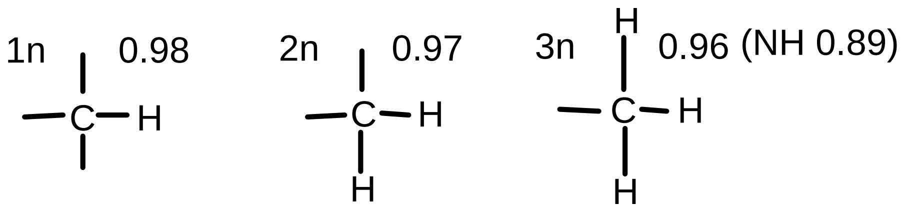
to restrain their distances to atoms not in the same rigid group (this was not allowed in SHELX-76).

A particularly useful constraint for the refinement of hydrogen atoms is the *riding model* (*n*=3):

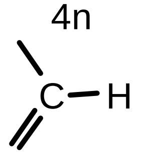
$$\mathbf{x}(\mathrm{H}) = \mathbf{x}(\mathrm{C}) + \mathbf{d}$$

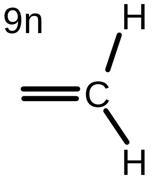
where **d** is a constant vector. Both atoms contribute to the derivative calculation and the same shifts are applied to both; the hydrogen atoms are re-idealized after each cycle (although this is scarcely necessary). The riding model constraint costs no extra parameters, and improves convergence of the refinement. SHELXL provides several variations of this riding model; for example the C-H distances (but not the XCH angles) may be allowed to refine (*n*=4; one extra parameter per group), the torsion angle of a methyl or hydroxyl group may be refined (*n*=7), or these two options may be combined (*n*=8).

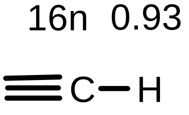
Fragments of known geometry may be fitted to target atoms (e.g. from a previous Fourier peak search), and the coordinates generated for any missing atoms. Four standard groups are available: regular pentagon (*m*=5), regular hexagon (*m*=6), naphthalene (*m*=11) and pentamethylcyclopentadienyl (*m*=10); any other group may be used simply by specifying orthogonal or fractional coordinates in a given cell (AFIX *mn* with *m*>16 and FRAG...FEND). This is usually, but not always, followed by rigid group refinement.



All H-C-X angles equal, H-C-H depends on X-C-X for AFIX 2n, tetrahedral for methyl groups.



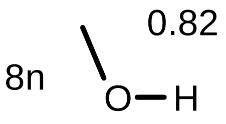




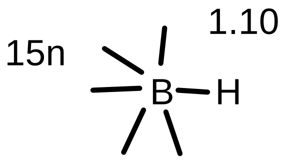
External bisector

HCH = 120 deg.

C-H 0.93 and N-H 0.86 for 4n and 9n.



XOH tetrahedral
(torsion for best H-bond)



B-H on minus
sum of unit vectors
to other atoms

## 4.6 Hydrogen atom generation and refinement

It is difficult to locate hydrogen atoms accurately using X-ray data because of their low scattering power, and because the corresponding electron density is smeared out, asymmetrical, and is not centered at the position of the nucleus. In addition hydrogen atoms tend to have larger librational amplitudes than other atoms. For most purposes it is preferable to calculate the hydrogen positions according to well-established geometrical criteria and then to adopt a refinement procedure which ensures that a sensible geometry is retained. The above table summarizes the options for generating hydrogen atoms; the hydrogen coordinates are re-idealized before each cycle. The distances given in this table are the values for room temperature, they are increased by 0.01 or 0.02 Å for low temperatures (specified by the TEMP instruction) to allow for the smaller librational correction at low temperature.

## 4.7 Special facilities for -$CH_3$ and -OH groups

Methyl and hydroxyl groups are difficult to position accurately (unless neutron data are available!). If good (low-temperature) x-ray data are available, the method of choice is HFIX 137 for -$CH_3$ and HFIX 147 for -OH groups; in this approach, a difference electron density synthesis is calculated around the circle which represents the locus of possible hydrogen positions (for a fixed X-H distance and Y-X-H angle). The maximum electron density (in the case of a methyl group after local threefold averaging) is then taken as the starting position for the hydrogen atom(s). In subsequent refinement cycles (and in further least-squares jobs) the hydrogens are re-idealized at the start of each cycle, but the current torsion angle is retained; the torsion angles are allowed to refine whilst keeping the X-H distance and Y-X-H angle fixed (*n*=7). If unusually high quality data are available, AFIX 138 would allow the refinement of a common C-H distance for a methyl group but not allow the group to tilt; a variable metric rigid group refinement (AFIX 9 for the carbon followed by AFIX 135 before the first hydrogen) would allow it to tilt as well, but still retain tetrahedral H-C-H angles and equal C-H distances within the group.

If the data quality is less good, then the refinement of torsion angles may not converge very well. In such cases the hydrogens can be positioned geometrically and refined using a riding model by HFIX 33 for methyl and HFIX 83 for hydroxyl groups. This staggers the methyl groups, and -OH groups attached to saturated carbons, as well as possible; -OH groups attached to aromatic rings are tested in one of the two positions with one hydrogen in the plane. In both cases the choice of hydrogen position is then determined by best hydrogen bond (to an N, O, Cl or F atom) that can be created. For disordered methyl groups (with two sites rotated by 60 degrees from one another) HFIX 123 is recommended, possibly with refinement of the corresponding site occupation factors via a 'free variable' so that their sum is unity (e.g. 21 and -21).

The choice of a suitable (default) O-H distance is very difficult. O-H internuclear distances for isolated molecules in the gas phase are about 0.96 Å (cf. 1.10 for C-H), but the appropriate distance to use for X-ray diffraction must be appreciably shorter to allow for the displacement of the center of gravity of the electron distribution towards the oxygen atom, and also for librational effects. Although the (temperature dependent) value assumed by the program fits reasonably well for O-H groups in predominantly organic molecules, appreciably longer O-H distances are appropriate for low temperature studies of strongly (cooperatively) hydrogen-bonded systems; short H...O distances are always associated with long O-H distances. If there are many such O-H groups and good quality data are available, HFIX 88 (or 148) plus SADI restraints to make all the O-H distances approximately equal (with an esd of say 0.02) is a good approach.

## 4.8 Further peculiarities involving hydrogen atoms

Hydrogen atoms are identified as such by their scattering factor numbers, which must correspond to a SFAC name H (or $H). The special treatment of hydrogens does not apply if they reference a different SFAC name (e.g. D !). Other elements that need to be specifically identified (e.g. so that HFIX 43 can use different default C-H and N-H distances) are defined similarly. However for the output of the PLAN instruction, hydrogen atoms are identified as those atoms with a radius of less the 0.4 Å. This is not as illogical as it may sound; the PLAN output is concerned with potential hydrogen bonds etc., not with the scattering power of an atom, and SHELXL has to handle neutron as well as X-ray data.

Hydrogen atoms may also 'ride' on atoms in rigid groups (unlike SHELX-76); for example HFIX 43 could reference carbon atoms in a rigid phenyl ring. In such a case further geometrical restraints (SADI, SAME, DFIX, FLAT) are not permitted on the hydrogen atoms; this is the only exception to the general rule that any number of restraints may be applied to any atom, whatever constraints are also being applied to it.

OMIT $H (or OMIT_* $H if residues are employed) combined with L.S. 0, FMAP 2 and PLAN -100 enables an 'omit map' to be calculated, in which the hydrogen atoms are retained but do not contribute to $F_c$. If a non-zero electron density appears in the 'Peak' column for a hydrogen atom in the Fourier output, then there was an actual peak in the difference electron density synthesis within 0.31 Å of the expected hydrogen position.

Sometimes it is known that the crystal contains a deuterated solvent molecule (e.g. $CDCl_3$) because it was crystallized in an n.m.r. tube. In such a case, an element 'D' may be added after 'H' on the SFAC instruction, and the appropriate numbers of H and D in the cell specified on the UNIT instruction. This enables the formula weight and density to be calculated correctly. The H and D atoms that follow in the *.ins* file should both be given the SFAC number corresponding to H, so that they are both treated as 'hydrogens' for all other purposes.

# 5. Restraints and Disorder

A *restraint* is incorporated in the least-squares refinement as if it were an additional experimental observation; $w(yt–y)^2$ is added to the quantity $\Sigma w(F_o^2–F_c^2)^2$ to be minimized, where a quantity *y* (which is a function of the least-squares parameters) is to be restrained to a target value *yt*, and the weight *w* (for either a restraint or a reflection) is $1/\sigma^2$. In the case of a reflection, $\sigma^2$ is estimated using a weighting scheme; for a restraint σ is simply the effective standard deviation. In SHELXL the restraint weights are multiplied by the mean value of $w(F_o^2–F_c^2)^2$ for the reflection data, which allows for the possibility that the reflection weights may be relative rather than absolute, and also gives the restraints more influence in the early stages of refinement (when the Goodness of Fit is invariably much greater than unity), which improves convergence. It is possible to use Brünger's $R_{free}$ test (Brünger, 1992) to fine-tune the restraint esds. In practice the optimal restraint esds vary little with the quality and resolution of the data, and the standard values (assumed by the program if no other value is specified) are entirely adequate for routine refinements. Default values for the various classes of restraint may be also set with DEFS instructions; there may be several DEFS instructions in the same .ins file: each applies to all restraints encountered before the next DEFS instruction (or the end of the file).

### 5.1 Floating origin restraints

Floating origin restraints are generated automatically by the program as and when required by the method of Flack & Schwarzenbach (1988), so the user should not attempt to fix the origin in such cases by fixing the coordinates of a heavy atom. These floating origin restraints effectively fix the X-ray 'center of gravity' of the structure in the polar axis direction(s), and lead to smaller correlations than fixing a single atom in structures with no dominant heavy atom. Floating origin restraints are not required (and will not be generated by the program) when CGLS refinement is performed.

### 5.2 Geometrical restraints

A particularly useful restraint is to make chemically but not crystallographically equivalent distances equal (subject to a given or assumed esd) without having to invent a value for this distance (SADI). The SAME instruction can generate SADI restraints automatically, e.g. when chemically identical molecules or residues are present. This has the same effect as making equivalent bond lengths and angles but not torsion angles equal (see also section 5.5).

The FLAT instruction restrains a group of atoms to lie in a plane (but the plane is free to move and rotate); the program achieves this by treating the restraint as a sum of chiral volume restraints with zero target volumes. Thus the restraint esd has units of $Å^3$. For comparison with other methods, the r.m.s. deviation of the atoms from their restraint planes is also calculated.

DFIX and DANG restrain distances to target values. DANG was introduced so that the default sigma for 1,3-distances could be made twice that for 1,2-distances (the first DEFS parameter). The DANG restraints are applied in exactly the same way as DFIX, but are also listed separately in the restraints summary tables.

CHIV restrains the *chiral volume* of an atom that makes three bonds; the chiral volume is the volume of the 'unit-cell' (i.e. parallelopiped) whose axes are represented by these three bonds. In the SHELXL-96, the sign of the chiral volume is determined by the alphabetical (ASCII) order of the atoms, rather than the order in the connectivity list (which caused some confusion in SHELXL-93).

When 'free variables' are used as the target values for DFIX, DANG and CHIV restraints, it is possible to restrain different distances etc. to be equal and to refine their mean value (for which an esd is thus obtained). ALL types of geometrical restraint may involve ANY atom, even if it is part of a rigid group or a symmetry equivalent generated using EQIV $n and referenced by _$n, except for hydrogen atoms which ride on rigid group atoms.

### 5.3 Anti-bumping restraints

Anti-bumping restraints are usually only necessary for lower resolution structures, e.g. of macromolecules. They may be applied individually, by means of DFIX distance restraints with the distance given as a negative number, or generated automatically by means of the BUMP instruction. In combination with the SWAT instruction for diffuse solvent, BUMP provides a very effective way of handling solvent water in macromolecules, and is also useful in preventing unreasonably close contacts between protein molecules.

DFIX restraints with negative distance d are ignored if the two atoms are further from one another than |d| in the current refinement cycle; if they are closer than |d|, a restraint is applied to increase the distance to |d| with the given (or assumed) esd. The automatic generation of anti-bumping restraints includes all possible symmetry equivalents, and has been substantially enhanced since the 1993 version of SHELXL. PART numbers are taken into account, and anti-bumping restraints are not applied if the sum of the occupancies of the two atoms is less than 1.1.

BUMP applies to all pairs of non-hydrogen atoms, provided that they are not linked by three or fewer bonds in the connectivity array. In addition, anti-bumping restraints are generated for all pairs of unreasonably close hydrogen atoms that are not bonded to the same atom. This discourages energetically unfavorable side-chain rotamers. If the BUMP esd is given as negative, the symmetry equivalents of bonds in the connectivity array are taken into account in applying the above rules, otherwise all short distances to symmetry generated atoms are potentially repulsive. The (default) positive esd action is usually the appropriate action for macromolecules, and prevents symmetry equivalents of one side-chain wandering too close to another, irrespective of whether spurious bonds between them have been (automatically) generated in the connectivity array. In contrast to SHELXL-93, the anti-bumping restraints are now regenerated each cycle.

The BUMP instruction also outputs a list of bonds and 1,3-distances in the connectivity array that have not been restrained in any way; this is a good way to detect spurious bonds and errors and omissions in the restraints. In some cases the lack of restraints is of course intentional, in which case the warnings can be ignored (e.g. for bonds involving metal atoms in a protein).

## 5.4 Restraints on anisotropic displacement parameters

Three different types of restraint may be applied to $U_{ij}$ values. DELU applies a *rigid-bond* restraint to $U_{ij}$-valus of two bonded (or 1,3-) atoms; the anisotropic displacement components of the two atoms along the line joining them are restrained to be equal. This restraint was suggested by Rollett (1970), and corresponds to the rigid-bond criterion for testing whether anisotropic displacement parameters are physically reasonable (Hirshfeld, 1976; Trueblood & Dunitz, 1983). Didisheim & Schwarzenbach (1987) have shown that in many but not all cases, rigid-bond restraints are equivalent to the TLS description of rigid body motion in the limit of zero esds; however this requires that (almost) all atom pairs are restrained in this way, which for molecules with conformational flexibility is unlikely to be appropriate. An extensive study (Irmer, 1990) has shown that the rigid bond condition is fulfilled within the experimental error for routine X-ray studies of bonds and 1,3-distances between two first-row elements (B to F inclusive), and so may be applied as a 'hard' restraint (low esd). A rigid bond restraint is not suitable for systems with unresolved disorder, e.g. $AsF_6^-$ anions and dynamic Jahn-Teller effects, although its failure may be useful in detecting such effects.

Isolated (e.g. solvent water) atoms may be restrained to be approximately isotropic, e.g. to prevent them going 'non-positive-definite'; this is a rough approximation and so should be applied as a 'soft' restraint with a large esd (ISOR). Similarly the assumption of 'similar' $U_{ij}$ values for spatially adjacent atoms (SIMU) causes the thermal ellipsoids to increase and change direction gradually going along a side-chain in a polypeptide, but this treatment is approximate and thus also appropriate only for a soft restraint; it is also useful for partially overlapping atoms of disordered groups. A simple way to apply SIMU to all such overlapping atoms (but not to others) is to give a SIMU instruction with no atoms (i.e. all atoms implied) and the third number set to a distance less than the shortest bond; additional SIMU restraints may be included in the same job. The default SIMU esd of 0.04 Å$^2$ is intended for anisotropic displacement parameters; SIMU may also be used for isotropic parameters (e.g. for refinement of a protein against 2 Å data) but in that slightly larger esd's, say 0.1 Å$^2$, might be more appropriate.

SHELXL does not permit DELU, SIMU and ISOR restraints to reference symmetry generated atoms, although this is allowed for all geometrical restraints. To permit such references for displacement parameter restraints as well would considerably complicate the program, and is rarely required in practice.

## 5.5 Non-crystallographic symmetry restraints

The new NCSY instruction provides a way of imposing *local non-crystallographic symmetry*. This is a very powerful restraint that holds remarkably well for many macromolecules, and it should be used whenever possible, especially when the resolution is not very high. The use of such restraints is slower than using NCS constraints (which involve performing a structure factor summation over just part of the structure, extending it to the whole structure by matrix operations) but has the advantage that no transformation matrix or real-space mask is required. The restraints make equivalent 1,4-distances (defined using the connectivity array) equal, and the isotropic *U*-values of equivalent atoms equal. Either of these restraints may be switched off, and any number of NCS domains may be defined. 1,2- and 1,3-distances are usually restrained using DFIX, DANG, SADI or SAME, so NCSY does not apply to them. The atoms to which NCS is applied are defined in a simple and flexible manner, so it is possible for example to leave out side-chains that deviate from NCS because they are involved in interaction with other (non-NCS related) molecules.

## 5.6 Shift limiting restraints

*Shift limiting restraints* (Watkin, 1994) may be applied in SHELXL by the Marquardt (1963) algorithm. Terms proportional to a 'damping factor' (the first parameter on the DAMP instruction) are added to the least-squares matrix before inversion. Shift limiting restraints are particularly useful in the refinement of structures with a poor data to parameter ratio, and for pseudosymmetric problems. The 'damping factor' should be reduced towards the end of the refinement, otherwise the least-squares estimates of the esds in the less well determined parameters will be too low (the program does however make a first order correction to the esds for this effect). The shifts are also scaled down if the maximum shift/esd exceeds the second DAMP parameter. In addition, if the actual and target values for a particular restraint differ by more than 100 times the given esd, the program will temporarily increase the esd to limit the influence of this restraint to that produced by a discrepancy of 100 times the esd. This helps to prevent a bad initial model and tight restraints from causing dangerously large shifts in the first cycle.

## 5.7 Restraints on linear combinations of free variables

Constraints may be applied to atom coordinates, occupation and displacement parameters, and to restrained distances (DFIX) and chiral volumes (CHIV), by the use of free variables. Linear combinations of free variables may in turn be restrained (SUMP). This provides a way of restraining the sum of the occupancies of a multi-component disorder to be (say) unity and of restraining the occupancies to fit the charge balance and chemical analysis of a mineral with several sites occupied by a mixture of cations. In the latter case, the atoms occupying the same site will also usually be constrained (using EXYZ and EADP) to have the same positional and displacement parameters.

## 5.8 Examples of restraints and constraints

A major advantage of applying chemically reasonable restraints is that a subsequent difference electron density synthesis is often more revealing, because the parameters were not allowed to 'mop up' any residual effects. The refinement of pseudosymmetric structures, where the X-ray data may not be able to determine all of the parameters, is also considerably facilitated, at the cost of making it much easier to refine a structure in a space group of unnecessarily low symmetry !

By way of example, assume that the structure contains a cyclopentadienyl (Cp) ring $\pi$-bonded to a metal atom, and that as a result of the high thermal motion of the ring only three of the atoms could be located in a difference electron density map. We wish to fit a regular pentagon (default C-C 1.42 Å) in order to place the remaining two atoms, which are input as dummy atoms with zero coordinates. Since the C-C distance is uncertain (there may well be an appreciable librational shortening in such a case) we refine the $C_5$-ring as a *variable metric rigid group*, i.e. it remains a regular pentagon but the C-C distance is free to vary. In SHELXL this may all be achieved by inserting one instruction (AFIX 59) before the five carbons and one (AFIX 0) after them:

```
AFIX 59                   ! AFIX mn with m = 5 to fit pentagon (default C-C
C1 1 .6755 .2289 .0763    ! 1.42 A) and n = 9 for v-m rigid-group refinement
C2 1 .7004 .2544 .0161
C3 1 0 0 0                ! the coordinates for C3 and C4 are obtained by the
C4 1 0 0 0                ! fit of the other 3 atoms to a regular pentagon
C5 1 .6788 .1610 .0766
AFIX 0                    ! terminates rigid group
```

Since $U_{ij}$ values were not specified, the atoms would refine isotropically starting from U = 0.05. To refine with anisotropic displacement parameters in the same or a subsequent job, the instruction:

```
ANIS C1 > C5
```

should be inserted anywhere before C1 in the '.ins' file. The SIMU and ISOR restraints on the $U_{ij}$ would be inappropriate for such a group, but:

```
DELU C1 > C5
```

could be applied if the anisotropic refinement proved unstable. The five hydrogen atoms could be added and refined with the 'riding model' by means of:

```
HFIX 43 C1 > C5
```

anywhere before C1 in the input file. For good data, in view of possible librational effects, a suitable alternative would be:

```
HFIX 44 C1 > C5
SADI 0.02 C1 H1 C2 H2 C3 H3 C4 H4 C5 H5
```

which retains a riding model but allows the C-H bond lengths to refine, subject to the restraint that they should be equal within about 0.02 Å.

In analogous manner it is possible to generate missing atoms and perform rigid group refinements for phenyl rings (AFIX 66) and Cp* groups (AFIX 109). Very often it is possible and desirable to remove the rigid group constraints (by simply deleting the AFIX instructions) in the final stages of refinement; there is good experimental evidence that the *ipso*-angles of phenyl rings differ systematically from 120° (Jones, 1988; Maetzke & Seebach, 1989; Domenicano, 1992).

As a second example, assume that the structure contains two molecules of poorly defined THF solvent, and that we have managed to identify the oxygen atoms. A rigid pentagon would clearly be inappropriate here, except possibly for placing missing atoms, since THF molecules are not planar. However we can *restrain* the 1,2- and the 1,3-distances in the two molecules to be similar by means of a 'similarity restraint' (SAME). Assume that the molecules are numbered O11 C12 ... C15 and O21 C22 ... C25, and that the atoms are given in this order in the atom list. Then we can either insert the instruction:

```
SAME O21 > C25
```

before the first molecule, or:

```
SAME O11 > C15
```

before the second. These SAME instructions define a group of five atoms that are considered to be the same as the five (non-hydrogen) atoms which immediately follow the SAME instruction. The entries in the connectivity table for the latter are used to define the 1,2- and 1,3-distances, so the SAME instruction should be inserted before the group with the best geometry. This one SAME instruction restrains five pairs of 1,2- and five pairs of 1,3-distances to be nearly equal, i.e.

```
d(O11-C12) = d(O21-C22),  d(C12-C13) = d(C22-C23),  d(C13-C14) = d(C23-C24),
d(C14-C15) = d(C24-C25),  d(C15-O11) = d(C25-O21),  d(O11-C13) = d(O21-C23),
d(C12-C14) = d(C22-C24),  d(C13-C15) = d(C23-C25),  d(C14-O11) = d(C24-O21),
and  d(C15-C12) = d(C25-C22).
```

In addition, it would also be reasonable to restrain the distances on opposite sides of the same ring to be equal. This can be achieved with one further SAME instruction in which we count the other way around the ring. For example we could insert:

```
SAME O11 C15 < C12
```

before the first ring. The symbol '<' indicates that one must count up the atom list instead of down. The above instruction is exactly equivalent to:

```
SAME O11 C15 C14 C13 C12
```

This generates 10 further restraints, but two of them [d(C13-C14) = d(C14-C13) and d(C12-C15) = d(C15-C12)] are identities and each of the others appears twice, so only four are independent and the rest are ignored. It is not necessary to add a similar instruction before the second ring, because the program also automatically generates all 'implied' restraints, i.e. restraints that can be derived by combining two existing distance restraints that refer to the same atom pair.

In contrast to other restraint instructions, the SAME instructions must be inserted at the correct positions in the atom list. These similarity restraints provide a very general and powerful way of exploiting non-crystallographic symmetry; in this example two instructions suffice to restrain the THF molecules so that they have (within an assumed standard deviation) twofold symmetry and are the same as each other. However we have not imposed planarity on the rings nor restricted any of the torsion angles.

To complicate matters, let us assume that the two molecules are two alternative conformations of a THF molecule disordered on a single site. We must then ensure that the site occupation factors of the two molecules add to unity, and that no spurious bonds linking them are added to the connectivity table. The former is achieved by employing site occupation factors of 21 (i.e. 1 times free-variable 2) for the first molecule and -21 {i.e. 1 times [1-fv(2)] } for the five atoms of the second molecule. Free variable 2 is then the occupation factor of the first molecule; its starting value must be specified on the FVAR instruction. The possibility of spurious bonds is eliminated by inserting 'PART 1' before the first molecule, 'PART 2' before the second, and 'PART 0' after it. Hydrogen atoms can be inserted in the usual way using the HFIX instruction since the connectivity table is 'correct'; they will automatically be assigned the site occupation factors of the atoms to which they are bonded.

Finally we would like to refine with anisotropic displacement parameters because the thermal motion of such solvent molecules is certainly not isotropic, but the refinement will be unstable unless we restrain the anisotropic displacement parameters to behave 'reasonably' by means of rigid bond restraints (DELU) and 'similar $U_{ij}$' restraints (SIMU); fortunately the program can set up these restraints automatically. DELU restrains the differences in the components of the displacement parameters of two atoms to zero along the 1,2- and 1,3-vector directions; these restraints are derived automatically with the help of the connectivity table. Since the SIMU restraints are much more approximate, we restrict them here to atoms which, because of the disorder, are almost overlapping (i.e. are within 0.7 Å of each other). Note that the SIMU restraints ignore the connectivity table and are based directly on a distance criterion specifically because the connectivity table does not link the disordered atoms. In order to specify a non-standard distance cut-off which is the third SIMU parameter, we must also give the first two parameters, which are the restraint esds for distances involving non-terminal atoms (0.02) and at least one terminal atom (0.04) respectively. The *.ins* file now contains:

```
HFIX 23 C12 > C15 C22 > C25
ANIS O11 > C25
DELU O11 > C25
SIMU O11 > C25 0.04 0.08 0.7
FVAR ..... 0.75
....
PART 1
SAME O21 > C25
SAME O11 C15 < C12
O11 4 ..... ..... ..... 21
C12 1 ..... ..... ..... 21
C13 1 ..... ..... ..... 21
C14 1 ..... ..... ..... 21
C15 1 ..... ..... ..... 21
PART 2
O21 4 ..... ..... ..... -21
C22 1 ..... ..... ..... -21
C23 1 ..... ..... ..... -21
C24 1 ..... ..... ..... -21
C25 1 ..... ..... ..... -21
PART 0
```

An alternative type of disorder common for THF molecules and proline residues in proteins is when one atom (say C14) can flip between two positions (i.e. it is the flap of an envelope conformation). If we assign C14 to PART 1, C14' to PART 2, and the remaining ring atoms to PART 0, then the program will be able to generate the correct connectivity, and so we can also generate hydrogen atoms for both disordered components (with AFIX, not HFIX):

```
SIMU C14 C14'
ANIS O11 > C14'
```

```
FVAR ..... 0.7
....
SAME O11 C12 C13 C14' C15
O11 4 ..... ..... .....
C12 1 ..... ..... .....
AFIX 23
H12A 2 ..... ..... .....
H12B 2 ..... ..... .....
AFIX 0
C13 1 ..... ..... .....
PART 1
AFIX 23
H13A 2 ..... ..... ..... 21
H13B 2 ..... ..... ..... 21
PART 2
AFIX 23
H13C 2 ..... ..... ..... -21
H13D 2 ..... ..... ..... -21
AFIX 0
PART 1
C14 1 ..... ..... ..... 21
AFIX 23
H14A 2 ..... ..... ..... 21
H14B 2 ..... ..... ..... 21
AFIX 0
PART 0
C15 1 ..... ..... .....
PART 1
AFIX 23
H15A 2 ..... ..... ..... 21
H15B 2 ..... ..... ..... 21
PART 2
AFIX 23
H15C 2 ..... ..... ..... -21
H15D 2 ..... ..... ..... -21
AFIX 0
C14' 1 ..... ..... ..... -21
AFIX 23
H14C 2 ..... ..... ..... -21
H14D 2 ..... ..... ..... -21
AFIX 0
PART 0
```

It will be seen that six hydrogens belong to one conformation, six to the other, and two are common to both. The generation of the idealized hydrogen positions is based on the connectivity table but also takes the PART numbers into account. These procedures should be able to set up the correct hydrogen atoms for all cases of two overlapping disordered groups. In cases of more than two overlapping groups the program will usually still be able to generate the hydrogen atoms correctly by making reasonable assumptions when it finds that an atom is 'bonded' to atoms with different PART numbers, but it is possible that there are rare examples of very complex disorder which can only be handled by using dummy atoms constrained (EXYZ and EADP) to have the same positional and displacement parameters as atoms with different PART numbers (in practice it may be easier - and quite adequate - to ignore hydrogens except on the two components with the highest occupancies !).

When the site symmetry is high, it may be simpler to apply similarity restraints using SADI or DFIX rather than SAME. For example the following three instruction sets would all restrain a perchlorate ion (CL,O1,O2,O3,O4) to be a regular tetrahedron:

```
SAME CL O2 O3 O4 O1
SADI  O1 O2  O1 O3
```

followed immediately by the atoms CL, O1... O4; the SAME restraint makes all the Cl-O bonds equal but introduces only FOUR independent restraints involving the O...O distances, which allows the tetrahedron to distort retaining only one $\bar{4}$ axis, so one further restraint must be added using SADI.

or:

```
SADI  CL O1  CL O2  CL O3  CL O4
SADI  O1 O2  O1 O3  O1 O4  O2 O3  O2 O4  O3 O4
```

or:

```
DFIX 31  CL O1  CL O2  CL O3  CL O4
DFIX 31.6330  O1 O2  O1 O3  O1 O4  O2 O3  O2 O4  O3 O4
```

in the case of DFIX, one extra least-squares variable (free variable 3) is needed, but it is the mean Cl-O bond length and refining it directly means that its esd is also obtained. If the perchlorate ion lies on a three-fold axis through CL and O1, the SADI method would require the use of symmetry equivalent atoms (EQIV $1 y, z, x and O2_$1 etc. for R3 on rhombohedral axes) so DFIX would be simpler (same DFIX instructions as above with distances involving O3 and O4 deleted) [the number 1.6330 in the above example is of course twice the sine of half the tetrahedral angle].

If you wish to test whether you have understood the full implications of these restraints, try the following problems:

(a) A C-O-H group is being refined with AFIX 87 so that the torsion angle about the C-O bond is free. How can we restrain it to make the 'best' hydrogen-bond to a specific Cl- ion, so that the H...Cl distance is minimized and the O-H...Cl angle maximized, using only one restraint instruction (it may be assumed that the initial geometry is reasonably good) ?

(b) Restrain a $C_6$ ring to an ideal chair conformation using one SAME and one SADI instruction. Hint: all 1-2, 1-3 and 1-4 distances are respectively equal for a chair conformation, which also includes a regular planar hexagon as a special case. A non-planar boat conformation does not have equal 1-4 distances. To force the ring to be non-planar, the ratio of the 1-2 and 1-3 distances would have to be restrained using DFIX and a free variable.

# 6. Refinement of Twinned Structures; Absolute Structure

A typical definition of a twinned crystal is the following: "Twins are regular aggregates consisting of crystals of the same species joined together in some definite mutual orientation" (Giacovazzo,1992). So for the description of a twin two things are necessary: a description of the orientation of the different species relative to each other (twin law) and the fractional contribution of each component. The *twin law* can be expressed as a matrix that transforms the *hkl* indices of one species into the other.

## 6.1 Twin refinement method

In SHELXL the twin refinement method of Pratt, Coyle & Ibers (1971) and Jameson, Schneider, Dubler & Oswald (1982) has been implemented. $F_c^2$ values are calculated by:

$$(F_c^2)^* = osf^2 \sum_{m=1}^{n} k_m F_{c_m}^2$$

where osf is the overall scale factor, $k_m$ is the fractional contribution of twin domain m and $F_{c_m}$ is the calculated structure factor of twin domain m. The sum of the fractional contributions $k_m$ must be unity, so (n-1) of them can be refined and $k_1$ is calculated by:

$$k_1 = 1 - \sum_{m=2}^{n} k_m$$

In SHELXL two kinds of twins are distinguished:

**(a)** For twins in which the reciprocal lattices exactly coincide (twinning by merohedry or pseudo-merohedry), the procedure is relatively simple. The command TWIN $r_{11}$ $r_{12}$ $r_{13}$ $r_{21}$ $r_{22}$ $r_{23}$ $r_{31}$ $r_{32}$ $r_{33}$ n defines the twin law. **R** as the matrix that transforms the hkl indices of one component into the other and n is the number of twin domains. **R** is applied (n-1) times; the default value of n is 2.

**(b)** In cases where only some reflections have contributions from more than one domain (non-merohedral twins or twinning by reticular merohedry) the *.hkl* file must be edited and the index transformations applied to individual contributors, which are also assigned component numbers. The code HKLF 5 is used to read in this file; no TWIN command should be used.

In both cases, starting values of the fractional contributions are input with the instruction BASF $k_2$ ... $k_n$; the $k_m$ values will be refined. Note that (in the new version of SHELXL) linear restraints may be applied to these k values by means of SUMP instructions; this can be very useful to prevent instabilities in the early stages of refinement. For this purpose $k_2$...$k_n$ are assigned parameter numbers immediately following the free variables.

## 6.2 Absolute structure

Even if determination of absolute configuration is not one of the aims of the structure determination, it is important to refine **every** non-centrosymmetric structure as the correct *absolute structure* in order to avoid introducing systematic errors into the bond lengths etc. In some cases the absolute structure will be known with certainty (e.g. proteins), but in others it has to be deduced from the X-ray data. Generally speaking, a single phosphorus or heavier atom suffices to determine an absolute structure using Cu-K$\alpha$ radiation, and with accurate high-resolution low-temperature data including Friedel opposites such an atom may even suffice for Mo-K$\alpha$.

In the course of the final structure factor calculation, the program estimates the absolute structure parameter *x* (Flack, 1983) and its esd. *x* is the fractional contribution of the inverted component of a 'racemic twin'; it should be zero if the absolute structure is correct, unity if it has to be inverted, and somewhere between 0 and 1 if racemic twinning is really present. Thus the above formulas apply with n=2 and **R** = (-1 0 0, 0 -1 0, 0 0 -1).

It is a bonus of the refinement against $F^2$ that this calculation is a 'hole in one' and doesn't require expensive iteration. A comparison of *x* with its esd provides an indication as to whether the refined absolute structure is correct or whether it has to be 'inverted'; the program prints a suitable warning should this be necessary. This attempt to refine *x* 'on the cheap' is reliable when the true value of *x* is close to zero, but may produce a (possibly severe) underestimate of *x* for structures which have to be inverted, because *x* is correlated with positional and other parameters which have not been allowed to vary. Effectively these parameters have adapted themselves to compensate for the wrong (zero) value of *x* in the course of the refinement, and need to be refined with *x* to eliminate the effects of correlation. These effects will tend to be greater when the correlation terms are greater, e.g. for polar space groups and for poor data to parameter ratios (say less than 8:1). *x* can be refined at the same time as all the other parameters using the TWIN instruction with the default matrix **R** = (-1 0 0, 0 -1 0, 0 0 -1) and BASF with one parameter (*x*); this implies racemic twinning and so is refined exactly as for other simple cases of twinning. Refinement of racemic twinning should normally only be attempted towards the end of the refinement after all non-hydrogen atoms have been located. If racemic twinning is refined in this way, the automatic calculation of the Flack x parameter in the final structure factor cycle is suppressed, since the BASF parameter is x.

For most space groups 'inversion' of the structure simply involves inserting an instruction 'MOVE 1 1 1 -1' before the first atom. Where the space group is one of the 11 enantiomorphous pairs [e.g. $P3_1$ and $P3_2$] the translation parts of the symmetry operators need to be inverted as well to generate the other member of the pair. There are seven cases for which, if the standard setting of the International Tables for Crystallography has been used, inversion in the origin does **not** lead to the inverted absolute structure. This problem was probably first described in print by Parthe & Gelato (1984) and Bernardinelli & Flack (1985), but had been investigated previously by D. Rogers (personal communication to GMS, ca. 1980).

The offending space groups and corresponding correct MOVE instructions are:

| | | | |
|---|---|---|---|
| **Fdd2** | **MOVE .25 .25 1 -1** | **$I4_1cd$** | **MOVE 1 .5 1 -1** |
| **$I4_1$** | **MOVE 1 .5 1 -1** | **$I\bar{4}2d$** | **MOVE 1 .5 .25 -1** |
| **$I4_122$** | **MOVE 1 .5 .25 -1** | **$F4_132$** | **MOVE .25 .25 .25 -1** |
| **$I4_1md$** | **MOVE 1 .5 1 -1** | | |

## 6.3 Refinement against powder data

Refinement of twinned crystals and refinement against $F^2$-values derived from powder data are similar in that several reflections with different indices may contribute to a single $F^2$ observation. For powder data this requires some small adjustments to the format of the *.hkl* file; the batch number becomes the multiplicity m, and where several reflections contribute to the same observation the multiplicity is made positive for the last reflection in the group and negative for the rest.

Although SHELXL may be useful for some high symmetry and hence reasonably well resolved powder and fibre diffraction patterns - the various restraints and constraints should be exploited in full to make up for the poor data/parameter ratio - for normal powder data a Rietveld refinement program would be much more appropriate.

For powder data the least-squares refinement fits the overall scale factor (osf$^2$ where osf is given on the FVAR instruction) times the multiplicity weighted sum of calculated intensities to $F_o^2$:

$$(F_c^2)^* = \mathrm{osf}^2\ [\ m_1\ F_{c1}^2 + m_2\ F_{c2}^2 + m_3\ F_{c3}^2 + \dots\ ]$$

where the multiplicities of the contributors are given in the place of the batch numbers in the *.hkl* file. Since it is then not possible to define batch numbers as well, BASF cannot be used with powder data.

## 6.4 Frequently encountered twin laws

The following cases are relatively common:

**(a)** Twinning by merohedry. The lower symmetry trigonal, tetragonal, hexagonal or cubic Laue groups may be twinned so that they look (more) like the corresponding higher symmetry Laue groups (assuming the c-axis unique except for cubic):

```
TWIN  0 1 0  1 0 0  0 0 -1
```

plus one BASF parameter if the twin components are not equal in scattering power. If they are equal, i.e. the twinning is perfect, as indicated by the $R_{int}$ for the higher symmetry Laue group, then the BASF instruction can be omitted and $k_1$ and $k_2$ are fixed at 0.5.

**(b)** Orthorhombic with **a** and **b** approximately equal in length may emulate tetragonal:

```
TWIN  0 1 0  1 0 0  0 0 -1
```

plus one BASF parameter for unequal components.

**(c)** Monoclinic with beta approximately 90° may emulate orthorhombic:

```
TWIN  1 0 0  0 -1 0  0 0 -1
```

plus one BASF parameter for unequal components.

