International Tables for Crystallography (2006). Vol. F, Section 25.2.1, pp. 695–705.

25.2. Programs and program systems in wide use

By W. Furey, K. D. Cowtan, K. Y. J. Zhang, P. Main, A. T. Brunger, P. D. Adams, W. L. DeLano, P. Gros, R. W. Grosse-Kunstleve, J.-S. Jiang, N. S. Pannu, R. J. Read, L. M. Rice, T. Simonson, D. E. Tronrud, L. F. Ten Eyck, V. S. Lamzin, A. Perrakis, K. S. Wilson, R. A. Laskowski, M. W. MacArthur, J. M. Thornton, P. J. Kraulis, D. C. Richardson, J. S. Richardson, W. Kabsch and G. M. Sheldrick

25.2.1. *PHASES* (W. Furey)

The program package *PHASES* came into being in the mid-to-late 1980s when it evolved largely from a series of independent computer programs written during the preceding decade for use in the Veterans Administration Medical Center's Biocrystallography Laboratory in Pittsburgh, PA, USA. The predecessor programs each carried out a particular task required in the processing, phasing and analysis of diffraction data from macromolecules, but these programs were usually computer, space-group and sometimes even protein specific. In addition, the programs were often poorly documented, if at all, and made use of incompatible data formats such that a battery of 'conversion' programs were required for transmitting information. While this was pretty much the situation in most laboratories at the time, it nevertheless unnecessarily complicated protein structure determination, particularly by graduate students and new postdoctoral workers. To overcome these problems, the original programs were rewritten (frequently combining several programs into one), generalized for all symmetries, modified to use a simple, standardized format and extensively documented. As methodologies developed, new programs and procedures were added, graphics programs were included, and the resulting package was optimized for use on interactive graphics workstations, which were then becoming the main computing resource in most laboratories. The first 'official' *PHASES* release was described at an American Crystallographic Association meeting (Furey & Swaminathan, 1990), although versions of the package had been in local use within the Pittsburgh laboratories for the preceding four years. There have been several major releases since the first as new features and strategies were incorporated, and the package as it existed in 1996 was extensively described in a *Methods in Enzymology* article (Furey & Swaminathan, 1997).

25.2.1.1. *Overall scope of the package*

The *PHASES* package was designed to deal with the major problem in macromolecular structure determination, *i.e.*, phasing the diffraction data. The package is not completely comprehensive, as it excludes programs for initial data reduction (production of a unique set of structure-factor amplitudes from the raw measurements), molecular-replacement calculations (rotation and translation functions), model building (interactive graphic fitting) and complete structure refinement (restrained least squares, conjugate gradient, molecular dynamics *etc.*). There are already excellent programs and packages available to accomplish these tasks; instead, the *PHASES* package focuses on the initial phasing of diffraction data from macromolecules by heavy-atom- and anomalous-scattering-based methods. Also included are programs and procedures for phase improvement by noncrystallographic symmetry averaging, solvent flattening, phase extension and partial structure phase combination. The programs and additional procedure scripts allow one to start with unique structure-factor amplitudes for native and/or derivative data sets and generate electron-density maps and skeletons that can be utilized in popular graphics programs for chain tracing and model building. The major methods incorporated in the package are listed below and will be described in more detail later.

25.2.1.1.1. *Isomorphous replacement, anomalous scattering and MAD phasing*

Heavy-atom-based phasing by the methods of isomorphous replacement (Green *et al.*, 1954) and/or anomalous scattering (Pepinsky & Okaya, 1956) are initiated by reading one or more 'scaled' files into the program *PHASIT*, along with estimates of the heavy-atom or anomalous-scatterer positional, occupancy and thermal parameters. Each input file can contain either isomorphous-replacement, derivative anomalous-scattering or native anomalous-scattering data. MAD (multiple-wavelength anomalous diffraction) data are treated as both isomorphous and anomalous-scattering data, in which case one simply inputs the scattering-factor differences (real part) appropriate for the wavelengths comprising the 'isomorphous' data sets and the actual scattering factors (imaginary part) appropriate for the 'anomalous' data sets. All possible combinations of isomorphous-replacement data, conventional anomalous-scattering data and MAD data are allowed and can be used simultaneously during phasing.

25.2.1.1.2. *Solvent flattening and negative-density truncation*

Solvent flattening and negative-density truncation are carried out following the strategy developed by Wang (1985); however, a reciprocal-space equivalent of the automated solvent-masking procedure is used (Furey & Swaminathan, 1997; Leslie, 1987). In addition, during solvent-mask construction all density near heavy-atom sites is automatically ignored, leading to more accurate masks. The complete process is fully automated and carries out three solvent-mask iterations with at least 16 solvent flattening and phase combination cycles. Optionally, an arbitrary number of additional cycles can be carried out for phase extension. A program is provided to interactively examine or edit the solvent mask or to create the mask by hand if desired.

25.2.1.1.3. *Noncrystallographic symmetry averaging*

Noncrystallographic (NC) symmetry averaging cases (Rossmann & Blow, 1963; Bricogne, 1974) are treated in direct space by operating on 'submaps', *i.e.* arbitrary regions in an electron-density map encompassing all of the molecules to be averaged that are unique by true crystal symmetry. Many of the averaging programs were derived from routines originally written by W. Hendrickson & J. Smith and have been described earlier (Bolin *et al.*, 1993), but they were substantially rewritten for incorporation into the *PHASES* package. Programs are supplied to: generate and examine the required submaps; refine the NC symmetry operators; interactively create averaging envelope masks; average density within the envelopes; convert the submaps to full-cell maps; invert the modified maps; and combine the phases with those from another source. An automated procedure is provided to carry out a specified number of averaging and phase combination cycles in addition to solvent flattening and negative-density truncation. This procedure allows for a gradual phase extension, if desired, extending by one reciprocal-lattice point in each direction for a given number of cycles.

25.2.1.1.4. *Partial structure phase combination and phase extension*

Several programs are included to carry out partial structure phase combination with a variety of weighting options as an aid to structure completion. If density modification (solvent flattening or negative-density truncation and/or NC symmetry averaging) is performed, then phase (and amplitude) extension is also possible by manual or automated procedures.

