Alphabetical list of SHELXL instructions

The following instructions may be used in the *.ins* file for SHELXL. 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 SHELXL-97 instructions TIME, HOPE and MOLE have been deprecated.

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*). This facility is primarily intended for long lists of restraints and LAUE instructions that would be the same for each refinement of a given structure. '++filename' also reads instructions from the named file but copies them to the *.res* file where appropriate. Such include files may call further files using '+filename' but not '++filename'. '+filename' is echoed to the *.res* file if it is in the *.ins* file but not if it is in a '+' or '++' file, '++filename' is never echoed.

The *.ins* file should always start with TITL, CELL, ZERR, [LATT], SYMM, [NEUT], SFAC, [DISP] and UNIT in that order and finish with HKLF and END. Instructions in square brackets are not always required. There may be more than one SYMM, SFAC and DISP instructions.

Atom instructions begin with an atom name (up to 4 characters, of which the first must be a letter) that do not correspond to any of the SHELXL instructions, 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 (in $Å^2$). Note that different program systems may differ in their order of U_{ij} components! The exponential factor takes the form $\exp(-8\pi^2 U[\sin(\theta)/\lambda]^2)$ for an isotropic displacement parameter U and:

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

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 combination of atom name, PART and RESI numbers must be unique. 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. For example an atom on a fourfold axis will usually have 10.25 in the sof position.

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•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•(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•fv_2 and 0.25•(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 recommended: the program will automatically work out and apply the appropriate positional, sof 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. If sof 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.

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 sofs 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 sofs 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 an isotropic 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 U-value is not refined independently but is updated after every least-squares cycle.

ABIN n1 n2

Reads h, k, l, A and B from the file *name.fab*, where A and B are the real and imaginary components of a partial structure factor. This file is read in free format (numbers separated by one or more spaces) with one reflection per line, and is terminated by the end of the file, a blank line, or a dummy reflection 0,0,0. Any following information in this file is ignored and may be used for comments. The reflections may be in any order, duplicates and systematic absences are ignored. Symmetry equivalents are generated automatically. At least one equivalent of each reflection used in the refinement, including all reflections in all twin components, should be present in this list, superfluous reflections in the *.fab* file (e.g. outside the resolution limits) are ignored. In the case of twinning, the A and B values should refer to the untwinned structure, but they are used to calculate the structure factors for all twin components. Thus the *.fcf* file created using the new **LIST 8**, which has already been

'detwinned' and merged (using Friedel's law only for centrosymmetric structures) but still contains the anomalous contributions, may be used as an aid to generating them.

The input A and B values are multiplied by $k.\exp(-8\pi^2 U\sin^2\theta/\lambda^2)$, where k is the value of free variable n_1 and U is the value of free variable n_2. If n_2 is omitted, U is set to zero, and if n_1 is also omitted, k is fixed at 1.0 (in which case the A and B values should be on an absolute scale of electrons per unit-cell). SUMP restraints may be applied to these free variables. The partial structure factor contributions might come from a solvent mask (for a macromolecule) or a blob of unresolved solvent density for a small molecule, e.g. in a channel along a cell axis, as modeled by the **squeeze** method in PLATON. In the latter case it might be appropriate to set the third L.S. parameter to the number of parameters that would have been required to model such a solvent region by fitting disordered solvent molecules to it, so that the standard uncertainties are estimated correctly. n_1 may be made negative to force the program not to assume Friedel's law when generating equivalents of the input A and B values; this is only required when the partial structure factors have significant anomalous contributions.

ACTA 2θ$_{full}$[#]

A CIF file is written to *.cif* for archiving and verification. ACTA automatically sets the BOND, FMAP 2, MERG 2, PLAN and LIST 4 instructions, and may not be used with other FMAP, MERG 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.

$2\theta_{full}$ is used to specify the value of 2θ for which the program calculates the completeness of the data. The default is the 2θ value at which $\sin(\theta)/\lambda=0.6$. In addition the completeness is calculated up to the maximum value of 2θ for the reflection data. SHELXL ignores systematic absences and the reflection 0,0,0 when calculating the completeness.

