Chapter 6.1.7.6

SHELXL-97

Least-squares Organisation

George W. Sheldrick
Dept. of Structural Chemistry
Tammannstraße 4
37077 Göttingen, Germany
email: gsheldr@shelx.uni-ac.gwdg.de

L.S. nls[0] nrf[0] nextra[0] maxvec[511]
nls cycles of full-matrix least-squares refinement are performed, followed by a structure factor calculation. When L.S. (or CGLS) is combined with BLOC, each cycle involves refinement of a block of parameters which may be set up differently in different cycles. If no L.S. or CGLS instruction is given, 'L.S. 0' is assumed.

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

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

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

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

maxvec refers to the maximum number of reflections processed simultaneously in the rate-determining calculations. Usually the program utilizes all available memory to process as many reflections as possible simultaneously, subject to a maximum of maxvec, which may not be larger than 511. For complicated reasons

involving the handling of suppressed and 'R_{free}' reflections and input/output buffering, some blocks may be smaller than the maximum, especially if the facilities for refinement against twinned or powder data are being used. It may be desirable to set maxvec to a smaller number than 511 to prevent unnecessary disk transfers when large structures are refined on virtual memory systems with limited physical memory.

CGLS nls[0] nrf[0] nextra[0] maxvec[511]
As L.S., but the Konnert-Hendrickson conjugate-gradient algorithm is employed instead of the full-matrix approach. Although BLOC may be used with CGLS, in practice it is much better to refine all parameters at once. CGLS is much faster than L.S. for a large number of parameters, and so will be the method of choice for most macromolecular refinements. The convergence properties of CGLS are good in the early stages (especially if there are many restraints), but cannot compete with L.S. in the final stages for structures which are small enough for full-matrix refinement. The major disadvantage of CGLS is that it does not provide estimated standard deviations, so that when a large structure has been refined to convergence using CGLS it may beworth performing a blocked full-matrix refinement (L.S./BLOC) to obtain the standard deviations in quantities of interest (e.g. torsion angles, in which case only xyz blocks would be required).The other parameters have the same meaning as with L.S.; CGLS is entirely suitable for R_{free} tests (negative nrf), and since it requires much less memory than L.S. there will rarely be any reason to change maxvec from its default value.

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

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

In addition to this optimization of the CGLS shift multiplication factor, the individual parameter shifts are monitored each L.S. or CGLS cycle, and the shift multiplication factors are reduced (to a value between 0.5 and 1) for parameters that tend to oscillate. This applies only to refinements in which BLOC is not used.

This produces an additional improvement in the convergence of the least-squares refinement, but (unlike Marquardt damping) has no effect on esds.

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

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

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

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

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

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

DAMP damp[0.7] limse[15]
The DAMP parameters take different meanings for L.S. and CGLS refinements. For L.S., damp is usually left at the default value unless there is severe correlation, e.g. when trying to refine a pseudo-centrosymmetric structure, or refining with few data per parameter (e.g. from powder data). A value in the range 1-10000 might then be appropriate. The diagonal elements of the least-squares matrix are multiplied by (1+damp/1000) before inversion; this is a version of the Marquardt (1963) algorithm. A side-effect of damping is that the standard deviations of poorly determined parameters will be artificially reduced; it is recommended that a final least-squares cycle be performed with little or no damping in order to improve these estimated standard deviations. Theoretically, damping only serves to improve the convergence properties of the refinement, and can be gradually reduced as the refinement converges; it should not influence the final parameter values. However in practice damping also deals effectively with rounding error problems in the (single-precision) least-squares matrix algebra, which can present problems when the number of parameters is large and/or restraints are used (especially when the latter have small esd's), and so it may not prove possible to lift the damping entirely even for a well converged refinement.

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

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

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

STIR sres step[0.01]
The STIR instruction allows a stepwise improvement in the resolution. In the first refinement cycle, the high-resolution limit (i.e. lowest d) is set at sres, in the next cycle to (sres–step), in the next (sres–2•step) etc. This continues until the limit of the data or the SHEL limit is reached, after which any remaining cycles to complete the number specified by CGLS or L.S. are completed with a constant resolution range. By starting at lower resolution and then gradually improving it, the radius of convergence for models with significant coordinate errors should be increased. This may be regarded as a primitive form of 'simulated annealing'; it could be useful in the early stages of refinement of molecular replacement

solutions, or for getting rid of bias for R_{free} tests (in cases where the solution of the struture was - possibly of ncessity - based on all the data).

WGHT a[0.1] b[0] c[0] d[0] e[0] f[.33333]
The weighting scheme is defined as follows:

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

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

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

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

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

FVAR osf[1] free variables
The overall scale factor is followed by the values of the 'free variables' fv(2) ... The overall scale factor is given throughout as the square root of the scale factor which multiplies F_c^2 in the least-squares refinement [to make it similar to the scale factor in SHELX-76 which multiplied F_c], i.e. $osf^2 F_c^2$ is fitted to F_o^2.

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

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

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