25.2.1.2. *Design principles*

The *PHASES* package was designed to be user-friendly with many of the programs being interactive, so that the user is prompted for all information needed. Other programs that are often run repeatedly as part of an iterative procedure are designed to execute as batch processes and are generally run from within command procedures or shell scripts. With the exception of atomic coordinate records, all user-supplied data can be input in free format. Space-group-symmetry information is given by explicitly providing a set of equivalent positions, which has the advantage of allowing non-standard space-group settings. The individual programs in the package can be run 'stand alone', but are often chained together through command procedures or shell scripts. Template scripts are provided for common iterative procedures, but the package design also allows program and option sequences to be combined in many ways, facilitating methodology development by advanced users.

25.2.1.2.1. *General program structure and data flow*

The current package includes 44 Fortran programs and one C subroutine, with the C subroutine used only to provide an interface between the Fortran programs and standard X-Window graphics-library routines. All programs communicate only through files with a simple common format. For the major programs, memory is allocated from a single large one-dimensional array which gets partitioned as required for each problem at run time. This greatly simplifies redimensioning if needed for very large problems, since at most only two lines of code need to be changed. All source code is provided, along with compilation procedures or shell scripts appropriate for most workstations, including Silicon Graphics, Sun, IBM R6000, ESV and DEC Alpha AXP (both OSF and OpenVMS). A flow chart illustrating the major programs and data flow for common phasing procedures is given in Fig. 25.2.1.1.

25.2.1.2.2. *Parameter and cumulative log files*

Vital data common to nearly all calculations, such as the cell dimensions, lattice type and space-group symmetry, are entered only once in a single 'parameter file'. All interactive programs prompt for the name of this file and for batch programs it is to be supplied on the first input line. The parameter file can also optionally contain the name of a 'running' log file. If used, the running log file is opened in 'append' mode by each program in the package, and a copy of all screen or printed output is added to the file along with a time and date stamp indicating what program was run and when. This allows the user to maintain a complete history of all calculations and results on a given problem in a single, chronologically accurate file.

25.2.1.3. *Merging and scaling native and derivative data*

The programs *CMBISO* and *CMBANO* (both interactive) are used to combine unique native and derivative data sets into a single file and place the derivative data set on the scale of the native. All common reflections are identified, paired together, scaled and output to a single 'scaled' file. With *CMBISO*, only mean structure-factor amplitudes are used for both native and derivative data, *i.e.* Bijvoet mates are deemed equivalent and averaged. *CMBANO* functions similarly, except that for the derivative data the individual Bijvoet mates are not averaged, and both values are output to the scaled file. The overall merging R factor is reported both on F and F^2, along with tables indicating the R factor as a function of resolution, F magnitude and $|F/\sigma(F)|$. A table is also output indicating the mean value of $F_{PH} - F_P$ as a function of resolution, where F_{PH} and F_P are the derivative and native structure-factor amplitudes, respectively. By default, scaling is initially carried out by the relative Wilson method (Wilson, 1949), with other optional procedures as outlined below to follow if desired.

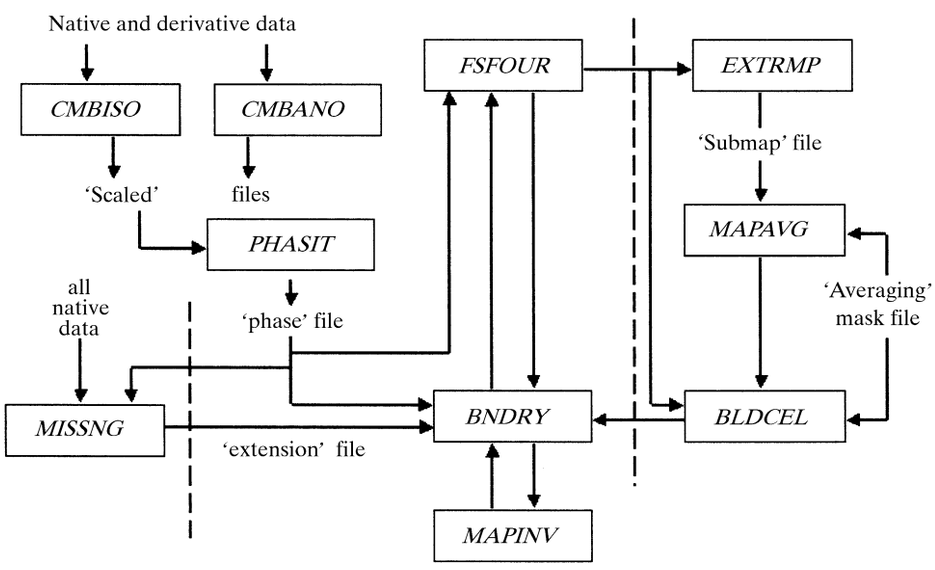

Fig. 25.2.1.1. Flow chart for the major phasing path encompassing native and derivative scaling, heavy-atom-based phasing, solvent flattening, negative-density truncation, and phase combination. Boxed entries represent programs while lines represent files. Optional paths for noncrystallographic symmetry averaging and phase extension are included by considering the additional programs offset from the main path by dashed lines.

25.2.1.3.1. *Relative Wilson scaling*

With this method, the derivative scattering, on average, is made equal to the native scattering by plotting

$$-\ln\left(\frac{\langle F_{PH}^2\rangle}{\langle F_P^2\rangle}\right) \textit{ versus } \left\langle\frac{\sin^2(\theta)}{\lambda^2}\right\rangle, \quad (25.2.1.1)$$

with the averages taken in corresponding resolution shells. A least-squares fit of a straight line to the plot yields a slope equal to $2(B_{PH} - B_P)$ (twice the difference between overall isotropic temperature parameters for derivative and native data sets) and an intercept of $\ln K^2$. From these values, the derivative data are put on the scale of the native by multiplying each derivative amplitude by

$$K \exp\left[(B_{PH} - B_P)\frac{\sin^2(\theta)}{\lambda^2}\right]. \quad (25.2.1.2)$$

25.2.1.3.2. *Global anisotropic scaling*

With this option, applied after relative Wilson scaling, the unique parameters of a symmetric 3×3 scaling tensor S are determined by two cycles of least-squares minimization of

$$\sum_{hkl} W_{hkl}(F_P - SF_{PH})^2 \qquad (25.2.1.3)$$

with respect to S, where W_{hkl} is a weighting factor,

$$S = S_{11}O_x^2 + S_{22}O_y^2 + S_{33}O_z^2 + 2(S_{12}O_xO_y + S_{13}O_xO_z + S_{23}O_yO_z) \qquad (25.2.1.4)$$

and O_x, O_y, O_z are direction cosines of the reciprocal-lattice vector expressed in an orthogonal system. The derivative data are then placed on the scale of the native by multiplying each derivative amplitude by the appropriate S.