**(d)** Monoclinic with **a** and **c** approximately equal and beta approximately 120 degrees may emulate hexagonal [$P2_1/c$ would give absences and possibly also intensity statistics corresponding to $P6_3$]. There are three components, so n must be specified on the TWIN instruction and the matrix is applied once to generate the indices of the second component and twice for the third component. In German this is called a 'Drilling' as opposed to a 'Zwilling' (with two components):

```
TWIN  0 0 1  0 1 0  -1 0 -1  3
```

plus TWO BASF parameters for unequal components.  If the data were collected using an hexagonal cell, then an HKLF matrix would also be required to transform them to a setting with b unique:

```
HKLF  4  1  1 0 0  0 0 1  0 -1 0
```

**(e)** Rhombohedral obverse/reverse twinning on hexagonal axes.

```
TWIN -1 0 0  0 -1 0  0 0 1
```

## 6.5  Combined general and racemic twinning

If general and racemic twinning are to be refined simultaneously, n (the last parameter on the TWIN instruction) should be doubled and given a negative sign, and there should be |n|-1 BASF twin component factors (or none, in the unlikely event that all are to be fixed as equal). The inverted components follow those generated using the TWIN matrix, in the same order. Sometimes it is necessary to use this approach to distinguish between possible twin laws for non-centrosymmetric structures, when they differ only in an inversion operator In a typical example (an organocesium compound), when the TWIN instruction was input as:

```
TWIN 0 1 0  1 0 0  0 0 -1  -4
```

The BASF parameters refined to:

```
BASF 0.33607 0.00001 0.00455
```

Which means that the last two components (the ones involving inversion) can be ignored, and the final refinement performed with the '-4' deleted from the end of the TWIN instruction, and a single BASF parameter. The introduction of twinning reduced the *R*1-value from 18% to 1.8% in this example. Note that the program does not allow the BASF parameters to become negative, since this would be physically meaningless (this explains the 0.00001 above).

## 6.6 Processing of twinned and powder data

The HKLF 5 and 6 instructions force MERG 0, i.e. neither a transformation of reflection indices into a standard form nor a sort-merge is performed before refinement. If twinning is specified using the TWIN instruction, any MERG instruction may be used and the default remains MERG 2. Although this is always safe for racemic twinning, there may be other forms of twinning for which it is not permissible to sort-merge first. Whether or not MERG is used, the program ignores all systematically absent contributions, with the result that a reflection is excluded from the data if it is systematically absent for all components.

For both powder (HKLF 6) and twinned data (HKLF 5 or TWIN with HKLF 4), the reflection data are reduced to the 'prime' component, by multiplying $F_o^2$ by the ratio of the $F_c^2$ for the prime reflection divided by the total $F_c^2$, before performing the analysis of variance and the Fourier calculations. Similarly 'OMIT h k l' refers to the indices of the prime component. The prime component is the one for which the indices have not been transformed by the TWIN instruction (i.e. m = 1 ), or in the case of HKLF 5 or HKLF 6 the component given with positive m (i.e. the last contributor to a given intensity measurement, not necessarily the one with |m| = 1).

## 6.7 The warning signs for twinning

Experience shows that there are a number of characteristic warning signs for twinning. Of course not all of them can be present in any particular example, but if one finds several of them the possibility of twinning should be given serious consideration.

**(a)** The metric symmetry is higher than the Laue symmetry.

**(b)** The $R_{int}$-value for the higher symmetry Laue group is only slightly higher than for the lower symmetry Laue group.

**(c)** The mean value for $|E^2\text{-}1|$ is much lower than the expected value of 0.736 for the non-centrosymmetric case. If we have two twin domains and every reflection has contributions from both, it is unlikely that both contributions will have very high or that both will have very low intensities, so the intensities will be distributed so that there are fewer extreme values.

**(d)** The space group appears to be trigonal or hexagonal.

**(e)** There are impossible or unusual systematic absences.

**(f)** Although the data appear to be in order, the structure cannot be solved.

**(g)** The Patterson function is physically impossible.

The following points are typical for non-merohedral twins, where the reciprocal lattices do not overlap exactly and only some of the reflections are affected by the twinning:

**(h)** There appear to be one or more unusually long axes, but also many absent reflections.

**(i)** There are problems with the cell refinement.

**(j)** Some reflections are sharp, others split.

**(k)** K = mean($F_o^2$) / mean($F_c^2$) is systematically high for the reflections with low intensity.

**(l)** For all of the 'most disagreeable' reflections, $F_o$ is much greater than $F_c$.

## 6.8 Conclusions

Twinning usually arises for good structural reasons. When the heavy atom positions correspond to a higher symmetry space group it may be difficult or impossible to distinguish between twinning and disorder of the light atoms; see Hoenle & von Schnering (1988). Since refinement as a twin usually requires only two extra instructions and one extra parameter, in such cases it should be attempted first, before investing many hours in a detailed interpretation of the 'disorder'! Indeed, it has been suggested by G.B. Jameson that all structures (including proteins) that are solved in space groups (such as $P3_1$) that could be merohedrally twinned without changing the systematic absences should be tested for such twinning (possible only present to a minor extent) by:

```
TWIN 0 1 0  1 0 0  0 0 -1
BASF 0.1
```

Refinement of twinned crystals often requires the full arsenal of constraints and restraints, since the refinements tend to be less stable, and the effective data to parameter ratio may well be low. In the last analysis chemical and crystallographic intuition may be required to distinguish between the various twinned and disordered models, and it is not easy to be sure that all possible interpretations of the data have been considered.

I should like to thank Regine Herbst-Irmer who wrote most of this chapter.

# 7. SHELXL Instruction Summary

This chapter lists the instructions that may be used in the *.ins* file for SHELXL-97. Defaults are given in square brackets; '#' indicates that the program will generate a suitable default value based on the rest of the available information. Continuation lines are flagged by '=' at the end of a line, the instruction being continued on the next line which must start with one or more spaces. Other lines beginning with spaces are treated as comments, so blank lines may be added to improve readability. All characters following '!' or '=' in an instruction line are ignored.

The *.ins* file may include an instruction of the form: +filename (the '+' character MUST be in column 1). This causes further input to be taken from the named file until an END instruction is encountered in that file, whereupon the file is closed and instructions are taken from the next line of the *.ins* file. The input instructions from such an 'include' file are not echoed to the *.lst* and *.res* file, and may NOT contain FVAR, BASF, EXTI or SWAT instructions or atoms (except inside a FRAG...FEND section) since this would prevent the *.res* file from being used unchanged for the next refinement job (after renaming as *.ins*).

The '+filename' facility enables standard fragment coordinates or long lists of restraints etc. to be read from the same files for each refinement job, and for different structures to access the same fragment or restraint files. One could also for example store the LATT and SYMM instructions for different space groups, or neutron scattering factors for particular elements, or LAUE instructions followed by wavelength-dependent scattering factors, in suitably named files. Since these 'include' files are not echoed, it is a good idea to test them as part of an *.ins* file first, to check for possible syntax errors. Such 'include' files may be nested; the maximum allowed depth depends upon the operating system and compiler used.

### 7.1 Crystal data and general instructions

**TITL [ ]**

Title of up to 76 characters, to appear at suitable places in the output. The characters '!' and '=', if present, are part of the title and are not specially interpreted.

**CELL λ a b c α β γ**

Wavelength and unit-cell dimensions in Å and degrees.

**ZERR Z esd(a) esd(b) esd(c) esd(α) esd(β) esd(γ)**

Z value (number of formula units per cell) followed by the estimated standard deviations in the unit-cell dimensions. Z is only required for the CIF output; the cell esds contribute to the estimated esds in bond lengths etc. after full-matrix refinement.

**LATT N[1]**

Lattice type: 1=P, 2=I, 3=rhombohedral obverse on hexagonal axes, 4=F, 5=A, 6=B, 7=C. N must be made negative if the structure is non-centrosymmetric.

**SYMM symmetry operation**

Symmetry operators, i.e. coordinates of the general positions as given in International Tables. The operator x, y, z is always assumed, so MUST NOT be input. If the structure is centrosymmetric, the origin MUST lie on a center of symmetry. Lattice centering and the presence of an inversion center should be indicated by LATT, not SYMM. The symmetry operators may be specified using decimal or fractional numbers, e.g. 0.5-x, 0.5+y, -z or Y-X, -X, Z+1/6; the three components are separated by commas.

**SFAC elements**

Element symbols which define the order of scattering factors to be employed by the program. The first 94 elements of the periodic system are recognized. The element name may be preceded by '$' but this is not obligatory (the '$' character is allowed for logical consistency but is ignored). The program uses the neutral atom scattering factors, f', f" and absorption coefficients from International Tables for Crystallography, Volume C (1992), Ed. A.J.C. Wilson, Kluwer Academic Publishers, Dordrecht: Tables 6.1.1.4(pp. 500-502), 4.2.6.8 (pp. 219-222) and 4.2.4.2 (pp. 193-199) respectively. The covalent radii stored in the program are based on experience rather than taken from a specific source, and are deliberately overestimated for elements which tend to have variable coordination numbers so that 'bonds' are not missed, at the cost of generating the occasional 'non-bond'. The default radii (not those set for individual atoms by CONN) are printed before the connectivity table.

**SFAC label a1 b1 a2 b2 a3 b3 a4 b4 c f' f" mu r wt**

Scattering factor in the form of an exponential series, followed by real and imaginary dispersion terms, linear absorption coefficient, covalent radius and atomic weight. Except for the 'label' and atomic weight the format is the same as that used in SHELX-76. label consists of up to 4 characters beginning with a letter (e.g. Ca2+) and should be included before a1; for consistency the first label character may be a '$', but this is ignored (note however that the '$', if used, counts as one of the four characters, leaving only three for the rest of the label). The two SFAC formats may be used in the same *.ins* file; the order of the SFAC instructions (and the order of element names in the first type of SFAC instruction) define the scattering factor numbers which are referenced by atom instructions. The units of mu should be barns/atom, as in Table 4.2.4.2 of International Tables, Volume C (see above). For neutrons this format should be used, with a1...b4 set to zero.

Hydrogen atoms are treated specially by SHELXL; they are recognized by having the scattering factor number that corresponds to 'H' on the SFAC instruction. For X-ray structures that contain both D and H, e.g. because the crystals were grown from a deuterated solvent in an n.m.r tube (a common source of good crystals!), both H and D should be included on the SFAC and UNIT instructions, but all the H and D atoms should employ the 'H' scattering factor number. In this way the density will be calculated correctly, but the D atoms may be idealized using HFIX etc.

**DISP E f' f" [#] mu [#]**

The DISP instruction allows the dispersion and (optionally) the absorption coefficient of a particular element (the name may be optionally prefaced by '$') to be read in without having to use the full form of the SFAC instruction. It will typically be used for synchrotron data where the wavelength does not correspond to the values (for Cu, Mo and Ag radiation) for which these terms are stored in the program. All other terms on the SFAC instruction are independent of the wavelength, so its short form may then be used. DISP instructions, if present, MUST come between the last SFAC and the UNIT instruction.

**UNIT n1 n2 ...**
Number of atoms of each type in the unit-cell, in SFAC order.

**LAUE E**
Wavelength-dependent values of f' and f" may be defined for an element E by means of the LAUE instruction, which is used in conjunction with the HKLF 2 reflection data format (in which the wavelength is given separately for each reflection). This is primarily intended for refinement of structures against Laue data collected using synchrotron radiation, but could also be used for refinement of a structure using data collected at different wavelengths for which some of the dispersion terms are significant (e.g. MAD data for macromolecules). There is no provision for handling overlapping reflection orders, and scaling for the source intensity distribution and Lp, absorption corrections etc. must have been performed before using SHELXL. A dummy wavelength of say 0.7 Å should be given on the CELL instruction, and the absorption coefficient estimated by the program should be ignored.

The element symbol may be preceded by '$' but this is optional; it must be followed by at least one blank or the end of the line. Any remaining information on the LAUE instruction line is ignored. The line immediately following the LAUE instruction is always ignored, and so may be used for headings. The following lines contain values of wavelength (in Å), f' and f" in FORMAT(F7.3,2F8.3); further information (e.g. the absorption coefficient μ) may follow on the same line but will be ignored. The wavelength values must be in ascending order and will be linearly interpolated; the wavelength intervals do not need to be equal (but it is more efficient if most of them are) and should indeed be smaller in the region of an absorption edge. This list is terminated by a record in which all three values are given as zero. There should only be one LAUE instruction for each element type; if a reflection wavelength is outside the range specified, the constant f' and f" values defined by the corresponding SFAC instruction are used instead.

A LAUE instruction must be preceded by (normal) SFAC and UNIT instructions referencing the elements in question, and by all atoms. Thus the LAUE instruction(s) are usually the last instructions before HKLF 2 (or -2) at the end of the *.ins* file (which facilitates editing). The +filename construction may conveniently be used to read long LAUE tables from 'include' files without echoing them.

**REM**
Followed by a comment on the same line. This comment is copied to the results file (*.res*). A line beginning with at least one blank may also be used as a comment, but such comments are only copied to the *.res* file if the line is completely blank; REM comments are always copied. Comments may also be included on the same line as any instruction following the character '!', and are copied to the *.res* file (except in the case of atoms and FVAR, EXTI, SWAT and BASF instructions).

**MORE m [1]**
MORE sets the amount of (printer) output; m takes a value in the range 0 (least) to 3 (most verbose). MORE 0 also suppresses the echoing to the *.lst* file of any instructions or atoms which follow it (until the next MORE instruction).

**TIME t [#]**

If the time t (measured in seconds from the start of the job) is exceeded, SHELXL performs no further least-squares cycles, but goes on to the final structure factor calculation followed by bond lengths, Fourier calculations etc. The default value of t is installation dependent, and is either set to 'infinity' or to a little less than the maximum time allocation for a particular class of job. Usually t is 'CPU time', but some some operating systems (e.g. MSDOS) the elapsed time may have to be used instead.

**`END`**

END is used to terminate an 'include' file, and may also be included after HKLF in the *.ins* file (for compatibility with SHELX-76).

### 7.2 Reflection data input

Before running SHELXL, a reflection data file *name.hkl* must have been prepared. The HKLF command tells the program which format has been chosen for this file, and allows the indices to be transformed using the 3x3 matrix $r_{11}$...$r_{33}$, so that the new *h* is $r_{11}$**h* + $r_{12}$**k* + $r_{13}$**l* etc. The program will not accept matrices with negative or zero determinants. It is essential that the cell, symmetry and atom coordinates in the *.ins* file correspond to the indices AFTER transformation using this matrix.

**`HKLF n[0] s[1] r11...r33[1 0 0 0 1 0 0 0 1] wt[1] m[0]`**

n is negative if reflection data follow, otherwise they are read from the *.hkl* file. The data are read in FORMAT(3I4,2F8.2,I4) (except for |n| < 3) subject to FORTRAN-77 conventions. The data are terminated by a record with *h*, *k* and *l* all zero (except |n| = 1, which contains a terminator and a checksum). In the reflection formats given below, BN stands for batch number. If BN is greater than one, $F_c$ is multiplied by the (BN-1)'th coefficient specified by means of BASF instructions (see below). If BN is zero or absent, it is reset to one. The multiplicative scale s multiplies both $F_o^2$ and $\sigma(F_o^2)$ (or $F_o$ and $\sigma(F_o)$ for n = 1 or 3). The multiplicative weight wt multiplies all $1/\sigma^2$ values and m is an integer 'offset' needed to read 'condensed data' (HKLF 1); both are included for compatibility with SHELX-76. Negative n is also only retained for upwards compatibility; it is much better to keep the reflection data in the *name.hkl* file, otherwise the data can easily get lost when editing *name.res* to *name.ins* for the next job.

**n = 1:** SHELX-76 condensed data (BN is set to one). 'Condensed data' impose unnecessary index restrictions and can introduce rounding errors; although they still have their uses (email!), SHELXL cannot generate condensed data and their use is discouraged.

**n = 2:** *h k l* $F_o^2$ $\sigma(F_o^2)$ BN [1] $\lambda$ [#] in FORMAT(3I4,2F8.2,I4,F8.4) for refinement based on singlet reflections from Laue photographs. The data are assumed to be scaled for source intensity distribution and geometric factors and (if necessary) corrected for absorption. If $\lambda$ is zero or absent the value from the CELL instruction is used. n = 2 switches off the merging of equivalent reflections BEFORE l.s. refinement (i.e. sets MERG 0); equivalents and measurements of the same reflections at different wavelengths are merged after least-squares refinement and the subsequent application of a dispersion correction, but before Fourier calculations.

The remaining options (n > 2) all require FORMAT(3I4,2F8.2,I4); other compatible formats (e.g. F8.0 or even I8) may be used for the floating point numbers provided that eight columns are used in all and a decimal point is present.

**n = 3:** $h\ k\ l\ F_o\ \sigma(F_o)$ BN [1] (if BN is absent or zero it is set to 1).The use of data corresponding to this format is allowed but is NOT RECOMMENDED, since the generation of $F_o$ and $\sigma(F_o)$ from $F_o^2$ and $\sigma(F_o^2)$ is a tricky statistical problem and could introduce bias.

**n = 4:** $h\ k\ l\ F_o^2\ \sigma(F_o^2)$ BN [1] is the standard reflection data file. Since $F_o^2$ is obtained as the difference of the experimental peak and background counts, it may be positive or slightly negative. BN may be made negative (e.g. by SHELXPRO) to flag a reflection for inclusion in the $R_{free}$ reference set (see CLGS and L.S. with a second parameter of -1).

**n = 5:** $h\ k\ l\ F_o^2\ \sigma(F_o^2)$ m where m is the twin component number. Each measured $F_o^2$ value is fitted to the sum of $k_{|m|}F_{c|m|}^2$ over all contributing components, multiplied by the overall scale factor. m should be given as positive for the last contributing component and negative for the remaining ones (if any). The values of $F_o^2$ and $\sigma(F_o^2)$ are taken from the last ('prime') reflection in a group, and may simply be set equal for each component, but the indices *h,k,l* will in general take on different values for each component. The starting values of the twin factors $k_2..k_{max(m)}$ are specified on BASF instruction(s); $k_1$ is given by one minus the sum of the other twin factors. Note that many simple forms of twinning can also be handled with HKLF 4 and a TWIN instruction to generate the indices of the remaining twin component(s); HKLF 5 is required if the reciprocal space lattices of the components cannot be superimposed exactly. HKLF 5 sets MERG 0, and may not be used with TWIN.

**n = 6:** $h\ k\ l\ F_o^2\ \sigma(F_o^2)$ m as for n = 5, there may be one or more sets of reflection indices corresponding to a single $F_o^2$ value. The last reflection in a group has a positive m value and the previous members of the group have negative m. The values of $F_o^2$ and $\sigma(F_o^2)$ are taken from the last ('prime') reflection in a group, and may simply be set to the same values for the others. m is here the reflection MULTIPLICITY, and is defined as the number of equivalent permutations of the given *h, k* and *l* values, not counting Friedel opposites. This is intended for fitting resolved powder data for high symmetry crystal systems. For example, in a powder diagram of a crystal in the higher cubic Laue class (m3m) the reflections 3 0 0 (with multiplicity 3) and 2 2 1 (multiplicity 12) would contribute to the same measured $F_o^2$. HKLF 6 sets MERG 0. HKLF 6 may not be used with BASF or TWIN.

THERE MAY ONLY BE ONE HKLF INSTRUCTION AND IT MUST COME LAST, except when HKLF -n is followed by reflection data in the *.ins* file, in which case the file is terminated by the end of the reflection data. Negative n is retained for compatibility with SHELX-76 but is not recommended!

**`OMIT s[-2] 2θ(lim)[180]`**

If s is given as negative, all reflections with $F_o^2 < 0.5s\sigma(F_o^2)$ are replaced by $0.5s\sigma(F_o^2)$; thus if no OMIT instruction is given the default action is to replace all $F_o^2$ values less than $-\sigma(F_o^2)$ by $-\sigma(F_o^2)$. If s is positive it is interpretated as a threshold for flagging reflections as 'unobserved'. Unobserved data are not used for least-squares refinement or Fourier calculations, but are retained for the calculation of *R*-indices based on all data, and may also appear (flagged with an asterisk) in the list of reflections for which $F_o^2$ and $F_c^2$ disagree significantly. Internally in the program s is halved and applied to $F_o^2$, so for positive $F_o^2$ the test is roughly equivalent to suppressing all reflections with $F_o < s\ \sigma(F_o)$, as required for consistency with SHELX-76. Note that s may be set to 0 or (as in the default setting) to a negative threshold (to modify very negative $F_o^2$). An OMIT instruction with a positive s value is NOT ALLOWED in combination with ACTA, because it may introduce a bias in the final refined parameters; individual aberrant reflections may still be suppressed using OMIT *h k l*, even when ACTA is used.

2θ(lim) defines a limiting 2θ above which reflections are totally ignored; they are rejected immediately on reading in. This facility may be used to save computer time in the early stages of structure refinement, and is also sometimes useful for macromolecules. The SHEL command may also be used to ignore reflections above or below particular limiting resolution values.

OMIT followed by atom names but no numbers may be used to calculate an 'omit map' and is described in the section 'Atom Lists ...'.

**OMIT *h k l***

The reflection *h*,*k*,*l* (the indices refer to the standard setting after data reduction, and correspond to those in the list of ‘disagreeable reflections’ after refinement) is ignored completely. Since there may be perfectly justified reasons for ignoring individual reflections (e.g. when a reflection is truncated by the beam stop) this form of OMIT is allowed with ACTA; however it should not be used indiscriminately. If MERG N with non-zero N is employed (or the (default) MERG 2 is assumed), all reflections which would generate the final indices *h*,*k*,*l* are ignored; if MERG 0 is specified, the indices must match those in the input *.hkl* file exactly.

**SHEL lowres[infinite] highres[0]**

Reflections outside the specified resolution range in Å are ignored completely. This instruction may be useful for macromolecules.

**BASF scale factors**

Relative batch scale factors are included in the least-squares refinement based on the batch numbers in the *.hkl* file. For batch number BN, the $F_c^2$ value is multiplied by the (BN-1)'th scale factor from the BASF instruction, as well as by the overall scale factor. For batch number one (or zero), $F_c$ is multiplied by the overall scale factor, but not by a batch scale factor. The least-squares matrix will be singular if there are no reflections with BN=1 (or zero), so the program considers this to be an error. Note that BASF scale factors, unlike the overall scale factor (see FVAR) are relative to $F^2$, not *F*. For twinned crystals, i.e. when either TWIN or HKLF 5 are employed, BASF specifies the fractional contributions of the various twin components. BASF parameters may also be used by the HOPE instruction. Except when they are used by HOPE, the program does not allow BASF parameters to become negative.

**TWIN 3x3 matrix [-1 0 0 0 -1 0 0 0 -1] n[2]**

n is the number of twin components (2 or greater) and the matrix is applied (iteratively if |n| > 2) to generate the indices of the twin components from the input reflection indices, which apply to the first (prime) component. If a transformation matrix is also given on the HKLF instruction, it is applied first before the (iterative) application of the TWIN matrix. This method of defining twinning allows the standard HKLF 4 format to be used for the *.hkl* file, but can only be used when the reciprocal lattices for all twinned components are metrically superimposable. In other cases HKLF 5 format must be used. The $F_o^2$ values are fitted to the sum of $k_{m*}F_{cm}^2$ multiplied by the overall scale factor, where $k_1$ is one minus the sum of $k_2$, $k_3$, .. and the starting values for the remaining twin fractions $k_2$, $k_3$, .. are specified on a BASF instruction. Only one TWIN instruction is allowed. If BASF is omitted the TWIN factors are all assumed to be equal (i.e. 'perfect' twinning).

If the racemic twinning is present at the same time as normal twinning, n should be doubled (because there are twice as many components as before) and given a negative sign (to indicate to the program that the inversion operator is to be applied multiplicatively with the specified TWIN matrix). The number of BASF parameters, if any, should be increased from m-1 to 2m-1 where m is the original number of components (equal to the new |n| divided by 2). The TWIN matrix is applied m-1 times to generate components 2 ... m from the prime reflection (component 1); components m+1 ... 2m are then generated as the Friedel opposites of components 1 ... m.

**EXTI x[0]**

An extinction parameter x is refined, where $F_c$ is multiplied by:

$$k\,[\,1 + 0.001\ x\ F_c^2\ \lambda^3\ /\ \sin(2\theta)\,]^{-1/4}$$

where k is the overall scale factor. Note that it has been necessary to change this expression from SHELX-76 (which used an even cruder approximation) and XLS in SHELXTL version 4 (which used 0.002 instead of $0.001\lambda^3$). The wavelength dependence is needed for HKLF 2 (Laue) data. The program will print a warning if extinction (or SWAT - see below) may be worth refining, but it is not normally advisable to introduce it until all the non-hydrogen atoms have been found. For twinned and powder data, the $F_c^2$ value used in the above expression is based on the total calculated intensity summed over all components rather than the individual contributions, which would be easier to justify theoretically (but makes little difference in practice). For the analysis of variance and *.fcf* output file, the $F_o^2$ values are brought onto the absolute scale of $F_c^2$ by dividing them by the scale factor(s) and the extinction factor. The above expression for the extinction is empirical and represents a compromise to cover both primary and secondary extinction; it has been shown to work well in practice but does not appear to correspond exactly to any of the expressions discussed in the literature. The article by Larson (1970) comes closest and should be consulted for further information.

**SWAT g[0] U[2]**

The SWAT option allows two variables g and U to be refined in order to model diffuse solvent using Babinet's principle (Moews & Kretsinger, 1975; the same formula is employed in the program TNT, but the implementation is somewhat different). The calculated intensity is modified as follows:

$$F_c^2(\mathrm{new}) = F_c^2(\mathrm{old})\ .\ (1 - g\ .\ \exp\,[\,-8\pi^2U(\sin\theta\,/\,\lambda)^2\,]\,)$$

A large value of U ensures that only the low theta $F_c^2$ values are affected. Subtracting the term in g in this way from the occupied regions of the structure is equivalent to adding a corresponding diffuse scattering term in the (empty) solvent regions in its effect on all calculated $F_c^2$ values except F(000). For proteins g usually refines to a value between 0.7 and unity, and U usually refines to a value between 2 and 5; for small molecules without significant diffuse solvent regions g should refine to zero. Since g and U are correlated, it is better to start the diffuse solvent refinement by giving SWAT with no parameters; the program will then invent suitable starting values. Note that a different formula was employed in SHELXL-93, and so parameter values from SHELXL-93 may well be unsuitable starting values for the new version.

Since both extinction and diffraction from diffuse solvent tend to affect primarily the strong reflections at low diffraction angle, they tend to show the same symptoms in the analysis of variance, and so a combined warning message is printed. It will however be obvious from the type of structural problem which of the two should be applied. The program does not permit the simultaneous refinement of SWAT and EXTI.

**`HOPE nh [1]`**

Refines 12 anisotropic scaling parameter as suggested by Parkin, Moezzi & Hope (1995). nh points to the BASF parameter that stores the value of the first HOPE parameter; if nb is negative the 12 parameters are fixed at their current values. These parameters are highly correlated with the individual atomic anisotropic displacement parameters, and so are only useful for structures that are refined isotropically , e.g. macromolecules at moderate resolution. To some extent they can also model absorption errors. If HOPE is given without any parameters and there are no BASF instructions, the program will generate appropriate starting values. If BASF parameters are needed for twin refinement or as scale factors for different batches of data, nh should be given an absolute value greater than one.

**`MERG n[2]`**

If n is equal to 2 the reflections are sorted and merged before refinement; if the structure is non-centrosymmetric the Friedel opposites are not combined before refinement (necessary distinction from SHELXS). If n is 1 the indices are converted to a 'standard setting' in which *l* is maximized first, followed by *k*, and then *h*; if n is zero, the data are neither sorted nor converted to a standard setting. n = 3 is the same as n = 2 except that Friedel opposites are also merged (this introduces small systematic errors and should only be used for good reason, e.g. to speed up the early stages of a refinement of a light atom structure before performing the final stages with MERG 2). Note that the reflections are always merged, and Friedel opposites combined, before performing Fourier calculations in SHELXL so that the (difference) electron density is real and correctly scaled. Even with n = 0 the program will change the reflection order within each data block to optimize the vectorization of the structure factor calculations (it is shuffled back into the MERG order for LIST 4 output). Note that MERG may not be used in conjunction with TWIN or HKLF 5 or 6. In SHELX-76, MERG 3 had a totally different meaning, namely the determination of inter-batch scale factors; in SHELXL, these may be included in the refinement using the BASF instruction.

MERG 4 averages all equivalents, including Friedel opposites, and sets all δf" values to zero; it is often used in refinement of macromolecules.

### 7.3 Atom lists and least-squares constraints

Atom instructions begin with an atom name (up to 4 characters that do not correspond to any of the SHELXL command names, and terminated by at least one blank) followed by a scattering factor number (which refers to the list defined by the SFAC instruction(s)), x, y, and z in fractional coordinates, and (optionally) a site occupation factor (s.o.f.) and an isotropic U or six anisotropic $U_{ij}$ components (both in $Å^2$). Note that different program systems may differ in their order of $U_{ij}$ components; SHELXL uses the same order as SHELX-76. The exponential factor takes the form $\exp(-8\pi^2 U[\sin(\theta)/\lambda]^2)$ for an isotropic displacement parameter U and:

$$\exp ( -2\pi^2 [ h^2(a^*)^2U_{11} + k^2(b^*)^2U_{22} + ... + 2hka^*b^*U_{12} ] )$$

for anisotropic $U_{ij}$. An atom is specified as follows in the *.ins* file:

atomname sfac x y z sof [11] U [0.05] or $U_{11}$ $U_{22}$ $U_{33}$ $U_{23}$ $U_{13}$ $U_{12}$

The atom name must be unique, except that atoms in different residues - see RESI - may have the same names; in contrast to SHELX-76 it is not necessary to pad out the atom name to 4 characters with blanks. To fix any atom parameter, add 10. Thus the site occupation factor is normally given as 11 (i.e. fixed at 1). The site occupation factor for an atom in a special position should be multiplied by the multiplicity of that position (as given in International Tables, Volume A) and divided by the multiplicity of the general position for that space group. This is the same definition as in SHELX-76 and is retained for upwards compatibility; it might have been less confusing to keep the multiplicity and occupation factor separate. An atom on a fourfold axis for example will usually have s.o.f. = 10.25.

If any atom parameter is given as (10*m+p), where abs(p) is less than 5 and m is an integer, it is interpreted as $p\bullet fv_m$, where $fv_m$ is the mth 'free variable' (see FVAR). Note that there is no $fv_1$, since this position on an FVAR instruction is occupied by the overall scale factor, and m=1 corresponds to fixing an atom by adding 10. If m is negative, the parameter is interpreted as $p\bullet(fv_{-m}-1)$. Thus to constrain two occupation factors to add up to 0.25 (for two elements occupying the same fourfold special position) they could be given as 20.25 and -20.25, i.e. $0.25\bullet fv_2$ and $0.25\bullet(1-fv_2)$, which correspond to p=0.25, m=2 and p=-0.25, m=-2 respectively.

In SHELX-76, it was necessary to use free variables and coordinate fixing in this way to set up the appropriate constraints for refinement of atoms on special positions. In SHELXL, this is allowed (for upwards compatibility) but is NOT NECESSARY: the program will automatically work out and apply the appropriate positional, s.o.f. and $U_{ij}$ constraints for any special position in any space group, in a conventional setting or otherwise. If the user applies (correct or incorrect) special position constraints using free variables etc., the program assumes that this has been done with intent, and reports but does not apply the correct constraints. Thus the accidental application of a free variable to a $U_{ij}$ term of an atom on a special position can lead to the refinement 'blowing up'! All that is necessary is to specify atomname, sfac, x, y and z, and leave the rest to the program; when the atom is (later) made anisotropic using the ANIS command, the appropriate $U_{ij}$ constraints will be added by the program. For a well-behaved structure, the list of atom coordinates (from direct methods and/or difference electron density syntheses) suffices. If the multiplicity factor (s.o.f.) is left out, it will be fixed at the appropriate value of 1 for a general position and less than 1 for a special position. Since SHELXL automatically generates origin restraints for polar space groups, no atom coordinates should be fixed by the user for this purpose (in contrast to SHELX-76).

It may still be necessary to apply constraints by hand to handle disorder; a common case is when there are two possible positions for a group of atoms, in which the first set should all have s.o.f.'s of (say) 21, and the second set -21, with the result that the sum of the two occupation factors is fixed at 1, but the individual values may refine as $fv_2$ and $1-fv_2$. Similarly if a special position with 2/m symmetry is occupied by $Ca^{2+}$ and $Ba^{2+}$, the two ions could be given the s.o.f.'s 30.25 and -30.25 respectively. In this case it would be desirable to use the EADP instruction to equate the $Ca^{2+}$ and $Ba^{2+}$ (anisotropic) displacement parameters.

If U is given as -T, where T is in the range $0.5 < T < 5$, it is fixed at T times the $U_{eq}$ of the previous atom not constrained in this way. The resulting value is not refined independently but is updated after every least-squares cycle.

### SPEC del[0.2]

All following atoms (until the next SPEC instruction) are considered to lie on special positions (for the purpose of automatic constraint generation) if they lie within del (Å) of a special position. The coordinates of such an atom are also adjusted so that it lies exactly on the special position.

### RESI class[ ] number[0] alias

Until the next RESI instruction, all atoms are considered to be in the specified 'residue', which may be defined by a class (up to four characters, beginning with a letter) or number (up to four digits) or both. The same atom names may be employed in different residues, enabling them to be referenced globally or selectively. The residue number should be unique to a particular residue, but the class may be used to refer to a class of similar residues, e.g. a particular type of amino acid in a polypeptide.

Residues may be referenced by any instruction that allows atom names; the reference takes the form of the character '_' followed by either the residue class or number without intervening spaces. If an instruction codeword is followed immediately by a residue number, all atom names referred to in the instruction are assumed to belong to that residue unless they are themselves immediately followed by '_' and a residue number, which is then used instead. Thus:

```
RTAB_4 Ang N H0 O_11
```

would cause the calculation of an angle N_4 - H0_4 - O_11, where the first two atoms are in residue 4 and the third is in residue 11.

If the instruction codeword is followed immediately by a residue class, the instruction is effectively duplicated for all residues of that class. '_* ' may be used to match all residue classes; this includes the default class '    ' (residue number 0) which applies until the first RESI instruction is encountered. Thus:

```
MPLA_phe CB > CZ
```

would calculate least-squares planes through atoms CB to CZ inclusive of all residues of class 'phe' (phenylalanine). In the special case of HFIX, only the FIRST instruction which applies to a given atom is applied. Thus:

```
HFIX_1 33 N
HFIX_* 43 N
```

would add hydrogens to the N-terminal nitrogen (residue 1) of a polypeptide to generate a (protonated) $-NH_3^+$ group, but all other (amide) nitrogens would become -NH-.

Individual atom names in an instruction may be followed by '_' and a residue number, but not by '_*' or '_' and a residue class. If an atom name is not followed by a residue number, the current residue is assumed (unless overridden by a global residue number or class appended to the instruction codeword). The symbols '_+' meaning 'the next residue' and '_-' meaning 'the preceding residue'(i.e. residues number n+1 and n-1 if the current residue number is n) may be appended to atom names but not to instruction codenames. Thus the instruction:

```
RTAB_* Omeg CA_+ N_+ C CA
```

could be used to calculate all the peptide $\omega$ torsion angles in a protein or polypeptide. If (as at the C-terminus in this example) some or all of the named atoms cannot be found for a particular residue, the instruction is simply ignored for that residue.

'_$n' does not refer to a residue; it uses the symmetry operation $n defined by a preceding 'EQIV $n' instruction to generate an equivalent of the named atom (see EQIV). alias specifies an alternative value of the residue number so that cyclic chains of residues may be created; for a cyclic pentapeptide (residue numbers 2,3,..6) it could be set to 1 for residue 6 and to 7 for residue 2. If more than one RESI instruction refers to the same number, alias only needs to be specified once. alias is referenced only by the _+ and _- operations (see above), and a value used for alias may not be used as a residue number on a RESI instruction. Note that if there is more than one cyclic peptide in the asymmetric unit, it is a good idea to leave a gap of TWO residue numbers between them. E.g. a cyclic pentapeptide with two molecules in the asymmetric unit would be numbered 2 to 6 and 9 to 13, with aliases 7 on RESI 2, 1 on RESI 6, 14 on RESI 9 and 8 on RESI 13. It will generally be found convenient for applying restraints etc. to use the same names for atoms in identical residues. Since SHELXL does not recognize chain ID’s (used in PDB format) it is normal to add a constant to the residue numbers to denote a different chain (e.g. chain A could be 1001 to 1234 and chain B 2001 to 2234). The auxiliary program SHELXPRO provides extensive facilities for handling residues.

**MOVE dx[0] dy[0] dz[0] sign[1]**

The coordinates of the atoms that follow this instruction are changed to: x = dx + sign * x, y = dy + sign * y, z = dz + sign * z until superseded by a further MOVE. MOVE should not be used at the same time as the specification of zero coordinates to indicate that an atom should not be used in fitting a fragment of known geometry (e.g. AFIX 66), because after the move the coordinates will no longer be zero!

**ANIS n**

The next n isotropic non-hydrogen atoms are made anisotropic, generating appropriate special position constraints for the $U_{ij}$ if required. Intervening atoms which are already anisotropic are not counted. A negative n has the same effect.

**ANIS names**

The named atoms are made anisotropic (if not already), generating the appropriate constraints for special positions. Note that names may include '$' followed by a scattering factor name (see SFAC); 'ANIS $CL' would make all chlorine atoms anisotropic. Since ANIS, like other instructions, applies to the current residue unless otherwise specified, ANIS_* $S would be required to make the sulfur atoms in all residues anisotropic (for example). ANIS MUST precede the atoms to which it is to be applied. ANIS on its own, with neither a number nor names as parameters, makes all FOLLOWING non-hydrogen atoms (in all residues) anisotropic. The L.S. and CGLS instructions provide the option of delaying the conversion to anisotropic of all atoms specified by ANIS until a given number of least-squares cycles has been performed.

**AFIX mn d[#] sof[11] U[10.08]**

AFIX applies constraints and/or generates idealized coordinates for all atoms until the next AFIX instruction is read. The digits mn of the AFIX code control two logically quite separate operations. Although this is confusing for new users, it has been retained for upwards compatibility with SHELX-76, and because it provides a very concise notation. m refers to geometrical operations which are performed before the first refinement cycle (hydrogen atoms are idealized before every cycle), and n sets up constraints which are applied throughout the least-squares refinement. n is always a single digit; m may be two, one or zero digits (the last corresponds to m = 0).