Two keywords may apper on the ACTA instruction in addition to numerical parameters. 'NOHKL' switches off the default embedding of the *.res*, *.fab* and *.hkl* into the *.cif* file. This should only be used for good reason, e.g. because the CIF file is intended for input to a graphics program that has problems with large input files. It is scientifically very desirable that the full data are deposited and archived. The keword 'TABS' attaches '_a' to atom names in PART 1, '_b' to atom names in PART 2 etc., exactly as employed in the *.lst* file. This makes the lists of bond lengths and angles etc. easier to understand, especially if the same atom names are used for different disorder components (standard practice in PDB format). The use of the TABS keyword may cause atom names to become too long for certain programs.

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. 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; 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 U_{eq} of the preceding normal atom). Each AFIX instruction must be followed by the required number of hydrogen or other atoms.

Although AFIX instructions were intended for placing hydrogen atoms, it is also possible to use AFIX 30, 120 or 130 to set up ideal CF_3 groups. However these may not be refined with a riding model because the fluorine atoms make a much larger contribution to the calculated structure factors that the carbon atom; the refinement would be unstable. The use of DFIX or SADI is recommended, e.g.

```
DFIX 1.328 C1 F1 C1 F2 C1 F3
DFIX 2.125 F1 F2 F1 F3 F2 F3
SADI 0.1 C2 F1 C2 F2 C2 F3
```

Note the relatively soft SADI restraint to the next atom to allow the CF_3 group to tilt a little. Alternatively, if no alternative conformation is involved, an AFIX 6 or AFIX 9 rigid group could be used. The above SADI instruction could be added if necessary.

```
C2 ...
AFIX 9
C1 ...
AFIX 135
F1 3 0 0 0
F2 3 0 0 0
F3 3 0 0 0
AFIX 0
```

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. 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 less than tetrahedral). This option is also suitable for a 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 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 Currently identical to m=6, reserved for other use in the future (e.g. OH_2).

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.

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-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 for one -CH_3 component, then the three for 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 without the hydrogen atoms and the torsion angle is set that maximizes the sum of the difference densities at the three hydrogen positions. This is usually followed by refinement of the torsion angle (AFIX 137).

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 that 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 zero coordinates 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, sof and U or U_{ij} are fixed.

n=2 The sof and U (or U_{ij}) are fixed, but the coordinates are free to refine.

n=3 The coordinates, but not the sof or U (or U_{ij}), ride on the previous non-riding atom. 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 (n=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.

n=5 The next atom(s) are 'dependent' atoms in a rigid group. 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 to all the X-H bonds to the same atom. 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 in 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.

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 some obvious cases, but in many cases this is not logically possible.

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 that 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.

ANSC six coefficients

Applies anisotropic scaling. This is only usually of practical use for isotropic refinements of macromolecules, because it would be 100% correlated with the individual anisotropic ADPs, though it might be applicable in cases where only the heavy atoms are refined anisotropically. In the first job, ANSC is entered without any parameters in the first run and is written out to the *.res* file with six parameters and can be reinput for the next refinement. Although six parameters are required, the program automatically applies the constraints appropriate for the crystal system, so usually less than six parameters are actually refined.

ANSR anres[0.001]

anres is the esd of a restraint that is applied with target values of zero to the six ANSC parameters to prevent instabilities in full-matrix refinement, especially when all atoms are also refined anisotropically, which would introduce correlations of 100%. This instruction will rarely be needed because the default restraint value is usually adequate.

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 volume contributions of the various twin components.The program now allows BASF parameters to become negative, though of course they should always be positive.

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).

BIND m n

Atoms in PART m may bond to atoms in PART n. This extends the PART rules and enables 'PARTs within PARTs' to be defined. These PART numbers may be positive or negative.

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 applies to all atoms 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 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 that does not refer to any atoms refines all atomic parameters (xyz, sof and U or Uij) in the specified cycles. Such a BLOC instruction takes priority over all other BLOC instructions, irrespective of their order in the *.ins* file. 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 referenced on the same BOND instruction. Angles are output for all pairs of such bonds involving a common atom. 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!