25.2.1.3.3. *Local scaling*

With this option, again applied after relative Wilson scaling, a scale factor for each reflection is also determined by minimizing equation (25.2.1.3) with respect to S, but here S is a scalar and the summation is taken only over neighbouring reflections within a sphere centred on the reflection being scaled. The sphere radius is initially set to include roughly 125 neighbours, and the scale factor is accepted if at least 80 are actually present. If insufficient neighbours are available, then the sphere size is increased incrementally and the process repeated until a preset maximum radius is encountered. If the maximum is reached, the process terminates with the message that the data set is too sparse for local scaling. Scaling is achieved by multiplying each derivative amplitude by the appropriate S.

25.2.1.3.4. *Outlier rejection*

Rejection of outliers is often desirable, as erroneously large isomorphous or anomalous differences can lead to streaks in difference-Patterson maps and complicate identification of heavy-atom or anomalous-scatterer sites. The interactive program *TOPDEL* facilitates identification and rejection of such outliers while selecting reflections for use in difference-Patterson calculations. An input 'scaled' file is read in, and user-supplied resolution and $F/\sigma(F)$ cutoffs are applied. The data are then sorted in descending order of magnitude of ΔF (either isomorphous or anomalous differences) and the largest differences are listed for examination. The user is then prompted to determine which, if any, of the large differences are to be rejected as outliers and to determine what percentage of the remaining largest differences are to be used in the Patterson-map synthesis. The appropriate Fourier-coefficient file is then created.

25.2.1.4. *Fourier-map calculations*

All Fourier maps, including native- and difference-Patterson maps, are computed by the program *FSFOUR*, which runs in batch mode and is a space-group-general variable-radix 3D fast Fourier transform program. Unique reflections are expanded to a hemisphere, and the calculation then proceeds in $P1$. The output map always spans one full unit cell.

25.2.1.4.1. *Submaps*

Selected regions of an electron-density map that are useful for NC symmetry applications can be extracted from the full-cell maps produced by *FSFOUR* with the programs *EXTRMAP* (batch) or *MAPVIEW* (interactive). The 'submap' regions can cover any arbitrary volume and cross multiple cell edges if desired.

25.2.1.4.2. *Orthogonal and skewed maps*

Programs *MAPORTH* and *SKEW* (both run in batch mode) are provided to modify submaps, as modification is sometimes useful or required with NC symmetry applications. *MAPORTH* simply converts the map to correspond to an orthogonal grid, which simplifies refinement of NC symmetry operators. *SKEW* also converts the map to an orthogonal grid, but changes the axis directions such that the new b axis can be arbitrarily oriented. This is useful in NC symmetry applications where one may want to examine maps looking directly down the NC symmetry rotation axis. Both programs compute density values at the new grid points by using a 64-point cubic spline interpolation and can also orthogonalize or skew masks to maintain correspondence with the modified submaps.

25.2.1.4.3. *Graphics maps and skeletonization*

Program *GMAP* (interactive) is used to extract any region from a *FSFOUR* map, possibly crossing multiple cell edges, and convert it to a form directly readable by the external interactive graphics programs *TOM* [SGI version of *FRODO* (Jones, 1978)], *O* (Jones *et al.*, 1991) or *CHAIN* (Sack, 1988). In addition to the output map file, one may also output a corresponding skeleton (Greer, 1974) file (for *TOM*) or skeleton data block (for *O*) to facilitate chain tracing.

25.2.1.4.4. *Peak search*

Program *PSRCH* (batch) lists the largest peaks in a Fourier map and is useful in identifying additional heavy-atom or anomalous-scatterer sites from a map phased by a tentative model. Either positive or negative peaks can be listed, with the latter sometimes useful in MAD phasing applications, depending on the assignment of 'native' and 'derivative' data sets. Only unique peaks are listed, and the peak positions are interpolated from the map.

25.2.1.5. *Structure-factor and phase calculations*

Several methods are used for structure-factor and phasing calculations depending on the nature of the model and how the results will be used. The methods available in the package are described below.

25.2.1.5.1. *By heavy-atom or anomalous-scattering methods*

Phasing by heavy-atom-based methods (isomorphous replacement and/or anomalous scattering) begins when one or more 'scaled' data sets are input to the program *PHASIT* (batch). User-specified rejection criteria are first applied to each data set, and structure factors corresponding to the heavy-atom or anomalous-scatterer substructure are computed from

$$F_{hkl} = Sc \sum_j O_j f_j \exp\{-B_j[\sin^2(\theta)/\lambda^2]\} \exp[2\pi i(hx_j + ky_j + lz_j)], \qquad (25.2.1.5)$$

where O_j is the occupancy, f_j is the (possibly complex) scattering factor, B_j is the isotropic temperature parameter and x_j, y_j, z_j are the fractional coordinates of the jth atom. Sc is a scale factor relating the calculated structure factor (absolute scale) to the scale of the observed data. The summation is taken over all heavy atoms or anomalous scatterers in the unit cell. Alternatively, anisotropic temperature parameters can be used for each atom if desired. A subset of reflections comprising all centric data (plus the largest 25% of the isomorphous or anomalous differences if there are insufficient centric data) is selected and used to estimate Sc by a least-squares fit to the observed differences. Initial estimates of the 'standard error' E (expected lack of closure) are determined from

this subset as a function of F magnitude, treating centric and acentric data separately. SIR (single isomorphous replacement) or SAS (single-wavelength anomalous scattering) phase probability distributions are given by

$$P(\varphi) = k \exp[-e(\varphi)^2/2E^2], \qquad (25.2.1.6)$$

where the lack of closure is defined by

$$e(\varphi) = F^2_{PH_{(\mathrm{obs})}} - F^2_{PH_{(\mathrm{calc})}}(\varphi) \qquad (25.2.1.7)$$

for isomorphous-replacement data and

$$e(\varphi) = [(F^+_{PH})^2 - (F^-_{PH})^2]_{\mathrm{obs}} - \{[F^+_{PH}(\varphi)]^2 - [F^-_{PH}(\varphi)]^2\}_{\mathrm{calc}} \qquad (25.2.1.8)$$

for anomalous-scattering data, with the + and − superscripts denoting members of a Bijvoet pair, and