The options for idealizing hydrogen atom positions depend on the connectivity table that is set up using CONN, BIND, FREE and PART; with experience, this can also be used to generate hydrogen atoms attached to disordered groups and to atoms on special positions. d determines the bond lengths in the idealized groups, and sof and U OVERRIDE the values in the atom list for all atoms until the next AFIX instruction. U is not applied if the atom is already anisotropic, but is used if an isotropic atom is to be made anisotropic using ANIS. Any legal U value may be used, e.g. 31 (a free variable reference) or -1.2 (1.2 times Ueq of the preceding normal atom). Each AFIX instruction must be followed by the required number of hydrogen or other atoms. The individual AFIX options are as follows; the default X-H distances depend on both the chemical environment and the temperature (to allow for librational effects) which is specified by means of the TEMP instruction.

**m = 0** No action.

**m = 1** Idealized tertiary C-H with all X-C-H angles equal. There must be three and only three other bonds in the connectivity table to the immediately preceding atom, which is assumed to be carbon. m = 1 is often combined with a riding model refinement (n = 3).

**m = 2** Idealized secondary $CH_2$ with all X-C-H and Y-C-H angles equal, and H-C-H determined by X-C-Y (i.e. approximately tetrahedral, but widened if X-C-Y is much less than tetrahedral). This option is also suitable for riding refinement (n = 3).

**m = 3** Idealized $CH_3$ group with tetrahedral angles. The group is staggered with respect to the shortest other bond to the atom to which the $-CH_3$ is attached. If there is no such bond (e.g. an acetonitrile solvent molecule) this method cannot be used (but m = 13 is still viable).

**m = 4** Aromatic C-H or amide N-H with the hydrogen atom on the external bisector of the X-C-Y or X-N-Y angle. m = 4 is suitable for a riding model refinement, i.e. AFIX 43 before the H atom.

**m = 5** Next five non-hydrogen atoms are fitted to a regular pentagon, default d = 1.42 Å.

**m = 6** Next six non-hydrogen atoms are fitted to a regular hexagon, default d = 1.39 Å.

**m = 7** Identical to m = 6 (included for upwards compatibility from SHELX-76). In SHELX-76 only the first, third and fifth atoms of the six-membered ring were used as target atoms; in SHELXL this will still be the case if the other three are given zero coordinates, but the procedure is more general because any one, two or three atoms may be left out by giving them zero coordinates.

**m = 8** Idealized OH group, with X-O-H angle tetrahedral. If the oxygen is attached to a saturated carbon, all three staggered positions are considered for the hydrogen. If it is attached to an aromatic ring, both positions in the plane are considered. The final choice is based on forming the 'best' hydrogen bond to a nitrogen, oxygen, chlorine or fluorine atom. The algorithm involves generating a potential position for such an atom by extrapolating the O-H vector, then finding the nearest N, O, F or Cl atom to this position, taking symmetry equivalents into account. If another atom that (according to the connectivity table) is bonded to the N, O, F or Cl atom, is nearer to the ideal position, the N, O, F or Cl atom is not considered. Note that m = 8 had a different effect in SHELX-76 (but was rarely employed).

**m = 9** Idealized terminal $X=CH_2$ or $X=NH_2^+$ with the hydrogen atoms in the plane of the nearest substituent on the atom X. Suitable for riding model refinement (AFIX 93 before the two H atoms).

**m = 10** Idealized pentamethylcyclopentadienyl (Cp*). This AFIX must be followed by the 5 ring carbons and then the 5 methyl carbons in cyclic order, so that the first methyl group (atom 6) is attached to the first carbon (atom 1). The default d is 1.42 Å, with the $C\text{-}CH_3$ distance set to 1.063d. A variable-metric rigid group refinement (AFIX 109) would be appropriate, and would allow for librational shortening of the bonds. Hydrogen atoms (e.g. with AFIX 37 or 127) may be included after the corresponding carbon atoms, in which case AFIX 0 or 5 (in the case of a rigid group refinement) must be inserted before the next carbon atom.

**m = 11** Idealized naphthalene group with equal bonds (default d = 1.39 A). The atoms should be numbered as a symmetrical figure of eight, starting with the alpha C and followed by the beta, so that the first six atoms (and also the last six) describe a hexagon in cyclic order. m = 11 is also appropriate for rigid group refinement (AFIX 116).

**m = 12** Idealized disordered methyl group; as m = 3 but with two positions rotated from each other by 60 degrees. The corresponding occupation factors should normally be set to add up to one, e.g. by giving them as 21 (i.e. 1*fv(2) ) and -21 ( 1*(1-fv(2)) ). If HFIX is used to generate an AFIX instruction with m=12, the occupation factors are fixed at 0.5. AFIX 12n is suitable for a *para* methyl on a phenyl group with no *meta* substituents, and should be followed by 6 half hydrogen atoms (first the three belonging to one $-CH_3$ component, then the three belonging to the other, so that hydrogens n and n+3 are opposite one another). The six hydrogens should have the same PART number as the carbon to which they are attached (e.g. PART 0).

**m = 13** Idealized $CH_3$ group with tetrahedral angles. If the coordinates of the first hydrogen atom are non-zero, they define the torsion angle of the methyl group. Otherwise (or if the AFIX instruction is being generated via HFIX) a structure-factor calculation is performed (of course only once, even if many hydrogens are involved) and the torsion angle is set that maximizes the sum of the electron density at the three calculated hydrogen positions. Since even this is not an infallible method of getting the correct torsion angle, it should normally be combined with a rigid or rotating group refinement for the methyl group (e.g. mn = 137 before the first H). In subsequent least-squares cycles the group is re-idealized retaining the current torsion angle

**m = 14** Idealized OH group, with X-O-H angle tetrahedral. If the coordinates of the hydrogen atom are non-zero, they are used to define the torsion angle. Otherwise (or if HFIX was used to set up the AFIX instruction) the torsion angle is chosen which maximizes the electron density (see m = 13). Since this torsion angle is unlikely to be very accurate, the use of a rotating group refinement is recommended (i.e. AFIX 147 before the H atom).

**m = 15** BH group in which the boron atom is bonded to either four or five other atoms as part of an polyhedral fragment. The hydrogen atom is placed on the vector that represents the negative sum of the unit vectors along the four or five other bonds to the boron atom.

**m = 16** Acetylenic C-H, with X-C-H linear. Usually refined with the riding model, i.e. AFIX 163.

**m > 16** A group defined in a FRAG...FEND section with code = m is fitted, usually as a preliminary to rigid group refinement. The FRAG...FEND section MUST precede the corresponding AFIX instruction in the '*.ins*' file, but there may be any number of AFIX instructions with the same m corresponding to a single FRAG...FEND section.

When a group is fitted (m = 5, 6, 10 or 11, or m > 16), atoms with non-zero coordinates are used as target atoms with equal weight. Atoms with all three coordinates zero are ignored. Any three or more non-colinear atoms may be used as target atoms.

'Riding' (n = 3, 4) and 'rotating' (n = 7, 8) hydrogen atoms, but not other idealized groups, are re-idealized (if m is 1, 2, 3, 4, 8, 9, 12, 13, 14, 15 or 16) before each refinement cycle (after the first cycle, the coordinates of the first hydrogen of a group are always non-zero, so the torsion angle is retained on re-idealizing). For n = 4 and 8, the angles are re-idealized but the (refined) X-H bond length is retained, unless the hydrogen coordinates are all zero, in which case d (on the AFIX instruction) or (if d is not given) a standard value which depends on the chemical environment and temperature (TEMP) is used instead.

**n = 0** No action.

**n = 1** The coordinates, s.o.f. and U or $U_{ij}$ are fixed.

**n = 2** The s.o.f. and U (or $U_{ij}$) are fixed, but the coordinates are free to refine.

**n = 3** The coordinates, but not the s.o.f. or U (or $U_{ij}$) 'ride' on the coordinates of the previous atom with n not equal to 3. The same shifts are applied to the coordinates of both atoms, and both contribute to the derivative calculation. The atom on which riding is performed may not itself be a riding atom, but it may be in a rigid group (m = 5, 6 or 9).

**n = 4** This constraint is the same as n = 3 except that the X-H distance is free to refine. The X-H vector direction does not change. This constraint requires better quality reflection data than n = 3, but allows for variations in apparent X-H distances caused by libration and bonding effects. If there is more than one equivalent hydrogen, the same shift is applied to each equivalent X-H distance (e.g. to all three C-H bonds in a methyl group). n = 4 may be combined with DFIX or SADI restraints (to restrain chemically equivalent X-H distances to be equal) or embedded inside a rigid (n = 6) group, in which case the next atom (if any) in the same rigid group must follow an explicit AFIX instruction with n = 5. Note that n = 4 had a different effect in SHELX-76.

**n = 5** The next atom(s) are 'dependent' atoms in a rigid group. Note that this is automatically generated for the atoms following an n = 6 or n = 9 atom, so does not need to be included specifically unless m has to be changed (e.g. AFIX 35 before the first hydrogen of a rigid methyl group with AFIX 6 or 9 before the preceding carbon).

**n = 6** The next atom is the 'pivot atom' of a NEW rigid group, i.e. the other atoms in the rigid group rotate about this atom, and the same translational shifts are applied to all atoms in the rigid group.

**n = 7** The following (usually hydrogen) atoms (until the next AFIX with n not equal to 7) are allowed to ride on the immediately preceding atom X and rotate about the Y-X bond; X must be bonded to one and only one atom Y in the connectivity list, ignoring the n = 7 atoms (which, if they are F rather than H, may be present in the connectivity list). The motion of the atoms of this 'rotating group' is a combination of riding motion (c.f. n = 3) on the atom X plus a tangential component perpendicular to the Y-X and X-H bonds, so that the X-H distances, Y-X-H and H-X-H angles remain unchanged. This constraint is intended for -OH, -$CH_3$ and possibly -$CF_3$ groups. X may be part of a rigid group, which may be resumed with an AFIX n = 5 following the n = 7 atoms.

**n = 8** This constraint is similar to n = 7 except that the X-H distances may also vary, the same shifts being applied along all the X-H bonds. Thus only the Y-X-H and H-X-H angles are held constant; the relationship of n = 8 to n = 7 corresponds to that of n = 4 to n = 3. DFIX and SADI restraints may be useful for the X-H distances. This constraint is useful for -$CF_3$ groups or for -$CH_3$ groups with good data.

**n = 9** The first (pivot) atom of a new 'variable metric' rigid group. Such a group retains its 'shape' but may shrink or expand uniformly. It is useful for $C_5H_5$ and $BF_4$ groups, which may show appreciable librational shortening of the bond lengths. Subsequent atoms of this type of rigid group should have n = 5, which is generated automatically by the program if no other AFIX instruction is inserted between the atoms. Riding atoms are not permitted inside this type of rigid group. Only the pivot atom coordinates may be fixed (by adding 10) or tied to free variables, and only the pivot atom may lie on a special position (for the automatic generation of special position constraints).

Although there are many possible combinations of m and n, in practice only a small number is used extensively, as discussed in the section on hydrogen atoms. Rigid group fitting and refinement (e.g. AFIX 66 followed by six atoms of a phenyl ring or AFIX 109 in front of a Cp* group) is particularly useful in the initial stages of refinement; atoms not found in the structure solution may be given zero coordinates, in which case they will be generated from the rigid group fit.

A rigid group or set of dependent hydrogens must ALWAYS be followed by 'AFIX 0' (or another AFIX instruction). Leaving out 'AFIX 0' by mistake is a common cause of error; the program is able to detect and correct some obvious cases, but in many cases this is not logically possible.

**HFIX mn U[#] d[#] atomnames**

HFIX generates AFIX instructions and dummy hydrogen atoms bonded to the named atoms, the AFIX parameters being as specified on the HFIX instruction. This is exactly equivalent to the corresponding editing of the atom list. The atom names may reference residues (by appending '_n' to the name, where n is the residue number), or SFAC names (preceded by a '$' sign). U may be any legal value for the isotropic temperature factor, e.g. 21 to tie a group of hydrogen U value to free variable 2, or -1.5 to fix U at 1.5 times U(eq) of the preceding normal atom. HFIX MUST precede the atoms to which it is to be applied. If more than one HFIX instruction references a given atom, only the FIRST is applied. 'HFIX 0' is legal, and may be used to switch off following HFIX instructions for a given atom (which is useful if they involve '_*' or a global reference to a residue class).

**FRAG code[17] a[1] b[1] c[1] α[90] β[90] γ[90]**

Enables a fragment to be input using a cell and coordinates taken from the literature. Orthogonal coordinates may also be input in this way. Such a fragment may be fitted to the set of atoms following an AFIX instruction with m = code (code must be greater than 16); there must be the same number of atoms in this set as there are following FRAG, and they must be in the same order. Only the coordinates of the FRAG fragment are actually used; atom names, sfac numbers, sof and $U_{ij}$ are IGNORED. A FRAG fragment may be given anywhere between UNIT and HKLF or END, and must be terminated by a FEND instruction, but must precede any AFIX instruction which refers to it. This 'rigid fit' is often a preliminary to a rigid group refinement (AFIX with n = 6 or 9).

**FEND**

This must immediately follow the last atom of a FRAG fragment.

**EXYZ atomnames**

The same x, y and z parameters are used for all the named atoms. This is useful when atoms of different elements share the same site, e.g. in minerals (in which case EADP will probably be used as well). The coordinates (and possibly free variable references) are taken from the named atom which precedes the others in the atom list, and the actual values, free variable references etc. given for the x, y and z of the other atoms are ignored. An atom should not appear in more than one EXYZ instruction.

**EADP atomnames**

The same isotropic or anisotropic displacement parameters are used for all the named atoms. The displacement parameters (and possibly free variable references) are taken from the named atom which precedes the others in the atom list, and the actual values, free variable references etc. given for the $U_{ij}$ of the other atoms are ignored. The atoms involved must either be all isotropic or all anisotropic. An atom should not appear in more than one EADP instruction. 'Opposite' fluorines of $PF_6$ or disordered $-CF_3$ groups are good candidates for EADP, e.g.

```
EADP F11 F14
EADP F12 F15
EADP F13 F16
C1 .......
PART 1
F11 ...... 21 ......
F12 ...... 21 ......
F13 ...... 21 ......
PART 2
F14 ...... -21 ......
F15 ...... -21 ......
F16 ...... -21 ......
PART 0
```

EADP applies an (exact) *constraint*. The SIMU instruction *restrains* the Uij components of neighboring atoms to be approximately equal with an appropriate (usually fairly large) esd.

### EQIV  $n  symmetry operation

Defines symmetry operation $n for referencing symmetry equivalent atoms on any instruction which allows atom names, by appending '_$n' (where n is an integer between 1 and 511 inclusive) to the atom name.  Such a symmetry operation must be defined beforeit is used; it does not have to be an allowed operation of the space group, but the same notation is used as on the SYMM instruction.  The same $n may not appear on two separate EQIV instructions. Thus:

```
EQIV $2 1-x, y, 1-z
CONF C1 C2 C2_$2 C1_$2
```

could be used to calculate a torsion angle across a crystallographic twofold axis (note that this may be required because CONF with no atom names only generates torsion angles automatically that involve the unique atom list and a one atom deep shell of symmetry equivalents).  If the instruction codeword refers to a residue, this is applied to the named atoms before any symmetry operation specified with '_$n'. Thus:

```
RTAB_23 O..O OG_12 O_$3
```

would calculate the (hydrogen bond) distance between OG_12 and (O_23)_$3, i.e. between OG in residue 12 and the equivalent obtained by applying the symmetry operation defined by EQIV $3 to the atom O in residue 23.

### OMIT  atomnames

The named atoms are retained in the atom list but ignored in the structure factor calculation and least-squares refinement.  This instruction may be used, together with L.S. 0 and FMAP 2, to create an 'OMIT map' to get a clearer picture of disordered regions of the structure; this concept will be familiar to macromolecular crystallographers.  In particular, 'OMIT $H' can be used to check the hydrogen atom assignment of -OH groups etc.  If an actual peak is present within 0.31 A of the calculated hydrogen atom position, the electron density appears in the 'Peak' column of the output created by PLAN with a negative first parameter. OMIT_* $H must be used for this if residues are employed.

## 7.4 The connectivity list

The connectivity list is a list of 'bonds' that is set up automatically, and may be edited using BIND and FREE. It is used to define idealized hydrogen atom positions, for the BOND and PLAN output of bond lengths and angles, and by the instructions DELU, CHIV, SAME and SIMU. Hydrogen atoms are excluded from the connectivity list (except when introduced by hand using BIND).

**`CONN bmax[12] r[#] atomnames or CONN bmax[12]`**

The CONN instruction fine-tunes the generation of the connectivity table and is particularly useful when π-bonded ligands or metal ions are present in the structure. For the purposes of the connectivity table (which is always generated), bonds are all distances between non-hydrogen atoms less than r1 + r2 + 0.5 Å, where r1 and r2 are the covalent radii of the atoms in question (taking PART into consideration as explained below). A shell of symmetry equivalent atoms is also generated, so that all unique bonds are represented at least once in the list. All bonds, including those to symmetry equivalent atoms, may be deleted or added using the FREE or BIND instructions.

Default values of r (identified by the scattering factor type) are stored in the program. These defaults may be changed (for both the connectivity table AND the PLAN -n output) by using the full form of the SFAC instruction. Alternatively the defaults may be overridden for the named atoms by specifying r on a CONN instruction, in which case r is used in the generation of the connectivity list but not by the PLAN instruction. '$' followed by an element name (the same as on a SFAC instruction) may also be employed on a CONN instruction (and also does not apply to PLAN). The second form of the CONN instruction may be used to change the maximum coordination number bmax for all atoms (which defaults to 12 if there is no CONN instruction).

If, after generating bonds as above and editing with FREE and BIND, there are more than bmax bonds to a given atom, the list is pruned so that only the bmax shortest are retained. A harmless side-effect of this pruning of the connectivity list is that symmetry operations may be stored and printed that are never actually used. Note that this option only removes one entry for a bond from the connectivity list, not both, except in the case of 'CONN 0' which ensures that there are no bonds to or from the named atoms. 'CONN 0' is frequently used to prevent the solvent water in macromolecular structures from making additional 'bonds' to the macromolecule which confuse the generation of idealized hydrogen atoms etc. In some cases it will be necessary to use FREE to remove a 'bond' from a light atom to an alkali metal atom (for example) in order to generate hydrogen atoms correctly. Refinements of macromolecules will often include BUMP and 'CONN 0 O_200 > LAST' (where the water happens to begin with residue 200). 'LAST' is used to indicate the last atom in the file, which saves trouble when adding extra waters.

The CONN instruction, like ANIS and HFIX, MUST precede the atoms to which it is to be applied. Repeated CONN instructions are allowed; the LAST relevant CONN preceding a particular atom is the one which is actually applied. CONN without atom names changes the default value of bmax for all following atoms. The following example illustrates the use of CONN:

```
CONN Fe 0
MPLA 5 C11 > C15 Fe
MPLA 5 C21 > C25 Fe
Fe  .....
C11 .....
.........
C25 .....
```

which would prevent bonds being generated from the iron atom to all 10 carbons in ferrocene. In this example, the distances of the iron atom from the two ring planes would be calculated instead.

### PART  n  sof

The following atoms belong to PART n of a disordered group. The automatic bond generation ignores bonds between atoms with different PART numbers, unless one of them is zero (the value before the first PART instruction). If a site occupation factor (sof) is specified on the PART instruction, it overrides the value on the following atom instructions (even if set via an AFIX instruction) until a further PART instruction, e.g. 'PART 0', is encountered).

If n is negative, the generation of special position constraints is suppressed and bonds to symmetry generated atoms with the same or a different non-zero PART number are excluded; this is suitable for a solvent molecule disordered on a special position of higher symmetry than the molecule can take (e.g. a toluene molecule on an inversion center). A PART instruction remains in force until a further PART instruction is read; 'PART 0' should be used to continue with the non-disordered part of the structure.

Some care is necessary in generating hydrogen atoms where disordered groups are involved. If the hydrogen atoms are assigned a PART number, then even if the atom to which they are attached has no part number (i.e. PART 0) the above rules may be used by the program to work out the correct connectivity for calculating the hydrogen atom positions. HFIX hydrogens are assigned the PART number of the atom to which they are attached. If the hydrogens and the atom to which they are attached belong to PART zero but the latter is bonded to atoms with non-zero PART, the LOWEST of these non-zero PART numbers is assumed to be the major component and is used to calculate the hydrogen positions. In general, if the same residue numbers and names and the same atom names but different PART numbers are used for different disorder components in a macromolecule, HFIX will generate hydrogen atoms correctly without any special action being required. For example the use of HFIX with the following disordered serine residue:

```
HFIX_Ser 33 N
HFIX_Ser 13 CA
HFIX_Ser 23 CB
HFIX_Ser 83 CG
 :
RESI 32 Ser
N  .....
CA .....
C  .....
O  .....
PART 1
CB   1  ...  ...  ...  21  ...
OG   4  ...  ...  ...  21  ...
PART 2
CB   1  ...  ...  ... -21  ...
OG   4  ...  ...  ... -21  ...
```

```
PART 0
```

would set up the AFIX hydrogens as if the following had been input. Note that only one, fully occupied, hydrogen is attached to CA; for this reason, and also to prevent small inconsistencies in the DFIX and DANG restraints, the disorder should be traced back one more atom than can be resolved (i.e. CB should be split even if it does not look as though this would be necessary in an electron density map):

```
RESI 32 Ser
N  .....
AFIX 43
H0   2  ...  ...  ...  11  -1.2
AFIX 0
CA .....
AFIX 13
HA   2  ...  ...  ...  11  -1.2
AFIX 0
C  .....
O  .....
PART 1
CB   1  ...  ...  ...  21  ...
AFIX 23
HB1  2  ...  ...  ...  21  -1.2
HB2  2  ...  ...  ...  21  -1.2
AFIX 0
OG   4  ...  ...  ...  21  ...
AFIX 83
HG   2  ...  ...  ...  21  -1.5
AFIX 0
PART 2
CB   1  ...  ...  ... -21  ...
AFIX 13
HB1  2  ...  ...  ... -21  -1.2
HB2  2  ...  ...  ... -21  -1.2
AFIX 0
OG   4  ...  ...  ... -21  ...
AFIX 83
HG   2  ...  ...  ... -21  -1.5
AFIX 0
PART 0
```

where free variable 2 is the occupation factor for PART 1 (say 0.7) and the occupation factor of the second component is tied to 1-fv(2) (i.e. 0.3). The value for this free variable is set on the FVAR instruction and is free to refine. If there were more than two components, a linear free variable restraint (SUMP) could be used to restrain the sum of occupation factors to unity. The addition of disorder components after including hydrogen atoms will require some hand editing and so is less efficient, but the auxiliary program SHELXPRO can be persuaded to do most of the work

### BIND atom1 atom2

The specified 'bond' (which may be of any length) is added to the connectivity list if it is not there already. Only one of the two atoms may be an equivalent atom (i.e. have the extension _$n).

### FREE atom1 atom2

The specified 'bond' is deleted from the connectivity list (if present). Only one of the two atoms may be an equivalent atom (i.e. have the extension _$n).

## 7.5 Least-squares restraints

**`DFIX d s[0.02] atom pairs`**

The distance between the first and second named atom, the third and fourth, fifth and sixth etc. (if present) is restrained to a target value d with an estimated standard deviation s. d may refer to a 'free variable', otherwise it is considered to be fixed. Fixing d by adding 10 is not allowed, so the value may lie between 0 and 15.

If d is given a negative sign, the restraint is applied ONLY if the current distance between the two atoms is LESS than |d|. This is an 'anti-bumping' restraint, and may be used to prevent solvent (water) molecules from approaching too close to one another or to a macromolecule. Antibumping restraints may also be generated automatically using the BUMP instruction (see below). The default value of s is 0.02. The default s may be changed by means of a preceding DEFS instruction (see below).

**`DANG d s[0.04] atom pairs`**

This instruction is interpreted in exactly the same way as DFIX, but the default value of s is twice the value of the first DEFS parameter (i.e. 0.04 if no DEFS instruction is used). The DFIX and DANG instructions appear separately in the table of restraint statistics. DANG is usually used for 1,3 or ‘angle distances’, i.e. distances between two atoms that are both bonded to the same atom. The distance between the first and second named atom, the third and fourth, fifth and sixth etc. (if present) is restrained to a target value d with an estimated standard deviation s. d may refer to a 'free variable', otherwise it is considered to be fixed. Fixing d by adding 10 is not allowed, so the value may lie between 0 and 15.

**`BUMP s [0.02]`**

'Anti-bumping' restraints are generated automatically for all distances involving two non-bonded C, N, O and S atoms (based on the SFAC type) that are shorter than the expected shortest non-bonded distances, allowing for the possibility of hydrogen bonds. All pairs of atoms that are not connected by one, two or three bonds in the connectivity table are considered to be non-bonded for this purpose. Anti-bumping restraints are also generated for short contacts between hydrogen atoms (if present) provided that the two hydrogen atoms are not bonded to the same atom; this should help to avoid energetically unfavorable side-chain conformations. If the sum of occupancies of the two atoms is less than 1.1, no restraint is generated; also if the atoms have different PART numbers and neither of them is zero no restraint is generated.

The default esd s is the first DEFS parameter (0.02 if there is no DEFS instruction). If s is given a negative sign, the absolute value is used as an esd, and symmetry equivalent atoms in the connectivity array are considered too in deciding which atoms are connected and so should not have anti-bumping restraints applied. Thus when s is positive (the default action if s is not specified on the BUMP instruction) short contacts between appropriate atoms in different asymmetric units ALWAYS result in anti-bumping restraints. This will be the normal procedure for macromolecular refinements (where it helps to eliminate accidental contacts between molecules in low-resolution refinements), but in the (unusual) case of a crystallographic twofold axis running through (say) a disulfide bond it will be necessary to make s negative to prevent the generation of anti-bumping restraints that would break the bond. Refinement with anti-bumping restraints provides a solvent model with acceptable hydrogen bonding distances that is consistent with the diffraction data. The anti-bumping restraints are regenerated before each refinement cycle. Anti-bumping restraints can also be added by hand using DFIX instructions with negative distances d.

**SAME s1[0.02] s2[0.02] atomnames**

The list of atoms (which may include the symbol '>' meaning all intervening non-hydrogen atoms in a forward direction, or '<' meaning all intervening non-hydrogen atoms in a backward direction) is compared with the same number of atoms which follow the SAME instruction. All bonds in the connectivity list for which both atoms are present in the SAME list are restrained to be the same length as those between the corresponding following atoms (with an effective standard deviation s1). The same applies to 1,3 distances (defined by two bonds in the connectivity list which share a common atom), with standard deviation s2. The default value of s1 is taken from the first DEFS parameter; the default value of s2 is twice this. s1 or s2 may be set to zero to switch off the corresponding restraints. The program automatically sets up the n*(n-1)/2 restraint equations required when n interatomic distances should be equal. This ensures optimum efficiency and avoids arbitrary unequal weights. Only the minimum set of restraints needs to be specified in the *.ins* file; redundant restraints are ignored by the program, provided that they have the same sigma values as the unique set of restraints. See also SADI and NCSY for closely related restraints.

The position of a SAME instruction in the input file is critical. This creates problems for programs such as SHELXPRO that provide a user interface to SHELXL, and for protein refinements SADI is to be preferred (e.g. to apply 4m local symmetry to a heme group); normally for proteins most of the 1,2- and 1,3-distances will be restrained to target values using DFIX and DANG respectivelly anyway. However SAME provides an elegant way of specifying that chemically identical but crystallographically independent molecules have the same 1,2 and 1,3 distances, e.g.

```
C1A
:
C19A
SAME C1A > C19A
C1B
:
C19B
SAME C1A > C19A
C1C
:
C19C
```

etc. This requires just n-1 SAME instructions for n equivalent molecules. In a more complicated example, assume that a structure contains several toluene solvent molecules that have been assigned the same atom names (in the same order!) and the same residue name (Tol) but different residue numbers, then one SAME instruction suffices:

```
SAME_Tol C1 > C7
```

This instruction may be inserted anywhere except after the last Tol residue; the program applies it as if it were inserted before the next atom that matches C1_Tol . This is convenient for proteins with repeated non-standard residues, since one command suffices to apply suitable restraints, and no target values are needed, for compatibility with SHELXPRO tis SAME instruction has to be placed before the FVAR instruction. This is an exception to the usual rule that the action of a SAME instruction is position dependent; but it might be best to put it before a toluene residue with good geometry, since the connectivity table for this residue will be used to define the 1,2- and 1,3-distances. In this case it would also be reasonable to impose local two-fold symmetry for each phenyl ring, so a further SAME instruction could be added immediately before one toluene residue (the ring is assumed to be labeled cyclicly C1 .. C6 followed by the methyl group C7 which is attached to C1):

```
SAME C1 C6 < C2 C7
```

which is equivalent to:

```
SAME C1 C6 C5 C4 C3 C2 C7
```

Note that these two SAME restraints are all that is required, however many PHE residues are present; the program will generate all indirectly implied 1,2 and 1,3 equal-distance restraints! In this case it would also be sensible to restrain the atoms of each tolune molecule to be coplanar by a FLAT restraint:

```
FLAT_Tol C1 > C7
```

### SADI s[0.02] atom pairs

The distances between the first and second named atoms, the third and fourth, fifth and sixth etc. (if present) are restrained to be equal with an effective standard deviation s. The SAME and SADI restraints are analyzed together by the program to find redundant and implied restraints. The same effect as is obtained using SADI can also be produced by using DFIX with d tied to a free variable, but the latter costs one more least-squares parameter (but in turn produces a value and esd for this parameter). The default effective standard deviations for SADI may be changed by means of a DEFS instruction before the instruction in question.

### CHIV V[0] s[0.1] atomnames

The chiral volumes of the named atoms are restrained to the value V (in Å$^3$) with standard deviation s. The chiral volume is defined as the volume of the tetrahedron formed by the three bonds to each named atom, which must be bonded to three and only three non-hydrogen atoms in the connectivity list; the (ASCII) alphabetical order of the atoms making these three bonds defines the sign of the chiral volume. Note that RTAB may be used to list chiral volumes defined in the same way but without restraining them. The chiral volume is positive for the alpha-carbon (CA) of an L-amino-acid if the usual names (N, CB and C) are used for the three non-hydrogen atoms bonded to it. It is also possible to define a chiral volume when two substituents are chemically eqivalent but have different names; this may be useful to ensure that CB of a valine retains a pyramidal geometry with the conventional labeling of CG1 and CG2. Note that 'CHIV 0' (or just CHIV since the default V is zero) may be used to impose a planarity restraint on an atom which is bonded to three other non-hydrogen atoms, by making its chiral volume zero. CHIV restraints with zero and non-zero target values are listed separately in the restraints summary printer out after each refinement cycle.

### FLAT s[0.1] four or more atoms

The named atoms are restrained to lie a common plane. This restraint is actually applied by restraining a sufficient number of tetrahedra involving the atoms in question to have (chiral) volumes of zero, using the same algorithm as CHIV. This way of applying a planarity restraint has good convergence properties because it does not fix the orientation of the plane in its current position. s should be given in $Å^3$ as for CHIV, but for comparison with other methods the r.m.s. deviation from the plane is also printed. The default values of s is set by the second DEFS parameter.

**DELU s1[0.01] s2[0.01] atomnames**

All bonds in the connectivity list connecting atoms on the same DELU instruction are subject to a 'rigid bond' restraint, i.e. the components of the (anisotropic) displacement parameters in the direction of the bond are restrained to be equal within an effective standard deviation s1. The same type of restraint is applied to 1,3-distances as defined by the connectivity list (atoms 1, 2 and 3 must all be defined on the same DELU instruction). If s2 is omitted it is given the same value as s1. A zero value for s1 or s2 switches off the corresponding restraint. If no atoms are specified, all non-hydrogen atoms are assumed. DELU is ignored if (in the refinement cycle in question) one or both of the atoms concerned is isotropic; in this case a 'hard' restraint is inappropriate, but SIMU may be used in the usual way as a 'soft' restraint. DELU without atom names applies to all non-hydrogen atoms (in the current residue); DELU_* without atoms applies to all non-hydrogen atoms in all residues. SFAC element names may also be referenced, preceded by the symbol '$'. The default values of s1 and s2 may be changed by means of a preceding DEFS instruction.

**SIMU s[0.04] st[0.08] dmax[1.7] atomnames**

Atoms closer than dmax are *restrained* with effective standard deviation s to have the same $U_{ij}$ components. If (according to the connectivity table, i.e. ignoring attached hydrogens) one or both of the two atoms involved is terminal (or not bonded at all), st is used instead as the esd. If s but not st is specified, st is set to twice s. If no atoms are given, all non-hydrogen atoms are understood. SIMU_* with no atoms applies to all non-hydrogen atoms in all residues. SFAC element names may also be referenced, preceded by '$'. The interatomic distance for testing against dmax is calculated from the atom coordinates without using the connectivity table (though the latter is used for deciding if an atom is terminal or makes no bonds).

Note that SIMU should in general be given a much larger esd (and hence lower weight) than DELU; whereas there is good evidence that DELU restraints should hold accurately for most covalently bonded systems, SIMU (and ISOR) are only rough approximations to reality. s or st may be set to zero to switch off the appropriate restraints.

SIMU is intended for use for larger structures with poorer resolution and data to parameter ratios than are required for full unrestrained anisotropic refinement. It is based on the observation that the $U_{ij}$ values on neighboring atoms in larger molecules tend to be both similar and (when the resolution is poor) significantly correlated with one another. By applying a very weak restraint of this type, we allow a gradual increase and change in direction of the anisotropic displacement parameters as we go out along a side-chain, and we restrain the motion of atoms perpendicular to a planar group (which DELU cannot influence). The use of a distance criterion directly rather than via the connectivity table enables the restraints to be applied automatically to partially overlapping disordered atoms, for which it is an excellent approach. dmax can be set so that coordination distances to metal ions etc. are excluded. Terminal atoms tend to show the largest deviations from equal $U_{ij}$'s and so st should be set higher than s (or made equal to zero to switch off the restraints altogether). SIMU restraints are NOT recommended for SMALL molecules and ions, especially if free rotation or torsion is possible (e.g. $C_5H_5$-groups, $AsF_6$- ions). For larger molecular fragments, the effective rotation angles are smaller, and the assumption of equal $U_{ij}$ for neighboring atoms is more appropriate:

both translation and libration of a large fragment will result in relatively similar $U_{ij}$ components on adjacent atoms. SIMU may be combined with ISOR, which applies a further soft but quite different restraint on the $U_{ij}$ components. SIMU may also be used when one or both of the atoms concerned is isotropic, in which case experience indicates that a larger esd (say 0.1 Å$^2$) is appropriate. The default value of s may be changed by a preceding DEFS instruction (st is then set to twice s).

**`DEFS sd[0.02] sf[0.1] su[0.01] ss[0.04] maxsof[1]`**

DEFS may be used to change the default effective standard deviations for the following DFIX, SAME, SADI, CHIV, FLAT, DELU and SIMU restraints, and is useful when these are to be varied systematically to establish the optimum values for a large structure (e.g. using $R_{free}$). sd is the default for s in the SADI and DFIX instructions, and also for s1 and s2 in the SAME instruction. sf is the default effective standard deviation for CHIV and FLAT, su is the default for both s1 and s2 in DELU, and ss is the default s for SIMU. The default st for SIMU is set to twice the default s.

maxsof is the maximum allowed value that an occupation factor can refine to; occupation factors that are fixed or tied to free variables are not restricted. It is possible to change this parameter (to say 1.1 to allow for hydrogen atoms) when refining both occupation factors and U's for solvent water in proteins (a popular but suspect way of improving the R factor).

**`ISOR s[0.1] st[0.2] atomnames`**

The named atoms are *restrained* with effective standard deviation s so that their $U_{ij}$ components approximate to isotropic behavior; however the corresponding isotropic U is free to vary. ISOR is often applied, perhaps together with SIMU, to allow anisotropic refinement of large organic molecules when the data are not adequate for unrestrained refinement of all the $U_{ij}$; in particular ISOR can be applied to solvent water for which DELU and SIMU are inappropriate. ISOR should in general be applied as a weak restraint, i.e. with relatively large sigmas, for the reasons discussed above (see SIMU); however it is also useful for preventing individual atoms from becoming 'non-positive-definite'. However it should not be used indiscriminately for this purpose without investigating whether there are reasons (e.g. disorder, wrong scattering factor type etc.) for the atom going n.p.d. If (according to the connectivity table, i.e. ignoring attached hydrogens) the atom is terminal (or makes no bonds), st is used instead as the esd. If s but not st is specified, st is set to twice s. If no atoms are given, all non-hydrogen atoms are understood. SFAC element names may also be referenced, preceded by '$'. s or st may be set to zero to switch offthe appropriate restraints. ISOR without atom names (or ISOR_* if residues are used) applies this restraint to all non-hydrogen atoms. Note also the use of the keyword 'LAST' to indicate the last atom in the .ins file; an anisotropic refinement of a macromolecule will often include:

```
ISOR 0.1 O_201 > LAST
```

assuming that the solvent water starts with O_201 and continues until the end of the atom list. ISOR should in general be given a much larger esd (and hence lower weight) than DELU; whereas there is good evidence that DELU restraints should hold accurately for most covalently bonded systems, ISOR (and SIMU) are only rough approximations to reality.

**`NCSY DN sd[0.1] su[0.05] atoms`**

The NCSY instruction applies local non-crystallographic symmetry restraints. In contrast to the widely used global NCS constraints, these do not save any CPU time but do not require the definition (and refinement) of a matrix transformation and mask. They are also very flexible, and can accommodate rotation of the molecule about hinges etc. Since for macromolecules at modest resolution the 1,2- and 1,3-distances are normally restrained to fixed target values by DFIX and DANG restraints, the NCS restraints are generated for equivalent 1,4-distances (if sd is non-zero or absent) and equivalent isotropic U-values (if su is non-zero or absent). The default sd is set to five times the first DEFS parameter, and the default su is equal to the fourth DEFS parameter.

For each atom the program attempts to find an ‘equivalent’ atom with the same name but with a residue number DN greater than the residue number of the named atom. If sd is greater than zero, the connectivity array is used to find 1,4-distances for which both atoms are specified in the same NCSY instruction; a SADI restraint is then created to make the distance equivalent to the same distance between the equivalent atoms. This is not quite the same as restraining torsion angles to be the same, because + and – gauche would have the same distance; however it is chemically plausible that equivalent side-chain conformations could differ in this way. If su is greater than zero (or absent), a SIMU restraint is generated to make the U-values approximately equal for each pair of ‘equivalent’ atoms, provided that both are isotropic. NCS restraints should be used whenever possible for isotropic (protein) refinement at modest resolution, since they increase the effective data to parameter ratio and so have a similar effect to that of increasing the resolution of the data. They are also very easy to set up; for example, to apply three-fold NCS restraints to a protein structure containing three equivalent chains numbered 1001-1109, 2001-2109 and 3001-3109, the following two instructions are all that is required:

```
NCSY 1000 N_1001 > OT2_1109
NCSY 2000 N_1001 > OT2_1109
```

The atom list may easily be modified to leave out particular loops, residues or side-chains. This is not only easier than specifying a transformation matrix and mask: it also will correspond more closely to reality, because the restraints are more flexible than constraints and also act *locally* rather than *globally*.