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.

CELL λ a b c α β γ

Wavelength and unit-cell dimensions in Å and degrees.

CGLS nls[0] nrf[0] nextra[0]

As L.S., but the 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 standard uncertainties, so that when a large structure has been refined to convergence using CGLS it may be worth 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 suitable for R_{free} tests (negative nrf).

The CGLS algorithm is based closely on the procedure described by Hendrickson & Konnert [*Computing in Crystallography* (1980) 13.01-13.25]. 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 that is characteristic of other conjugate gradient algorithms [Tronrud, *Acta Cryst.* **A48** (1992) 912-916].

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.

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 of these three atoms are chemically equivalent 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 printed out after each refinement cycle.

CONF atomnames max_d[1.9] max_a[170]

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 for which the central bond is shorter than max_d and both the bond angles are less than max_a. 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 follow the conventions defined by Allen & Rogers, *Acta Cryst.* **B25** (1969) 1326-1330.

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. 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 .....
```

this would prevent bonds being generated from the iron atom to all 10 carbons in ferrocene. This also illustrates the calculation of the distances of the iron atom from the two ring planes.

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 [*J. Soc. Ind. Appl. Math.* **11** (1963) 431-441] 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.

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.

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).

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. 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. For many purposes DELU has been superceded by RIGU.

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).

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 Ga, Cu, Mo and AgKα 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.

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 first atom in the atom list that is linked to other atoms by EADP. 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. '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 U_{ij} components of neighboring atoms to be approximately equal with an appropriate (usually fairly large) esd.

END

END is used to terminate an 'include' file, and may also be included after HKLF in the .*ins* file.

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 before it 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.

EXTI x[0]

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

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

where k is the overall scale factor. The wavelength dependence is needed for HKLF 2 (Laue) data. The program will print a warning if extinction (or SWAT) 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 [*Crystallographic Computing* (1970) 291-294] comes closest and should be consulted for further information.

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 possible free variable references are taken from the

EXYZ atom that comes first in the atom list.

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

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.

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 is 1, 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. Reflections with σ(F) relatively large compared with $|F_c|$ are weighted down, this helps to reduce noise. The r.m.s. 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 listed, sorted on the absolute value of the peak height. This is useful for neutron diffraction data.

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

code=3: Electron density synthesis with coefficients F_o and phases φ(calc).

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

code=5: Sim-weighted $(2mF_o-DF_c)$ Fourier.

code=6: Sim-weighted $(2mF_o-DF_c)$ Fourier, coefficients sharpened by multplying with √(E/F).

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. Atoms with zero coordinates are not fitted, but new coordinates are generated for these atoms. The atom names, sfac numbers, sof and U_{ij} of the FRAG fragment are ignored, only the coordinates are used. A FRAG fragment may be given anywhere between UNIT and HKLF or END or in an 'include' file, 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 6).

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).

FVAR osf[1] free variables

The overall scale factor is followed by the values of the 'free variables' fv_2 ... For historical reasons, 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. 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 $\sigma(F_o^2)$ are brought onto an absolute scale by dividing by the squared 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).

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. sl and dl need not be integral. A 103 x 103 x nl grid is only available when it is set automatically by the program. It is almost always better to let the program define this grid.

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 come before 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 can be useful if they involve '_* ' or a global reference to a residue class).

HKLF N[0] S[1] r11...r33[1 0 0 0 1 0 0 0 1] sm[1] m[0]

N defines the format of the data in the *.hkl* file, the scale factor S multiplies both F_o^2 and $\sigma(F_o^2)$ (or F_o and $\sigma(F_o)$ for N = 1 or 3) and the indices are 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 matrix must have a positive determinant. It is essential that the cell, symmetry and atom coordinates in the *.ins* file correspond to the indices after transformation using this matrix. N is negative if reflection data follow in the same *.ins* file (deprecated), 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 conventions. They are terminated by a blank line or 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 is the batch (or twin component) number; if it is zero or absent, it is reset to one. sm multiplies all σ-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 historical reasons; 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, extremely compact but lossy and deprecated.

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

The remaining options (N>2) all require Fortran FORMAT(3I4,2F8.2,I4); other compatible formats (e.g. F8.0 or even I8) may be used.

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, it may be positive or slightly negative. BN may be made negative (e.g. by the Bruker program XPREP) to flag a reflection for inclusion in the R_{free} reference set (see CGLS 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 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 is used to input an indexed powder diffraction pattern. 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. In contrast to N=5, 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 group (m-3m) the reflections 3 0 0 (which has multiplicity 3) and 2 2 1 (multiplicity 12) would contribute to the same measured F_o^2. HKLF 6 sets MERG 0 and may not be used with BASF or TWIN.