$$F^2_{PH_{(\mathrm{calc})}}(\varphi) = F^2_P + F^2_H + 2F_P F_H \cos(\varphi - \varphi_H), \qquad (25.2.1.9)$$

with φ denoting the protein phase, and F_H and φ_H denoting the heavy-atom structure-factor amplitude and phase, respectively. The distributions, however, are cast in the A, B, C, D form (Hendrickson & Lattman, 1970). After all input data sets are processed in this manner, the individual phase probability distributions for common reflections are combined *via*

$$P(\varphi)_{\mathrm{comb}} = k \exp[\cos(\varphi) \sum_j A_j + \sin(\varphi) \sum_j B_j + \cos(2\varphi) \sum_j C_j + \sin(2\varphi) \sum_j D_j], \qquad (25.2.1.10)$$

with k as a normalization constant and the sums taken over all contributing data sets. The resulting combined distributions are then integrated to yield a centroid phase and figure of merit for each reflection. The standard error estimates, E, as a function of structure-factor magnitude are then updated for each data set, this time using all reflections and a probability-weighted average over all possible phase values for the contribution from each reflection (Terwilliger & Eisenberg, 1987). With these updated standard error estimates, the individual SIR and/or SAS phase probability distributions are recomputed for all reflections and combined again to yield an improved centroid phase and figure of merit for each reflection. The resulting phases, figures of merit and probability distribution information are then available for use in map calculations or for further parameter or phase refinement. This method is used to produce MIR (multiple isomorphous replacement), SIRAS (single isomorphous replacement with anomalous scattering) MIRAS (multiple isomorphous replacement with anomalous scattering) and MAD phases as well as other possible phase combinations.

25.2.1.5.2. *Directly from atomic coordinates*

Structure-factor amplitudes and phases for a macromolecular structure can be computed directly from atomic coordinates corresponding to a tentative model with the programs *PHASIT* and *GREF* (both run as batch processes). This allows one to obtain structure-factor information from an input model typically derived from a partial chain trace or from a molecular-replacement solution. Equation (25.2.1.5) is used, but this time the sum is taken over all known atoms in the cell, and the scale factor is refined by least squares against the native amplitudes rather than against the magnitudes of isomorphous or anomalous differences. The computed structure factors may be used directly for map calculations, including 'omit' maps, or for combination with other sources of phase information. One can output probability distribution information for the calculated phases, if desired, as well as coefficients for various Fourier syntheses, including those using sigma_A weighting (Read, 1986) for the generation of reduced-bias native or difference maps.

25.2.1.5.3. *By map inversion*

For the purpose of improving phases by density-modification methods, such as solvent flattening, negative-density truncation and/or NC symmetry averaging, one must compute structure factors by Fourier inversion of an electron-density map rather than from atomic coordinates. The program *MAPINV* (batch) is a companion program to *FSFOUR* and carries out this inverse Fourier transform. It accepts a full-cell map in *FSFOUR* format and inverts it to produce amplitudes and phases for a selected set of reflections when given the target range of Miller indices. A variable-radix 3D fast Fourier transform algorithm is used. Optionally, the program can modify the density prior to inversion by truncation below a cutoff and/or by squaring the density values. Other types of density modification are handled by different programs in the package and are carried out prior to running *MAPINV*. The indices, calculated amplitude and phase are written to a file for each target reflection.

25.2.1.6. *Parameter refinement*

Several methods are provided for refinement of heavy-atom or anomalous-scatterer parameters and scaling parameters, depending on the desired function to be minimized. In all cases, the structure factor F_H corresponding to the heavy atom or anomalous scatterer is given by equation (25.2.1.5). The options available are briefly described below.

25.2.1.6.1. *Against amplitude differences*

The simplest procedure is to refine against the magnitudes of isomorphous or anomalous structure-factor amplitude differences, which can be carried out with the program *GREF* (batch mode). In this case, one minimizes

$$\sum_j W_j \left(|F_{PH_j} - F_{P_j}| - F_{H_j}\right)^2 \qquad (25.2.1.11)$$

for isomorphous-replacement data or

$$\sum_j W_j \left(|F^+_{PH_j} - F^-_{PH_j}| - 2F_{H_j}\right)^2 \qquad (25.2.1.12)$$

for anomalous-scattering data with respect to the desired parameters contributing to F_H, where W_j is a weighting factor. For anomalous-scattering data, only the imaginary component of the scattering factors is used during the F_H structure-factor calculation. For isomorphous-replacement data, the summation is taken only over centric reflections, plus the strongest 25% of differences for acentric reflections if insufficient centric data are present. For anomalous-scattering data, the summation is taken only over the strongest 25% of Bijvoet differences. An advantage of these methods is that only data from the derivative being refined are used (plus the native with isomorphous data), hence there is no possibility of feedback between other derivatives which may not be truly independent. A disadvantage is that, apart for the centric reflections, the target value in the minimization is only an approximation to the true F_H. The accuracy of this approximation is improved by restricting the summations to the strongest differences.

25.2.1.6.2. *By minimizing lack of closure*

An alternative procedure available in the program *PHASIT* (batch) is to refine against the observed derivative amplitudes. In this case, one minimizes the 'lack of closure' (now based on amplitudes instead of intensities) with respect to the desired

parameters contributing to F_{PH}, including the derivative-to-native scaling parameters. In all cases, the calculated derivative amplitudes $F_{PH_{(\mathrm{calc})}}$ are obtained from equation (25.2.1.9). To use this procedure, one must have an estimate of the protein phase φ. Several variations of this method, all available in *PHASIT*, are described below and are generally referred to as 'phase refinement'.

25.2.1.6.2.1. *'Classical' phase refinement*

With this option, one minimizes

$$\sum_j W_j[F_{PH_{(\mathrm{obs})}} - F_{PH_{(\mathrm{calc})}}(\varphi)]^2 \qquad (25.2.1.13)$$

for isomorphous-replacement data or

$$\sum_j W_j\{(F^+_{PH} - F^-_{PH})_{\mathrm{obs}} - [F^+_{PH}(\varphi) - F^-_{PH}(\varphi)]_{\mathrm{calc}}\}^2 \qquad (25.2.1.14)$$

for anomalous-scattering data with respect to the desired parameters. Typically, the weights are taken as the reciprocal of the 'standard error' (expected lack of closure) or its square. The summations are taken over all reflections for which the protein phase is thought to be reasonably valid, usually implied by a figure of merit of 0.4 or higher. The protein phase estimate usually comes from the centroid of the appropriate combined phase probability distribution given by equation (25.2.1.10); however, one has the option of including all data sets when combining the distributions, or including all *except* that for the derivative being refined. Once new heavy-atom and scaling parameters are obtained, new individual SIR or SAS phase probability distributions are computed and combined to provide new protein phases, and these phases are used to update the standard error estimates as described earlier. Then the individual distributions are recomputed once more using the new standard error estimates, and these distributions are combined again to give new protein phase estimates. The process is then iterated using the new phases and new heavy-atom parameters to start another round of refinement. After several iterations, the heavy-atom parameters, standard error estimates and protein phase estimates converge to their final values.