**SUMP c sigma c1 m1 c2 m2 ...**

The linear restraint: c = c1*fv(m1) + c2*fv(m2) + ... is applied to the specified free variables. This enables more than two atoms to be assigned to a particular site, with the sum of site occupation factors restrained to be a constant. It also enables linear relations to be imposed between distances used on DFIX restraints, for example to restrain a group of atoms to be collinear. sigma is the effective standard deviation. By way of example, assume that a special position on a four-fold axis is occupied by a mixture of sodium, calcium, aluminium and potassium cations so that the average charge is +2 and the site is fully occupied. The necessary restraints and constraints could be set up as follows (the program will take care of the special position constraints on the coordinates and $U_{ij}$ of course):

```
SUMP 1.0 0.01 1.0 2 1.0 3 1.0 4 1.0 5   ! site fully occupied
SUMP 2.0 0.01 1.0 2 2.0 3 3.0 4 1.0 5   ! mean charge = +2
EXYZ Na1 Ca1 Al1 K1                     ! common x, y and z coordinates
EADP Na1 Ca1 Al1 K1                     ! common U or Uij
FVAR ... 0.20 0.30 0.35 0.15     ! starting values for free variables 2..5
...
Na1 ... ... ... ... 20.25 ...    ! 0.25 * fv(2)  [the 0.25 is required for
Ca1 ... ... ... ... 30.25 ...    ! 0.25 * fv(3)  a special position on a
Al1 ... ... ... ... 40.25 ...    ! 0.25 * fv(4)  four-fold axis, i.e. site
K1  ... ... ... ... 50.25 ...    ! 0.25 * fv(5)  symmetry 4]
```

This particular refinement would probably still be rather unstable, but the situation could be improved considerably by adding weak SUMP restraints for the elemental analysis. Such SUMP restraints may be used when elements are distributed over several sites in minerals so that the elemental composition corresponds (within suitable standard deviations) to an experimental chemical analysis.

SUMP may also be applied to BASF, EXTI and BASF parameters, including parameters used to describe twinning (TWIN) and anisotropic scaling (HOPE). The parameters are counted in the order overall scale and free variables, EXTI, then BASF.

## 7.6 Least-squares organization

**`L.S.  nls[0]  nrf[0]  nextra[0]  maxvec[511]`**

nls cycles of full-matrix least-squares refinement are performed, followed by a structure factor calculation. When L.S. (or CGLS) is combined with BLOC, each cycle involves refinement of a block of parameters which may be set up differently in different cycles. If no L.S. or CGLS instruction is given, 'L.S. 0' is assumed.

If nrf is positive, it is the number of these cycles that should be performed before applying ANIS. This two-stage refinement is particularly suitable for the early stages of least-squares refinement; experience indicates that it is not advisable to let everything go at once!

Negative nrf indicates which reflections should be ignored during the refinement but used instead for the calculation of free *R*-factors in the final structure factor summation; for example L.S. 4 –10 would ignore every 10th reflection for refinement purposes. It is desirable to use the same negative value of nrf throughout, so that the values of '*R*1(free)' and 'w*R*2(free)' are not biased by the 'memory' of the contribution of these reflections to earlier refinements. These independent *R*-factors (Brünger, 1992) may be used to calibrate the sigmas for the various classes of restraint, and provide a check as to whether the data are being 'over-refined' (primarily a problem for macromolecules with a poor data to parameter ratio). In SHELXL, these ignored reflections are not used for Fourier calculations.

nrf=–1 selects the $R_{free}$ reference set that is flagged (with negative batch numbers) in the *.hkl* file (SHELXPRO may be used to do this). The division of the data into reference and working set is then independent of the space group and the MERG, OMIT and SHEL settings. However on merging reflections, to play safe a reflection is retained in the reference set only if all equivalents have the $R_{free}$ flag set. Thus if equivalents are present, it is a good idea to use the SHELXPRO option to set the $R_{free}$ flag in thin shells, so that all equivalents of a particular unique reflection are either all in the reference set or all in the working set. nrf=–1 is the recommended way of applying the $R_{free}$ test in SHELXL.

nextra is the number of additional parameters which were derived from the data when performing empirical absorption corrections etc. It should be set to 44 for DIFABS [or 34 without the theta correction; Walker & D. Stuart (1983)]. It ensures that the standard deviations and GooF are estimated correctly; they would be underestimated if the number of extra parameters is not specified. nextra is zero (and so can be omitted) if extra information in the form of indexed crystal faces or psi-scan data was used to apply an absorption correction.

maxvec refers to the maximum number of reflections processed simultaneously in the rate-determining calculations. Usually the program utilizes all available memory to process as many reflections as possible simultaneously, subject to a maximum of maxvec, which may not be larger than 511. For complicated reasons involving the handling of suppressed and '$R_{free}$' reflections and input/output buffering, some blocks may be smaller than the maximum, especially if the facilities for refinement against twinned or powder data are being used. It may be desirable to set maxvec to a smaller number than 511 to prevent unnecessary disk transfers when large structures are refined on virtual memory systems with limited physical memory.

**`CGLS nls[0] nrf[0] nextra[0] maxvec[511]`**

As L.S., but the Konnert-Hendrickson conjugate-gradient algorithm is employed instead of the full-matrix approach. Although BLOC may be used with CGLS, in practice it is much better to refine all parameters at once. CGLS is much faster than L.S. for a large number of parameters, and so will be the method of choice for most macromolecular refinements. The convergence properties of CGLS are good in the early stages (especially if there are many restraints), but cannot compete with L.S. in the final stages for structures which are small enough for full-matrix refinement. The major disadvantage of CGLS is that it does not provide estimated standard deviations, so that when a large structure has been refined to convergence using CGLS it may beworth performing a blocked full-matrix refinement (L.S./BLOC) to obtain the standard deviations in quantities of interest (e.g. torsion angles, in which case only xyz blocks would be required).The other parameters have the same meaning as with L.S.; CGLS is entirely suitable for $R_{free}$ tests (negative nrf), and since it requires much less memory than L.S. there will rarely be any reason to change maxvec from its default value.

The CGLS algorithm is based closely on the procedure described by Hendrickson & Konnert (1980). The structure-factor derivatives contribute only to the diagonal elements of the least-squares matrix, but all 'additional observational equations' (restraints) contribute in full to diagonal and off-diagonal terms, although neither the l.s. matrix A nor the Jacobean J are ever generated. The preconditioning recommended by Hendrickson & Konnert is used to speed up the convergence of the internal conjugate gradient iterations, and has the additional advantage of preventing the excessive damping of poorly determined parameters characteristic of other conjugate gradient algorithms (Tronrud,1992).

A further refinement in the CGLS approach is to save the parameter shifts from the previous CGLS cycle, and to use them to improve the estimated parameter shifts in the current cycle. Since this is only possible in the second and subsequent cycles, an initial shift multiplier of 0.7 is assumed in the first cycle. If the refinement proves to be unstable, this starting value can be reset using the first DAMP parameter.

In addition to this optimization of the CGLS shift multiplication factor, the individual parameter shifts are monitored each L.S. or CGLS cycle, and the shift multiplication factors are reduced (to a value between 0.5 and 1) for parameters that tend to oscillate. This applies only to refinements in which BLOC is not used. This produces an additional improvement in the convergence of the least-squares refinement, but (unlike Marquardt damping) has no effect on esds.

**BLOC n1 n2 atomnames**

If n1 or n2 are positive, the x, y and z parameters of the named atoms are refined in cycle |n1| or |n2| respectively.. If n1 or n2 are negative, the occupation and displacement parameters are refined in the cycle. Not more than two such cycle numbers may be specified on a single BLOC instruction, but the same atoms may be mentioned in any number of BLOC instructions. To refine both x, y and z as well as displacement parameters for an atom in the same block, n1 and n2 should specify the same cycle number, but with opposite signs. A BLOC instruction with no atom names refines all atoms (in residue 0) in the specified cycles. The pattern of blocks is repeated after the maximum block number has been reached if the number of L.S. refinement cycles is larger than the maximum BLOC |n1| or |n2|. If a cycle number less than the maximum |n1| or |n2| is not mentioned in any BLOC instruction, it is treated as full-matrix. The overall scale, batch/twin scale factors, extinction coefficient, SWAT g parameter, HOPE parameters and free variables (if present) are refined in every block. Riding (hydrogen) atoms and atoms in rigid groups are included in the same blocks as the atoms on which they ride.

For example, a polypeptide consisting of 30 residues (residue numbers 1..30 set by RESI instructions) could be refined efficiently as follows (all non-hydrogen atoms assumed anisotropic):

```
BLOC 1
BLOC -2 N_1 > O_16
BLOC -3 N_14 > O_30
```

which would ensure 3 roughly equally sized blocks of about 800 parameters each and some overlap between the two anisotropic blocks to avoid problems where they join. The geometric parameters would refine in cycles 1,4,7 .. and the anisotropic displacement parameters in the remaining cycles. In this example it is assumed that the first atom in each residue is N and the last is O. An alternative good blocking strategy would be to divide the structure into three overlapping blocks of xyz and $U_{ij}$ parameters, and to add a fourth cycle in which all xyz but no $U_{ij}$ values are refined (these four blocks would then also each contain about 800 parameters), i.e.:

```
BLOC 1 -1 N_1 > O_11
BLOC 2 -2 N_10 > O_21
BLOC 3 -3 N_20 > O_30
BLOC 4
```

A BLOC instruction with no parameters fixes all atomic parameters (xyz, sof and U or $U_{ij}$). Such a BLOC instruction takes priority over all other BLOC instructions, irrespective of their order in the *.ins* file.

**DAMP damp[0.7] limse[15]**

The DAMP parameters take different meanings for L.S. and CGLS refinements. For L.S., damp is usually left at the default value unless there is severe correlation, e.g. when trying to refine a pseudo-centrosymmetric structure, or refining with few data per parameter (e.g. from powder data). A value in the range 1-10000 might then be appropriate. The diagonal elements of the least-squares matrix are multiplied by (1+damp/1000) before inversion; this is a version of the Marquardt (1963) algorithm. A side-effect of damping is that the standard deviations of poorly determined parameters will be artificially reduced; it is recommended that a final least-squares cycle be performed with little or no damping in order to improve these estimated standard deviations. Theoretically, damping only serves to improve the convergence properties of the refinement, and can be gradually reduced as the refinement converges; it should not influence the final parameter values. However in practice damping also deals effectively with rounding error problems in the (single-precision) least-squares matrix algebra, which can present problems when the number of parameters is large and/or restraints are used (especially when the latter have small esd's), and so it may not prove possible to lift the damping entirely even for a well converged refinement.

Note the use of 'DAMP 0 0' to estimate esds but not apply shifts, e.g. when a final L.S. 1 job is performed after CGLS refinement.

For CGLS refinements, damp is the multiplicative shift factor applied in the first cycle. In subsequent CGLS cycles it is modified based on the experience in the previous cycles. If a refinement proves unstable in the first cycle, damp should be reduced from its default value of 0.7.

If the maximum shift/esd for a L.S. refinement (excluding the overall scale factor) is greater than limse, all the shifts are scaled down by the same numerical factor so that the maximum is equal to limse. If the maximum shift/esd is smaller than limse no action is taken. This helps to prevent excessive shifts in the early stages of refinement. limse is ignored in CGLS refinements.

**`STIR sres step[0.01]`**

The STIR instruction allows a stepwise improvement in the resolution. In the first refinement cycle, the high-resolution limit (i.e. lowest d) is set at sres, in the next cycle to (sres–step), in the next (sres–2•step) etc. This continues until the limit of the data or the SHEL limit is reached, after which any remaining cycles to complete the number specified by CGLS or L.S. are completed with a constant resolution range. By starting at lower resolution and then gradually improving it, the radius of convergence for models with significant coordinate errors should be increased. This may be regarded as a primitive form of 'simulated annealing'; it could be useful in the early stages of refinement of molecular replacement solutions, or for getting rid of bias for $R_{\mathrm{free}}$ tests (in cases where the solution of the struture was - possibly of ncessity - based on all the data).

**`WGHT a[0.1] b[0] c[0] d[0] e[0] f[.33333]`**

The weighting scheme is defined as follows:

$$w = q / [\ \sigma^2(F_o^2) + (a*P)^2 + b*P + d + e*\sin(\theta)\ ]$$

where P = [ f * Maximum of (0 or $F_o^2$) + (1-f) * $F_c^2$ ]. It is possible for the experimental $F_o^2$ value to be negative because the background is higher than the peak; such negative values are replaced by 0 to avoid possibly dividing by a very small or even negative number in the expression for w. For twinned and powder data, the $F_c^2$ value used in the expression for P is the total calculated intensity obtained as a sum over all components. q is 1 when c is zero, $\exp[c*(\sin(\theta)\lambda)^2]$ when c is positive, and $1 - \exp[c*(\sin(\theta)/\lambda)^2]$ when c is negative.

The use of P rather than (say) $F_o^2$ reduces statistical bias (Wilson 1976). The weighting scheme is NOT refined if a is negative (contrast SHELX-76). The parameters can be set by trial and error so that the variance shows no marked systematic trends with the magnitude of $F_c^2$ or of resolution; the program suggests a suitable WGHT instruction after the analysis of variance. This scheme is chosen to give a flat analysis of variance in terms of $F_c^2$, but does not take the resolution dependence into account. It is usually advisable to retain default weights (WGHT 0.1) until all atoms have been found and the refinement is essentially complete, when the scheme suggested by the program can be used for the next refinement job by replacing the WGHT instruction (if any) by the one output by the program towards the end of the *.res* file. This procedure is adequate for most routine refinements.

It may be desirable to use a scheme which does not give a flat analysis of variance to emphasize particular features in the refinement; for example c = +10 or -10 would weight up data at higher 2θ, e.g. to perform a 'high-angle' refinement (uncontaminated by hydrogen atoms which contribute little at higher diffraction angle) prior to a difference electron density synthesis (FMAP 2) to locate the hydrogens. The exponential weights which are obtained when c is positive were advocated by Dunitz & Seiler (1973). Weighting up the high angle reflections will in general give X-ray atomic coordinates which are closer to those from neutron diffraction.

Refinement against $F^2$ requires different weights to refinement against $F$; in particular, making all the weights equal ('unit weights'), although useful in the initial stages of refinement against F, is NEVER a sensible option for $F^2$. If the program suspects that an unsuitable WGHT instruction has been accidentally retained for a structure which had been refined previously with SHELX-76 or the XLS program in version 4 of the SHELXTL system, it will output a warning message.

**`FVAR  osf[1]  free variables`**

The overall scale factor is followed by the values of the 'free variables' fv(2) ... The overall scale factor is given throughout as the square root of the scale factor which multiplies $F_c^2$ in the least-squares refinement [to make it similar to the scale factor in SHELX-76 which multiplied $F_c$], i.e. $\mathrm{osf}^2 F_c^2$ is fitted to $F_o^2$.

SHELXL goes to some trouble to ensure that the initial value of the scale factor has very little influence. Firstly, if the initial scale is exactly 1.0, a quick structure factor summation with a small fraction of the total number of reflections is performed to estimate a new scale factor. If the values differ substantially then the new value is used. Secondly the scale factor is factored out of the least-squares algebra so that, although it is still refined, the only influence the previous value has is an indirect one via the weighting scheme and extinction correction.

Before calculating electron density maps and the analysis of variance, and writing the structure factor file (*name.fcf*), the observed $F^2$ values and esds are brought onto an absolute scale by dividing by the scale factor.

The free variables allow extra constraints to be applied to the atoms, e.g. for common site occupation factors or isotropic displacement parameters, and may be used in conjunction with the SUMP, DFIX and CHIV restraints. If there is more than one FVAR instruction, they are concatenated; they may appear anywhere between UNIT and HKLF (or END).

## 7.7 Lists and tables

The esds in bond lengths, angles and torsion angles, chiral volumes, Ueq, and coefficients of least-squares planes and deviation of atoms from them, are estimated rigorously from the full correlation matrix (an approximate treatment is used for the angles between least-squares planes). The errors in the unit-cell dimensions (specified on the ZERR instruction) are taken into account exactly in estimating the esds in bond lengths, bond angles, torsion angles and chiral volumes. Correlation coefficients between the unit-cell dimensions are ignored except when determined by crystal symmetry (so that for a cubic crystal the cell esds contribute to errors in bond lengths and chiral volumes but not to the errors in bond angles or torsion angles). The (rather small) contributions of the unit-cell errors to the esds of quantities involving least-squares planes are estimated using an isotropic approximation.

For full-matrix refinement, the esds are calculated after the final refinement cycle. In the case of BLOC'ed refinement, the esds are calculated after every cycle (except that esds in geometric parameters are not calculated after pure Uij/sof cycles etc.), and the maximum estimate of each esd is printed in the final tables. This prevents some esds being underestimated because not all of the relevant atoms were refined in the last cycle, but at the cost of overestimating all the esds if the R-factor drops appreciably during the refinement. Thus large structures should first be refined almost to convergence (either by CGLS or L.S./BLOC), and then a separate final blocked refinement job performed to obtain the final parameters and their esds. It is important that there is sufficient overlap between the blocks to enable every esd to be estimated with all contributing atoms refining in at least one of the refinement cycles.

**BOND atomnames**

BOND outputs bond lengths for all bonds (defined in the connectivity list) that involve two atoms named on the same BOND instruction. Angles are output for all pairs of such bonds involving a common atom. Numerical parameters on a BOND instruction are ignored, but not treated as errors (for compatibility with SHELX-76). A BOND instruction with no parameters outputs bond lengths (and the corresponding angles) for ALL bonds in the connectivity table, and 'BOND $H' on its own includes all bonds to hydrogens as well (but since the hydrogens are not included in the connectivity table, bonds involving symmetry equivalent hydrogens are not included). Other element names may also be referenced globally by preceding them with a '$' on a BOND instruction. BOND is set automatically by ACTA, and the bond lengths and angles are written to the *.cif* file. Note that the best way to calculate B-H-B angles is with RTAB !

**CONF atomnames**

The named atoms define a chain of at least four atoms. CONF generates a list of torsion angles with esd's for all torsion angles defined by this chain. CONF is often used to specify an n-membered ring, in which case the first three atoms must be named twice (n+3 names in all). If no atoms are specified, all possible torsion angles not involving hydrogen are generated from the connectivity array. The torsion angles generated by CONF are also written to the .cif file if an ACTA instruction is present. All torsion angles calculated by SHELXL follow the conventions defined by Allen & Rogers (1969).

**MPLA na atomnames**

A least-squares plane is calculated through the first na of the named atoms, and the equation of the plane and the deviations of all the named atoms from the plane are listed with estimated standard deviations (from the full covariance matrix). The angle to the previous least-squares plane (if any) is also calculated, but some approximations are involved in estimating its esd. na must be at least 3. If na is omitted the plane is fitted to all the atoms specified.

**`RTAB codename atomnames`**

Chiral volumes (one atomname), bonds (two), angles (three) and torsion angles (four atomnames) are tabulated compactly against residue name and number. codename is used to identify the quantity being printed; it must begin with a letter and not be longer than 4 characters (e.g. 'Psi' or 'omeg'). There may not be more than 4 atom names. It is assumed that the atoms have the same names in all the required residues. For chiral volumes only, the necessary bonds must be present in the connectivity list (the same conventions are employed as for CHIV). Since the atoms do not themselves have to be in the same residue (it is sufficient that the names match), the residue name (if any) is printed as that of the first named atom for distances, the second for angles, and the third in the case of torsion angles. The latter should be consistent with generally accepted conventions for proteins. A typical application of RTAB for small-molecule structures is the tabulation of hydrogen-bonded distances and angles (with esd's) since these will not usually appear in the tables created automatically by BOND. For an example of this see the 'sigi' test job in chapter 3.

If RTAB refers to more than one residue (e.g. RTAB_*), it is ignored for those residues in which not all the required atoms can be found (e.g. some of the main chain torsional angles for the terminal residues in a protein).

**`HTAB dh[2.0]`**

The new HTAB instruction provides an analysis of the hydrogen bonds. A search is made over all polar hydrogens (i.e. hydrogen bonded to electronegative elements) present in the structure, and hydrogen bonds printed for which: **H•••A < r(A)+dh** and **<DHA > 110º**. If it appears likely that the hydrogens have been assigned wrongly (e.g. two -OH groups have been assigned to the same O•••O vector) a suitable warning message appears. This output should be checked carefully, since the algorithms used by HFIX/AFIX to place hydrogens are by no means infallible! To obtain esd's on the distances and angles involved in the hydrogen bond, the second form of the HTAB instruction (and if necessary EQIV) should be used (see below); HTAB without atom names is used first to find the necessary symmetry transformations for EQIV..

**`HTAB  donor-atom  acceptor-atom`**

The second form of the HTAB instruction is required to generate the esds and the CIF output records. The donor atom D and acceptor A should be specified; the program decides which of the hydrogen atoms (if any) makes the most suitable hydrogen bond linking them. Only the acceptor atom may specify a symmetry operation (_$n) because this standard CIF entry for publication in Acta Crystallographica requires this.

**`LIST m[#] mult[1]`**

**m = 0**: No action.

**m = 1**: Write *h,k,l*, $F_o$, $F_c$ and phase (in degrees) to .fcf in X-PLOR format. Only unique reflections after removing systematic absences, scaling [to an absolute scale of F(calc)], applying dispersion and extinction or SWAT corrections (if any), and merging equivalents including Friedel opposites are included. If $F_o^2$ was negative, $F_o$ is set to zero. Reflections suppressed by OMIT or SHEL [or reserved for R(free)] are not included.

**m = 2**: List *h,k,l*, $F_o$, $\sigma(F_o)$ and phase angle in degrees in FORMAT(3I4,2F8.2,I4) for the reflection list as defined for m = 1.

**m = 3**: List *h,k,l*, $F_o$, $\sigma(F_o)$, A(real) and B(imag) in FORMAT(3I4,4F8.2), the reflections being processed exactly as for m = 2.

**m = 4**: List *h,k,l*, $F_c^2$, $F_o^2$, $\sigma(F_o^2)$ and a one-character status flag. $F_o^2$ are scaled to $F_c^2$ and possibly corrected for extinction, but no corrections have been made for dispersion and no further merging has been performed. FORMAT (3I4,2F12.2,F10.2,1X,A1) is employed. The status flag is 'o' (observed), 'x' [observed but suppressed using 'OMIT *h k l*', SHEL or reserved for R(free)], or '<' ($F_o^2$ is less than t.$\sigma(F_o^2)$, where t is one half of the *F*-threshold s specified on an OMIT instruction).

**m = 5**: Write *h,k,l*, $F_o$, $F_c$, and $\phi$ (phase angle in degrees) in FORMAT(3I4,2F10.2,F7.2) for the reflection list as defined for m = 1. Like the m = 1 option, this is intended for input to somestandard macromolecular FFT programs (such as W. Furey's PHASES program), thereby providing a possible route to a graphical display of the electron density.

**m = 6**: Write a free-format CIF file containing *h,k,l*, $F_o^2$, $\sigma(F_o^2)$, $F_c$ and $\phi$ (phase angle in degrees) for the reflection list as defined for m = 1. This is the recommended format for the deposition of reflection data with the PDB, and is also the format required for the generation of refinement statistics and electron density maps using SHELXPRO.

For m = 4 only, mult is a constant multiplicative factor applied to all the quantities output (except the reflection indices!), and may be used if there are scaling problems. For other m options mult is ignored. For m = 2,3 or 4 only a blank line is included at the end of the file as a terminator. The reflection list is written to the file *name.fcf*, which is in CIF format for n = 3, 4 or 6; however the actual reflections are always in fixed format except for n = 1 or 6. The program CIFTAB can - amongst other options - read the m = 4 output and print $F_o/F_c/\sigma(F)$ tables in compact form on an HP-compatible laser printer. n = 4 is the standard archive format for small-molecule structures, n = 6 for macromolecules (with Friedel opposites averaged). Since the final refinement is normally performed on all data (including the $R_{free}$ reference set) the LIST 6 output is not able to flag the $R_{free}$ reflections.

**ACTA 2thetafull[#]**

A 'Crystallographic Information File' file *name.cif* is created in self-defining STAR format. This ASCII file is suitable for data archiving, network transmission, and (with suitable additions) for direct submission for publication. ACTA automatically sets the BOND, FMAP 2, PLAN and LIST 4 instructions, and may not be used with other FMAP or LIST instructions or with a positive OMIT s threshold. A warning message appears if the cell contents on the UNIT instruction are not consistent with the atom list, because they are used to calculate the density etc. which appears in the *.cif* output file.

2thetafull is used to specify the value of 2θ for which the program calculates the completeness of the data for the CIF output file as required by Acta Crystallographica. If no value is given, the program uses the maximum value of 2θ for the reflection data. If the data were collected to a specific limiting 2θ, or if a limit was imposed using SHEL, this would be a good choice. Otherwise the choice of 2thetafull is a difficult compromise; if it is too low, the paper will be rejected because the resolution of the data is not good enough; if it is higher, the lower completeness might lead to rejection by the automatic Acta rejection software! SHELXL calculates the completeness by counting reflections after merging Friedel opposites and eliminating systematic absences (and the reflection 0,0,0).

**`SIZE dx dy dz`**

dx, dy and dz are the three principal dimensions of the crystal in mm, as usually quoted in publications. This information is written to the *.cif* file. If a SIZE instruction is present in the *.ins* file, SHELXL uses it to write the estimated minimum and maximum transmission to the *.cif* file. This should give order of magnitude estimates that should be replaced by the values from the actual absorption correction if these were applied. The empirical SHELXL estimates take into account that most of the diffraction from strongly absorbing crystals takes place at the edges and corners; these estimates of the actual absorption of the crystal may be a little smaller than those from psi-scan and other semi-empirical routines that include absorption by the mounting fibre and glue or oil.

**`TEMP T[20]`**

Sets the temperature T of the data collection in degrees Celsius. This is reported to the *.cif* file and used to set the default isotropic U values for all atoms. TEMP must come before all atoms in the *.ins* file. TEMP also sets the default X-H bond lengths (see AFIX) which depend slightly on the temperature because of librational effects. The default C-H bond lengths and default U-values are rounded to two decimal places so that they may be quoted more easily.

**`WPDB n[1]`**

Writes the refined coordinates to a *.pdb* file. If n is positive hydrogen atoms are omitted; if |n| is 1 all atoms are converted to isotropic and ATOM statements generated, and if |n| is 2 ANISOU statements are also generated (but the equivalent B value is still used on the ATOM statement). The atom names and residue classes and numbers should conform to PDB conventions. This provides a direct link to X-PLOR and other programs which use (more of less) the official (Brookhaven) dialect of the PDB format. Note that SHELXPRO can be used to extend the PDB output file to include refinement details etc. (from the .lst file) for deposition with the PDB, and also to modify disordered residues so that they can be interpreted by programs such as O that cannot read the full standard PDB format.

## 7.8 Fouriers, peak search and lineprinter plots

**`FMAP code[2] axis[#] nl[53]`**

The unique unit of the cell for performing the Fourier calculation is set up automatically unless specified by the user using FMAP and GRID; the value of axis must be non-zero to suppress the automatic selection. The program chooses a 53 x 53 x nl or 103 x 103 x nl grid depending on the resolution of the data. axis is1, 2 or 3 to define the direction perpendicular to the layers. Dispersion corrections are applied (so that the resulting electron density is real) and

Friedel opposites are merged after the least-squares refinement and analysis of variance but before calculating the Fourier synthesis. This will improve the map (and bring the maximum and minimum residual density closer to zero) compared with SHELX-76. In addition, since usually all the data are employed, reflections with $\sigma(F)$ relatively large compared with $F_c$ are weighted down. This should be better than the use of an arbitrary cutoff on $F_o/\sigma(F)$. The rms fluctuation of the map relative to the mean density is also calculated; in the case of a difference map this gives an estimate of the 'noise level' and so may be used to decide whether individual peaks are significant. Usually FMAP 2 is employed to find missing atoms, but if a significant part of the structure is missing, FMAP 5 or 6 may be better. ACTA requires FMAP 2 so that the difference density is on an absolute scale.

If code is made negative, both positive and negative peaks are included in the list, sorted on the absolute value of the peak height. This is intended to be useful for neutron diffraction data.

**code = 2:** Difference electron density synthesis with coefficients ($F_o$–$F_c$) and phases $\phi$(calc).

**code = 3:** Electron density synthesis with coefficients $F_o$ and phases $\phi$(calc).

**code = 4:** Electron density synthesis with coefficients (2$F_o$–$F_c$) and phases $\phi$(calc). $F$(000) is included in the Fourier summations for code = 3 and 4.

**code = 5:** Sim-weighted (2m$F_o$-$F_c$) Fourier (Giacovazzo, 1992).

**code = 6:** Sim-weighted (2m$F_o$-$F_c$) Fourier with coefficients sharpened by multplying with $\sqrt{E/F}$.

**GRID sl[#] sa[#] sd[#] dl[#] da[#] dd[#]**

Fourier grid, when not set automatically. Starting points and increments multiplied by 100. s means starting value, d increment, l is the direction perpendicular to the layers, a is across the paper from left to right, and d is down the paper from top to bottom. Note that the grid is 53 x 53 x nl points, i.e. twice as large as in SHELX-76, and that sl and dl need not be integral. The 103 x 103 x nl grid is only available when it is set automatically by the program (see above).

**PLAN npeaks[20] d1[#] d2[#]**

If npeaks is positive a Fourier peak list is printed and written to the *.res* file; if it is negative molecule assembly and line printer plots are also performed. Distances involving peaks which are less than r1+r2+d1 (the covalent radii r are defined via SFAC; 1 and 2 refer to the two atoms concerned) are printed and used to define 'molecules' for the line printer plots. Distances involving atoms and/or peaks which are less than r1+r2+|d2| are considered to be 'non-bonded interactions'; however distances in which both atoms are hydrogen or at least one is carbon (recognised by SFAC label 'C') are ignored. These non-bonded interactions are ignored when defining molecules, but the corresponding atoms and distances are included in the line printer output. Thus an atom or peak may appear in more than one map, or more than once on the same map. A table of the appropriate coordinates and symmetry transformations appears at the end of each molecule.

Negative d2 includes hydrogen atoms in the line printer plots, otherwise they are left out (but included in the distance tables). For the purposes of the PLAN instruction, a hydrogen atom is one with a radius of less than 0.4 Å. Peaks are assigned the radius of SFAC type 1, which is usually set to carbon. Peaks appear on the printout as numbers, but in the .res file they are given names beginning with 'Q' and followed by the same numbers. Peak heights are also written to the .res file (after the sof and dummy U values) in electrons Å$^{-3}$. See also MOLE for forcing molecules (and their environments) to be printed separately.

A default npeaks of +20 is set by FMAP; to obtain line printer plots, an explicit PLAN instruction with negative npeaks is required. If npeaks is positive the nearest unique atoms to each peak are tabulated, together with the corresponding distances. A table of shortest distances between peaks is also produced. For macromolecules and for users of the Siemens' SHELXTL system npeaks will almost always be positive! If npeaks is positive d1 and d2 have a different meaning. The default of d1 is then -1 and causes the full peaklist to appear in the .res file. If it is positive (say 2.3) then the full peaklist is still printed in the *.lst* file, but only suitable candidates for (full occupancy) water molecules appear in the *.res* file (with SFAC 4 and U set to 0.75). The water molecules must be less than 4 Å from an atom which begins with 'O', 'N' or 'W', and may not be nearer than d2 (default 3.0) from any atom which does not begin with 'O', 'N', 'W' or 'H', and may not be nearer than d1 to any 'O', 'N' or 'W' atom or to other potential waters which have larger peak heights. This facility is intended for extending the water structure of proteins in connection with BUMP and SWAT. To include the waters in the next refinement job, their names need to be changed and they need to be moved to before the HKLF instruction at the end of the atom list in the new *.ins* file. This can be performed automatically using SHELXPRO. It is recommended that the last water be called 'LAST' on the ISOR and CONN instructions so that its name does not need to be updated each job.

The heights and positions of the highest (difference) electron density maximum and the deepest minimum are output irrespective of the PLAN parameters.

**MOLE n**

Forces the following atoms, and atoms or peaks that are bonded to them, into molecule n of the PLAN output. n may not be greater than 99. n = 99 has a special meaning: the 'lineprinter plot' is suppressed for the following atoms, but the table of distances is still printed. This is sometimes useful for saving paper.

# 8. Strategies for Macromolecular Refinement

SHELXL is designed to be easy to use and general for all space groups and uses a conventional structure-factor calculation rather than a FFT summation; the latter would be faster, but in practice involves some small approximations and is not very suitable for the treatment of dispersion or anisotropic thermal motion. The price to pay for the extra generality and precision is that SHELXL is much slower than programs written specifically for macromolecules, but this is to some extent compensated for by the better convergence properties, reducing the amount of manual intervention required (and also the *R*-factor).

Recent advances in cryogenic techniques, area detectors, and the use of synchrotron radiation enable macromolecular data to be collected to higher resolution than was previously possible. In practice this tends to complicate the refinement because it is possible to resolve finer details of the structure; it is often necessary to model alternative conformations, and in a few cases even anisotropic refinement is justified. Although SHELXL provides a number of other features not found in many macromolecular refinement programs, it is probably the flexible treatment of disorder and the facilities for restrained anisotropic refinement that are most likely to be of immediate interest to macromolecular crystallographers.

An auxiliary program SHELXPRO (Chapter 9) is provided as an interface to other macromolecular programs. SHELXPRO is able to generate an *.ins* file from a PDB format file, including the appropriate restraints etc. SHELXPRO can also generate map files for the program O and can display the refinement results in the form of Postscript plots, as well as including the updated coordinates in the *.ins* file for the next refinement.. SHELXL produces PDB and CIF format files that can be read by SHELXPRO and used for archiving.

### 8.1 The radius of convergence

A crucial aspect of any macromolecular refinement program is the radius of convergence. A larger radius of convergence reduces the amount of time-consuming manual intervention using interactive graphics. Many claims that SHELXL gives *R*-factors one or two percent lower than other programs have been tracked down either to subtle differences in the model or to not getting trapped in local minima. The differences in the model include the treatment of diffuse solvent and hydrogen atoms, and the ability to refine common occupancies for disordered groups. The inclusion of dispersion terms and the use of a conventional rather than a FFT structure factor summation are also more precise; the approximations in the FFT summation may become significant for high resolution data and atoms with small displacement parameters. There are probably a number of contributing factors to the good convergence typically observed for SHELXL, e.g. the refinement against ALL data, the inclusion of important off-diagonal terms in the least-squares algebra, the ability to refine all parameters at once (i.e. coordinates and displacement parameters in the same cycle), and the restriction to unimodal restraint functions; multimodal restraint functions such as torsion angles or hydrogen bonds tend to increase the number of spurious local minima. It is much better to reserve the multimodal chemical information such as torsion angles for verifying the structure with an independent program such as PROCHECK (Laskowski, MacArthur, Moss & Thornton, 1993), and to use the unimodal information as restraints. The errors in the FFT calculation of derivatives are larger that those in the structure factors (for the same grid intervals); this would also impede convergence.

## 8.2 Residues

Macromolecular structures are conventionally divided up into *residues*, for example individual amino-acids. In SHELXL residues may be referenced either individually, by '_' followed by the appropriate residue number, or as all residues of a particular class, by '_' followed by the class. For example 'DFIX 2.031 SG_9 SG_31' could be used to restrain a disulfide distance between two cystine residues, whereas 'FLAT_PHE CB > CZ' would apply planarity restraints to all atoms between CB and CZ inclusive in all PHE (phenylalanine) residues. Plus and minus signs refer to the next and previous residue numbers respectively, so 'DFIX_* 1.329 C_– N' applies a bond length restraint to all peptide bonds ('_*' after the command name applies it to all residues). This way of referring to atoms and residues is in no way restricted to proteins; it is equally suitable for oligonucleotides, polysaccharides, or to structures containing a mixture of all three. It enables the necessary restraints and other instructions to be input in a concise and relatively self-explanatory manner. These instructions are checked by the program for consistency and appropriate warnings are printed.

## 8.3 Constraints and restraints for macromolecules

The lower data to parameter ratio for macromolecules makes the use of constraints and especially restraints essential. Rigid group constraints enable a structure to be refined with very few parameters, especially when the (thermal) displacement parameters are held fixed (BLOC 1). After a structure has been solved by molecular replacement using a rather approximate model for the whole protein or oligonucleotide, it may well be advisable to divide the structure up into relatively rigid domains (using a few AFIX 6 and AFIX 0 instructions) and to refine these as rigid groups, initially for a limited resolution shell (e.g. SHEL 8 3), then stepwise extending the resolution, e.g. using the STIR instruction. Restraints may still be required to define flexible hinges and prevent the units from flying apart. In view of the small number of parameters and the high correlations introduced by rigid group refinement, L.S. (full-matrix refinement) should be used for this stage (but CGLS will be necessary for the subsequent refinement). After this initial step which exploits the large convergence radius of rigid group refinement, in general the more flexible restraints will be used in preference to constraints for the rest of the refinement.

SHELXL provides distance, planarity and chiral volume restraints, but not torsion angle restraints or specific hydrogen bond restraints. For oligonucleotides, good distance restraints are available for the bases (Taylor & Kennard, 1982), but for the sugars and phosphates it is probably better to assume that chemically equivalent 1,2- and 1,3-distances are equal (using the SAME and SADI restraints) without the need to specify target values. In this way the effect of the pH on the protonation state of the phosphates and hence the P-O distances does not need to be predicted, but it is assumed the whole crystal is at the same pH. For proteins, since some amino-acid residues occur only a small number of times in a given protein, it is probably better to use 1,2- and 1,3-target distances based on the study of Engh and Huber (1991); these are employed in the restraints added by SHELXPRO to the *.ins* file.

The three bonds to a carbonyl carbon atom can be restrained to lie in the same plane by means of a *chiral volume restraint* (Hendrickson & Konnert, 1980) with a target volume of zero (e.g. 'CHIV_GLU 0 C CD' to restrain the carbonyl and carboxyl carbons in all glutamate residues to have planar environments). The planarity restraint (FLAT) restrains the chiral volumes of a sufficient number of atomic tetrahedra to zero; in addition the r.m.s. deviation of the atoms from the best planes is calculated. Chiral volume restraints with non-zero targets are useful to prevent the inversion of $\alpha$-carbon atoms and the $\beta$-carbons of Ile and Thr, e.g. 'CHIV_ILE 2.5 CA CB'. It is also useful to apply chiral volume restraints to non-chiral atoms such as CB of valine and CG of leucine in order to ensure conformity with conventional atom-labeling schemes (from the point of view of the atom names, these atoms could be considered to be chiral !).