There may be only one HKLF instruction and it must come last, except when HKLF -N is followed by reflection data, in which case the file is terminated by the end of the data.

HTAB dh[2.0]

The HTAB instruction provides an analysis of the hydrogen bonds. A search is made for all hydrogen bonds 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 and writes the appropriate HTAB (with atom names) and EQIV instructions to the end of the *.res* file. Non-classical C–H•••O hydrogen bonds are generated and should be checked carefully since they may be controversial!

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 CIF requires this.

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 useful for water molecules for which RIGU, 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 (NPD). 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 NPD. 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 off the 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 or RIGU; whereas there is good evidence that DELU and RIGU restraints should hold accurately for most covalently bonded systems, ISOR (and SIMU) are only rough approximations to reality.

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.

LAUE E

Wavelength-dependent values of f', f" and optionally mu 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. There is no provision for handling overlapping reflection orders, scaling for the source intensity distribution, Lp and absorption corrections. A dummy wavelength (e.g. 1.0) should be given on the CELL instruction and the absorption coefficient output by the program should be ignored.

The element symbol may be preceded by '$' but this is optional. 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 Fortran FORMAT(F7.3,2F8.3); further information 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 blank line. 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.

LIST m[#] mult[1]

Write reflection lists to the *.fcf* file. Only one LIST instruction is allowed.

m=0: No action.

m=1: List h,k,l, F_o, F_c and phase (in degrees) 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 Fortran 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 Fortran 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. Fortran 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 $t.s(F_o^2)$, 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 f (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 h,k,l, F_o^2, $\sigma(F_o^2)$, F_c and φ(phase angle in degrees) for the reflection list as defined for m=1. This is the format required for input to shelXle or Coot.

m=7: List h,k,l, F_o^2, $\sigma(F_o^2)$ followed on the same line by F_c^2 for each twin component (-1 if a twin component makes no contribution to the reflection).

m=8: List h,k,l, F_o^2, $\sigma(F_o^2)$, F_c^2, φ(phase angle in degrees), d-spacing in Å and $1/\sqrt{w}$ where w is the weight derived from the weighting scheme (WGHT) and used in the refinement. For weak reflections $1/\sqrt{w}$ should be only a little larger than $\sigma(F_o^2)$. This list is on an absolute scale and is detwinned, merged (according to the point group of the crystal) and sorted, but without eliminating the anomalous contributions (except in the calculation of φ, so the corresponding electron density is real.

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 output 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 m=3,4,6 or 8. m=4 was the standard archive format for small-molecule structures, and m=6 for macro-molecules. However now that all the IUCr journals and many others accept the new CIF format that contains embedded *.res* and *.hkl* files, the *.cif* file should be preferred for archiving the structure. This has the advantage that the refinement can be repeated exactly (after extracting the files with SHREDCIF) and that it can be used with all HKLF data formats. And there are no problems with reflections flagged for use in the free R-factor.

L.S. nls[0] nrf[0] nextra[0]

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, i.e. structure factors are still calculated.

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. However the recommended value is -1, to use the R_{free} set defined in the *.hkl* file, which is independent of the space group and the MERG, OMIT and SHEL settings. R_{free} [Brunger, *Nature* **355** (1992) 472-475] provides a check as to whether the structure is being over-refined. To avoid bias, the R_{free} reflections are not used for Fourier calculations in SHELXL and are also not included in the LIST 6 output that can be used to make a map with the program Coot.

nextra is the number of additional parameters that were derived from the data when 'squeezing' the structure etc. 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 should be left at the default of zero except when 'squeeze' has been used.

MERG n[2]

If n is equal to 2 the reflections are sorted and merged before refinement, but if the structure is non-centrosymmetric the Friedel opposites are not combined before refinement. 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 for non-centrosymmetric structures and should only be used for good reason. Note that the reflections are always merged, and Friedel opposites combined, before performing Fourier calculations so that the (difference) electron density is real and correctly scaled. Even with n=0 the program may change the reflection order within each data block to optimize the vectorization of the structure factor calculations. Note that MERG may not be used in conjunction with HKLF 5 or 6.

MERG 2 is now the standard for small molecules and is required for the production of CIF files. MERG 4 averages all equivalents including Friedel opposites, and in contrast to MERG 3 also sets all f" values to zero; it is often used in refinement of macromolecules.

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). If m is negative, lists of parameters and the full covariance matrix are written to the *.mat* file.

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!

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.

NCSY DN sd[0.1] su[0.05] atoms

The NCSY instruction applies local non-crystallographic symmetry restraints. In contrast to global NCS constraints, these do not save CPU time but do not require the definition (and refinement) of a matrix transformation and mask. They are very flexible, and can accommodate rotation of domains relative to each other 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, 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 a corresponding 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 rotamers 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. 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
```