25.2.1.6.2.2. *Approximate-likelihood method*

This variation, also available in *PHASIT*, is similar to the classical phase refinement described above, except that instead of using only a single value for the protein phase φ during the calculation of F_{PH}, all possible values are considered, with each contribution weighted by the corresponding protein phase probability (Otwinowski, 1991). One minimizes

$$\sum_j W_j \sum_i P_i[F_{PH_{(\mathrm{obs})}} - F_{PH_{(\mathrm{calc})}}(\varphi_i)]^2 \qquad (25.2.1.15)$$

with respect to the desired parameters for isomorphous-replacement data, where P_i is the protein phase probability and the inner summation is over all allowed protein phase values, stepped in intervals of 5° (or 180° for centric reflections). For anomalous-scattering data, a similar modification is made to equation (25.2.1.14). The weights may be as in the classical phase refinement case or unity. Since each contribution is weighted by its phase probability regardless, there is no need to use a high figure-of-merit cutoff, as was done earlier. In fact, very good results are usually obtained using unit weights for W_j (that is, only the probability weighting) and a figure-of-merit cutoff of around 0.2 for inclusion of reflections in the summations. This variation has been found to increase stability in the refinement and works considerably better than conventional phase refinement when the phase probability distributions are strongly multimodal. Parameter refinement and phasing iterations proceed as described earlier. The combination of probability weighting during refinement with probability weighting during standard error estimation enables the key features of maximum-likelihood refinement to be carried out, although only approximately.

25.2.1.6.2.3. *Using external phase information*

When using either the conventional phase refinement or approximate-likelihood methods, protein phase estimates are required. In the former case, only a single value is used, whereas in the latter, information about all possibilities is provided by way of the phase probability distribution. Normally, this information comes from a prior phasing calculation; thus, the estimates are typically SIR, SAS, MIR *etc.* phases. However, in *PHASIT*, an option allows one to read in the protein phase information from an external source. This enables parameter refinement (by either conventional or approximate-likelihood methods) using protein phase estimates that are improvements over the initial ones. For example, one could get the best phases by one of the previously described methods, but then improve them by density-modification procedures, such as solvent flattening or negative-density truncation and/or NC symmetry averaging. Using these improved phases in the calculation of F_{PH} when refining should then lead to more accurate heavy-atom and scaling parameters, which in turn will produce still better protein phases. These new protein phases can either be treated as final and used to produce an electron-density map for interpretation, or be used to initiate another round of phase improvement by density modification. There are several cases where this type of refinement has been beneficial, and it is particularly useful for the refinement of derivative-to-native scaling parameters.

25.2.1.6.3. *Rigid-group refinement*

Although *GREF* can be used to refine individual heavy-atom or anomalous-scatterer parameters against isomorphous or anomalous structure-factor difference magnitudes, it is actually a group refinement program. Thus, all entities to be refined are treated as rigid bodies such that only group orientations, positions, scaling and temperature parameters can be refined. The groups, however, can be defined arbitrarily. For individual heavy-atom sites, they are simply defined as single atom 'groups', and no orientation parameters are selected for refinement. This enables the program to serve two additional roles. In the case where the heavy-atom reagent is known to contain a rigid group, it can be properly treated. Also, if one chooses the target values to be native structure-factor amplitudes instead of difference magnitudes and inputs an entire protein molecule or domain, then conventional rigid-body or segmented rigid-body refinement can be carried out. The output consists of the refined parameters and a Fourier-coefficient file suitable for map or phase combination calculations.

25.2.1.7. *Origin and hand correlation, and completing the heavy-atom substructure*

Several programs are provided to enable the computation and analysis of various types of difference-Fourier maps as an aid to completing the heavy-atom structure by picking up additional sites. They are also used to correlate the origin and hand between derivatives and to determine the absolute configuration. During phasing calculations in *PHASIT*, files suitable for isomorphous or Bijvoet difference-Fourier calculations are automatically produced for each derivative or data set and can be used directly in program *FSFOUR*. The procedures used are described below.

25.2.1.7.1. *Difference and cross-difference Fourier syntheses*

The files produced by *PHASIT* for isomorphous data sets contain the information needed to produce the Fourier coefficients

$$[F_{H_{(\mathrm{obs})}} - F_{H_{(\mathrm{calc})}}]\exp[i\varphi_{H_{(\mathrm{calc})}}], \qquad (25.2.1.16)$$

where $F_{H_{(\mathrm{calc})}}$ and $\varphi_{H_{(\mathrm{calc})}}$ are the calculated heavy-atom structure-factor amplitude and phase, respectively, and $F_{H_{(\mathrm{obs})}}$ is computed from

$$F^2_{H_{(\mathrm{obs})}} = F^2_{PH} + F^2_P - 2F_{PH}F_P\cos(\varphi_{PH} - \varphi_P) \qquad (25.2.1.17)$$

where φ_{PH} and φ_P are the current derivative and native phases, respectively. These coefficients are more accurate than using simple isomorphous difference magnitudes to approximate $F_{H_{(\mathrm{obs})}}$ and can be computed once phasing has begun, since estimates of the required phase differences are then available. Alternatively, the program *MRGDF* (interactive) can be used to produce Fourier coefficients of the form

$$m(F_{PH} - F_P)\exp(i\varphi_P), \qquad (25.2.1.18)$$

where m is the current figure of merit. This method suffers somewhat as phase differences are ignored, but it has the advantage that the amplitude difference does not necessarily involve any derivative previously used in the computation of φ_P. If amplitudes from a new derivative and from the native are used, then peaks in the resulting 'cross-difference' Fourier synthesis for the new derivative will automatically correspond to the same origin and hand as prior sites used in the phasing process, although the hand may still be incorrect. Finally, *GREF* can be used to generate the Fourier coefficients

$$(|F_{PH} - F_P|)\exp[i\varphi_{H_{(\mathrm{calc})}}] \quad \text{or} \quad [|F_{PH} - F_P| - F_{H_{(\mathrm{calc})}}]\exp[i\varphi_{H_{(\mathrm{calc})}}], \qquad (25.2.1.19)$$

with the second set producing a map similar to that obtained using equation (25.2.1.16). Both coefficient sets in equation (25.2.1.19) are lacking in that the phase difference is ignored, but the second set [and also those in equation (25.2.1.16)] has the advantage that heavy-atom sites already in the model are subtracted away, allowing any remaining minor sites to stand out in the resulting map.