*Anti-bumping restraints* are distance restraints that are only applied if the two atoms are closer to each other than the target distance. They can be generated automatically by SHELXL, taking all symmetry equivalent atoms into account. Since this step is relatively time-consuming, in the 1993 release it was performed only before the first refinement cycle, and the anti-bumping restraints were generated automatically only for the solvent (water) atoms (however they could be inserted by hand for any pairs of atoms). In practice this proved to be too limited, so in later releases the automatic generation of anti-bumping restraints was extended to all C, N, O and S atoms (with an option to include H•••H interactions) and was performed each refinement cycle. Anti-bumping restraints are not generated automatically for (a) atoms connected by a chain of three bonds or less in the connectivity array (unless separated by more than a specified number of residues), (b) atoms with different non-zero PART numbers, and (c) pairs of atoms for which the sum of occupancies is less than 1.1. The target distances for the O...O and N...O distances are less than for the other atom pairs to allow for possible hydrogen bonds.

## 8.4 Restrained anisotropic refinement

There is no doubt that macromolecules are better described in terms of anisotropic displacements, but the data to parameter ratio is very rarely adequate for a free anisotropic refinement. Such a refinement often results in 'non-positive definite' (NPD) displacement tensors, and at the best will give probability ellipsoids that do not conform to the expected dynamical behavior of the molecule. Clearly constraints or restraints must be applied to obtain a chemically sensible model. It is possible to divide a macromolecule up into relatively rigid domains, and to refine the 20 TLS parameters of rigid body motion for each domain (Driessen, Haneef, Harris, Howlin, Khan & Moss, 1989). This is a good model for the bases in oligonucleotides and for the four aromatic side-chains in proteins, but otherwise macromolecules are probably not sufficiently rigid for the application of TLS constraints, or they would have to be divided up into such small units that too many parameters would be required. As with the refinement of atomic positions, restraints offer a more flexible approach.

The *rigid bond restraint* (DELU) assumes that the components of the anisotropic displacement parameters (ADPs) along bonded (1,2-) or 1,3-directions are zero within a given esd. This restraint should be applied with a low esd, i.e. as a 'hard' restraint. Didisheim & Schwarzenbach (1987) showed that for many non-planar groups of atoms, rigid bond restraints effectively impose TLS conditions of rigid body motion. Although rigid-bond restraints involving 1,2- and 1,3-distances reduce the effective number of free ADPs per atom from 6 to less than 4 for typical organic structures, further restraints are often required for the successful anisotropic refinement of macromolecules.

The *similar ADP restraint* (SIMU) restrains the corresponding $U_{ij}$-components to be approximately equal for atoms which are spatially close (but not necessarily bonded because they may be in different components of a disordered group). The isotropic version of this restraint has been employed frequently in protein refinements. This restraint is consistent with the characteristic patterns of thermal ellipsoids in many organic molecules; on moving out along side-chains, the ellipsoids become more extended and also change direction gradually.

Neither of these restraints are suitable for isolated solvent (water) molecules. A linear restraint (ISOR) restrains the ADP's to be *approximately isotropic*, but without specifying the magnitude of the corresponding equivalent isotropic displacement parameter. Both SIMU and ISOR restraints are clearly only approximations to the truth, and so should be applied as 'soft' restraints with high esds. When all three restraints are applied, structures may be refined anisotropically with a much smaller data to parameter ratio, and still produce chemically sensible ADP's. Even when more data are available, these restraints are invaluable for handling disordered regions of the structure.

Constraints and restraints greatly increase the radius and rate of convergence of crystallographic refinements, so they should be employed in the early stages of refinement wherever feasible. The difference electron density syntheses calculated after such restrained refinements are often more revealing than those from free refinements. In large small-molecule structures with poor data to parameter ratios, the last few atoms can often not be located in a difference map until an anisotropic refinement has been performed with geometrical and ADP restraints. Atoms with low displacement parameters that are well determined by the X-ray data will be relatively little affected by the restraints, but the latter may well be essential for the successful refinement of poorly defined regions of the structure. Premature removal or softening the restraints (to improve the *R*-value !) is probably the most common cause of unstable macromolecular refinements with SHELXL.

## 8.5 The free *R*-factor

The questions of whether the restraints can be removed in the final refinement, or what the best values are for the corresponding esds, can be resolved elegantly by the use of $R_{free}$ (Brünger, 1992). To apply this test, the data are divided into a working set (about 95-90% of the reflections) and a reference set (about 5-10%). The reference set is only used for the purpose of calculating a conventional *R*-factor that is called $R_{free}$. It is very important that the structural model is not in any way based on the reference set of reflections, so these are left out of ALL refinement and Fourier map calculations. If the original model was in any way derived from the same data, then many refinement cycles are required to eliminate memory effects. This ensures that the *R*-factor for the reference set provides an objective guide as to whether the introduction of additional parameters or the weakening of restraints has actually improved the model, and not just reduced the *R*-factor for the data employed in the refinement ('*R*-factor cosmetics'). The best way to set up the $R_{free}$ test is to use SHELXPRO to flag reflections in the *.hkl* file for use in the reference set, and to set the second CGLS parameter to '-1'. If NCS or twinning is anticipated, it is advisable to use the 'thin shells' method of flagging the reflections for $R_{free}$ in SHELXPRO.

$R_{\mathrm{free}}$ is invaluable in deciding whether a restrained anisotropic refinement is significantly better than an isotropic refinement. Experience indicates that both the resolution and the quality of the data are important factors, but that restrained anisotropic refinement is unlikely to be justified for crystals that do not diffract to better than 1.5 Å. An ensemble distribution created by molecular dynamics is an alternative to the harmonic description of anisotropic motion (Gros, van Gunsteren & Hol, 1990; Clarage & Phillips, 1994), and may be more appropriate for structures with severe conformational disorder that do not diffract to high resolution.

Despite the overwhelming arguments for using $R_{\mathrm{free}}$ to monitor macromolecular refinements, it is only a single number, and is itself subject to statistical uncertainty because it is based on a limited number of reflections. Thus $R_{\mathrm{free}}$ may be insensitive to small structural changes, and small differences in $R_{\mathrm{free}}$ should not be taken as the last word; one should always consider whether the resulting geometrical and displacement parameters are *chemically reasonable*. The final refinement and maps should always be calculated with the ***full*** data, but without introducing additional parameters or changing the weights of the restraints. $R_{\mathrm{free}}$ is most useful for establishing refinement protocols; for a series of closely similar refinements (e.g. for mutants to similar resolution) the $R_{\mathrm{free}}$ tests only need to be applied to the first.

## 8.6 Disorder in macromolecules

To obtain a chemically sensible refinement of a disordered group, we will probably need to constrain or restrain a sum of occupation factors to be unity, to restrain equivalent interatomic distances to be equal to each other or to standard values (or alternatively apply rigid group constraints), and to restrain the displacement parameters of overlapping atoms. In the case of a tight unimodal distribution of conformations, restrained anisotropic refinement may provide as good a description as a detailed manual interpretation of the disorder in terms of two or more components, and is much simpler to perform. With high-resolution data it is advisable to make the atoms anisotropic BEFORE attempting to interpret borderline cases of side-chain disorder; it may well be found that no further interpretation is needed, and in any case the improved phases from the anisotropic refinement will enable higher quality difference maps to be examined.

Typical warning signs for disorder are large (and pronounced anisotropic) apparent thermal motion (in such cases the program may suggest that an atom should be split and estimate the coordinates for the two new atoms), residual features in the difference electron density and violations of the restraints on the geometrical and displacement parameters. This information in summarized by the program on a residue by residue basis, separately for main-chain, side-chain and solvent atoms. In the case of two or more discrete conformations, it is usually necessary to model the disorder at least one atom further back than the maps indicate, in order that the restraints on the interatomic distances are fulfilled. The different conformations should be assigned different PART numbers so that the connectivity array is set up correctly by the program; this enables the correct rigid bond restraints on the anisotropic displacement parameters and idealized hydrogen atoms to be generated automatically even for disordered regions (it is advisable to model the disorder before adding the hydrogens).

Several strategies are possible for modeling disorder with SHELXL, but for macromolecules the simplest is to include all components of the disorder in the same residues and use the same atom names, the atoms belonging to different components being distinguished only by their different PART numbers. This procedure enables the standard restraints etc. to be used unchanged, because the same atom and residue names are used. No special action is needed to add the disordered hydrogen atoms, provided that the disorder is traced back one atom further than it is visible (so that the hydrogen atoms on the PART 0 atoms bonded to the disordered components are also correct). Note that this very simple and effective treatment of disorder was not available in the original 1993 release of SHELXL.

### 8.7 Automatic water divining

It is relatively common practice in the refinement of macromolecular structures to insert water molecules with partial occupancies at the positions of difference electron density map peaks in order to reduce the *R*-factor (another example of ' *R*-factor cosmetics'). Usually when two different determinations of the same protein structure are compared, only the most tightly bound waters, which usually have full occupancies and smaller displacement parameters, are the same in each structure. The refinement of partial occupancy factors for the solvent atoms (in addition to their displacement parameters) is rarely justified by $R_{\text{free}}$, but sometimes the best $R_{\text{free}}$ value is obtained for a model involving some water occupancies fixed at 1.0 and some at 0.5.

Regions of diffuse solvent may be modeled using *Babinet's principle* (Moews & Kretsinger, 1975); the same formula is employed in the program TNT, but the implementation is somewhat different. In SHELXL it is implemented as the SWAT instruction and usually produces a significant but not dramatic improvement in the agreement of the very low angle data. Anti-bumping restraints may be input by hand or generated automatically by the program, taking symmetry equivalents into account. After each refinement job, the displacement parameters of the water molecules should be examined, and waters with very high values (say *U* greater than 0.8 Å$^2$, corresponding to a *B* of 63) eliminated. The $F_o$-$F_c$ map is then analyzed automatically to find the highest peaks which involve no bad contacts and make at least one geometrically plausible hydrogen bond to an electronegative atom. These peaks are then included with full occupancies and oxygen scattering factors in the next refinement job. This procedure is repeated several times; in general $R_{\text{free}}$ rapidly reaches its minimum value, although the conventional *R*-index continues to fall as further waters are added. It should be noted that the automatic generation of anti-bumping restraints is less effective when the water occupancies are allowed to have values other than 1.0 or 0.5. This approach provides an efficient way of building up a chemically reasonable (but not necessarily unique) network of waters that are prevented from diffusing into the protein, thus facilitating remodeling of disordered side-chains etc. The occupancies of specific waters may also be tied (using free variables) to the occupancies of particular components of disordered side-chains where this makes chemical sense. This procedure may be facilitated by using SHELXPRO to convert the *.res* output file from one refinement job to the *.ins* file for the next, or fully automated using the program SHELXWAT that calls SHELXL repeatedly. A similar but much more sophisticated approach (ARP) described by Lamzin & Wilson (1993) may also be used in conjunction with SHELXL.

### 8.8 Refinement of structures at modest resolution

Although the unique features of SHELXL are primarily useful for refinement against very high resolution data, tests indicated that only small changes would be required to the original release to extend its applicability to medium resolution data (say 2.5Å). The most important of these changes were improved diagnostics and more sophisticated anti-bumping restraints (see above), and the addition of non-crystallographic symmetry (NCS) restraints. The use of NCS restraints considerably improves the effective data to parameter ratio, and the resulting Fourier maps often look as though they were calculated with higher resolutiondata than were actually used (because the phases are more accurate). Two types of NCS restraint may be generated automatically with the help of the NCSY instruction. The first type uses the connectivity table to define equivalent 1,4-distances, which are then restrained to be equal. The second restrains the isotropic *U*-values of equivalent atoms to be equal. It is not normally necessary to restrain equivalent 1,2- and 1,3-distances to be equal because the DFIX and DANG restraints will have this effect anyway; but SAME may be used to add such restraints in the absence of DFIX and DANG. The use of *restraints* rather than applying NCS as an exact constraint (e.g. in the structure factor calculation) is more flexible (but slower) and does not require the specification of transformation matrices and real-space masks. Experience indicates that NCS restraints should be used wherever possible; it is not difficult to relax them later (e.g. for specific side-chains involved in interactions with other non-NCS related molecules) should this prove to be necessary.

### 8.9 A typical SHELXL refinement using high resolution data

An example of a typical SHELXL refinement against high resolution data, for an inhibited form of serine protease, is summarized in Table 5.2. Data were collected at 120 K on a synchrotron to 0.96 Å resolution with an overall mean I/ of 15.2 and a $R_{\text{merge}}$ (based on intensities) of 3.7%. Molecular dynamics refinement using X-PLOR (Brünger, Kuriyan & Karplus, 1987) from initial *R*-values of 42.5% produced the results shown as Job 1. A reference set consisting of 10% of the reflections and a working set of the remaining 90% were used throughout the X-PLOR and SHELXL refinement. The final X-PLOR and initial SHELXL refinements were performed with the resolution range restricted to 1.1 to 8 Å (48495 working set reflections) to save computer time. 10 conjugate gradient cycles were performed in each of the SHELXL refinement jobs; where new atoms were introduced they were always refined isotropically for 2 cycles before making them anisotropic. The CPU times (150 MHz Silicon Graphics R4400 processor) varied from 6.1 hours for job 2 to 21.7 for job 13. The weighting scheme was fixed at 'WGHT 0.2' until jobs 12 and 13, where the two-parameter scheme with values suggested by the program was employed. The restraints (DEFS 0.015 0.2 0.01 0.025) were made tighter than usual to make the refinements more comparable with X-PLOR; the mean distance deviation was 0.009 Å for X-PLOR and 0.014 Å for the final SHELXL job.

Introduction of the diffuse solvent parameter in job 3 (which started from the same parameters as job 2) was not significant, although usually it reduces $R_{\text{free}}$ by about 0.5%; probably this was a consequence of leaving out the low angle data at this stage. Making all atoms anisotropic resulted in almost a 3% drop in $R_{\text{free}}$, but from experience of similar structures we believe that the drop would have been larger if all the data had been used at this stage. This helps to explain the further drop in $R_{\text{free}}$ on using all the reflection data (job 8), and the fact that the difference between $R_1$ and $R_{free}$ was about 3% for jobs 4 to 7 and about 2% for the remaining jobs. Particularly noteworthy is the drop in the $R$-factors on introducing hydrogens (no extra parameters); a parallel job using exactly the same model but excluding hydrogens showed that 1.25% of the drop in $R_{free}$ was contributed by the hydrogens. On the other hand the drop in job 12 is caused almost entirely by the improvements to the model; the same job with the original weights gave an $R_{free}$ of 10.90%. After using $R_{free}$ to monitor the refinement as discussed here, a final refinement was performed against all 80102 unique reflections without any further changes to the model; this converged to $R_1$ = 8.77%, essentially identical to the final $R_1$ for the working set.

**SHELXL refinement of a serine protease (188 residues)**

| Job | Action taken | NP | NH | NW/NW½/NX | $N_{par}$ | $R_1$ | $R_{free}$ |
|---|---|---|---|---|---|---|---|
| 1 | Final X-PLOR, 1.1-8Å | 1337 | 0 | 176 / 0 / 19 | 6129 | 19.47 | 21.14 |
| 2 | Same atoms, SHELXL | 1337 | 0 | 176 / 0 / 19 | 6129 | 17.15 | 18.96 |
| 3 | SWAT added | 1337 | 0 | 176 / 0 / 19 | 6130 | 17.07 | 18.95 |
| 4 | All atoms anisotropic | 1337 | 0 | 176 / 0 / 19 | 13790 | 12.96 | 16.10 |
| 5 | Disorder, added solvent | 1376 | 0 | 207 / 0 / 34 | 14565 | 11.46 | 14.20 |
| 6 | More disorder and solvent | 1422 | 0 | 214 / 2 / 39 | 14831 | 11.35 | 14.22 |
| 7 | Disorder, half occ. waters | 1447 | 0 | 213 / 20 / 37 | 15478 | 11.13 | 14.10 |
| 8 | Resolution: 0.96Å-Inf. | 1447 | 0 | 218 / 28 / 37 | 15595 | 10.75 | 12.95 |
| 9 | Riding Hydrogens added | 1451 | 1088 | 220 / 38 / 40 | 15769 | 9.58 | 11.56 |
| 10 | Minor adjustments | 1477 | 1052 | 222 / 48 / 40 | 16114 | 9.15 | 11.19 |
| 11 | Minor adjustments | 1491 | 1042 | 211 / 64 / 48 | 16173 | 9.29 | 11.31 |
| 12 | Weighting changed | 1491 | 1029 | 222 / 84 / 38 | 16357 | 8.74 | 10.85 |
| 13 | Further refinement | 1499 | 1025 | 212 / 96 / 38 | 16353 | 8.76 | 10.79 |

NP = Number of protein atoms (including partially occupied atoms), NH = Number of hydrogens (all fully occupied), NW = Number of fully occupied waters, $NW_{½}$ = Number of half occupied waters, NX = Number of other atoms (inhibitor, formate, glycerol, some of them partially occupied), and $N_{par}$ = Number of least-squares parameters.

## 8.10 Summary of useful SHELXL keywords for macromolecular refinement

The more important keywords for macromolecular refinement are summarized in the following table (* indicates significant changes from SHELXL-93):

| | |
|---|---|
| **DEFS** | **Set global restraint esd defaults.** |
| **DFIX** | **Restrain 1,2-distance to target (which may be a free variable).** |
| **DANG*** | **Restrain 1,3-distance to target (which may be a free variable).** |
| **SADI** | **Restrain distances to be equal without specifying target.** |
| **SAME** | **Generate SADI automatically for 1,2- and 1,3-distances using connectivity.** |
| **CHIV** | **Restrain chiral volume to target (default zero; may be a free variable).** |
| **FLAT*** | **Planarity restraint.** |
| **DELU** | **Generate rigid bond $U_{ij}$ restraints automatically using connectivity.** |
| **SIMU** | **Generate similar $U$ (or $U_{ij}$) restraints automatically using distances.** |
| **ISOR** | **'Approximately isotropic' restraints.** |
| **BUMP*** | **Generate anti-bumping restraints automatically (incl. symm. equivalents).** |
| **NCSY*** | **Generate non-crystallographic symmetry restraints.** |
| **FVAR** | **Starting values for overall scale factor and free varaibles.** |
| **SUMP** | **Restrain linear combination of free variables.** |
| **PART** | **Atoms with different non-zero PART numbers not connected by program.** |
| **AFIX** | **Riding H, rigid groups and other constraints on individual atoms.** |
| **HFIX** | **Generate hydrogens and suitable AFIX instructions for their refinement.** |
| **MERG** | **'MERG 4' averages equiv. reflns., incl. Friedel opp., sets all $f$ " to 0.** |
| **SHEL** | **Set maximum and minimum resolution (data ignored outside range).** |
| **STIR*** | **Stepwise increase of resolution.** |
| **SWAT** | **Refine diffuse solvent parameter (Babinet's principle).** |
| **WGHT** | **Weighting scheme, probably best left at default 'WGHT 0.1' throughout.** |
| **CGLS** | **No. of cycles conjugate gradient least-squares, select $R_{free}$ reflections.** |
| **BLOC, L.S.** | **Blocked-matrix least-squares (for esds).** |
| **RTAB, MPLA, HTAB*** | **Tables of bonds, angles, torsions, planes, H-bonds etc.** |
| **WPDB*, ACTA, LIST*** | **Output PDB and CIF files for archiving and data transfer..** |

I am grateful to Thomas R. Schneider for providing the serine protease example and helping with this chapter.

# 9. SHELXPRO: Protein Interface to SHELX-97

A new program **SHELXPRO** has been added as an interactive user interface between SHELXL and other programs often used by protein crystallographers. It is designed to be self-explanatory so that it can be used without constant reference to a manual. It is started by:

```
shelxpro name
```

When started, SHELXPRO creates a log file *name.pro* and a Postscript output file *name.ps*. These may be printed after exiting from SHELXPRO and provide text and graphical summaries of the operations performed. Many options in SHELXPRO expect that the files *name.lst*, *name.fcf*, *name.pdb*, *name.res* etc. have been generated in a SHELXL job using the LIST 6 and WPDB instructions. A menu of possible options is displayed by SHELXPRO; choosing a particular option by typing the appropriate letter (upper or lower case) produces a detailed description of that option, after which the user has the choice of typing <Enter> to continue or N<Enter> to return to the menu. The menu consists of:

```
[F] New output filename                    [V] R(free) files
[A] Anisotropic scaling (Hope & Parkin)    [I] .ins from PDB file
[P] Progress of LS refinement diagram      [L] Luzzati plot
[T] Thermal displacement analysis          [E] Esd analysis
[U] Update .res (and .pdb) to .ins file    [N] NCS analysis
[R] Ramachandran Phi-Psi plot              [K] Kleywegt NCS plot
[M] Map file for O from .fcf               [O] PDB file for O
[H] .hkl file from other data formats      [B] PDB deposition
[D] Convert DENZO/SCALEPACK .sca to .hkl   [C] Color plots (now on)
[X] Write XTALVIEW map coefficients        [W] Write Turbo-Frodo map
[S] Reflection statistics from .fcf        [Z] Least-squares fit
[G] Generate PDB file from .res or .pdb    [Q] Quit

Enter option:
```

The various options will now be discussed individually. Several of them add Postscript plots to the file *name.ps*. In these plots, the main-chain atoms are often color-coded according to the secondary structure, which the user is prompted for (blue for alpha-helix, green for beta-sheet and red for others). The side-chains are often color-coded according to residue characteristics:

| | |
|---|---|
| Yellow | = Cys, Met |
| Green | = Phe, Tyr, Trp, His |
| Cyan | = Gly, Ala, Leu, Ile, Val, Pro |
| Red | = Glu, Asp |
| Blue | = Arg, Lys |
| Purple | = Gln, Asn |
| Gray | = Ser, Thr |

## 9.1 Outline of the available features

The options provided by SHELXPRO can be divided into three general groups.

*(a) Files and communication with other protein programs*

[H] *.hkl* file from other data formats. This provides general interactive reformatting of reflection data files, avoiding the need to write a FORTRAN program or UNIX shell-script each time it is necessary to reformat reflection data.

[D] Convert DENZO/SCALEPACK *.sca* to *.hkl*. This is often the safest and quickest way of generating the *.hkl* reflection data file for SHELXL, SHELXS etc.

[V] R(free) files. This adds an $R_{free}$ flag to selected reflections in an *.hkl* file; they may be chosen at random or in thin shells. This is the preferred method of calculating a *free R-factor* using SHELXL, and requires the SHELXL instructions CGLS n –1 or L.S. n –1.

[I] *.ins* from PDB file. This will normally be used when a structure is transferred from another program to SHELXL for the first time. It generates most of the restraints and other extra instructions automatically as well as converting the atoms to fractional coordinates in SHELX format. For editing and updating between SHELXL refinement cycles the following [U] option should be used instead.

[U] Update *.res* (and *.pdb*) to *.ins* file. This should be used to read the *.res* output file from a SHELXL refinement job and update it to create the *.ins* input file for the next job. Alterations such as extra residues or disorder components may be added from a PDB format file written by a graphics program such as O or XtalView.

[G] Generate PDB file from *.res* or *.pdb*. Although SHELXL can write a PDB format file directly, this option provides for more user interaction, e.g. for setting up a PDB format file containing symmetry equivalents or modified temperature factors for use with molecular replacement programs such as AMoRe.

[B] PDB deposition. Collects the information needed for PDB deposition from the *.lst* and *.pdb* files written by SHELXL and creates a file according to the current specifications for deposition with the Brookhaven PDB. The resulting file contains all the compulsory records, but still requires some hand editing e.g. to include information about the data collection..

[F] New output filename. New *.ps* and *.pro* files are started and the previous *.ps* and *.pro* files closed. This enables the Postscript plots to be viewed in another window without leaving SHELXPRO etc.

[C] Color plots (now on). This option toggles color on or off in the Postscript output files. For some journals it may be necessary to produce black and white diagrams rather than color.

[Q] Quit. Terminates SHELXPRO and returns to the command line prompt.

*(b) Creation of map (and pdb) files for various graphics packages*

[M] Map file for O from *.fcf*. This creates a map file that can be read by O and some versions of FRODO. A variety of maps may be created, including Sigma-A maps. SHELXPRO reads the *.fcf* file written by SHELXL (it contains calculated structure factors and phases) and the *.pdb* file (in order to work out the extent of the map).

[W] Write Turbo-Frodo map. Very similar to the corresponding option for O.

[O] PDB file for O. The otherwise exemplary program O is unfortunately not able to read standard PDB format files (as written by e.g. SHELXL) when they contain disordered groups. This option provides a (not very elegant) work-around.

[X] Write XtalView map coefficients. Writes a *.phs* file with coefficients for various types of map including Sigma-A maps for input to XtalView. XtalView should be instructed to calculate an $F_o$-map whatever type of map is actually required! This produces MUCH better maps than inputting the atoms from SHELXL as a *.pdb* file into XtalView and repeating the structure factor calculation in XtalView (because of various incompatibilities such as the solvent model, anisotropic temperature factors, complex scattering factors as well as approximations made by XtalView in the structure factor calculation).

### *(c) Analysis of a structure after refinement with SHELXL*

[P] Progress of LS refinement diagram. Produces a diagram of the *R*-factor as a function of the refinement cycle, with special action for automated water divining (SHELXWAT). The *R*-factors are extracted from the REM instructions in the current *.res* file, which are accumulated there when the U option in SHELXPRO is used to update the *.res* file written by one refinement job to create the *.ins* file for the next.

[T] Thermal displacement analysis. Creates bar-plots to show the variation of B-value (and anisotropy) with residue number for main-chain and side-chain atoms.

[R] Ramachandran Phi-Psi plot. A Ramachandran plot is created and the outliers listed. Reads the *.lst* file that must contain the necessary torsion angles calculated in SHELXL using RTAB instructions.

[K] Kleywegt NCS plot. A Kleywegt plot is a Ramachandran plot with NCS-related residues joined by straight lines. The lines cross the edges of the plot and reappear at the other side if necessary. If the plot is too hairy you may be in trouble..

[N] NCS analysis. Creates bar-plots of differences in B-values and various torsion angles between NCS related monomers. These are read from the *.lst* file so the torsion angles should have been calculated using RTAB instructions in SHELXL.

[S] Reflection statistics from *.fcf*. *R*-factors, data completeness, mean(I/sigma) etc. may be calculated for user-specified resolution ranges.

[L] Luzzati plot. Similar to [S] but the resolution ranges are fixed by the program and a Luzzati plot of *R*-factor against resolution is created as well as the statistics.

[E] Esd analysis. Graphical analysis of the esds estimated by a (blocked) full-matrix refinement using SHELXL.

[Z] Least-squares fit. Allows parts of one or more structures to be fitted to each other and r.m.s. deviations calculated. The deviations may be plotted against residue number as bar plots and superimposed structures may be output in suitable format for preparing diagrams with MOLSCRIPT or the XP program in SHELXTL.

[A] Anisotropic scaling (Hope & Parkin). Applies an anisotropic scaling analysis to the .fcf file output from SHELXL using LIST 6. It is similar to the action of the HOPE instruction in SHELXL, but is much faster. This instruction may be used as a quick check to see whether the introduction of the HOPE instruction would be justified.

## 9.2 Communication with other programs

The various options will now be described in more detail. Much of this information is provided by the program when an option is chosen. This section contains useful information on the best ways of using SHELXL for protein refinements.

*[H] .hkl file from other data formats*

The program can read a variety of reflection data file formats and write a *.hkl* file in SHELX *.hkl* format. If the original file contained *F*-values, the *.hkl* file should be read into SHELXL with HKLF 3; if the original file contained intensities, HKLF 4 is appropriate. The input file should contain one reflection per line, but lines may be stripped from the beginning and end, e.g. to process data transferred by email. On reading the file, the first line is displayed. To skip this line and move to the next, hit the <Enter> key. To read *h,k,l, F* (or $F^2$) and $\sigma(F)$ [or $\sigma(F^2)$] from this and subsequent lines in free format, enter the character ∗ followed by <Enter>; to read in fixed format, fill the positions under these quantities with H,K,L,F or S. Thus to read a correctly formatted *.hkl* file, enter the line:

```
HHHHKKKKLLLLFFFFFFFFSSSSSSSS
```

For technical reasons, the following option [D] should always be used instead of [H] to read files produced by SCALEPACK.

*[D] Convert DENZO/SCALEPACK .sca to .hkl*

The SCALEPACK *.sca* and SHELXL *.hkl* formats look very similar, but there are some subtle differences. The .sca file has three lines of header information but *.hkl* has no header. The *.hkl* file may be terminated by a line with all items zero that is not present in the .sca file; however both are also terminated by the end of the file. Unlike *.hkl*, the *.sca* file may contain floating-point numbers in 'I8' format. If the 'anomalous' flag was applied, the .sca file may contain reflections $\mathbf{h}_+$ and $\mathbf{h}_-$ on the same line, with dummy values if not measured. The [D] option handles these differences and may also be used to extract anomalous $\Delta F$ values (with esds) for heavy-atom location using Patterson or direct methods in SHELXS.

*[V] R(free) files*

This command is used to flag say 5 or 10% of the reflections in the *.hkl* file for use as a reference set in calculating free *R*-values (Brünger, 1992). As a rule of thumb, at least 500 reflections or 5% of the total number should be flagged, whichever is larger. It is difficult to obtain statistically meaningful free *R*-values for datasets containing a total of less than 5000 reflections before division into reference and working sets. The flag is applied by making the 'batch number' at the end of each line in the *.hkl* file negative. The unflagged reflections constitute the working set. The *.hkl* file is read into SHELXL in the normal way using HKLF 4 (or 3), and the flags are ignored (i.e. all reflections are used for refinement and no free *R* is calculated) unless the second number on the CGLS (or L.S.) instruction is –1, in which case only the working set is used for the refinement, and only the reference set is used to calculate the free *R*-values. It is customary to perform the final refinement using all the data, but not increasing the number of independent parameters or reducing the weights of the restraints. This may be done by simply deleting the second number on the CGLS or L.S. instruction.

The reference set may either be chosen at random or in thin shells. The latter option is strongly recommended if a twinned structure is being refined or if NCS restraints are applied, because otherwise the reference and working sets will not be independent. When the reflections are averaged in SHELXL, they are included in the final reference set only if all contributors have the $R_{\text{free}}$ flag set, otherwise they are used in the working set. In such a case it is advisable to use thin shells rather than flagging the reflections at random, otherwise there will not be many reflections left in the reference set after averaging!

Note that if the second CGLS (or L.S.) parameter is negative (–N) with N not equal to 1, SHELXL will generate its own reference set consisting of every N'th reflection (after merging) irrespective of the flags in the *.hkl* file. This possibility is retained for upwards compatibility with SHELXL-93, but is NOT RECOMMENDED, because the reference set may possibly change if a different space group. resolution range, merging procedure or a different version of SHELXL is used, and because it is inappropriate for problems involving NCS or twinning.

*[I] .ins from PDB file*

Usually, when SHELXL is used for a high-resolution refinement, a low-resolution or preliminary refinement will already have been performed with another program, or a model will be available from molecular replacement or map interpretation in the form of a PDB file. SHELXPRO can read PDB files taken from the Brookhaven database as well as files written by X-PLOR and other widely used protein programs. The [I] option incorporates standard Engh & Huber (1991) restraints, and other instructions needed for a refinement job, into the .ins file. The program applies some consistency checks and searches for disulfide bridges, generating the necessary restraints automatically. The user may renumber the residues and must specify the residue numbers for N- and C-termini so that appropriate action can be taken. Since SHELXL does not recognize chains, these must be flagged by adding (e.g.) 1000, 2000, ... to the residue numbers (note that the [B] and [G] options in SHELXPRO provide the reverse transformation). It is advisable to ignore hydrogen atoms in the input PDB file because it is better to regenerate and refine them using the riding model in SHELXL.

It is almost inevitable that some hand editing of the resulting *.ins* file will still be necessary. For example, SHELXPRO is not able to define restraints, torsion angles and hydrogen atoms for residues that it doesn't recognize. Bad initial geometry may require the addition of FREE or BIND instructions so that the connectivity array is generated correctly by SHELXL, and chain breaks, ligands or solvent molecules other than water may require special action. The [I] option, followed by any necessary hand editing, should be used once per structure before the first SHELXL refinement. Thereafter it is much more convenient to use the [U] option in SHELXPRO to update the *.res* file from one refinement job to produce the *.ins* file for the next., because special restraints and other instructions are retained, and because there are extra facilities for defining and checking disorder, solvent molecules, etc. The restraints incorporated into the *.ins* file are stored internally in SHELXPRO, so no dictionary file is required (in contrast to the now obsolete program PDBINS supplied with SHELXL-93, which used a dictionary file *shelxl.dic*).

*[U] Update .res (and .pdb) to .ins file*

This option converts a SHELXL .res file to a new *.ins* file by including new or changed atoms from PDB format files such as those written by the graphics programs O, Turbo-Frodo and XtalView. All other SHELXL commands are retained unchanged. This option also provides for setting up disorder refinement and updating the list of solvent molecules. The .res file should not contain instructions other than RESI, AFIX, PART and atoms between FVAR and HKLF, and both FVAR and HKLF must be present. Note that although it is possible to set up threefold or multiple disorders in this way, the necessary SUMP restraints must be edited into the .ins file later by hand; no extra editing is needed for twofold disorders. The [U] option may also be used without a *.pdb* file to update the *.res* file to *.ins* and apply various checks. It is recommended that the *.res* file is always updated to .ins in this way rather than by using an editor, so that the REM records that contain a summary of the course of the refinement are accumulated correctly; if necessary the resulting *.ins* file can then be edited further with a text editor before rerunning SHELXL.

*[G] Generate PDB file from .res or .pdb*

The WPDB instruction in SHELXL is normally used to write PDB format files, but the [G] option in SHELXPRO provides additional editing facilities that are particularly useful for the creation of PDB format files for use as molecular replacement search models, and are also sometimes useful before calculating least-squares fits etc An *.ins*, *.res* or PDB format file serves as input. B-values may be reset automatically to typical values, disordered atoms, solvent molecules and H-atoms may be removed, chain ID's (not recognized by SHELXL) may be (re)inserted, and multiple copies of chains may be generated using (non-)crystallographic symmetry. In the resulting PDB file all atoms are isotropic.

*[B] PDB deposition*

The [B] option reads files *.pdb* and *.lst* files written by the 'final' SHELXL refinement job and creates a file with the default extension *.ent* in PDB format suitable for deposition in Brookhaven. Some of this file is in the form of a template suitable for hand editing, e.g. to include literature references, experimental details, special features of the structure and refinement, etc. The user is prompted for details of chains and possible renumbering of the residues; except for structures consisting of a single chain, chain ID's should be (re)inserted in this way before deposition. The resulting file should contain all the compulsory records, but some of them will need completion by subsequent hand editing. The following notation is used to redefine residue numbers and chains. When prompted by the program, the new chain ID letter (the character '$' should be used if a blank chain ID is required) is followed by the first and last old residue numbers and the first new residue number. One chain should be specified per input line, and the list of chains is terminated by a blank line. Thus if there were two chains numbered 1001-1189 and 2001-2189, followed by waters with residue numbers 1-111, the following three lines should be entered:

```
A 1001 1189 1
B 2001 2189 1
$ 1 111 201
```

For example, residue 1001 in this example would become chain A residue 1. Similarly, residue 2189 becomes chain B residue 189. The solvent water that used to start at residue 1 now starts at residue 201.

For the deposition of reflection data, the CIF format *.fcf* file written by SHELXL may be used directly.

## 9.3 Creation of map (and pdb) files for various graphics packages

In a computer utopia, interactive graphics packages would all read the CIF format *.fcf* file written by SHELXL directly; this contains all the information necessary for generating maps. For the couple of years before this comes to pass, SHELXPRO provides the necessary generation of maps or (in the case of XtalView) coefficients. For the programs O and Turbo-Frodo, it is also necessary to define the region of space for which the map is calculated; SHELXPRO does this by scanning a PDB file to find the maximum and minimum atomic coordinates in each direction. Furthermore, O is liable to be confused by disordered residues even if these are specified exactly according to the PDB rules (as SHELXL does), so it is also necessary for SHELXPRO (option [O]) to be able to modify the PDB file so that all disorder components are given separate residue numbers. Note that the option [U] provides the reverse procedure, i.e. separate residues obtained using O may be recombined as different disorder components of the same residue for refinement using SHELXL. SHELXPRO does not make the changes that may be required to the *all.dat* connectivity file read by O.

The [M], [O], [W] and [X] options should be self-explanatory. The following questions are asked by the program; usually the answers suggested by the program are suitable, so most of the questions are answered by <Enter>.

Name of .fcf file created using SHELXL and LIST 6 [name.fcf]:

Enter name of PDB file [name.pdb]:

Include all waters in the volume covered by map? [Y]:

Number of grid points per cell in x, y and z (the first two MUST be powers of 2, and the last MUST be a multiple of 8) [64 64 88]:

Origin of map along x, y and z (grid points) [-32 -24 24] (must all be multiples of 8):

Extent of map along x, y and z (grid points) [128 136 88] (must all be multiples of 8):

Fourier type (-3=mFo-DFc (Sigma-A difference map), -2=2mFo-DFc (Sigma-A map), -1=Fo-Fc, 0=Fc, 1=Fo, 2=2Fo-Fc, n=nFo-(n-1)Fc [-2]:

Enter reference/working set Sigma-A ratio from SHELXL [0.97]:

Apply sharpening (Y or N) ? [N]:

Enter name of map file [sigmaa.map]:

For XtalView, the questions about the grid are skipped. Note that there is a choice of maps. Thus the input '3' for the Fourier type generates a $3F_o$-$2F_c$ map; '4' gives a $4F_o$-$3F_c$ map, etc. The sigma-A ratio is calculated in each SHELXL job that uses the free *R*-factor; it is designed to correct the sigma-A weight for overfitting. For refinement at low resolution this might be about 0.8, for medium resolution 0.9; the default is appropriate for structures with a high ratio of data to parameters. If the free *R*-factor was not used in the refinement, a estimated value should be input. 'Sharpening' multiplies the coefficients by $<F^2>^{1/4}$, where $<F^2>$ is the mean reflection intensity in the appropriate resolution shell (this factor is used in preference to the almost identical factor $\sqrt{(E/F)}$ because the latter involves a statistical factor for certain reflections that is inappropriate for this application). Finally, the program outputs the maximum and minimum electron density (in sigma units as well as -electrons per cubic Ångstöm) and electron density histogram.

Note that XtalView MUST be told to do an $F_o$ synthesis, whatever type of map the coefficients actually represent !