These atom lists 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.

NEUT

The NEUT instruction takes no parameters and is designed to facilitate refinement against neutron diffraction data. If present, it should normally appear just before the first SFAC instructions. It has three effects:

1. The special treatment of H and D atoms is switched off except for the generation and refinement of hydrogen atoms with HFIX and AFIX. AFIX 87 and 137 take the negative scattering length of H into account when interpreting the circular difference density map.

2. If NEUT comes *before* a SFAC instruction that contains atom names but not numbers, neutron scattering lengths and absorption coefficients are used for those elements. Except for D that is assumed to be a pure isotope, the scattering lengths are the weighted mean values for natural isotopic abundances. DISP can be added as required, e.g. for cadmium. The full form of the SFAC instruction may still be used to input mean scattering lengths for non-natural isotopic mixtures etc., or for atomic numbers greater than 94.

3. If the isotropic U of an atom (usually H or D) is given a value -k where -0.5 > -k > -5.0, it is set to k times the U-value of the last normal atom, but in contrast to the similar action when NEUT is not set, both atoms contribute to the calculation of the derivatives used in the least-squares calculations, so both atoms must be isotropic. This significantly reduces the number of degrees of freedom of the refinement and so would be expected to reduce the gap between $R1_{free}$ and R1 for a macromolecule, especially when the number of neutron data is limited. A relatively large k value (say 2.5) appears to be appropriate for D_2O and H_2O molecules when refining against neutron data.

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.

OMIT s[-2] 2θ(lim)[180]

If s is given as zero or 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 $F_o < s\sigma(F_o)$. 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. The SHEL command may also be used to ignore reflections above or below particular limiting resolution values.

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 that would generate the final indices h,k,l are ignored. It is best to use the indices exactly as given in the table of the 'most disagreeable reflections' in the *.lst* file; it may be necessary to omit both members of a Friedel pair.

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 default value until a PART instruction is read). 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, so it is recommended that the hydrogen atoms should be introduced with HFIX after the disorder has been fully accounted for. 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 OG
 :
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
HO   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 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 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.

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 (this option is clearly only of historical interest).. 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 *.lst* file. 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 $Å^{-3}$.

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

PRIG p[#]

This sets the parameter p used in the weighting scheme for the RIGU restraint [Thorn, Dittrich & Sheldrick, *Acta Cryst.* A**68** (2012) 448-451]. It will very rarely be necessary to change p from its default value.