25.2.1.7.2. *Bijvoet difference and cross-Bijvoet difference Fourier syntheses*

The files produced by *PHASIT* for anomalous-scattering data sets contain the information needed to produce the Fourier coefficients

$$(F^+_{PH} - F^-_{PH})_{\mathrm{obs}}\exp[i(\varphi^+_P - \pi/2)] \qquad (25.2.1.20)$$

or

$$[(F^+_{PH} - F^-_{PH})_{\mathrm{obs}} - (F^+_{PH} - F^-_{PH})_{\mathrm{calc}}]\exp[i(\varphi^+_P - \pi/2)], \qquad (25.2.1.21)$$

where φ^+_P is the protein phase used when computing F^+_{PH}. The coefficients in equation (25.2.1.20) correspond to a conventional Bijvoet difference Fourier map, which should show large positive peaks at the locations of anomalous-scattering sites when the hand is correct. The coefficients in equation (25.2.1.21) correspond to the case in which contributions from known anomalous scatterers are subtracted out. As in the isomorphous-replacement case, a program *MRGBDF* (interactive) is also provided to generate the Fourier coefficients

$$m(F^+_{PH} - F^-_{PH})_{\mathrm{obs}}\exp[i(\varphi^+_P - \pi/2)], \qquad (25.2.1.22)$$

where the Bijvoet difference does not necessarily have to come from a derivative used in the phasing. If the difference doesn't come from a derivative used in phasing, then a 'cross-Bijvoet difference' Fourier map is obtained, which should produce large positive peaks at anomalous-scatterer locations in the new derivative when the original hand is correct. Additionally, *GREF* can be used to generate the Fourier coefficients

$$(|F^+_{PH} - F^-_{PH}|_{\mathrm{obs}})\exp(i\varphi^+_H) \text{ or } [|F^+_{PH} - F^-_{PH}|_{\mathrm{obs}} - F^+_{H_{(\mathrm{calc})}}]\exp(i\varphi^+_H), \qquad (25.2.1.23)$$

where F^+_H and φ^+_H are the heavy-atom structure-factor amplitude and phase, used when computing F^+_{PH}. These coefficients can also be used to identify additional anomalous-scatterer sites, but they are insensitive to the hand. As in equation (25.2.1.21), if the second set in equation (25.2.1.23) is used, then contributions from anomalous scatterers already included in the phasing will be subtracted out.

Finally, the program *HNDCHK* (interactive) is provided to determine the enantiomorph by examination of a Bijvoet difference Fourier map. One inputs the map along with the anomalous-scatterer positions used in the phasing. The program then uses a 64-point cubic spline interpolation algorithm to obtain the density precisely at the input coordinates and also at coordinates related to them by a centre of symmetry. If the input heavy-atom configuration had the correct hand, large positive peaks should occur exactly at the input locations. If the hand is incorrect, even larger *negative* peaks occur at the true positions, *i.e.* those related to the input positions by a centre of symmetry.

25.2.1.8. *Solvent flattening and negative-density truncation*

Solvent flattening with negative-density truncation is efficiently carried out by the programs *BNDRY*, *FSFOUR*, *MAPINV* and *RMHEAVY*, all of which are run in batch mode with multiple iterations under the control of a command procedure or shell script. The various aspects of the process as implemented are described below.

25.2.1.8.1. *Mask construction*

Solvent-mask construction follows the procedure suggested by Wang (1985), with the exception that electron density in the vicinity of heavy-atom sites is temporarily ignored during the mask-building process. This allows one to use a tight solvent mask, which maximizes the phasing power of the method while preventing artificial extension of the protein envelope into the solvent region in the vicinity of surface-bound heavy-atom sites. Failure to do this has occasionally been found to deplete the protein region elsewhere to compensate for the incorrectly extended region.

25.2.1.8.1.1. *Automated mask construction*

An electron-density map produced by *FSFOUR* is passed to the program *RMHEAVY* along with a set of heavy-atom locations and a blanking radius. A copy of the map is then made that is identical to the original except that density values within the blanking radius of any heavy-atom site are set to zero. The modified map is then passed to program *MAPINV*, which sets to zero all density values that were negative (note that the F_{000} coefficient is *not* included in program *FSFOUR*) and then computes the corresponding set of structure factors by Fourier inversion. These structure factors are then passed to program *BNDRY* along with a resolution-dependent averaging radius R to compute the Fourier transform of the direct-space weighting function,

$$W(r) = 1 - r/R \text{ if } r \le R \quad \text{and} \quad W(r) = 0 \text{ if } r > R, \qquad (25.2.1.24)$$

where $W(r)$ is the weighting function and r is the distance from the map grid point being evaluated. R is typically 2.5–3 times the minimum d spacing in the data set. Each unique structure factor obtained from map inversion is then multiplied by the transform of $W(r)$, $f(s)$, given by

$$f(s) = 4\pi R^3\{2[1-\cos(A)] - A\sin(A)\}/A^4, \qquad (25.2.1.25)$$

where

$$A = 4\pi R\sin(\theta/\lambda). \qquad (25.2.1.26)$$

These weighted structure factors are then input to *FSFOUR* to compute a 'smeared' map, which corresponds to convolution of all non-negative density in the original map with the weighting function $W(r)$. The 'smeared' map is then passed to *BNDRY* along with an estimate of the solvent fractional volume. The fractional volume is converted to the corresponding number of grid points occupied by solvent, and a histogram is constructed identifying the number of grid points associated with each density value. Starting with the lowest observed density, a threshold value is increased incrementally, and a running sum is maintained identifying the current number of grid points with density values below the threshold. When the number of points accumulated reaches the expected number in the solvent region, the corresponding threshold indicates the density value for the protein–solvent boundary contour level. A mask map having a one-to-one correspondence with the map grid is then constructed such that if the density in the smeared map is less than the contour level, the grid point is deemed to be in the solvent region; otherwise, it is assigned to the protein region. The mask is then written to a file.