## 9.4 Analysis of the refined structure

The *.lst* file produced by SHELXL contains a great deal of important information, but for proteins (in contrast to small molecules) it is not very economical to print it out and read it after every job. Many of the following options are designed to summarize the essential information in more digestible form, e.g. as Postscript plots. Usually the *.lst* and/or *.fcf* and sometimes the *.res* or *.pdb* files are required from a SHELXL refinement job in which the LIST 6, FMAP 2 and WPDB instructions were employed.

*[P] Progress of LS refinement diagram*

At the end of each refinement job, and after each SHELXL stage in the SHELXWAT water divining procedure, SHELXL outputs three lines of remarks to the *.res* file containing current R-values etc. If the *.res* file is edited to the next *.ins* file in such a way as to retain these remarks, they provide a convenient summary of the course of the refinement. The remarks are written after the HKLF instruction so they must be moved ahead of this instruction in order to be preserved; if the [U] option is used to update from *.res* to *.ins* this happens automatically. The [P] option extracts the *R*-factors from these remarks and prepares a Postscript plot of *R*-factor against refinement job number. Points that were part of the SHELXWAT water divining procedure are plotted with a smaller horizontal gap between them. This plot provides a convenient summary of the course of refinement; it can be seen at a glance which stage produced the biggest drop in free *R*-factor, and whether *R* continues to fall but the free *R*-factor rises again, indicating over-refinement.

*[T] Thermal displacement analysis*

This reads a SHELXL *.lst* file from an isotropic or anisotropic refinement and prepares Postscript bar plots of the mean (equivalent) B and (optionally) anisotropy (minimum eigenvalue divided by maximum eigenvalue) against residue number. The refinement should have been performed with FMAP 2, so that the residue diagnostics table is present in the *.lst* file. Unless black and white Postscript output is set, the main-chain plots are color coded according to secondary structure (it is useful to run PROCHECK first to obtain this information) and the side-chain plots by residue type. The color schemes are defined in the .pro output file.

Alpha-helices and beta-strands are entered one per line with 'A n1 n2' or 'B n1 n2' respectively, where n1 and n2 are the first and last residues of the helix or strand. The letters may be upper or lower case. The list is terminated with a blank line. Thus:

```
a 21 45
b 48 55
a 67 108
```

would define two alpha-helices (residues 21 to 45 and 67 to 108 resp.) and one beta-strand (48 to 55). The alpha-helix regions are colored blue, the beta-strands green, and the rest red. There may be up to four diagrams on one page, starting at the top. Each should be defined by entering three characters: a symbol to label the diagram, then either B (B-values) or A (anisotropy), followed by M (main-chain) or S (side-chain) and then the numbers of the first and last residues. END terminates the list. The program will suggest suitable parameters. A typical sequence, selecting these defaults by <Enter> each time, would be:

```
Next diagram [aBM 1 204]:
Maximum value and step for vertical scale [50 10]:
Next diagram [bAM 1 204]:
Next diagram [cBS 1 204]:
Maximum value and step for vertical scale [60 10]:
Next diagram [dAS 1 204]:
```

Note that no scale needs to be specified for the anisotropy, because the range is always from 0 to 1.

### *[R] Ramachandran Phi-Psi plot*

The [R] option reads the SHELXL .lst output file and extracts the psi and phi torsion angles to make Ramachandran plots. If the main-chain is disordered, only the PART 1 (and of course PART 0) atoms are used. Glycines are included optionally as open squares; prolines are treated as normal residues. A list of outliers appears on the screen and in the *.pro* file. Residues are color-coded according to residue type unless black and white Postscript has been specified (option [C] in the main menu). The refinement should have been performed with appropriate RTAB instructions for the phi and psi torsion angles and with FMAP 2, so that the residue diagnostics table is present in the .lst file. See Kleywegt & Jones (1996), who kindly provided the distribution table used in SHELXPRO.

*[K] Kleywegt NCS plot*

This is the same as the normal Ramachandran plot (option [R] above) except that the phi/psi dots for each residue are smaller and residues related by non-crystallographic symmetry (NCS) are joined by lines (Kleywegt, 1996). The lines may cross the edges of the plot and reappear at the other side if this makes the differences between the angles smaller. Ramachandran outliers (as defined by Kleywegt and Jones) are also reported. This plot gives an immediate indication of how well NCS is obeyed for the main-chain atoms, and is also a good indicator of the overall quality of the structure. If the main-chain is disordered, only PART 0 and PART 1 atoms are considered. Glycines are optionally included as open squares; prolines are treated as normal residues. Unless color has been switched off (option [C]) the dots and lines are color-coded according to residue type. The refinement should have been performed with FMAP 2 and the RTAB instructions needed to calculate the phi and psi torsion angles in SHELXL.

*[N] NCS analysis*

This option provides a detailed analysis of deviations from non-crystallographic symmetry (NCS). The Kleywegt plot [K] can also be used to provide an overall picture of how well NCS is obeyed by the main-chain torsion angles. Before using these options, a SHELXL refinement should be performed in which RTAB is used to calculate the phi, psi, omega and chi1...chi4 torsion angles. The instruction FMAP 2 is also required so that the *.lst* file contains the residue diagnostics table. It is also useful to have secondary structure assignments to hand for color coding of the NCS bar plots; many standard protein programs such as PROCHECK are able to supply this information.

Differences (2 NCS related components) and maximum deviations and r.m.s. deviations (if there are more than two components) are plotted and tabulated as a function of the base residue number (i.e. the residue number minus the offset such as 1000, 2000 ... that SHELXL uses instead of a chain ID). Because of the large number of factors involved this option requires some attention to detail.

Alpha-helices and beta-strands are entered one per line as 'A n1 n2' or 'B n1 n2' where n1 and n2 are the first and last residues of the helix or strand. Base residue numbers should be used and the list is terminated with a blank line. Then the numbers that have to be added to the base residue numbers to generate the NCS related units are defined in answer to a prompt by the program. For fourfold NCS the usual SHELXL convention of numbering equivalent chains 1001..., 2001... etc. would require the input '1000 2000 3000 4000' here. The program then requests the minimum deviations in angles (deg.) and B for output to .pro file; 0 would print all and 999 would not print any.:

There may be up to four diagrams on one page, starting at the top. Each should be defined by entering three characters: a symbol to label the diagram, then either D (absolute difference [rms absolute difference from mean if more than 2 components]), M (maximum absolute deviation [from mean if more than 2]) or A (average), followed by the letter H (phi), Y (psi), P (phi and psi), O (omega), C (chi1), T (all chi), M (main-chain B) or S (side-chain B) and then the numbers of the first and last base residues. Note that A is only allowed with S or M and that P or T must be preceded by M. END terminates the list.

The default diagrams are:

aMH (diagram a; maximum absolute deviation of phi angles)
bMY (diagram b; maximum absolute deviation of psi angles)
cMO (diagram c; maximum absolute deviation of omega angles)
dMT (diagram d; maximum absolute deviation of all chi angles)
eMM (diagram e; maximum absolute deviation of main-chain B)
fMS (diagram f; maximum absolute deviation of side-chain B)
gAM (diagram g; average main-chain B)
hAS (diagram h; average side-chain B)

### *[S] Reflection statistics from .fcf*

This option creates reflection statistics from a *.fcf* file written by SHELXL in response to a LIST 6 instruction.. The user must specify the resolution ranges, e.g. to be the same as those used for data reduction. A table of data completeness, *R*-factors etc. is written to the console and to the .pro output file.

### *[L] Luzzati plot*

This plots the resolution vs. *R*1. The *.fcf* file must have been created using LIST 6 in SHELXL. SHELXPRO outputs a Postscript Luzzati (1952) plot, which gives estimates of the average errors in atomic coordinates for an incompletely refined structure assuming perfect data, NOT (as widely assumed by people who have not read this paper which happens to be in French) estimates of the esds in the atomic positions. For small proteins and high resolution data, esds in individual bond lengths and atomic positions may be estimated rigorously using SHELXL (see the [E] option in SHELXPRO described below). Nevertheless, a plot of *R*-factor against resolution is always entertaining.

### *[E] Esd analysis*

This option reads SHELXL *.lst* file and prepares Postscript scatter-plots of esds in atom positions and bond lengths against (equivalent) B values. The refinement should normally have been performed with the SHELXL instructions L.S. 1, DAMP 0 0, BLOC 1 and BOND. If geometrical restraints were used in the refinement the bond length esds will be very low, but high resolution data are required to perform such a refinement without restraints. Similarly the damping has to be switched off because this can also lead to underestimated esds. Disordered atoms, atoms on special positions, and atoms other than C, N and O are not included in the diagrams. Such atoms are recognized by the first letter of their names in the atom coordinate table, so it may be necessary to remove calcium and other atoms that might be mistakenly identified from this table by editing the *.lst* file before running SHELXPRO.

A quadratic may be fitted to the atom radial esds, which enables the results to be compared with the formula suggested by Cruickshank (1996). Note that this formula predicts positional esds in one direction, which should be a factor of √3 smaller than the radial esds output by SHELXL.

*[Z] Least-squares fit*

The [Z] option may be used to perform a least-squares fit of two molecules, taken from the same or different structures. The iterative quaternion method is employed. This option is of necessity rather complex, and it is important to read each request for information by the program carefully because the default action (<Enter>) may well not be suitable and an incorrect answer can lead to complications.

It is necessary first to define the first molecule (called 'current structure'), which is extracted from a PDB format file. The a second molecule ('model') is obtained from another (or possibly the same) PDB file. Both PDB files may be as output by SHELXL or may be taken directly from the PDB databank, so 'chains' may be present. Since the residues may be numbered differently in the two molecules, it is necessary to convert the residue numbers in both molecules to a matching set of residue numbers referred to as SHELXPRO residue numbers. These numbers are also used to annotate the plots etc. The set of residues used for fitting is in general a subset of those used for the plots and calculation of r.m.s. esds.

After performing the fit for specified atoms in each of the specified residues, the program prints the r.m.s. deviation of the atoms fitted and the largest individual deviations (greater than $2\sigma$). Then appears the question:

New current structure (C), new model (M), Repeat fit (R), write PDB file (P), XP file (X), Postscript bar plot of differences (D) or exit (E) [E]:

'R' repeats the fit (possibly using different residues and atoms) of the 'model' (second molecule) to the 'current structure' (first molecule). 'M' replaces the 'model' but keeps the 'current structure'.. 'C' starts again with a new 'current structure'. 'P' writes a new PDB format file that contains the two molecules as two separate chains with the SHELXPRO residue numbers; this can be used as input to the program MOLSCRIPT. 'X' writes an orthogonal coordinate file that can be read by the Siemens' SHELXTL program XP and used to make a (stereo) Cα-trace of the superposition. 'D' prepares a Postscript bar plot of the differences between the two molecules, using all stored residues, not just those that were fitted.

*[A] Anisotropic scaling (Hope & Parkin)*

This option reads an *.fcf* file created using the LIST 6 instruction in SHELXL, and writes a NEW *.hkl* file after application of anisotropic scaling by the method of Parkin, Moezzi & Hope (1995). The modification of the observed structure factors in this way is scientifically suspect and is intended for testing purposes only. It is much better to use the HOPE instruction in SHELXL so that parameter correlations are taken into account and the observed data are not modified. The SHELXPRO correction provides a quick test as to whether HOPE in SHELXL will result in a significant improvement; in this case the question about the filename for corrected data should be answered with <Enter>. A 'local' $R_{\text{free}}$ test is applied to establish how many parameters [none(!), 12, 18 or 24] may justifiably be fitted. A significant improvement is not to be expected if anisotropic refinement has been performed or if a large number of symmetry equivalents were merged in the data reduction.

# 10. SHELXWAT: Automated Water Divining

A simple program **SHELXWAT** has been added that iteratively recycles SHELXL to provide automatic water divining. This may be regarded as a cheap and inadequate imitation of ARP (V. Lamzin & K.S. Wilson, *Acta Cryst.* **D49** (1993) 129-147), but is relatively easy to use and useful if you intend to take a holiday. SHELXWAT is started by means of a command line with OPTIONAL UNIX-type switches (the filename must come last):

```
shelxwat name
```

or e.g.

```
shelxwat -n10 -s4 -u0.6 -r0.8 -m50 -f name
```

These are the default settings for the switches -n (number of overall cycles), -s (scattering factor number for oxygen), -u (starting isotropic U for new waters), -r (water rejected if U refines to greater than this value), -m (maximum number of waters to be added in one cycle) and -h (half/full occupancies) or -f (full occupancies only). All switches present must come before 'name'.

Standard SHELXL files *name.ins* and name.hkl are required; the *.ins* file should contain 'CGLS 3 -20', 'FMAP 2', 'PLAN 200 2.4' or 'PLAN 200 -2.4' (half occupancies allowed), 'CONN 0 O_501 > LAST', 'BUMP' or similar instructions (the free R test is not obligatory) and MUST include at least one water at the end of the atom list. The waters will then be assigned dynamical residue numbers starting with the residue number of this water (501 in the above example) and should all have residue class 'HOH' and atom name 'O' with one atom per residue and no PART numbers. On starting, SHELXWAT makes a backup copy (*name.bak*) of the *.ins* file, since the *.ins* file is repeatedly overwritten during the recycling. The recycling may be terminated tidily before the preset number of iterations has been performed by creating a file *name.end* in the same directory; this operates like the *name.fin* file for SHELXL, but is 'deleted' by SHELXWAT once per iteration.

SHELXWAT calls SHELXL once each cycle, then edits the resulting *.res* file to prepare the *.ins* file for the next cycle. The $R1$ (and $R1_{\text{free}}$, if present) indices are extracted from the *.lst* file and included in the *.res* files as remarks; these and other remarks (REM) provide a protocol of the refinement, and may be converted to a Postscript plot using the "P" option in SHELXPRO. Note that the SHELXPRO option "U" provides the facilities necessary to update that solvent etc. interactively, in much the same way that SHELXWAT does automatically.

By changing the PLAN instruction to (say) 'PLAN 200 1 1' and leaving out the BUMP instruction it might be possible to emulate ARP in its structure extension mode; this has yet to be tested, but might be useful for completing high resolution (better than 2Å) structures.

# 11. Examples of Macromolecular Refinement

The following extracts from the file *6rxn.ins* (provided together with *6rxn.hkl* as an example) illustrate a number of points. The structure was determined by Stenkamp, Sieker & Jensen, (1990) who have kindly given permission for it to be used in this way. As usual in *.ins* files, comments may be included as REM instructions or after exclamation marks. The resolution of 1.5Å does not quite justify refinement of all non-hydrogen atoms anisotropically ('ANIS' before the first atom would specify this), but the iron and sulfur atoms should be made anisotropic as shown below. Note that it would be better to flag the $R_{\text{free}}$ reflections randomly using SHELXPRO rather than just using every twelth reflection.

```
TITL Rubredoxin in P1 (from 6RXN in PDB)
CELL 1.54178 24.920 17.790 19.720 101.00 83.40 104.50  ! Lambda & cell
ZERR       1  0.025  0.018  0.020   0.05  0.05   0.05  ! Z & cell esds
LATT -1                        ! Space group P1
SFAC  C   H   N  O   S  FE     ! Scattering factor types and
UNIT  224 498 55 136 6  1      ! unit-cell contents

DEFS 0.02 0.2 0.01 0.05        ! Global default restraint esds

CGLS 10 -12    ! 10 Conjugate gradient cycles, every 12th reflection
SHEL 999 0.1   ! for R(free), all other data used for refinement
FMAP 2         ! Difference Fourier
PLAN 200 2.3   ! Peaksearch and identification of potential waters

LIST 6    ! Output phased reflection file to generate maps etc.
WPDB      ! Write PDB output file
HTAB      ! Output analysis of hydrogen bonds (requires H-atoms !)

DELU $C_* $N_* $O_* $S_*    ! Rigid bond retraints - ignored for iso.

SIMU 0.1 $C_* $N_* $O_* $S_*  ! Similar U restraints - iso. or anis.
    ! Esd should be changed to ca. 0.05 if whole structure is anis.

ISOR 0.1 O_201 > LAST   ! Approximate isotropic restraints for waters;
                        ! ignored for isotropic

ANIS_* FE SD SG         ! Make iron and all sulfur atoms anisotropic

CONN 0 O_201 > LAST    ! Don't include water in connectivity array and
BUMP                   ! generate antibumping restraints automatically

SWAT                   ! Diffuse water model

REM HOPE               ! Anisotropic scaling not included

MERG 4 ! Remove MERG 4 if Friedel opposites should not be merged

MORE 1 ! MORE 0 for minimum, 2 or 3 for more output for diagnostics
```

```
REM Special restraints etc. specific to this structure follow:

REM HFIX 43 C1_1       !
DFIX C1_1 N_1 1.329    ! O=C(H)- (formyl) on N-terminus
DFIX C1_1 O1_1 1.231   ! incorporated into residue 1
DANG N_1 O1_1 2.250    !
DANG C1_1 CA_1 2.435   !

DFIX_52 C OT1 C OT2 1.249     !
DANG_52 CA OT1 CA OT2 2.379   ! Ionized carboxyl at C-terminus
DANG_52 OT1 OT2 2.194         !

SADI_54 0.04 FE SG_6 FE SG_9 FE SG_39 FE SG_42 ! Equal but unknown Fe-
S
SADI_54 0.08 FE CB_6 FE CB_9 FE CB_39 FE CB_42 ! distances around Fe

REM HFIX 83 SG_38 SG_138  ! -SH for remaining cysteine (disordered)

DFIX C_18 N_26 1.329            ! Patch break in numbering - residues
DANG O_18 N_26 2.250            ! 18 and 26 are bonded but there is a
DANG CA_18 N_26 2.425           ! gap in numbering for compatibility
DANG C_18 CA_26 2.435           ! with other rubredoxins that have an
FLAT 0.3 O_18 CA_18 N_26 C_18 CA_26  ! extra loop
RTAB Omeg CA_18 C_18 N_26 CA_26      !
RTAB Phi C_18 N_26 CA_26 C_26        !
RTAB Psi N_18 CA_18 C_18 N_26        !

REM DFIX from CSD and R.A.Engh & R.Huber, Acta Cryst. A47 (1991) 392.
REM Remove 'REM ' before HFIX to activate H-atom generation

REM HFIX_ALA 43 N
REM HFIX_ALA 13 CA
REM HFIX_ALA 33 CB

REM HFIX_ASN 43 N
REM HFIX_ASN 13 CA
REM HFIX_ASN 23 CB
REM HFIX_ASN 93 ND2

REM HFIX_ASP 43 N
REM HFIX_ASP 13 CA
REM HFIX_ASP 23 CB

   ... etc ...

REM HFIX_VAL 43 N
REM HFIX_VAL 13 CA CB
REM HFIX_VAL 33 CG1 CG2

REM Peptide standard torsion angles and restraints
```

```
RTAB_* Omeg CA C N_+ CA_+
RTAB_* Phi C_- N CA C
RTAB_* Psi N CA C N_+
RTAB_* Cvol CA

DFIX_* 1.329 C_- N
DANG_* 2.425 CA_- N
DANG_* 2.250 O_- N
DANG_* 2.435 C_- CA

FLAT_* 0.3 O_- CA_- N C_- CA

REM Standard amino-acid restraints etc.

CHIV_ALA C
CHIV_ALA 2.477 CA

DFIX_ALA 1.231 C O
DFIX_ALA 1.525 C CA
DFIX_ALA 1.521 CA CB
DFIX_ALA 1.458 N CA
DANG_ALA 2.462 C N
DANG_ALA 2.401 O CA
DANG_ALA 2.503 C CB
DANG_ALA 2.446 CB N

RTAB_ASN Chi N CA CB CG

CHIV_ASN C CG
CHIV_ASN 2.503 CA

DFIX_ASN 1.231 C O  CG OD1
DFIX_ASN 1.525 C CA
DFIX_ASN 1.458 N CA
DFIX_ASN 1.530 CA CB
DFIX_ASN 1.516 CB CG
DFIX_ASN 1.328 CG ND2
DANG_ASN 2.401 O CA
DANG_ASN 2.462 C N
DANG_ASN 2.455 CB N
DANG_ASN 2.504 C CB
DANG_ASN 2.534 CA CG
DANG_ASN 2.393 CB OD1
DANG_ASN 2.419 CB ND2
DANG_ASN 2.245 OD1 ND2

RTAB_ASP Chi N CA CB CG

CHIV_ASP C CG
CHIV_ASP 2.503 CA

DFIX_ASP 1.231 C O
DFIX_ASP 1.525 C CA
DFIX_ASP 1.530 CA CB
DFIX_ASP 1.516 CB CG
DFIX_ASP 1.458 CA N
DFIX_ASP 1.249 CG OD1 CG OD2
DANG_ASP 2.401 O CA
DANG_ASP 2.462 C N
DANG_ASP 2.455 CB N
DANG_ASP 2.504 C CB
DANG_ASP 2.534 CA CG
DANG_ASP 2.379 CB OD1 CB OD2
DANG_ASP 2.194 OD1 OD2

RTAB_CYS Chi N CA CB SG

CHIV_CYS C
CHIV_CYS 2.503 CA

DFIX_CYS 1.231 C O
DFIX_CYS 1.525 C CA
DFIX_CYS 1.458 N CA
DFIX_CYS 1.530 CA CB
DFIX_CYS 1.808 CB SG
DANG_CYS 2.401 O CA
DANG_CYS 2.504 C CB
DANG_CYS 2.455 CB N
DANG_CYS 2.462 C N
DANG_CYS 2.810 CA SG

  ... etc ...

RTAB_VAL Chi N CA CB CG1
RTAB_VAL Chi N CA CB CG2

CHIV_VAL C
CHIV_VAL 2.516 CA

DFIX_VAL 1.231 C O
DFIX_VAL 1.458 N CA
DFIX_VAL 1.525 C CA
DFIX_VAL 1.540 CA CB
DFIX_VAL 1.521 CB CG2 CB CG1
DANG_VAL 2.401 O CA
DANG_VAL 2.462 C N
DANG_VAL 2.497 C CB
DANG_VAL 2.515 CA CG1 CA CG2
DANG_VAL 2.479 N CB
DANG_VAL 2.504 CG1 CG2

WGHT    0.100000
FVAR       1.00000   0.5   0.5   0.5   0.5

RESI   1  MET
C1      1  -0.01633   0.35547   0.44703  11.00000   0.11817
O1      4   0.01012   0.32681   0.48491  11.00000   0.17896

N       3   0.00712   0.35446   0.37983  11.00000   0.11863
CA      1   0.05947   0.33273   0.35391  11.00000   0.06229
CB      1   0.07411   0.33732   0.27909  11.00000   0.15678
CG      1   0.03196   0.28864   0.22872  11.00000   0.14569
SD      5   0.04907   0.31846   0.14359  11.00000   0.23570
CE      1   0.11380   0.29170   0.12261  11.00000   0.21476
C       1   0.10634   0.38738   0.39766  11.00000   0.09178
O       4   0.10329   0.45513   0.41972  11.00000   0.16480
RESI   2  GLN
N       3   0.14741   0.35678   0.40741  11.00000   0.08599
CA      1   0.18940   0.39931   0.45565  11.00000   0.09291
CB      1   0.22933   0.34643   0.45886  11.00000   0.13253
CG      1   0.27354   0.38674   0.51173  11.00000   0.09866
CD      1   0.24547   0.38838   0.58387  11.00000   0.05748
OE1     4   0.22482   0.32772   0.60689  11.00000   0.16301
NE2     3   0.24704   0.46053   0.62045  11.00000   0.10164
C       1   0.22198   0.47895   0.43826  11.00000   0.08193
O       4   0.25019   0.48377   0.38408  11.00000   0.10402

RESI   3  LYS
N       3   0.21781   0.54034   0.48673  11.00000   0.07413
CA      1   0.25088   0.62006   0.47934  11.00000   0.05181
CB      1   0.21991   0.68311   0.51795  11.00000   0.09646
CG      1   0.16130   0.66288   0.49255  11.00000   0.10455
CD      1   0.12843   0.72146   0.52924  11.00000   0.22324
CE      1   0.10532   0.70085   0.60053  11.00000   0.26354
NZ      3   0.05943   0.74195   0.62796  11.00000   0.40338
C       1   0.30678   0.63497   0.50917  11.00000   0.05714
O       4   0.31462   0.59598   0.55179  11.00000   0.07986

  ... etc ...

RESI  12  GLU
N       3   0.41413   1.09215   0.48246  11.00000   0.06790
CA      1   0.37955   1.01183   0.48195  11.00000   0.05761
PART   1
CB      1   0.32666   1.01321   0.52971  21.00000   0.12219
CG      1   0.29679   0.93111   0.54638  21.00000   0.15333
CD      1   0.25357   0.93709   0.60700  21.00000   0.20272
OE1     4   0.24346   1.00278   0.63210  21.00000   0.26315
OE2     4   0.23012   0.87537   0.63031  21.00000   0.21375
```

```
PART   2
CB       1   0.32549   1.01718   0.52772 -21.00000   0.12065
CG       1   0.27756   0.94582   0.50954 -21.00000   0.15928
CD       1   0.22547   0.95184   0.55635 -21.00000   0.20457
OE1      4   0.20774   0.90241   0.59575 -21.00000   0.22329
OE2      4   0.20259   1.00588   0.55325 -21.00000   0.31441
PART   0
C        1   0.36477   0.97439   0.40859  11.00000   0.04768
O        4   0.34317   1.00861   0.37369  11.00000   0.06890

   ... etc ...

RESI  38  CYS
N        3   0.77141   0.92674   0.00625  11.00000   0.10936
CA       1   0.78873   0.97402   0.07449  11.00000   0.13706
PART   1
CB       1   0.83868   1.04271   0.05517  41.00000   0.11889
SG       5   0.89948   1.00271   0.02305  41.00000   0.18205
PART   2
CB       1   0.84149   1.03666   0.06538 -41.00000   0.14933
SG       5   0.83686   1.10360   0.01026 -41.00000   0.17328
PART   0
C        1   0.74143   1.01670   0.10383  11.00000   0.08401
O        4   0.70724   1.02319   0.06903  11.00000   0.10188

RESI  39  CYS
N        3   0.74699   1.04547   0.17051  11.00000   0.08888
CA       1   0.70682   1.09027   0.20876  11.00000   0.06869
CB       1   0.72588   1.11964   0.28230  11.00000   0.04269
SG       5   0.67932   1.17560   0.33481  11.00000   0.08016
C        1   0.70922   1.16093   0.17333  11.00000   0.06208
O        4   0.75427   1.20325   0.15858  11.00000   0.07437

   ... etc ...

RESI  52  ALA
N        3   0.33596   0.63469   0.69557  11.00000   0.04662
CA       1   0.30961   0.68882   0.74487  11.00000   0.08939
CB       1   0.34040   0.77357   0.74194  11.00000   0.13277
C        1   0.24852   0.67507   0.73435  11.00000   0.09032
OT1      4   0.22236   0.72170   0.77321  11.00000   0.11368
OT2      4   0.22682   0.61667   0.69191  11.00000   0.08341

RESI  54  FE
FE       6   0.72017   1.22290   0.43784  11.00000   0.07929

REM Only the waters with high occupancies and low U's have been
REM retained, and all the occupancies have been reset to 1, with
REM a view to running the automatic water divining.  Water
REM residue numbers have been changed to start at 201.
```

```
RESI 201  HOH
O       4   0.13450   0.53192   0.60802  11.00000   0.13132
RESI 202  HOH
O       4   0.84795   0.53873   0.69488  11.00000   0.15273
RESI 203  HOH
O       4   0.27771   0.95750   0.25086  11.00000   0.11315
RESI 204  HOH
O       4   0.37066   0.71872   0.90376  11.00000   0.10854

   ... etc ...

RESI 233  HOH
O       4   0.27813   1.38725   0.25914  11.00000   0.10698

HKLF 3
END
```

# 12. SHELXS - Structure Solution

SHELXS is primarily designed for the solution of ‘small moiety’ (1-200 unique atoms) structures from single crystal at atomic resolution, but is also useful for the location of heavy atoms from macromolecular isomorphous or anomalous $\Delta F$ data. The use of the program with SIR, OAS or MAD $F_A$ data is described in Chapter 15. SHELXS is general and efficient for all space groups in all settings, and there are no arbitrary limits to the size of problems which can be handled, except for the total memory available to the program. Instructions and data are taken from two standard (ASCII) text files, compatible to those used for SHELXL, so that input files can easily be transferred between different computers.

## 12.1 Program and file organization

The way of running SHELXS and the conventions for filenames will of course vary for different computers and operating systems, but the following general concept should be adhered to as much as possible. SHELXS may be run on-line by means of the command:

**shelxs name**

where *name* defines the first component of the filename for all files which correspond to a particular crystal structure. On some systems, *name* may not be longer than 8 characters. On UNIX systems, all filenames (including SHELXS) MUST be given in ***lower case***. Batch operation will normally require the use of a short batch file containing the above command etc.

Before starting SHELXS, at least one file - *name.ins* - MUST have been prepared; it contains instructions, crystal and atom data etc. It will usually be necessary to prepare a *name.hkl* file as well which contains the reflection data; the format of this file (3I4,2F8.2) is the same as for all versions of SHELX. This file should be terminated by a record with all items zero. The reflection order is unimportant. This *.hkl* file is read each time the program is run; unlike SHELX-76, there is no facility for intermediate storage of binary data. This enhances computer independence and eliminates several possible sources of confusion. SHELXS requires a single set of input data, and ignores batch numbers, direction cosines or wavelengths if they are present at the end of each record in the *name.hkl* file.

A brief summary of the progress of the structure solution appears on the console (i.e. the standard FORTRAN output), and a full listing is written to a file *name.lst*, which can be printed or examined with a text editor. After structure solution a file *name.res* is written; this contains crystal data etc. as in the *name.ins* file, followed by potential atoms. It may be copied or edited to *name.ins* for structure refinement using SHELXL or partial structure expansion with SHELXS (Chapter 14).

Two mechanisms are provided for interaction with a SHELXS job which is already running. The first, which it is not possible to implement for all computer systems, applies to 'on-line' runs. If the <ctrl-I> key combination is hit, the job terminates almost immediately (but without the loss of output buffers etc. which can happen with <ctrl-C> etc.). If the <Esc> key is hit during direct methods, the program does not generate any further phase permutations but completes the current batch of phase refinement and then procedes to *E*-Fourier recycling etc. If the <Esc> key is hit during Patterson interpretation, the program stops after completing the calculations for the current superposition vector. Otherwise <Esc> has no effect. On computer consoles with no <Esc> key, <F11> or <Ctrl-[> usually have the same effect.

The second mechanism requires the user to create the file *name.fin*; the program tries at regular intervals to delete this file, and if it succeeds it takes the same action as after <Esc>. The file is also deleted (if found) at the start of a job in case it has been accidentally left over from a previous job. This approach may be used with batch jobs, but may prove difficult to implement on certain systems. The output files are also 'flushed' at regular intervals (if permitted by the operating system) so that they can be examined whilst a batch job is running (if permitted).

The UNIX version of SHELXS is able to read the *.ins* and *.hkl* files in either UNIX or DOS format, and may be compiled under UNIX so as to write the *.res* file in DOS format (see the comments near the start of the program source), so that PC's can access such files via a shared disk without the need for conversion programs such as DOS2UNIX etc. However the compiled programs are supplied with this option switched off, i.e. they write standard UNIX format files. The *.lst* file is always in the local format for reasons of efficiency. The MSDOS program SPRINT supplied with SHELX can print from both MSDOS or UNIX formats.

### 12.2 The .ins instruction file

Three types of general calculation may be performed with SHELXS. The structure of the *.ins* file is extremely similar for all three (and the *.hkl* file is always the same). The *.ins* file always begins with the instructions TITL..UNIT in the order given below. There follows TREF (for direct methods), PATT (for Patterson interpretation) or TEXP plus atoms (for partial structure expansion). The final instruction is usually HKLF.

```
Direct Methods:  Patterson Interp.:  Partial Structure Exp.:
--------------   -----------------   ----------------------
TITL ...           TITL ...            TITL ...
CELL ...           CELL ...            CELL ...
ZERR ...           ZERR ...            ZERR ...
LATT ...           LATT ...            LATT ...
SYMM ...           SYMM ...            SYMM ...
SFAC ...           SFAC ...            SFAC ...
UNIT ...           UNIT ...            UNIT ...
TREF               PATT                TEXP
HKLF               HKLF                atoms
                                       HKLF
```

Although these standard settings should be appropriate for a wide range of circumstances, various parameters may be specified for TREF, PATT or TEXP, and further instructions may be included between UNIT and HKLF for 'fine tuning' in the case of difficult structures. The parameter summary printed out after the data reduction in every job should be consulted before this is attempted, since the default settings for parameters that are not specified depend on the space group, the size of the structure, and the parameters that are actually specified (this is sometimes referred to as 'artificial intelligence' !).

All instructions commence with a four (or less) letter word (which may be an atom name); numbers and other information follow in free format, separated by one or more spaces. Upper and lower case input may be freely mixed; with the exception of the text strings input using TITL it is all converted to upper case for internal use in SHELXS. The TITL, CELL, ZERR, LATT, SYMM, SFAC and UNIT instructions must be given in that order; all remaining instructions, atoms, etc. should come between UNIT and the last instruction, which is almost always HKLF (to read in reflection data).

Defaults are given in square brackets in this documentation; '#' indicates that the program will generate a suitable default value based on the rest of the available information. Continuation lines are flagged by '=' at the end of a line, the instruction being continued on the next line which must start with at least one space. Other lines beginning with one or more spaces are treated as comments, so blank lines may be added to improve readability. All characters following '!' or '=' in an instruction line are ignored, except after TITL or SYMM (for which continuation lines are not allowed). AFIX, RESI and PART instructions may be present in the *.ins* file for compatibility with SHELXL but are ignored.

### 12.3 Instructions common to all modes of structure solution

**TITL [ ]**

Title of up to 76 characters, to appear at suitable places in the output. The characters '!' and '=' may form part of the title. The title could include a chemical formula and/or space group, but one must be careful to update these if the UNIT or SYMM instructions are later changed !

**CELL $\lambda$ a b c $\alpha$ $\beta$ $\gamma$**

Wavelength and unit-cell dimensions in Angstroms and degrees.

**ZERR Z esd(a) esd(b) esd(c) esd($\alpha$) esd($\beta$) esd($\gamma$)**

Z value (number of formula units per cell) followed by the estimated errors in the unit-cell dimensions. This information is not actually required by SHELXS but is allowed for compatibility with SHELXL.

**LATT N [1]**

Lattice type: 1=P, 2=I, 3=rhombohedral obverse on hexagonal axes, 4=F, 5=A, 6=B, 7=C. N must be made negative if the structure is non-centrosymmetric.

**SYMM symmetry operation**

Symmetry operators, i.e. coordinates of the general positions as given in International Tables. The operator X, Y, Z is always assumed, so may NOT be input. If the structure is centrosymmetric, the origin MUST lie on a center of symmetry. Lattice centering should be indicated by LATT, not SYMM. The symmetry operators may be specified using decimal or fractional numbers, e.g. 0.5-x, 0.5+y, -z or Y-X, -X, Z+1/6; the three components are separated by commas. At least one SYMM instruction must be present unless the structure is triclinic.

**SFAC elements**

These element symbols define the order of scattering factors to be employed by the program. The first 94 elements of the periodic system are recognized. The element name may be preceded by '$' but this is not obligatory (the '$' character is allowed for logical consistency with certain SHELXL instructions but is ignored). The program uses absorption coefficients from International Tables for Crystallography (1991), Volume C. For organic structures the first two SFAC types should be C and H, in that order; the E-Fourier recycling generally assigns the first SFAC type (i.e. C) to peaks.

**SFAC a1 b1 a2 b2 a3 b3 a4 b4 c df' df" mu r wt**

Scattering factor in the form of an exponential series, followed by real and imaginary corrections, linear absorption coefficient, covalent radius and atomic weight. Except for the atomic weight the format is the same as that used in SHELX-76. In addition, a 'label' consisting of up to 4 characters beginning with a letter (e.g. Ca2+) may be included before a1 (the first character may be a '$', but this is not obligatory). The two SFAC formats may be used in the same *.ins* file; the order of the SFAC instructions (and the order of element names in the first type of SFAC instruction) define the scattering factor numbers which are referenced by atom instructions. Not all numbers on this instruction are actually used by SHELXS, but the full data must be given for compatibility with SHELXL. For neutron data, c should be the scattering length (which may be negative) and a1..b4 will usually all be zero.

**UNIT n1 n2 ...**

Number of atoms of each type in the cell, in SFAC order.

**REM**

Followed by a comment on the same line. This comment is ignored by the program but is copied to the results file (*.res*). Note that comments beginning with one or more blanks are only copied to the *.res* file if the line is completely blank; REM comments are always copied.

**MORE verbosity [1]**

More sets the amount of (printer) output; verbosity takes a value in the range 0 (least) to 3 (most verbose).

**TIME t [#]**

If the time t (measured in seconds from the start of the job) is exceeded, SHELXS performs no further blocks of phase permutations (direct methods), but goes on to the final *E*-map recycling etc. In the case of Patterson interpretation, no further vector superpositions are performed after this time has expired. The default value of t is installation dependent, and is usually set to a little less than the maximum time allocation for a particular job class. Usually t is 'CPU time', but on some simpler computer systems (eg. MSDOS) the elapsed time has to be used instead.

**OMIT s [4] 2θ(lim) [180]**

Thresholds for flagging reflections as 'unobserved'. Note that if no OMIT instruction is given, ALL reflections are treated as 'observed'. Internally in the program s is halved and applied to $F_o^2$, so the test is roughly equivalent to suppressing all reflections with $F_o < s\sigma(F_o)$, as required for consistency with SHELX-76. Note that s may be set to 0 (to suppress reflections with negative $F_o^2$) or even to a negative threshold (to suppress very negative $F_o^2$) which has no equivalent in SHELX-76. If 2θ(lim) is POSITIVE, it specifies a 2θ value above which the data are treated as 'unobserved'; if it is negative, the absolute value is used as a lower 2θ cutoff.

**OMIT h k l**

The reflection h k l is flagged as 'unobserved' in the list of merged reflections after data reduction. It will not be used directly in phase refinement or Fourier calculations, but is retained for statistical purposes and as a possible cross-term in a negative quartet. Thus if it is known that a strong reflection has been included accidentally in the *.hkl* file with a very small intensity (e.g. because it was cut off by the beam stop), it is advisable to delete it from the *.hkl* file rather than using OMIT (which is intended for imprecisely measured data rather than blunders).