REM

Followed by a comment on the same line. This comment is copied to the *.res* file. 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 in any instruction following the character '!'.

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. Residues of the same class may have different residue numbers but each number must always have the same class.

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 duplicate the instruction for all residues; 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 protein 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 ω 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 at least two residue numbers between them, so a cyclic pentapeptide with two molecules in the asymmetric unit could 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 chain (e.g. chain A could be 1001 to 1234 and chain B 2001 to 2234).

RIGU s1[0.004] s2[0.004] atomnames

Apply enhanced rigid bond restraints with esds s1 for 1,2-distances and s2 for 1,3 [Thorn, Dittrich & Sheldrick, *Acta Cryst.* A**68** (2012) 448-451]. This may be considered a hard restraint and so these low esds are appropriate and will rarely need changing. Often it will replace DELU, since it generates 3 or 6 restraints rather than 1 or 2, but in some cases it might be worth combining it with a DELU restraint that has a smaller esd. A zero value for s1 or s2 switches off the corresponding restraint. If no atoms are specified, all non-hydrogen atoms are assumed. RIGU 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. RIGU without atom names applies to all non-hydrogen atoms. SFAC element names may also be referenced, preceded by the symbol '$'.

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 (except for RTAB D2CG). 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, and the same sign 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. 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 residue in a protein).

RTAB D2CG atom1 atom2 .. atomN

This special form of the RTAB instruction calculates the distance (with esd) of atom1 to the unweighted centroid of atom2 .. atomN. This is useful for finding distances to mid-points of cyclopentadiene rings for example.

SADI s[0.02] pairs of atoms

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 standard uncertainty for this parameter). The default effective standard deviations for SADI may be changed by means of a DEFS instruction before the instruction in question.

A SADI instruction without any atom names causes SADI instructions that are equivalent to all the SAME instructions to be written to end of the *.res* file. This is useful if it is necessary to model a disorder involving atoms related by SAME.

SAME s1[0.02] s2[0.04] 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. 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 or larger 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. 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. 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 Tol 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 toluene molecule to be coplanar by a FLAT restraint:

```
FLAT_Tol C1 > C7
```

SAME_XYZ, where XYZ is a residue name, now operates differently to SAME_N (where N is a residue number) or SAME without a residue number, both of which must be immediately followed by atoms in the same order as on the SAME instruction (H atoms are ignored). SAME_XYZ no longer uses the following atoms but is applied to all residues with the name XYZ, so the atoms must have the same names in the same order in each of these residues so that the program knows which are equivalent. For example, if we have six THF solvent molecules which should be restrained in this way, the THF molecules are preceded by 'RESI 1 THF', 'RESI 2 THF' .. 'RESI 6 THF' and are each followed by RESI 0 (which can be omitted if another RESI instruction follows immediately). In this case the THF atoms should have the same names in the same order in each THF residue. Only two SAME instruction are required and may (now) appear anywhere in the .ins file:

```
SAME_THF O1 > C4
SAME_THF O1 C4 < C1
```

Hydrogen atoms can be added later with a single instruction:

```
HFIX_THF 23 C1 C2 C3 C4
```

Alternately 'SAME_1 O1 > C4' and 'SAME_1 O1 C4 < C1' could be inserted before the first atom of each THF residue after the first, that would be less elegant but would not require the same atom names to be used in each THF residue. Since hydrogen atoms are ignored by SAME, SADI not SAME should be used for H_2O. 'SAME_WAT OW HW1 HW2' will not have any effect even if each water molecule is in a separate residue with the name 'WAT' and atoms OW HW1 and HW2, but either:

```
SADI_WAT OW HW1 OW HW2
SADI_WAT HW1 HW2
```

or:

```
DFIX_WAT 0.84 OW HW1 OW HW2
DFIX_WAT 1.33 HW1 HW2
```

should work. These examples illustrate the way in which residues can be used to make the *.ins* file much simpler and easier to understand. Note however that because of a design

limitation in PLATON/CheckCIF, the number of characters in the atom name plus the number of in the residue number (plus two if ACTA TABS is set) should not be greater than 7.