25.2.1.8.1.2. *Masks from atomic coordinates*

In some instances, it may be desirable to create masks, either for solvent flattening or NC symmetry averaging, from a set of atomic coordinates. The interactive program *MDLMSK* can be used for this as it accepts a set of atomic coordinates along with a masking radius, mask number and map region. It then creates a mask file spanning the requested map region such that all grid points within the region that are also within the masking radius of any model atom are assigned the specified mask value, and all other points a solvent mask value. If multiple masks are required, the interactive program *MRGMSK* can be used to combine separate mask files created by *MDLMSK* into a single mask file. For NC symmetry averaging purposes, one generally creates mask files separately for each independent molecule, using an average van der Waals radius as the masking radius, and then combines them with *MRGMSK*. This mask is then edited in the program *MAPVIEW* (see below) to maintain the outer boundary, but to fill in holes within the molecular interior. This mask can be used directly for NC symmetry averaging. If it is to be used for solvent flattening, then it must first be expanded to correspond to a full unit cell by the program *BLDCEL*.

25.2.1.8.1.3. *Mask verification and manual editing*

Both solvent masks and NC symmetry averaging masks can be examined and edited interactively using the program *MAPVIEW*. The program reads an electron-density map and (possibly) the corresponding mask. It then displays map sections contoured at any desired level. It can also be used to view the mask superimposed on the contoured map. One can scroll through all sections of the map one at a time, examining the corresponding mask assignment. If desired, one can manually edit the mask by tracing out the protein boundary using a cursor tied to a mouse, or even create the entire mask from scratch in this manner. Other features of *MAPVIEW* will be described later.

25.2.1.8.2. *The flattening and truncation procedure*

Once a solvent mask is constructed, solvent flattening and negative-density truncation is carried out using the program *BNDRY*. An electron-density map and corresponding mask are input along with an empirical constant S, which is used to estimate the value of F_{000}/V on the scale of the input map. The estimation follows the procedure of Wang (1985) and is based on the assumption that for typical solvent conditions and proteins not containing heavy metals, the ratio of mean solvent electron density to maximum protein electron density is constant, although for phasing purposes the optimum values are resolution-dependent. Typical values of S are supplied in the package. One simply couples the value of S taken from known structures with density values obtained from the experimental maps to estimate F_{000}/V on the appropriate (but unknown) map scale by solving the equation

$$\frac{\langle\rho\rangle_{\rm solvent} + F_{000}/V}{\rho_{\rm max,\ protein} + F_{000}/V} = S \qquad (25.2.1.27)$$

for F_{000}/V. Once the estimate of F_{000}/V is obtained, solvent flattening and negative-density truncation are carried out simultaneously by resetting all map values according to the relationships

$$\begin{aligned} \rho &= \langle\rho_{\rm solvent}\rangle + F_{000}/V && \text{if in the solvent region,} \\ \rho &= \max(\rho_{\rm input} + F_{000}/V, 0) && \text{if in the protein region.} \end{aligned} \qquad (25.2.1.28)$$

25.2.1.9. *Phase combination and extension procedures*

Phase combination, either during density-modification procedures or to make use of partial structure information, is carried out by the *BNDRY* program (batch). For standard phase combination, two structure-factor files are input. The first file, called the 'anchor' phase set, contains structure-factor information along with phase probability distributions in the form of A, B, C, D coefficients and usually corresponds to MIR, SIR, or MAD phases. The other file contains only 'calculated' structure-factor amplitudes and phases and is usually obtained either from Fourier inversion of a modified electron-density map or from a structure-factor calculation based on atomic coordinates from a partial structure. Common reflections in both files are identified, and the 'calculated' amplitudes are scaled to those in the anchor set by least squares. For phase combination, a variety of options are available, with the most important described below.

25.2.1.9.1. *Modified Sim weights*

The scaled data are sorted into bins according to d spacing, and a three-term polynomial is fitted to the mean values of $|F^2_{\rm obs} - F^2_{\rm calc}|$ as a function of resolution. For each reflection, a unimodal phase probability distribution is constructed using a modification (Bricogne, 1976) of the Sim (1959) weighting scheme *via*

$$P(\varphi_P) = k\exp\left[\frac{2F_{\rm obs}F_{\rm calc}\cos(\varphi_P - \varphi_{\rm calc})}{\langle|F^2_{\rm obs} - F^2_{\rm calc}|\rangle}\right], \qquad (25.2.1.29)$$

where the average in the appropriate resolution range is determined from the polynomial. This distribution is cast in the A, B, C, D form with

$$\begin{aligned} A &= W\cos(\varphi_{\rm calc}), \\ B &= W\sin(\varphi_{\rm calc}), \\ C &= 0 \\ D &= 0 \ \text{ and} \\ W &= \frac{2F_{\rm obs}F_{\rm calc}}{\langle|F^2_{\rm obs} - F^2_{\rm calc}|\rangle}. \end{aligned} \qquad (25.2.1.30)$$

Phase combination with the anchor set then proceeds according to equation (25.2.1.10), and the combined distributions are integrated to give a new phase and figure of merit for each reflection.

25.2.1.9.2. *σ_A weights*

As an alternative to the procedure above, in the *BNDRY* program the weights, W, used when constructing the unimodal probability distributions in equations (25.2.1.30) can be computed according to

$$W = \frac{2\sigma_A E_{\rm tot} E_{\rm par}}{1 - \sigma_A}, \qquad (25.2.1.31)$$

where $E_{\rm tot}$ and $E_{\rm par}$ are normalized structure-factor amplitudes for the observed and calculated structure factors, respectively, and σ_A is determined by the procedure described by Read (1986). For acentric reflections, equation (25.2.1.31) is used whereas for centric reflections, W is one half the value given by equation (25.2.1.31).

25.2.1.9.3. *Damping contributions*

Normally, the distributions constructed for the calculated phases are combined with those for the anchor set with full weight in equation (25.2.1.10). However, in *BNDRY*, one can supply a damping factor in the range 0–1 to down-weight the contributions of the anchor set. The damping factor simply multiplies the distribution coefficients such that a factor of 1 (default) indicates no damping, and values less than one place more emphasis on the map-inverted or partial structure phases. If set to zero, the calculated phases are accepted as they are, since there is effectively no phase combination with the anchor set.