**ESEL Emin [1.2] Emax [5] dU [.005] renorm [.7] axis [0]**

Emin sets the minimum *E*-value for the list of largest *E*-values which the program normally retains in memory; it should be set so as to give more than enough reflections for TREF etc. It is also the threshold used for tangent expansion and 'peak-list optimisation'. It is advisable to reduce Emin to about 1.0 for triclinic structures and pseudosymmetry problems. If Emin is negative, acentric triclinic data are generated for use in ***all*** calculations. The other parameters control the normalisation of the *E*-values:

$$\mathbf{new}(E) = \mathbf{old}(E) \bullet \mathbf{exp}[\ 8\mathbf{dU}\ (\pi \sin\theta/\lambda)^2\ ] \ /\ [\ \mathbf{old}(E)^{-4} + \mathbf{Emax}^{-4}\ ]^{0.25}$$

renorm is a factor to control the parity group renormalisation; 0.0 implies no renormalisation, 1.0 sets full renormalisation, i.e. the mean value of $E^2$ becomes unity for each parity group.

If axis is 1, 2 or 3, an additional similar renormalisation is applied for groups defined by the absolute value of the *h*, *k* or *l* index respectively. If axis is set to zero, no such additional renormalisation is applied.

**EGEN d(min) d(max)**

All missing reflections in the resolution range d(min) to d(max) Å (the order of d(min) and d(max) is unimportant) are generated on a statistical basis, assuming that they were skipped during the data collection because a prescan indicated that they were weak. These reflections will then be flagged as 'unobserved', but improve the estimation of the remaining *E*-values and enable an increased number of negative quartets to be identified. d(min) should be safely inside the resolution limit of the data and d(max) should be set so that there is no danger of regenerating strong reflections (as weak) which were cut off by the beam stop etc.

**LIST m [0]**

m = 1 and m = 2 write *h*, *k*, *l*, *A* and *B* lists to the *name.res* file, where *A* and *B* are the real and imaginary parts of a point atom structure factor respectively. If m = 1 the list corresponds to the phased *E*-values for the 'best' direct methods solution, before partial structure expansion (if any). If m = 2 the list is produced after the final cycle of partial structure expansion, and corresponds to weighted *E*-values used for the final Fourier synthesis. These options enable other Fourier programs to be used, e.g. for graphical display of 3D-Fouriers for data to less than atomic resolution.

After data reduction and merging equivalent reflections, a list of *h*, *k*, *l*, $F_o$ and $\sigma(F_o)$ (for m = 3) or *h*, *k*, *l*, $F_o^2$ and $\sigma(F_o^2)$ (for m = 4) is written to the *name.res* file. This provides a useful input file for programs such as DIRDIF and MULTAN, which do not include sort/merge and rejection of systematic absences etc. SHELXS always averages Friedel opposites. In all four cases the output format is (3I4,2F8.2), and the list is terminated by a dummy reflection 0,0,0.

**FMAP  code [#]  axis [#]  nl [#]**

The unique unit of the cell for performing the Fourier calculation is set up automatically unless specified by the user using FMAP and GRID. The program chooses a 53 x 53 x nl or 103 x 103 x nl grid depending the the resolution of the data, provided sufficient memory is available in the latter case.

code = 1 ($F^2$-Patterson), 3 (Patterson with coefficients input using HKLF 7; negative coefficients are allowed. 4 (*E*-map without peak-list optimisation, e.g. because the peaks correspond to unequal atoms), 5 (Fourier with *A* and *B* coefficients input using HKLF 3), 6 (*EF* Patterson), code > 6 (E-map followed by [code–6] cycles peak-list optimization). Note that the peak-list optimization assigns very strong peaks to heavy atoms (if specified by SFAC) and all remaining peaks to scattering factor type 1, so for many structures this should be specified as carbon on a SFAC instruction. FMAP 4 may be used with atoms but without TEXP etc. for an *E*-map based on calculated phases.

**GRID  sl [#]  sa [#]  sd [#]  dl [#]  da [#]  dd [#]**

Fourier grid, when not set automatically. Starting points and increments are multiplied by 100. s means starting value, d increment, l is the direction perpendicular to the layers, a is across the paper from left to right, and d is down the paper from top to bottom. Note that the grid is 53 x 53 x nl points, i.e. twice as large as in SHELX-76, and that sl and dl need not be integral. The 103 x 103 x nl grid is only available when it is set automatically by the program (see above).

**PLAN  npeaks [#]  d1 [0.5]  d2 [1.5]**

If npeaks is positive it is the number of highest unique Fourier peaks which are written to the *.res* and *.lst* files; the remaining parameters are ignored. If npeaks is given as negative, the program attempts to arrange the peaks into unique molecules taking the space group symmetry into account, and to 'plot' a projection of each such molecule on the printer (i.e. the .lst file). Distances involving peaks which are less than r1+r2+d1 (the covalent radii r are defined via SFAC; 1 and 2 refer to the two atoms concerned) are considered to be 'bonds' for purposes of the molecule assembly and tables. Distances involving atoms and/or peaks which are less than r1+r2+d2 are considered to be 'non-bonded interactions'. Such interactions are ignored when defining molecules, but the corresponding atoms and distances are included in the line-printer output. Thus an atom may appear in more than one map, or more than once on the same map. Negative d2 includes hydrogen atoms in these non-bonds, otherwise they are ignored (the absolute value of d2 is used in the test). Peaks are always always assigned the radius of SFAC type 1, which is usually set to carbon. Peaks appear on the printout as numbers, but in the *.res* file they are given names beginning with 'Q' and followed by the same numbers.

To simplify interpretation of the lineprinter plots, extra symmetry-generated atoms are added, so that atoms or peaks may appear more than once. A table of the appropriate coordinates and symmetry transformations appears at the end of the output. See also MOLE for forcing molecules (and their environments) to be printed separately.

**MOLE  n [#]**

Forces the following atoms, and atoms or peaks that are bonded to them, into molecule n of the PLAN output. n may not be greater than 99.

**`HKLF n [0] s [1] r11...r33 [1 0 0 0 1 0 0 0 1] wt [1] m [0]`**

Before running SHELXS, a reflection data file *name.hkl* must usually be prepared. The HKLF command tells the program which format has been chosen for this file, and allows the indices to be reorientated using a 3x3 matrix r11..r33 (which should have a positive determinant). n is negative if reflection data follow, otherwise they are read from the *.hkl* file. The data are read in fixed format 3I4,2F8.2 (except for n = 1) subject to FORTRAN-77 conventions. The data are terminated by a record with *h*, *k* and *l* all zero (except n=1, which contains a terminator and checksum). If batch numbers, direction cosines or wavelengths are present in the *.hkl* file (e.g. for use with SHELXL) they will be ignored. The multiplicative scale s multiplies both $F^2$ and $\sigma(F^2)$ (or *F* and $\sigma(F)$ for n = 1 or 3). The multiplicative weight wt multiplies all $1/\sigma^2$ values and m is an integer 'offset' needed to read 'condensed data' (HKLF 1); both are included only for compatibility with SHELX-76. Usually simply 'HKLF 4' is all that will be required.

**n = 1:** SHELX-76 condensed data. Although now obsolete this format is both ASCII and compact, and contains a checksum, so is sometimes used for network transmission and testing purposes.

**n = 3:** $h$ $k$ $l$ $F_o$ $\sigma(F_o)$ or $h$ $k$ $l$ $A$ $B$ depending on FMAP setting. In the first case the sign of $F_o$ is ignored (for use with macromolecular $\Delta F$ data). This format should NOT be used for routine structure determination purposes because the approximation(s) required for the derivation of *F* and $\sigma(F_o)$ degrade the quality of the data.

**n = 4:** $h$ $k$ $l$ $F^2$ $\sigma(F^2)$. The recommended format for nearly all purposes (for macromolecular isomorphous or anomalous $\Delta F$ HKLF 3 is suitable).

**n = 7:** $h$ $k$ $l$ $E$ or $h$ $k$ $l$ $P$ (Patterson coefficient) depending on FMAP.

***There may only be one* HKLF *instruction and it must come last*** !

**`END`**

This is the last instruction in the rare cases when the .ins file is not terminated by the HKLF instruction.

### 12.4 Instructions for writing and reading files for the program PATSEE

**`SPIN phi1 [0] phi2 [0] phi3 [0]`**

The following fragment (which should begin with a FRAG instruction) is rotated by the specified angles (in radians). This instruction is used to reinput angles from Patterson search programs (in particular PATSEE).

**`FRAG code [#] a [1] b [1] c [1] alpha [90] beta [90] gamma [90]`**

FRAG enables the PATSEE search fragment to be read in using the original cell or orthogonal coordinates. This instruction will usually be preceded by SPIN and MOVE commands to give the rotation angles and translation (same conventions as for PATSEE), and followed by a list of atoms. FRAG, SPIN and MOVE instructions remain in force until superseded by another instruction of the same type. code is ignored by SHELXS but is included for compatibility with PATSEE and SHELXL (where it is used for different purposes).

**PSEE  m [200]  2θ(max) [#]**

The largest |m| *E*-values and the complete Patterson map are dumped into the *name.res* file in fixed format for use by Patterson search programs (in particular PATSEE) etc.  2θ(max) should be used to limit the resolution of the *E*-values generated; the default value uses sinθ= λ/2.  The 2θ(max) value is also written to the *.res* file, so it is possible to restrict the resolution of the *E*-values actually used by PATSEE to a lower 2θ(max) by editing this file without rerunning SHELXS; of course the *E*-values with higher 2θ than the value used in SHELXS were not written to the *.res* file and so cannot be recovered in this way.  When m is negative a 'super-sharp' Patterson with coefficients √($E^3F$) is used; if m is positive a standard sharpened Patterson with coefficients ($EF$) is employed.  The resulting *name.res* file must be renamed *name.inp* (or *name.pat* if the search fragment and encoded Patterson are to be read from separate files) for use by PATSEE.  After a PSEE instruction, UNIT is followed by the strongest *E*-values and the full Patterson map in this output file (which may be rather long !).

# 13. Structure Solution by Direct Methods

### 13.1 Routine structure solution

Usually direct methods will be initiated with the single SHELXS command TREF; for large structures brute force (e.g. TREF 5000) may prove necessary. In fact there are a large number of parameters which can be varied, though the program is based on experience of many thousands of structures and can usually be relied upon to choose sensible default values. A summary of these parameters appears after the data reduction output, and should be consulted before attempting any direct methods options other than 'TREF n'.

### 13.2 Facilities for difficult structures

The phase refinement of multiple random starting phase sets takes place in three stages, controlled by the INIT, PHAN and TREF instructions. The 'best' solution is then expanded further by tangent expansion and *E*-Fourier recycling (see the section on partial structure expansion).

```
INIT  nn [#]  nf [#]  s+ [0.8]  s- [0.2]  wr [0.2]
```

The first stage involves five cycles of weighted tangent formula refinement (based on triplet phase relations only) starting from nn reflections with random phases and weights of 1. Single phase seminvariants which have $\Sigma_1$-formula $P_+$ values less that s- or greater than s+ are included with their predicted phases and unit weights. All these reflections are held fixed during the INIT stage but refined freely in the subsequent stages. The remaining reflections also start from random phases with initial weights wr, but both the phases and the weights are allowed to vary.

If nf is non-zero, the nf 'best' (based on the negative quartet and triplet consistency) phase sets are retained and the process repeated for (npp–nf) parallel phase sets, where npp is the previous number of phase sets processed in parallel (often 128). This is repeated for nf fewer phase sets each time until only a quarter of the original number are processed in parallel. This rather involved algorithm is required to make efficient use of available computer memory. Typically nf should be 8 or 16 for 128 parallel permutations.

The purpose of the INIT stage is to feed the phase annealing stage with relatively self-consistent phase sets, which turns out to be more efficient than starting the phase annealing from purely random phases. If TREF 0 is used to generate partial structure phases for all reflections, the INIT stage is skipped. To save time, only ns reflections and the strongest mtpr triplets for each reflection (or less, if not so many can be found) are used in the INIT stage; these numbers are given on the PHAN instruction.

```
PHAN  steps [10]  cool [0.9]  Boltz [#]  ns [#]  mtpr [40]  mnqr [10]
```

The second stage of phase refinement is based on 'phase annealing' (Sheldrick, 1990). This has proved to be an efficient search method for large structures, and possesses a number of beneficial side-effects. It is based on steps cycles of tangent formula refinement (one cycle is a pass through all ns phases), in which a correction is applied to the tangent formula phase. The phase annealing algorithm gives the magnitude of the correction (it is larger when the 'temperature' is higher; this corresponds to a larger value of Boltz), and the sign is chosen to give the best agreement with the negative quartets (if there are no negative quartets involving the reflection in question, a random sign is used instead). After each cycle through all ns phases, a new value for Boltz is obtained by multiplying the old value by cool; this corresponds to a reduction in the 'temperature'. To save time, only ns reflections are refined using the strongest mtpr triplets and mnqr quartets for each reflection (or less, if not so many phase relations can be found). The phase annealing parameters chosen by the program will rarely need to be altered; however if poor convergence is observed, the Boltz value should be reduced; it should usually be in the range 0.2 to 0.5. When the 'TEXP 0 / TREF' method of multisolution partial structure refinement is employed, Boltz should be set at a somewhat higher value (0.4 to 0.7) so that not too many solutions are duplicated.

**`TREF np [100] nE [#] kapscal [#] ntan [#] wn [#]`**

np is the number of direct methods attempts; if negative, only the solution with code number |np| is generated (the code number is in fact a random number seed). Since the random number generation is very machine dependent, this can only be relied upon to generate the same results when run on the same model of computer. This facility is used to generate *E*-maps for solutions which do not have the 'best' combined figure of merit. No other parameter may be changed if it is desired to repeat a solution in this way. For difficult structures, it may well be necessary to increase np (e.g. TREF 5000) and of course the computer time allocated for the job.

nE reflections are employed in the full tangent formula phase refinement. Values of nE that give fewer than 20 unique phase relations per reflection for the full phase refinement are not recommended.

kapscal multiplies the products of the three *E*-values used in triplet phase relations; it may be regarded as a fudge factor to allow for experimental errors and also to discourage overconsistent (uranium atom) solutions in symorphic space groups. If it is negative the cross-term criteria for the negative quartets are relaxed (but all three cross-term reflections must still be measured), and more negative quartets are used in the phase refinement, which is also useful for symorphic space groups.

ntan is the number of cycles of full tangent formula refinement, which follows the phase annealing stage and involves all nE reflections; it may be increased (at the cost of CPU time) if there is evidence that the refinement is not converging well. The tangent formula is modified to avoid overconsistency by applying a correction to the resulting phase of cos-1($<\alpha>/\alpha$) when $<\alpha>$ is less than $\alpha$; the sign of the correction is chosen to give the best agreement with the negative quartets (a random sign is used if there are no negative quartets involving the phase in question). This tends to drive the figures of merit $R_\alpha$ and Nqual simultaneously to desirable values. If ntan is negative, a penalty function of $(<\Sigma_1> - \Sigma_1)^2$ is added to CFOM (see below) if and only if $\Sigma_1$ is less than its estimated value $<\Sigma_1>$. $\Sigma_1$ is a weighted sum of the products of the expected and observed signs of one-phase seminvariants, normalized so that it must lie in the range -1 to +1. This is useful (i.e. better than nothing) if no negative quartets have been found or if they are unreliable, e.g. when macromolecular $\Delta F$ data are employed (see below).

wn is a parameter used in calculating the combined figure of merit CFOM: CFOM = R$\alpha$ (NQUAL < wn) or $R_\alpha$ + (wn–NQUAL)$^2$ (NQUAL $\geq$ wn); wn should be about 0.1 more negative than the anticipated value of NQUAL. If it is known that the measurements of the weak reflections are unreliable (i.e. have high standard deviations), e.g. because data were collected using the default options on a CAD-4 diffractometer, then the NQUAL figure of merit is less reliable. If the space group does not possess translation symmetry, it is essential to obtain good negative quartets, i.e. to measure ALL reflections for an adequate length of time.

Only the TREF instruction is essential to specify direct methods; appropriate INIT, PHAN, FMAP, GRID and PLAN instructions are then generated automatically if not given.

## 13.3 What to do when direct methods fail

If direct methods fail to give a clearly correct answer, the diagnostic information printed out during the data reduction at the start of the name.lst file should first be carefully reexamined.

After reading the SFAC and UNIT instructions the program uses the unit-cell contents and volume to calculate the volume per non-hydrogen atom, which is usually about 18 for typical oganic structures. Condensed aromatic systems can reduce this value (to about 13 in extreme cases) and higher values (20-30) are observed for structures containing heavier elements. The estimated maximum single weight Patterson vector may be useful (in comparison with the Patterson peak-list) in deciding whether the expected heavy atoms are in fact present. However in general the program is rather insensitive to the given unit-cell contents; the assignment of atom types in the *E*-Fourier recycling (after direct methods when heavier atoms are present) and in the Patterson interpretation do however assume that the elements actually present are those named on the SFAC instructions.

Particularly useful checks are the values of 2θ(max) and the maximum values of the (unsigned) reflection indices *h*, *k* and *l*; for typical small-molecule data the latter should be a little greater than the corresponding unit-cell dimensions. If not, or if 2θ(max) does not correspond to the value used in the data collection, there must be an error in the CELL or HKLF instructions, or possibly in the reflection data.

The $R_{\text{int}}$ value may be used as a test of the Laue group provided that appropriate equivalent reflections have been measured. Generally $R_{\text{int}}$ should be below 0.1 for the correct assignment. $R_{\text{sigma}}$ is simply the sum of $\sigma(F^2)$ divided by the sum of $F^2$; a value above 0.1 indicates the the data are very weak or that they have been incorrectly processed.

The mean values of $|E^2\text{-}1|$ show whether the *E*-value distribution for the full data and for the 0*kl*, *h*0*l* and *hk*0 projections are centric or acentric; this provides a check on the space group assignment, but such statistics may be unreliable if heavy atoms are present (especially when they lie on special positions) or if there are very few reflections in one of these three projections. Twinned structures may give an acentric distribution even when the true space group is centrosymmetric, or a mean $|E^2\text{-}1|$ value less than 0.7 for non-centrosymmetric structures.. These numbers may also show up typing errors in the LATT and SYMM instructions; although the program checks the LATT and SYMM instructions for internal consistency, it is not possible to detect all possible errors in this way.

Direct methods are based on the assumption of 'equal resolved atoms'. If the data do not suffice to 'resolve' the atoms from each other, direct methods are doomed to failure. A good empirical test of resolution is to compare the number of reflections 'observed' in the 1.1 to 1.2

Å range with the number theoretically possible (assuming that OMIT is at its default setting of 4) as printed out by the program. If this ratio is less than one half, it is unlikely that the structure will be ever be solved by direct methods. This criterion may be relaxed somewhat for centrosymmetric structures and those containing heavy atoms. It also does not apply to the location of heavy atoms from macromolecular $\Delta F$ data because the distances between the 'atoms' are much larger. If the required resolution has not been reached, there is little point in persuing direct methods further; the only hope is to recollect the data with a larger crystal, stronger radiation source, longer measurement times, area detector, real-time profile fitting and lower temperature, or at least as many of these as are simultaneously practicable.

If the data reduction diagnostics give no grounds for suspicion and no direct methods solution gives good figures of merit, a brute force approach should be applied. This takes the form of TREF followed by a large number (e.g. TREF 5000); it may also be necessary to set a larger value for TIME. If either of the methods for interrupting a running job are available (see above), an effectively infinite value may be used (TREF 999999). Any change in this number of phase permutaions will also change the random number sequence employed for the starting phases. It may also be worth increasing the second TREF parameter (WE) in steps of say 10%.

If more than one solution has good $R_\alpha$ and Nqual values, it is possible that the structure has been solved but the program has chosen the wrong solution. The list of one-phase seminvariant signs printed by the program can be used to decide whether two solutions are equivalent or not. In such a case other solutions can be regenerated without repeating the complete job by means of 'TREF -n', where n is a solution code number (in fact the random number seed). Because of the effect of small rounding errors the 'TREF -n' job must be performed on the same computer as the original run. No other parameters should be changed when this option is used.

In cases of pseudosymmetry is may be necessary to modify the *E*-value normalization (i.e. by increasing the renorm parameter on the ESEL instruction to 0.9, or by setting a non-zero value of axis on the same instruction). E(min) should be set to 1.0 or a little lower in such cases.

When direct methods only reveal a fragment of the structure, it may well be correctly oriented but incorrectly translated relative to the origin. In such cases a non-centrosymmetric triclinic expansion with 'ESEL -1' may enable the symmetry elements and hence the correct translation (and perhaps the correct space group) to be identified.

Finally, if any heavier (than say sodium) elements are present, automatic Patterson interpretation should be tried.

# 14. Patterson Interpretation and Partial Structure Expansion

The Patterson superposition procedure in SHELXS was originally designed for the location of heavier atoms in small moiety structures, but it turns out that it can also be used to locate  heavy atom sites for macromolecular $\Delta F$ data (see Chapter 15).  For further details and examples see Sheldrick (1996) and Sheldrick, Dauter, Wilson, Hope & Sieker (1993).

## 14.1 Patterson interpretation algorithm

The algorithm used to interpret the Patterson to find the heavier atoms in the new version of SHELXS is totally different to that used in SHELXS-86; it may be summarized as follows:

1. One peak is selected from the sharpened Patterson (or input by means of a VECT instruction) and used as a superposition vector.  This peak must correspond to a correct heavy-atom to heavy-atom vector otherwise the method will fail.  The entire procedure may be repeated any number of times with different superposition vectors by specifying 'PATT nv', with |nv| > 1, or by including more than one VECT instruction in the same job.

2. The Patterson function is calculated twice, displaced from the origin by +U and -U, where U is the superposition vector.  At each grid point the lower of the two values is taken, and the resulting 'superposition minimum function' is interpolated to find the peak positions.  This is a much cleaner map than the original Patterson and contains only 2N (or 4N etc. if the superposition vector was multiple) peaks rather than $N^2$.  The superposition map should ideally consist of one image of the structure and its inverse; it has an effective 'space group' of $P\overline{1}$ (or $C\overline{1}$ for a centered lattice etc.).

3. Possible origin shifts are found which place one of the images correctly with respect to the cell origin, i.e. most of the symmetry equivalents can be found in the peak-list. The SYMFOM figure of merit (normalized so that the largest value for a given superposition vector is 99.9) indicates how well the space group symmetry is satisfied for this image.

4. For each acceptable origin shift, atomic numbers are assigned to the potential atoms based on average peak heights, and a 'crossword table' is generated.  This gives the minimum distance and Patterson minimum function for each possible pair of unique atoms, taking symmetry into account.  This table should be interpreted by hand to find a subset of the atoms making chemically sensible minimum interatomic distances linked by consistently large Patterson minimum function values.  The PATFOM figure of merit measures the internal consistency of these minimum function values and is also normalised to a maximum of 99.9 for a given superposition vector.  The Patterson values are recalculated from the original $F_o$ data, not from the peak-list.  For high symmetry space groups the minimum function is calculated as an average of the two (or more) smallest Patterson densities.

5. For each set of potential atoms a 'correlation coefficient' (Fujinaga and Read, 1987) is calculated as a measure of the agreement between $E_o$ and $E_c$, and expressed as a percentage. This figure of merit may be used to compare solutions from different superposition vectors.

## 14.2 Instructions for Patterson Interpretation

**`PATT  nv [#]  dmin [#]  resl [#]  Nsup [#]  Zmin [#]  maxat [#]`**

nv is the number of superposition vectors to be tried; if it is negative the search for possible origin shifts is made more exhaustive by relaxing various tolerances etc. dmin is the minimum allowed length for a heavy-atom to heavy-atom vector; it affects ONLY the choice of superposition vector. If it is negative, the program does not generate any atoms on special positions in stage 4 (useful for some macromolecular problems). resl is the effective resolution in Å as deduced from the reflection data, and is used for setting various tolerances. If the data extend further than the crystal actually diffracted, or if the outer data are incomplete, it may well be worth increasing this number. This parameter can be relatively critical for macromolecular structures. Nsup is the number of unique peaks to be found by searching the superposition function. Zmin is the minimum atomic number to be included as an atom in the crossword table etc. (if this is set too low, the calculation can take appreciably longer). maxat is the maximum number of potential atoms to be included in the crossword table, and can also appreciably affect the time required for PATT.

**`VECT  X  Y  Z`**

A superposition vector (with coordinates taken from the Patterson peak-list) may be input by hand by a VECT instruction, in which case the first two numbers on the PATT instruction are ignored (except for their signs !), and a PATT instruction will be automatically generated if not present in the .ins file. There may be any number of VECT instructions.

In the unlikely event of a routine PATT run failing to give an acceptable solution, the best approach - after checking the data reduction diagnostics carefully as explained above - is to select several potential heavy-atom to heavy-atom vectors by hand from the Patterson peak-list and specify them on VECT instructions (either in the same job or different jobs according to local circumstances) for use as superposition vectors. The exhaustiveness of the search can also be increased - at a significant cost in computer time - by making the first PATT parameter negative and/or by increasing the value of resl a little. The sign of the second PATT parameter (a negative sign excludes atoms on special positions) and the list of elements which might be present (SFAC/UNIT) should perhaps also be reconsidered.

## 14.3 Instructions for partial structure expansion

**`TEXP  na [#]  nH [0]  Ek [1.5]`**

na PHAS reflections with $E_o$ > Ek and the largest values of $E_c/E_o$ are generated for use in partial structure expansion or direct methods. The first nH atoms (heavy atoms) in the atom list are retained during partial structure expansion, the rest are thrown away after calculating phases. At least one atom MUST be given! TEXP automatically generates appropriate FMAP, GRID and PLAN instructions.

TEXP (and/or PHAS) may be used in conjunction with TREF to generate fixed phases for use in direct methods; the special TEXP option na = 0 provides point atom phases for ALL reflections, which are then refined during the phase annealing and tangent expansion stages of direct methods (as specified on the PHAN and TREF instructions). It is not necessary to use different starting phases for the different phase sets, because the phase annealing stage itself introduces (statistically distributed) random phase shifts! This is a powerful method of partial structure expansion for cases when the phasing power of the partial structure is not quite adequate, e.g. when it consists of only one atom (say P or S in a large organic structure). If at least 5 atoms have been correctly located then TEXP alone should suffice.

When TEXP is used without TREF a tangent formula expansion (to all reflections with *E* > Emin as specified on the ESEL instruction) is first performed, followed by several cycles (see FMAP) of *E*-Fouriers and peak-list optimization. TEXP is particularly useful for cases in which several not very heavy atoms (e.g. P, S) have been located by PATT followed by hand interpretation of the resulting 'crossword table'. In such cases nH should be set to the number of such atoms and na to about half the number of reflections with *E* > 1.5 (see the first page of the SHELXS-96 output).

**`PHAS h k l phi`**

A fixed phase for structure expansion or direct methods. PHAS may be used to fix single phase seminvariants that have been obtained from other programs or derived by examination of the best TREF solutions. The phase angle phi must be present, and should be given in degrees.

**`atomname sfac x y z sof [1] U (or U11 U22 U33 U23 U13 U12)`**

Atom instructions begin with an atom name (up to 4 characters which do not correspond to any of the SHELXS command names, and terminated by at least one blank) followed by a scattering factor number (which refers to the list defined by the SFAC instruction(s)), x, y, and z in fractional coordinates, and (optionally) a site occupation factor (s.o.f.) and an isotropic U or six anisotropic $U_{ij}$ components (both in $Å^{-2}$). The U or $U_{ij}$ values are ignored by SHELXS but may be included for compatibility with SHELXL.

When SHELXS writes the *.res* output file, a dummy U value is followed by a peak height (unless an atom type has been assigned by the program before the *E*-Fourier recycling). Both the dummy U and the peak height are ignored if the atom is read back into SHELXS (e.g. for partial structure expansion). SHELXL also ignores the peak height if found in the .ins file. In contrast to SHELX-76 it is not necessary to pad out the atom name to 4 characters with blanks, but it should be followed by at least one blank. References to 'free variables' and fixing of atom parameters by adding 10 as in SHELX-76 and SHELXL will be interpreted correctly, but SHELXL AFIX, RESI and PART instructions are simply ignored (so idealized hydrogen atoms etc. are NOT generated). The site occupation factor for an atom in a special position should be divided by the number of atoms in the general position that have coalesced to give the special position. It may also be found by dividing the multiplicity of the special position (as as given in International Tables) by the multiplicity of the general position. Thus an atom on a fourfold axis will usually have s.o.f. = 10.25 (i.e. 0.25, fixed by adding 10).

**`MOVE dx [0] dy [0] dz [0] sign [1]`**

The coordinates of the following atoms are changed to: x = dx + sign * x, y = dy + sign * y, z = dz + sign * z (after applying FRAG and SPIN - if present - according to PATSEE conventions); MOVE applies to all following atoms until superseded by a further MOVE. MOVE is normally used in conjunction with SPIN and FRAG (see below) but is also useful on its own for applying origin shifts.

TEXP may be used in conjunction with ESEL -1 for a partial structure expansion in the effective space group P1 (C1 etc. if the lattice is centered). This can be very effective if it is suspected that a fragment is correctly oriented but translated from its real position, or if the space group cannot be unambiguously assigned. Hand interpretation of the resulting *E*-map is then however necessary to locate the positions of the crystallographic symmetry elements.

# 15. Location of Heavy Atoms from Protein Δ*F* Data

In principle both the Patterson interpretation and direct methods are suitable for the location of heavy atoms from protein or oligonucleotide isomorphous or anomalous Δ*F* data-sets.

## 15.1 Data preparation

For both the anomalous and isomorphous cases the user must prepare a file name.hkl containing *h*, *k*, *l*, Δ*F* and σ(Δ*F*) [or $(\Delta F)^2$ and $\sigma((\Delta F)^2)$] in the usual format (3I4,2F8.2), terminated by the dummy reflection with *h* = *k* = *l* = 0. The sign of Δ*F* is ignored. The auxiliary program SHELXPRO provides some facilities for the generation of this file, as does for example the CCP4 system.

Careful scaling of the derivative and native data, pruning of statistically unreasonable Δ*F*-values, and good estimated standard deviations are essential to the success of this approach. It should be emphasised that treating Δ*F* as if it were *F* involves an approximation which, at best, will add appreciable 'noise'.

SHELXS-96 will usually recognize that it has been given macromolecular Δ*F* data (from the cell volume and contents) and will then set appropriate defaults, so as with small molecules the *.ins* file will often simply consist of TITL..UNIT, then TREF (for direct methods) or PATT (Patterson interpretation) and finally HKLF 3 (because the *.hkl* file contains Δ*F* (HKLF 3) or $(\Delta F)^2$ (HKLF 4). The UNIT instruction should contain the correct number of heavy atoms and the ***square root*** of the number of light atoms in the cell; they may conveniently be assumed to be nitrogen. The mean atomic volume and density printed by the program should of course be ignored. It is strongly recommended that these standard TREF and PATT jobs are tried first before any parameters are varied.

## 15.2 Limitations of Δ*F*-data

Unfortunately there are two fundamental difficulties with the application of direct methods to Δ*F* data. The first is that the negative quartets are meaningless, because the Δ*F*-values represent lower bounds on their true values, and so are unsuitable for identifying the very small *E*-values which are required for the cross-terms of the negative quartets. On the other hand the Δ*F* values do correctly identify the ***largest*** *E*-values, and so the old triplet formula works well. The second problem is that the estimation of probabilities for the triplet formula for the use in figures of merit: what should replace the 1/*N* term (where *N* is the number of atoms per cell) when Δ*F*-data are used?

## 15.3 Direct methods

Most of the recent advances in direct methods exploit either the weak reflections or more sophisticated formulas for probability distributions, so are wasted on Δ*F* data. Nevertheless, direct methods will tend to perform better in space groups with (a) translation symmetry (not counting lattice centering), (b) a fixed rather than a floating origin and (c) no special positions; thus $P2_12_12_1$ (the only space group to fulfill all three criteria) is good but P1, C2, R3 and I4 are unsuitable.

If the standard direct methods run fails to find convincing heavy-atom sites, it should first be checked that the program has put out a comment that it has set the defaults for macromolecular data. The number of phase permutations may have to be increased (the first TREF parameter) or the number of large *E*-values for phase refinement may have to be changed (one should aim for at least 20 triplets per refined phase), but if too many phases are refined the performance is degraded because the $\Delta F$-values only identify the strongest *E*-values reliably. The probability estimates may be changed by modifying the UNIT instruction, or more simply by changing the third TREF parameter, which multiplies the products of the three *E*-values in the triplet probability formula; for small molecules a value in the range 0.75 to 0.95 gives the best probability estimates, but it may be necessary to go outside this range for $\Delta F$-data.

## 15.4 Patterson interpretation

For location of the heavy-atom site by Patterson interpretation of $\Delta F$-data it may well be necessary to increase the number of superposition vectors to be tried (the first parameter on the PATT instruction), since the heavy-atom to heavy-atom vectors may be well down the Patterson peak-list. This number can be made negative to increase the 'depth of search' at the cost of a significant increase in computer time. The second number (the minimum vector length for the superposition vector) should be set to at least 8 Å (and to a larger value if the cell is large), and it can usually be made negative to indicate that special positions are not to be considered as possible heavy atom sites. An advantage of Patterson as opposed to direct methods is that such false solutions can be eliminated at a much earlier stage.

The third PATT parameter is also fairly critical for macromolecular $\Delta F$-data; it is the apparent resolution, and is used to set the tolerances for deconvoluting the superposition map. If - as can easily happen with area detector data - a few $\Delta F$-values are at appreciably higher resolution than the rest of the data, this may fool the program into setting too high an effective resolution. In such cases it is worth experimenting with several different values, e.g. 3.5 Å instead of 3.0 etc. The only other parameter which may need to be altered is maxat, if more than 8 sites are expected.

A typical $\Delta F$ PATT run (e.g. PATT 10 -12 2.5) will produce a relatively large number of possible solutions, some of which may be equivalent. The 'correlation coefficient' (which is defined in the same way as in most molecular replacement programs) is the only useful figure of merit for comparison purposes. Hand interpretation of the 'crossword table' is not as easy as for small molecules, because the minimum interatomic distances are not so useful; it is however still necessary to find a set of atoms for which the Patterson minimum function values are consistently high for at least most of the pairs of sites involved. This information tends to be more decisive for the higher symmetry space groups, because when there are more vectors between symmetry equivalents, it is unlikely that all will be associated with large Patterson values simultaneously by accident.

# 16. CIF, CIFTAB and Electronic Publication

## 16.1 CIF archive format

The *CIF* format represents a major step forward in the archiving, publication and communication of crystallographic data. At last it is possible to publish crystal structures and incorporate structural data into the crystallographic databases without the expensive and error-prone retyping of tables by hand. CIF format also provides a convenient method of transferring data from one program system to another. The ACTA instruction instructs SHELXL to write two CIF-format files: *name.fcf* contains the reflection data and 'name.cif' all other data. These files contain all the items needed for archiving the structure; those answers not known to SHELXL (e.g. the color of the crystal) are left as a question mark. In general the final 'name.cif' file should be edited using CIFTAB or any text editor to replace most of these question marks. The file is then suitable for deposition in the CSD (organic) and ICSD (inorganic crystal structure) databases.

For publication of a routine structure determination via electronic mail it will normally be necessary to add the authors' names, title, text etc., which may also be done in CIF-format; this is followed by the edited contents of one or more *.cif* files each describing one structure (or possibly the same structure at different temperatures etc.). In general SHELXL provides all the CIF identifiers required by Acta Cryst. except those that begin with '_publ'. Further details are given below, and an example of a paper submitted to Acta Cryst. in this way may be found in the file *example.cif* (it has been brought up to date for the 1997 requirements for authors; whether it would pass the new stricter quality controls is another matter!). SHELXL users are strongly recommended to familiarize themselves with the definitive paper by the I.U.Cr. Commission on Crystallographic Data by Hall, Allen & Brown (1991), and with the current Acta Crystallographica Instructions for Authors.

Since the archiving of macromolecular data in CIF format is still being debated, SHELXL only creates a standard 'small molecule' CIF file, suitable for Acta Cryst. etc.; a macromolecular CIF file is likely to contain much more information. However the LIST 6 instruction in the new version of SHELXL does produce a CIF format reflection data file suitable for archiving with the PDB. This file also contains all the information necessary for the calculation of electron density maps, though as yet it appears that no standard macromolecular graphics package is able to read CIF format. Macromolecular coordinates etc. should be deposited in PDB format; SHELXPRO provides the necessary facilities for extending the *.pdb* file produced by SHELXL so that it can be used as a template for deposition.

## 16.2 The auxiliary program CIFTAB

**CIFTAB** is a simple program that reads CIF files and convert them into tables. It may prove useful for padding out Ph.D. theses and for submission of table to old-fashioned journals. It is also intended as an example of how to read CIF files, and it is hoped that SHELX users will be able to modify it for their own purposes. CIFTAB is started by the command:

```
CIFTAB name
```

where name is the first component of the filenames for the structure in question. CIFTAB enables tables to be produced from the *.cif* or *.fcf* files written by SHELXL and provides the following facilities, which may be selected from a simple menu.

Tables of crystal data, atom parameters, bond lengths and angles, anisotropic displacement parameters and hydrogen atom coordinates may be produced in a format specified in a file *ciftab.???* (where *???* is any three letter combination). A standard ASCII file *ciftab.def* is provided; users may use it as a model for preparing standard ASCII tables files for input to word processors etc.

The format file is simply copied to the output file, except that directives (lines beginning with '?' or '$') have a special meaning, '\n\' (where n is a number) is replaced by the ASCII character n (e.g. \12\ starts a new page), and CIF identifiers (which begin with the character '_') are replaced by the appropriate number or string from the CIF file. CIF identifiers may optionally be followed (without an intervening space) by one or more of: '<n', '>n', ':n' and '=n' where n is an integer; the CIF identifier (including qualifier) must be terminated by one space that is not copied to the output file. '<n' left justifies the CIF item so that it starts in column n, and is usually used for strings. '>n' right justifies a string or justifies a number so that the figure immediately to the left of the decimal point appears in column n; if there is no decimal point then the last digit appears in column n. In either case the standard deviation (if any) extends to the right with brackets but without intervening spaces. If '<n' and '>n' are both absent, the CIF item is inserted at the current position. If ':n' is absent the item is treated as a string (see above), otherwise it is treated as a number; n is the power of 10 with which the CIF item should be multiplied, and is useful for converting Å to pm or printing coordinates as integers; n may be negative, zero or positive. '=n' rounds the CIF item (after application of ':n') so that there are not more than n figures after the decimal point; n must be zero or positive.

A line beginning with 'loop_' is repeated until the corresponding loop in the CIF file is exhausted; all the CIF items in the line must be in the same loop in the CIF input file

A line containing at least 4 consecutive underscores is copied to the output file unchanged, and may be used for drawing a horizontal line. There are also two pseudo-CIF-identifiers: '_tabno' is the number of the table, and '_comno' is a number or text string to identify the compound. Both may be set via the CIFTAB menu. '_tabno' but not '_comno' is incremented each time it is used.