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 (and D) 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 E 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. The element label E 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 either the NEUT instruction should come before an SFAC instruction that only contains atom names, or this format can 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' or 'D' on the SFAC instruction. This special treatment of H and D does not apply if NEUT is given before SFAC.

SHEL lowres[infinite] highres[0]

Reflections outside the specified resolution range in Å are ignored completely. This instruction is often used for macromolecules.

SIMU s[0.04] st[0.08] dmax[2.0] 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 RIGU and DELU; whereas there is good evidence that RIGU and 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 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 and RIGU 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 Uij'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, BF_4^- ions). For larger molecular fragments, the effective rotation angles are smaller, and the assumption of equal Uij for neighboring atoms is more appropriate: both translation and libration of a large fragment will result in relatively similar Uij components on adjacent atoms. SIMU may be combined with ISOR, which applies a further soft but quite different restraint on the Uij 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).

A SIMU restraint with dmax set to say 0.7 may be used to stabilize refinements in which there are overlapping disorder components with different PART numbers. This is complementary to RIGU which uses the connectivity table, and hence the PART rules to decide which atom pairs may be restrained.

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. These estimates take into account that most of the diffraction from strongly absorbing crystals takes place at the edges and corners.

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.

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.

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
EADP Na1 Ca1 Al1 K1                     ! common U or Uij
FVAR ... 0.20 0.30 0.35 0.15    ! starting values for free variables
Na1 ... ... ... ... 20.25 ...   ! 0.25*fv(2)
Ca1 ... ... ... ... 30.25 ...   ! 0.25*fv(3)
Al1 ... ... ... ... 40.25 ...   ! 0.25*fv(4)
K1  ... ... ... ... 50.25 ...   ! 0.25*fv(5)
```

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 EXTI and BASF parameters, including parameters used to describe twinning (TWIN). The parameters are counted in the order overall scale and free variables, EXTI, then BASF.

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 calculated intensity is modified as follows:

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

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.

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.

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 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.

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.

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.

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 of the original and twin-related components are 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 |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.

TWST N[0]

Twin component number to be used for the completeness and Friedel completeness statistics. Only single or composite reflections containing this twin component are used for these statistics. It applies to both TWIN+HKLF 4 and HKLF 5 data, but is most useful for the latter. The default N=0 causes all components to be used. If there is no twinning, this parameter has no effect.

UNIT n1 n2 ...

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

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

The weighting scheme is defined as follows:

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

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, *Acta Cryst.* **A32** (1976) 994-996]. The parameters should 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 existing WGHT instruction 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, *Acta Cryst.* **B29** (1973) 589-595. 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 used it will output a warning message.

WIGL del[0.2] dU[0.2]

If WIGL is used, all atoms not on special positions are displaced by an average distance del Å in a random direction. These shifts are applied after generating the connectivity table (which would otherwise be compromised) and before generating hydrogen atoms. Fixed coordinates (e.g. for special positions) are not changed. U_{iso} and U_{11}, U_{22} and U_{33}, but not U_{23}, U_{13} and U_{12}, are multiplied by a random factor so that they change by an average of 100dU%. WIGL is useful for removing R_{free} memory effects and checking convergence properties. Shifts greater than about 0.5 Å can result in some atoms moving out of density and not finding their way back home.

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 B_{eq} is still used on the ATOM statement). It is up to the user to ensure that the residue and atom names conform to PDB rules.

XNPD Umin[-0.001]

Sets a lower bound for the eigenvalues of the U_{ij} tensor of all anisotropic atoms or the U of an isotropic atom. The default (i.e. assumed if XNPD is not given) is 'XNPD -0.001'. This has the effect that non-positive-definite (NPD) atoms are still detected and reported, but they are prevented from causing the refinement to explode. XNPD without a number sets Umin to +0.001, in which case no atom should be reported as NPD. The numbers of 'may be split' and NPD atoms are output to the console at the end of the refinement.

ZERR Z esd(a) esd(b) esd(c) esd(a) esd(b) esd(g)

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.