An underscore '_' followed by a space may be used to continue on the next line without creating a new line in the output file. Lines beginning with question marks are output to the console (without the leading question mark) as questions; if the answer to the question is not 'Y' or 'y', everything in the format file is skipped until the next line which begins with a question mark. Lines beginning with a dollar '$' are not interpreted as text, but are scanned for the following strings (upper or lower case, quotes not essential):

**'xtext':** output should be formatted for the Siemens SHELXTL XCIF program (which now incorporates XTEXT, which was a separate program in version 4 of SHELXTL).

**'xtext,deutsch':** as above, but translated into German.

The above directive, if present, should be the first line of the format file.

The directive $symops:n, where n is an integer, prints the symmetry operations used to generate equivalent atoms, starting each line of text in column n. These operators are referenced by '#m' (where m is an integer) after the atom name. The line beginning '$symops:n' usually follows the tables of selected bond lengths and angles, torsion angles and hydrogen bonds.

The remaining directives may appear at any point in the format file except immediately after a continuation line marker, but always on a line beginning with '$'.

**'h=none':** leave out all hydrogen atoms.

**'h=only':** leave out all non-hydrogen atoms.

**'h=free':** leave out riding or rigid group hydrogens but include the rest.

**'h=all':** include all hydrogen and all other atoms.

The hydrogen atom directives apply only to tables of coordinates; hydrogen atoms are recognized by the .._type_symbol 'H'. A common user error on writing format files is to forget that 'h=only' etc. applies until it is replaced by another 'h=...' directive! The publication flags can be used to control which hydrogen atoms appear in tables of bond lengths, angles etc.

**'brack':** Atom names should include brackets (if present in the CIF file).

**'nobrack':** Brackets are deleted from the atom names.

**'flag':** Only output items for which the publication flag is 'Y' or 'y'.

**'noflag':** Output all items, ignoring the publication flag.

The default settings are '$h=none,brack,flag'. The standard tables file *ciftab.def* illustrates the use of most of these facilities. CIFTAB extends some of the standard CIF codes to make them more suitable for tables, and also takes special action when items such as _refine_ls_extinction_coef are missing or undefined.

The above description refers to the version of CIFTAB distributed with SHELXL. The simplest method of altering the contents and format of results tables is to create a different ciftab.??? format file (or a collection of such files for various purposes), using the standard file ciftab.def as a starting model. Thus the output can be tailored to different journals, doctoral theses, reports, etc.

The more ambitious user may wish to make some changes in the CIFTAB program itself, to incorporate additional options not provided by the program as distributed. The flexibility of the format file, however, provides most of the facilities that are likely to be needed, and the standard CIFTAB does include a procedure for replacing undefined data items by values taken from one or more other files conforming to CIF rules. Thus items such as diffractometer or area detector operating parameters, details of absorption corrections, and crystal color, which are unknown to SHELXL, can be incorporated from separate files. This is  more reliable than using a text editor.

### 16.3  Using SHELXL CIF files for publication in Acta Crystallographica

The process of converting a virgin SHELXL CIF output file into an electronic manuscript submission for Acta Cryst. Section C may seem at first rather complex and daunting, but the journal's Instructions for Authors are very detailed, and much of the conversion is routine and can be semi-automated; it can soon become an accustomed habit!

The important first step is to be properly informed of what is involved.  The I.U.Cr. makes a variety of useful information available, and it can conveniently be accessed in its most up-to-date form at the World Wide Web location http://www.iucr.ac.uk/welcome.html by any standard Web browser.  Printed Instructions for Authors can be found each year in the journal itself, and copies are available on request from The Managing Editor, I.U.Cr., 5 Abbey Square, Chester CH1 2HU, England.  The Chester office can also supply copies of a technical account of how a CIF becomes a printed paper (reprinted from McMahon, 1993), and of 'A Guide to CIF for Authors' (published in 1995).

For a manuscript describing a single structure, the SHELXL CIF output needs only the addition of a well-defined set of publication information (the items that begin with '_publ'), itself in correct CIF format.  A template for this can be obtained by ftp from I.U.Cr., and the SHELXL CIF output is attached to the end of it.  Into the template are inserted (by any standard text editor) items such as manuscript title, authors' names and addresses, descriptive text, some extra experimental details as necessary (such as chemical synthesis and crystallization details, and a description of hydrogen atom refinement procedures), literature references, acknowledgments, and figure captions.  There is also a place for inserting a formal submission letter.  Some parts of the SHELXL output need changing; in particular, bond lengths and angles to be printed in the journal must be identified by changing their publication flag from '.?'  to 'yes'.

When the CIF appears to be ready for submission, its completeness and validity can be checked anonymously by e-mailing it to the address checkcif@iucr.ac.uk; a report will be automatically generated and returned by e-mail listing and CIF syntax errors and any unrecognized data items.  If there are no errors, the file is also checked for completeness and for some aspects of self-consistency (geometry is checked against coordinates, and possible higher symmetry is searched for).  Any errors or omissions should be corrected and the checkcif procedure repeated, until everything is correct.

Beware of adding anything to e-mailed CIF submissions which does not accord with the syntax rules.  In particular, there must be no non-CIF lines at the beginning or end of the message, and this includes automatically appended e-mail signatures!  These should be disabled or, safer, set up such that every line begins with the # character, which signals a CIF comment line to be ignored.

There is also a facility for previewing a manuscript in the form which will be produced from the CIF.  Sending the CIF by e-mail to printcif@iucr.ac.uk will produce, as a reply message, a

PostScript file of the manuscript; this can be printed or viewed by appropriate software. A useful feature is the highlighting (in bold) of any items which may subsequently be queried by editorial staff, and it may be possible to deal with these potential problems now, before final submission.

When everything is ready and checked, the CIF is e-mailed to med@iucr.ac.uk; after automatic checking is complete, a reply will list any problems requiring attention, will give a Co-Editor reference, and will ask for further material to be sent. This includes structure factor data, figures (diagrams), a copyright transfer form, and a formal signed letter of submission. The last two must still be sent by normal mail, but the others can be transferred electronically (ftp), using the method specified in the Instructions for Authors and the submission acknowledgment. None of these items should be sent until the acknowledgment and reference code arrive.

If these instructions are followed carefully, the editorial process should proceed smoothly! I am grateful to Bill Clegg for writing much of this chapter.

# 17. SHELXA: Empirical Absorption Corrections

The program **SHELXA** has been kindly donated to the system by an **anonymous user**. This applies "absorption corrections" by fitting the observed to the calculated intensities as in the program DIFABS. SHELXA is intended for **EMERGENCY USE ONLY**, eg. when the world's only crystal falls off the diffractometer before there is time to make proper absorption corrections by indexing crystal faces or by determining an absorption surface experimentally by measuring equivalent reflections at different azimuthal angles etc.

SHELXA reads an *.fcf* file written by SHELXL (using LIST 4 or LIST 6 and any combination of MERG, OMIT etc.) and a .raw file in SHELX HKLF 4 format containing "direction cosines", and writes a new SHELX *.hkl* file in HKLF 4 format**. THIS WILL OVERWRITE AN EXISTING .hkl FILE !** A SHELXL-93 .fcf file is not suitable because some information is missing. The following restrictions apply to the use of SHELXA:

**(a)** The structure should not be twinned (racemic twinning is allowed), the data should have been collected from one crystal (inter-batch scale factors should not have been refined), and there may not be a re-orientation matrix on the HKLF instruction. Otherwise there are no restrictions on the type of structure (SHELXA is equally (un)suitable for proteins) or the instructions used in the SHELXL refinement.

**(b)** It is understood that any structure determined by means of this scientifically dubious procedure **WILL NEVER BE PUBLISHED !** The anonymous author of SHELXA has no intention of ever writing a paper about it that could be cited and thereby ruin his reputation.

The absorption is modeled by spherical harmonic functions using full-matrix least-squares more or less by the method of Blessing (1995); nb. it is not this model that should be regarded as dubious, just the way SHELXA misuses it. Data are used for parameter determination if the I/σ(I) ratios for both the observed and calculated intensities exceed a given (by the -t switch) or assumed threshold (equal to 5.0). The -u switch specifies an artificial $\Delta U/\lambda^2$ value that is applied to the calculated intensities; this helps to prevent atoms going NPD, but the default value is zero. The -e and -o switches specify the highest even and odd order spherical harmonics to employ; the refinement could be unstable if these are too high, especially if only part of reciprocal space is sampled, eg. because only an asymmetric unit was collected for a high symmetry structure. Allowed values are (0,2,4,6,8) and (0,1,3,5,7) respectively. Thus:

```
shelxa -t3 -u0.002 -e4 -o1 baddata
```

would read baddata.raw and baddata.fcf and write baddata.hkl, with data with $I>3\sigma(I)$ used to fit the absorption parameters, a $\Delta U/\lambda^2$ of 0.002 effectively added to all current isotropic displacement parameters, and highest even and odd harmonics 4 and 1 respectively. Such UNIX switches will also be recognized under MSDOS, VMS etc.; no spaces are allowed between the letter and value. The values employed for these switches are summarized by the program (on the standard output device). The filename stem (here *baddata*) must come last. Usually the default values should prove sensible, ie:

```
shelxa baddata
```

The data may be re-processed when, for example, extra atoms are added; however, as with DIFABS, best results are obtained if the procedure is last run with the final ISOTROPIC model; re-running it after anisotropic refinement will result in a deterioration of the structure and (most important) the *R*-factors.  The ΔU fudge should not be used repetitively, because the effects will be cumulative !

Note that all esd's estimated by SHELXL using data "corrected" in this way will be invalid unless the number of parameters used in the absorption model is input as the third L.S. parameter.  This number depends on the settings of the -e and -o switches and is output by SHELXA.

The program can read either standard SHELX direction cosines (relative to the crystal reciprocal axes), or orthogonal direction cosines calculated by the method given in Blessing's paper.  Siemens and Stoe write the SHELX .raw (HKLF 4) format as standard, for CAD4 diffractometers a suitable data reduction program is available from Klaus Harms at the University of Marburg.  Users of other makes of diffractometer and area detectors will enjoy writing their own programs to generate direction cosines using Blessing's method; the anonymous author of SHELXA is of course not able to enter into any correspondence about this!  For very large structures it may be necessary to change the number of reflections the program can handle by increasing the values of MR and MF in the PARAMETER statement at the start of the main program, and recompiling it.

# 18. Frequently Asked Questions

**Q1:** Please send me a copy of SHELX-76. I am afraid that I cannot use the new version because **my diffractometer measures *F*-values, not intensities**.

**A:** Buy a CCD detector. They measure intensities! [In fact, diffractometers measure intensities too. You just need the right data reduction program. If you are desperate you can even feed SHELXL with *F*-values using HKLF 3.]

**Q2:** When I start SHELXL on my PC the disk rattles loudly for several hours and smoke comes out of the back. Is this a bug?

**A:** You must be trying to run SHELX under some version of **WINDOWS**! The best solution is to reformat the hard disk and install LINUX. If you are running WINDOWS-95 an inferior alternative is to 'Reboot to DOS' (as recommended for games programs).

**Q3:** The **referee rejected my paper** because the weighted *R*-factor was too high and because the stupid program had forgotten to fix the y coordinate of one atom to fix the origin in space group $P2_1$. What should I do?

**A:** Try another journal; if you emphasize the 'biological relevance' enough, they may not notice the *R*-factor! Note that wR2 (based on intensities and all data) is of necessity 2 to 3 times higher than wR1 (based on *F* and leaving out reflections with say $F<4\sigma$). Unfortunately SHELXL cannot work out wR1, because the weighting scheme for intensities does not apply to *F*-values. It is better to quote the *unweighted* R1 (with or without a 4σ threshold) anyway, because it is too easy to cheat on wR2 by modifying the weights!

It is no longer necessary or desirable to fix the origin by fixing coordinates, the program applies appropriate *floating origin restraints* automatically when they are needed.

**Q4:** The program tells me to refine **extinction**, this does reduce the *R*-factor but the extinction parameter becomes very large although my crystal could hardly be described as 'perfect'. Is this reasonable?

**A:** No. The most likely causes of large apparent extinction are: (a) you have input *F* with HKLF 4, (b) A few reflections that should be very strong have been measured as weak because they were cut off by the beam-stop, (c) your counter was saturating and an inadequate dead-time correction was made (in the case of an image plate this is an 'overload'), or (d) your counter was defective or the energy discrimination was set wrongly. Overloads may be eliminated by 'OMIT *h k l*' if necessary.

**Q5:** The structure could only be solved in **P1**, not $P\overline{1}$, but on refinement some of the bond lengths and U-values are wildly different in the two molecules. If I use SAME the geometries of the two molecules become very similar but how do I restrain the $U_{ij}$ components of equivalent atoms to be the same?

**A:** You could use EADP, but it might be better to look for the inversion center instead, otherwise you will probably be **'marshed'**.

**Q6:** I included batch numbers in the *.hkl* file and BASF parameters in the *.ins* file, but the stupid program still **didn't refine the batch scale factors**!?

**A:** You need MERG 0 (the default MERG 2 will average the batch numbers).

**Q7:** How do I obtain the molecular replacement program **PATSEE**?

**A:** PATSEE has been maintained by its author, Ernst Egert, since he moved from Göttingen to the University of Frankfurt. He can be contacted by fax (+49-69-7982-9128) or email (bolte@chemie.uni-frankfurt.d400.de).

**Q8:** What should I do about **'may be split'** warnings?

**A:** Probably nothing. The program prints out this warning whenever it might be possible to interpret the anisotropic displacement of an atom in terms of two discrete sites. Such atoms should be checked (e.g. with the help of an ORTEP plot) but in many cases the single-site anisotropic description is still eminently suitable.

**Q9:** I get the message ' ** **UNSET FREE VARIABLE** FOR ATOM ... **' but I haven't used any 'free variables'!?

**A:** There is a typo in your atom coordinates, e.g. a decimal point missing or replaced by a comma.

**Q10:** After using SHELXPRO to prepare the .ins file from a PDB file and then running SHELXL, I get the message: ' ** **No match for 2 atoms in DFIX** ** ' !?

**A:** This message probably refers to the fact that SHELXPRO labels the oxygens of the carboxy-terminus OT1 and OT2 so that special restraints can be applied, so there is no atom called 'O' in this residue. This is normal and can be safely ignored. Other similar messages, also messages about bad CHIV or AFIX connectivity, should be investigated (by checking the extra information, including the connectivity table, given in the *.lst* file) to see if they can be ignored safely or not. If the initial geometry is poor, it may be necessary to edit the automatically generated connectivity table with BIND and FREE.

**Q11:** The program prints out a **Flack *x* parameter** of 0.3 with an esd of 0.05. Is the crystal racemically twinned?

**A:** Not necessarily! The Flack parameter estimated by the program in the final structure factor calculation ignores correlations with all other parameters (except the overall scale factor). Since these parameters may have refined so as best to fit a wrong absolute structure, it is quite possible to get an estimate of about 0.3 for the Flack parameter when the true value is 1, i.e. the structure needs to be inverted and is not racemically twinned. On the other hand a value close to zero with a small esd is a strong indication that the absolute structure is correct. If there is any doubt the Flack parameter should be refined together with all the other parameters using TWIN and BASF.

**Q12:** Neither direct methods nor Patterson interpretation in **SHELXS** can find the 24 selenium atoms from the **MAD data** of my selenomethionine labeled protein.

**A:** I'm not surprised.

**Q13:** How does one set up **restraints for a non-standard residue** in a protein for SHELXL?

**A:** First find a suitable fragment in a database such as the CSD, then calculate all 1,2- and 1,3-distances and turn them into DFIX and DANG instructions resp. FLAT and (zero chiral volume) CHIV restraints can easily be added by hand. If the structure contains a number of identical units such as sulfate ions, SADI or SAME can be used instead, then it is not necessary to invent any target values.

**Q14:** What is the **worst resolution** that is acceptable for: (a) solution of a structure by direct methods using SHELXS, (b) refinement with SHELXL?

**A:** Direct methods assume randomly distributed resolved atoms. Direct methods are crucially dependent on having atomic resolution data, say better than 1.2Å. A good rule of thumb is that a least one half of the theoretically possible number of reflections between 1.1 and 1.2Å should have been measured with $I>2\sigma$ for direct methods to be successful, though this rule can be relaxed somewhat for centrosymmetric structures and structures containing heavier atoms. In particular the resolution is not so critical for the location of heavy atoms from $\Delta F$-data, provided that the minimum distance betwen heavy atoms is much greater than the resolution.

SHELXL lacks the energy terms used by e.g. X-PLOR for refinement against low-resolution data. This imposes an effective limit of about 2.5Å, but this limit may be extended a little to lower resolution if NCS restraints can be used.

# 19. SHELX-97 Installation

Before trying to install the programs, it is worth checking with the SHELX homepage at http://shelx.uni-ac.gwdg.de/SHELX/ to see if there are any last-minute changes and whether other users have encountered problems on particular machines. The ftp site and CDROM contain the following files and subdirectories:

***readme*** - Installation instructions, last-minute changes, changes since SHELXL-93 etc..

***shelx.htm*** and ***shelxman.htm*** - On-line help in HTML format: requires a browser such as Netscape. *shelx.htm* contains the same general information as in README, and calls *shelxman.htm* that includes summaries of commands etc. ***applfrm.htm*** is the application form in html format. The file extensions will need to be changed to .html to active these files. three-letter extensions are used in the release for compatibility with MSDOS.

Subdirectory ***'unix'*** contains the sources of all programs for relatively standard UNIX systems. These should also compile successfully on many other operating sytems too (except VMS).

Subdirectory ***'vms'*** contains the VMS sources for Digital computers.

Subdirectory ***'doc'*** cotains the full manual in WINWORD 6 format, one file per chapter. It is designed to print on letter sized paper.

Subdirectory ***'ps'*** cotains the full manual in Postscript format, one file per chapter. It is designed to print on letter sized paper.

Subdirectory ***'egs'*** contains the test jobs and other examples files.

Subdirectory ***'ibm'*** contains the IBM RS6000 executables (these also execute on the IBM Power-PC series).

Subdirectory ***'sgi'*** contains the SGI IRIX executables; they should run under IRIX 5.3 or later with the R4000 series processors. For other systems it is desirable to recompile to obtain programs that execute faster even if the precompiled versions run correctly.

Subdirectory ***'linux'*** contains the LINUX executables for Intel processors.

Subdirectory ***'dos'*** contains the pure MSDOS executables. These may or may not run in the DOS windows under WINDOWS or OS2.

In addition, the ftp login directory contains gzipped tar files of the above subdirectories (e.g. ***unix.tgz***). These are convenient for down-loading with ftp as shown in the next section. The current sizes of these files in bytes are given on the SHELX homepage and should be checked to ensure that transmission is complete.

### 19.1 Installing the precompiled versions

In many cases it will be possible to use the precompiled versions provided. The executable programs (and the file ciftab.def) should simply be copied from the appropriate directory on the CDROM or ftp site to a directory on your machine. This directory should be specified in the 'PATH' so that the executables can be found. On UNIX systems the lazy way is to copy the programs into /usr/bin; on MSDOS systems they are usually copied to C:\EXE and this directory name is then added to the PATH specified in AUTOEXEC.BAT. You may also wish to copy the documentation and examples files.

As an example we shall take a PC running Linux; the following files should be fetched to your working directory by ftp (binary transfer !); for most other UNIX systems the installation procedure is similar:

*linux.tgz, ps.tgz, egs.tgz*, *shelx.htm* and *shelxman.htm*

The three gzipped tar files can then be expanded:

```
gunzip *.tgz
tar -xvf ps.tar
tar -xvf egs.tar
tar -xvf linux.tar
```

which will create the subdirectories ***ps, egs*** and ***linux***. The executables can be copied to */usr/bin* (needs system manager priviledges !):

```
cp linux/* /usr/bin
```

Under LINUX it is particularly easy to print the documentation, because lpr can recognize and print Postscript even on a non-Postscript printer:

```
lpr ps/*.ps
```

The on-line help files *shelxl.htm* and *shelxman.htm* should be renamed (mv) to *shelxl.html* and *shelxman.html* (the three-letter extension was needed for MSDOS systems !) and copied to a generally accessible directory; they may then be viewed with Netscape or any other HTML browser. These files are NOT copyrighted and you are welcome to improve and extend them as you wish for non-commercial purposes. *shelx.htm* calls *shelxman.htm* and applfrm.htm (the application form) It contains all the information from 'README' (which is a plain ASCII text file) plus a summary of the documentation (the full documention is available in WINWORD 6 format in subdirectory ***'doc'*** and in Postscript form in subdirectory ***'ps'***).

### 19.2 Program compilation under UNIX (and other operating systems)

The UNIX version has been designed to be easy to compile on a wide rangeof UNIX (and other) systems. The resulting compiled programs do not needany environment variables or hidden files to run; it is simply necessarythat the executable program is accessible via the PATH or an alias. The simplest way is to copy the executables into */usr/bin*.

The SGI executable of SHELXL was compiled under IRIX 5.3 as follows:

```
f77 shelxl.f -O3 -c
f77 shelxlv.f -O3 -c
f77 shelxl.o shelxlv.o -o shelxl
```

The compilation for other UNIX systems should be similar. **IT IS NECESSARY TO BE VERY CAREFUL ABOUT OPTIMIZATION**. The safest is to compile without any optimization first (-O0 rather than -O3 in this case), run the ags4 and 6rxn tests, and rename the resulting output files **.res, *.lst, *.fcf* and **.pdb*. Then recompile with highest optimization (-O3), rerun the tests, and use the UNIX diff instruction to compare the results with those from the unoptimized version. Small differences in the last decimal place do not matter, and of course the CPU times will differ, but if there are significant differences then the optimization level should be lowered and the tests repeated. For some systems (including certain SG Challenge and Digital Alpha systems), only the shelxlv.f file (containing the rate-determining routines) can be compiled with the highest optimization level; shelxl.f must be compiled at a lower level.

*The shelxl*.f source contains the following routines that may be different or not available for some FORTRAN compilers:

IARGC and GETARG: these have always worked so far; if necessary the standard C routines could be adapted since the specifications are the same.

EXIT and FLUSH: if these cause problems they can safely be commented out in the source or replaced by the dummy FORTRAN subroutines provided. EXIT is used in 2 places to tidy up before terminating, FLUSH(6) occurs once to flush the logfile so that a batch job can be watched as it runs (eg. with tail -f).

ETIME and FDATE: most UNIX FORTRAN compilers will recognize these routines. For compilers that do not, both FORTRAN and C substitutes are provided. Usually at least one substitute will work, but the following points should be checked carefully:

Some FORTRAN compilers add an underscore to the end of procedure names before searching them in a library (this avoids confusion with standard C routines that happen to have the same names). The C versions are provided both with underscores (files *fdate_.c* and *etime_.c*) and without (*fdate.c* and *etime.c*).

The FORTRAN substitute for FDATE (*fdate.f*) calls FORTRAN routines TIME and DATE. Some compilers link in the C procedure 'time' instead, with strange results because the parameters may be different. The alternative fdate.c is safer.

The C replacement for ETIME (*etime.c*) may suffer from time 'wrap-around' if a large value for CLOCKS_PER_SEC (say 1000000) is combined with the use of a 32-bit or shorter integer to pass the time (!). Check the type time_t and CLOCKS_PER_SEC in */usr/lib/sys/time.h* (you may need to consult a Guru).

The IBM RS6000 executable was compiled as follows; note that *fdate.f* cannot be used for the reason given above, and that the underscore is not needed after fdate in the C subroutine. The FLUSH routine was replaced by the dummy.

```
xlf shelxl.f -O -c
xlf shelxlv.f -O -c
xlf etime.f -c
xlf flush.f -c
xlc fdate.c -c
xlf shelxl.o shelxlv.o fdate.o etime.o flush.o -o shelxl
```

The Linux executable was compiled using the GNU FORTRAN and C compilers as follows. The Absoft compiler is not recommended because optimization gives bad numerical results, and f2c produces slower code. The two C routines require underscores.

```
g77 shelxl.f shelxlv.f etime.c fdate.c -O3 -ffast-math -o shelxl
```

SHELXS uses the same routines and should be compiled just like SHELXL.The rate-determining routines are in *shelxsv.f*, the rest in *shelxs.f*.

One commented line near the start of SHELXL and SHELXS needs to be changed if these programs should write MSDOS format ASCII text files rather than UNIX format when run on a UNIX system. This is useful for a heterogeneous UNIX/MSDOS network, because the UNIX versions of all SHELX programs can read MSDOS format files. but not vice versa.

The remaining programs do not require optimization (except possibly SHELXA and SHELXPRO) and do not require FDATE, ETIME, FLUSH and EXIT, so they are easier to compile. For example under IRIX 5.3:

```
f77 shelxpro.f -O3 -o shelxpro
f77 shelxwat.f -o shelxwat
f77 ciftab.f -o ciftab
f77 shelxa.f -O3 -o shelxa
```

Unlike SHELXL and SHELXS, there are some intentional deviations from the strict FORTRAN-77 standard in these programs. REAL*8 and list-directed reading of internal files are used in several cases, and SHELXPRO uses types INTERGER*2 and BYTE in order to produce binary map files for O. Most FORTRAN compilers have no problems with these extensions, but may output warning messages.

Note that CIFTAB will search the current directory for a specified format file, and if it doesn't find it there it will look for it it a directory that is defined in the source. Unless this is edited before compiling, the directory is set to /usr/bin, so if the executable programs are located in /usr/bin the file ciftab.def (the default format file) should be there too.

### 19.3 Program compilation under VMS

The following instructions may be tried for compilation of the VMS sources under OpenVMS:

```
$fort/opt/ass=(noac,nodu)/align=all shelxs+shelxsv
$link shelxs
$fort/opt/ass=(noac,nodu)/align=all shelxl+shelxlv
$link shelxl
$fort/noopt shelxpro
$link shelxpro
$fort/noopt shelxwat
$link shelxwat
$fort/noopt shelxa
$link shelxa
$fort/noopt ciftab
$link ciftab
```

It may be necessary to split up the programs into subroutines to prevent the compiler running out of virtual memory. The files produced by the test jobs for SHELXL and SHELXS MUST be compared with those obtained using unoptimized versions of these programs (compiled with /noopt instead of /opt; note that /opt is usually the default) since optimizing errors are common for Digital compilers; there is a DIFF instruction in VMS that can be used for this. The remaining programs are not very CPU-intensive and so should not be optimized. If optimization causes errors, it is worth trying just to optimize *shelxsv.f* and *shelxlv.f* (which contain the rate determining routines) but not the rest. The executables need to be defined as follows:

**shelxs :== $ disk:[directory]shelxs** etc.

where 'disk' and 'directory' should be replaced by the appropraite local names and the programs are run (after preparing the files name.ins and name.hkl) by e.g.

**shelxl name**

SHELXWAT and SHELXA accept UNIX-type switches (even under VMS); they MUST come before the filename, e.g.

**shelxwat -h name**

No other files or parameter settings are required to run the programs, except that the file ciftab.def or a user-produced format definition file should be in the current directory when CIFTAB is run; if this file cannot be found in the current diectory, CIFTAB searches for it in a directory specified in the source.

## 19.4 Parallel and vector machines

SHELXL and SHELXS are designed to run very efficiently on vector computers (such as older Cray and Convex machines); no changes should be needed to the code. Unfortunately the crystallographic algorithms involved are less suitable for parallel computers (or multiprocessor systems); in such cases the avaliable computer resources are more efficiently used by running several jobs simultaneously, one per processor.

## 19.5 SHELXH - version of SHELXL for very large structures

SHELXH is a special version of SHELXL for the refinement of very large structures (with more than about 10000 unique atoms). The only difference between shelxh.f and shelxl.f is the first FORTRAN statement in which the array dimensions are specified by means of a PARAMETER statement; shelxh wascompiled (using shelxlv.f etc.) exactly as described above for shelxl. Large versions of shelxs, shelxpro and shelxa may be created in the same way, but it is rather unlikely that they will ever be required. Further details are provided by comments in the respective sources.

SHELXL will print a suitable error message if it is necessary to increase the dimensions of the large arrays A or B. An additional warning sign is the 'maximum vector length' printed in the .lst file at the beginning of each refinement cycle; if it is too small (say less than 32) the program will still run, but with reduced efficiency. This applies to all computers but is especially serious on a vectorizing computer such as an older Cray or Convex.

A little care and fine-tuning is required so that such large structures can be refined efficiently. If the computer does not have enough physical memory available, or if the 'maximum vector length' is set too large, shelxh will run in disk exercising mode. This 'maximum vector length' refers to the number of reflections that are processed in one vector run, which may be smaller than the number in the input/output buffer. Some trial and error is needed to set the maximum allowed value so that the physical memory is fully exploited with a minimum of disk I/O for the virtual memory swap file. This number is set as the fourth parameter on the L.S. or CGLS instruction, and should be a multiple of 8; a good value to try for a 64MB computer is 64 (the third number on the L.S. or CGLS instruction is almost always zero). The array B is used as working space for these vectors (CGLS and L.S.) as well as for the least-squares matrix (L.S.). If the array B is not big enough, the program will use a smaller maximum vector run.

A further point to note for refinement of structures with more than 10000 atoms is that the SIMU and DELU instructions need to be broken up into several overlapping instructions, because the maximum number of atoms that can be referenced on any single instruction was arbitrarily set to 10000 (I never expected that this limit would be reached!).

# 20. SHELX-97 Application Form

**SHELXL-97 LICENSE REGISTRATION FORM**

Title/name:

Postal address:

Fax:

-----------------------------------------------------------------------------------------------------------

Email (legible!):

-----------------------------------------------------------------------------------------------------------

[ ] I wish to license SHELX-97 for use at the following for-profit firm or institution. I agree that within two months I will either destroy all copies of the programs in my possession or pay the license fee of US$2499. This license fee covers the use of the complete SHELX-97 for an unlimited time on an unlimited number of computers of any type at one geographical location:

-----------------------------------------------------------------------------------------------------------

[ ] I wish to license SHELX-97 for exclusively non-commercial purposes at the following not-for-profit institution only:

-----------------------------------------------------------------------------------------------------------

Please tell me how to obtain SHELX-97 by ftp [ ]; I already possess a copy of SHELX-97 [ ]

Please supply it on CDROM* [ ] / 100MB ZIP diskette* [ ] (*$99 for academic users)

[ ] I agree to cite SHELX-97 in all publications for which it was useful.

[ ] I agree that the author has no liability for any damage or loss caused by the programs.

[ ] Please send me a receipt for enclosed cheque [ ] Please send me an invoice

[ ] Please send direct bank transfer information [ ] No payment required (academic/ftp)

-----------------------------------------------------------------------------------------------------------

Signed: Date:

This form should be returned to George Sheldrick, Institut Anorg. Chemie, Tammannstr. 4, D37077 Göttingen, Germany by post or fax (+49-551-392582). Unsigned, emailed, incomplete (are the right boxes ticked?) or illegible forms will be returned by normal post for completion!

# References

Allen, F. H. & Rogers, D. (1969). *Acta Cryst.* **B25**, 1326 -1330.

Arnberg, L., Hovmöller, S. & Westman, S.(1979). *Acta Cryst.* **A35,** 497 - 499

Armstrong, D. R., Herbst-Irmer, R., Kuhn, A., Moncrieff, D., Paver, M. A., Russell, C. A., Stalke, D., Steiner, A. & Wright, D. S. (1993). *Angew. Chem. Int. Ed. Engl.* **32**, 1774 - 1776.

Bernardinelli, G. & Flack, H. D. (1985). *Acta Cryst*. **A41**, 500 - 511.

Blessing, R. (1995). *Acta Cryst.* **A51**, 33-38.

Brünger, A. T. (1992). *Nature (London),* **355**, 472 -.475.

Brünger, A. T. Kurijan, J. & Karplus, M. (1987). *Science*, **235**, 458 - 460.

Buerger, M. J. (1959). *Vector Space*, Chapter 11. Wiley, New York.

Clarage, J. B. & Phillips, G. N. (1994). *Acta Cryst.* **D50**, 24 - 36.

Cruickshank, D. W. J. (1996). CCP4 Meeting, Chester.

DeLaMatter, D., McCullough, J. J. & Calvo, C. (1973). *J. Phys. Chem.* **77**, 1146 - 1148.

Didisheim, J. J. & Schwarzenbach, D. (1987). *Acta Cryst.* **A43**, 226 - 232.

Domenicano, A. (1992). *Accurate Molecular Structures*, edited by. A. Domenicano & I. Hargitttai, Chapter 18. Oxford University Press: Oxford, UK.

Driessen, H., Haneef, M. I. J., Harris, G. W., Howlin, B., Khan, G. & Moss, D. S. (1989). *J. Appl. Cryst.* **22**, 510 - 516.

Dunitz, J. D. & Seiler, P. (1973). *Acta Cryst***.** **B29**, 589 - 595.

Engh, R. A. & Huber, R. (1991). *Acta Cryst.* **A47**, 392 - 400.

Flack, H. D. (1983). *Acta Cryst.* **A39***,* 876 - 881.

Flack, H. D. & Schwarzenbach, D. (1988). *Acta Cryst.* **A44**, 499 - 506.

Frazão, C., Soares, C. M., Carrondo, M. A., Pohl, E., Dauter, Z., Wilson, K. S., Hervás, M., Navarro, J. A., De la Rosa, M. A. & Sheldrick, G. M. (1995). *Structure* **3**, 1159-1169.

Fujinaga, M. & Read, R. J. (1987). *J. Appl. Cryst.* **20**, 517 - 521.

Giacovazzo, C. ed. (1992). *Fundamentals in Crystallography*, I.U.Cr. & O.U.P.: Oxford, UK.

Gros, P., van Gunsteren, W. F. & Hol, W. G. (1990). *Science,* **249,** 1149 - 1152.

Hall, S. R., Allen, F. H. & Brown, I. D. (1991). *Acta Cryst*. **A47**, 655 - 685.

Hendrickson, W. A. & Konnert, J. H. (1980). *Computing in Crystallography,* edited by R. Diamond, S. Ramaseshan & K. Venkatesan. pp. 13.01 - 13.25. I.U.Cr. and Indian Acad. Sci.: Bangalore, India.

Hirshfeld, F. L. (1976). *Acta Cryst.* **A32**, 239 - 244.

Hirshfeld, F.L. & Rabinovich, D. (1973). *Acta Cryst.* **A29,** 510 - 513.

Hoenle, W. & von Schnering, H. G. (1988). *Z. Krist.* **184**, 301 - 305.

Irmer, E. (1990). Ph.D. Thesis, University of Göttingen. Germany.

Jameson, G. B., Schneider, R., Dubler, E. & Oswald, H. R. (1982). *Acta Cryst.* **B38**, 3016 - 3020.

Jones, P. G. (1988). *J. Organomet. Chem*. **345**, 405.

Kilimann, U., Noltemeyer, M., Schäfer, M., Herbst-Irmer, R., Schmidt, H. G. & Edelmann F. T. (1994). *J. Organomet. Chem.* **469**, C27 - C30.

Kleywegt, G. J. (1996). *Acta Cryst*. **D52**, 842 - 857.

Kleywegt, G. J. & Jones, T. A. (1996). *Structure* **4**, 1395-1400.

Lamzin, V. & Wilson, K. S. (1993). *Acta Cryst.* **D49**, 129 - 147.

Langridge, R., Marvin, D. A., Seeds, W. E., Wilson, H. R., Hooper, C. W., Wilkins, M. H. F. & Hamilton, L. D. (1960). *J. Mol. Biol.* **2**, 38 - 64.

Larson, A. C. (1970). *Crystallographic Computing,* edited by F. R. Ahmed, S. R. Hall & C. P. Huber, pp. 291 - 294. Copenhagen, Munksgaard.

Laskowski, R. A., MacArthur, M. W., Moss, D. S. & Thornton J. M. (1993). *J. Appl. Cryst.* **26**, 283 - 291.

LePage, Y. (1982). *J. Appl. Cryst.* **15**, 255 - 259.

Luzzati, P. V. (1952). *Acta Cryst*. **5**, 802-810.

Marquardt, D. W. (1963). *J. Soc. Ind. Appl. Math*. **11**, 431-441.

Maetzke T. & Seebach, D. (1989). *Helv. Chim. Acta* **72**, 624 - 630.

McMahon, B. (1993). *Acta Cryst.* **C49**, 418-423.

Parkin, S., Moezzi, B. & Hope, H. (1995). *J.Appl.Cryst.* **28**, 53-56.

Pathe, E. & Gelato, L. M. (1984). *Acta Cryst*. **A40**,169 - 183.

Peterson, S. W., Gebert, E., Reis Jr., A. H., Druyan, M. E. & Peppard, D. F (1977). *J. Phys. Chem.* **81**, 466 - 471.

Pratt, C. S., Coyle, B. A. & Ibers J. A. (1971). *J. Chem. Soc.* 2146 - 2151.

Richardson, J.W. & Jacobson, R.A. (1987). *Patterson and Pattersons*, edited by J. P. Glusker, B. K. Patterson & M. Rossi, 310 - 317. I.U.Cr. and O.U.P.: Oxford.

Roesky, H. W., Gries, T., Schimkowiak, J. & Jones, P. G. (1986). *Angew. Chem. Int. Edn*. **25**, 84 - 85.

Rollett, J. S. (1970). *Crystallographic Computing,* edited by F. R. Ahmed, S. R. Hall & C. P. Huber, pp. 167 - 181. Copenhagen, Munksgaard.

Sheldrick, G.M. (1990). *Acta Cryst.* **A46**, 467 - 473.

Sheldrick, G.M. (1992). *Crystallographic Computing*, edited by D. Moras, A. D. Podjarny & J. C. Thierry, pp. 145 - 157. I.U.Cr. and O.U.P.: Oxford, UK.

Sheldrick, G. M., Dauter, Z., Wilson, K. S., Hope, H. & Sieker, L. C. (1993). *Acta Cryst.* **D49**, 18 - 23.

Sheldrick, G. M. & and R. O. Gould, R. O. (1995). *Acta Cryst.* **B51**, 423-431.

Stenkamp, R. E., Sieker, L. C. & Jensen, L. H. (1990). *Proteins, Struct. Funct. Genet.* **8**, 352 - 364.

Taylor, R. & Kennard, O. (1982). *J. Mol. Struct.* **78**, 1 -28.

Tronrud D. E. (1992). *Acta Cryst.* **A48**, 912 - 916.

Trueblood, K. N. & Dunitz, J. D. (1983). *Acta Cryst.* **B39**, 120 - 133.

Walker, N. & Stuart, D. (1983). *Acta Cryst.* **A39**, 158 - 166.

Watkin, D. (1994). *Acta Cryst.* **A50**, 411 - 437.

Wilson, A. J. C. (1976). *Acta Cryst.*, **A32**, 994 - 996.

# Index