25.2.1.9.4. *Phase extension*

If phase extension is requested during the phase combination step, an additional file (prepared by the interactive program *MISSNG*) is also supplied to the *BNDRY* program. This file contains unique reflections absent from the anchor set but for which observed amplitudes (and possibly phase probability distribution coefficients) are available. Phase combination then proceeds exactly as above, except that for any extended reflections lacking phase probability information, the calculated phases are accepted as they are. Phase extension is required when phasing purely by SAS methods as it is the only way to phase centric reflections. As a final option, phase and amplitude extension is possible, in which case both the calculated amplitude and phase are accepted as they are for reflections having only indices provided on the extension file. This is sometimes desirable to include low-resolution reflections that may have been obscured by the beam stop.

25.2.1.10. *Noncrystallographic symmetry calculations*

Several programs are provided to carry out noncrystallographic symmetry averaging within submaps and are briefly described below.

25.2.1.10.1. *Operator representation and definitions*

NC symmetry operators are specified in terms of the parameters φ, ψ, χ, O_x, O_y, O_z and t, which refer to a Cartesian coordinate system in Å, obtained by orthogonalization of the unit cell as in the Protein Data Bank (Bernstein *et al.*, 1977). The angles φ and ψ determine the direction of the NC rotation axis, while χ determines the amount of rotation about it. O_x, O_y and O_z are coordinates of a point through which the rotation axis passes, and t is a post-rotation translation parallel to the rotation axis. The relationships between the angles, orthogonal reference axes X, Y, Z and the unit cell are given in Fig. 25.2.1.2. Coordinates for a pair of points related by NC symmetry are then expressed in the orthogonal system by

$$P_2 = R_{\varphi,\psi,\chi}(P_1 - O) + O + tD_{\varphi,\psi}, \qquad (25.2.1.32)$$

where P_1 and P_2 are three-element column vectors containing coordinates for the related points, $R_{\varphi,\psi,\chi}$ is a 3×3 rotation matrix derived from the angles, O is a three-element column vector containing coordinates for a point through which the rotation axis passes, t is the post-rotation translation scalar in Å and $D_{\varphi,\psi}$ is a three-element column vector containing direction cosines of the rotation axis. This type of parameterization simplifies transfer of information from self-rotation functions, which are usually calculated in spherical polar angles anyway, and also makes obvious pseudo-space-group symmetry type operations such as pseudo-screw axes. For convenience, a program *O_TO_SP* is provided to convert from a 3×3 rotation matrix and 1×3 column vector representation of the NC symmetry operation, as used in some programs, to the parameters described here.

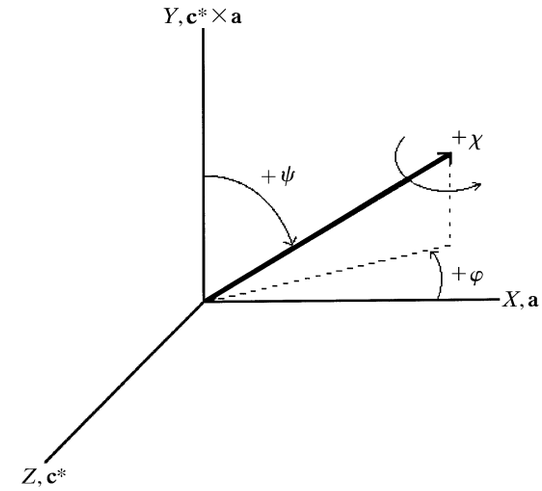

Fig. 25.2.1.2. Relationships between noncrystallographic symmetry rotation axis direction, orthogonal reference system axes X, Y, Z and crystallographic axes. The X axis is aligned with the crystal **a**. The Y axis is parallel to $\mathbf{a} \times \mathbf{c}^*$. The Z axis is parallel to $\mathbf{X} \times \mathbf{Y}$, *i.e.* $\mathbf{c}^*$. ψ is the angle between the NC rotation and $+Y$ axes. φ is the angle between the projection of the NC rotation axis in the XZ plane and the $+X$ axis, with $+\varphi$ counterclockwise when viewed from $+Y$ toward the origin. χ is the amount of rotation about the directed axis, with $+\chi$ clockwise when viewed from the axis toward the origin.

25.2.1.10.2. *Operator refinement*

Refinement of the NC symmetry operator parameters is achieved by least-squares minimization of the squared difference in electron density for all NC-symmetry-related points. Thus, one minimizes

$$\sum\{\rho(r) - \rho[R_{\varphi,\psi,\chi}(r - O) + O + tD_{\varphi,\psi}]\}^2 \qquad (25.2.1.33)$$

with respect to the operator parameters, where the sum is taken over all points within the appropriate averaging envelope(s). One starts refinement with low-resolution data (~6 Å) on a coarse (~2 Å) grid and monitors progress by following the correlation coefficient between the related electron-density values. Once convergence is obtained, the calculation is resumed with higher-resolution data on a finer grid. Typically, a correlation coefficient of around 0.4 or higher (for a 3 Å MIR map, 1 Å grid) indicates that the operator has been correctly located. The operator refinement is confined to submaps and is facilitated by use of an orthogonal grid. A submap containing the molecules to be averaged is obtained from the programs *MAPVIEW* or *EXTRMAP* and can be converted to an orthogonal grid, if needed, by the program *MAPORTH*, as described earlier.

25.2.1.10.2.1. *Simple rotational symmetry*

For 'proper' NC symmetry, only pure *n*-fold rotations are involved with *n* a small integer, *i.e.* twofold, threefold *etc.* In this case, only a single envelope mask encompassing all of the molecules to be averaged is needed for averaging and operator refinement, since one does not have to differentiate between molecules within the aggregate. Initial operator refinement can use a simple spherical mask of appropriate radius, with the sphere centred near the aggregate centre of mass and on the rotation axis. One can also use a mask created either by hand (described below) or created from atomic coordinates, as described earlier. For averaging purposes, however, a mask created by hand is usually desired. The NC symmetry operator refinement is carried out within the program *LSQROT* (batch).

25.2.1.10.2.2. *Complex rotational and/or translational symmetry*

For 'improper' NC symmetry, where there are translational components and/or arbitrary rotation angles involved, separate envelope masks must be assigned to each molecule in the aggregate for both NC symmetry operator refinement and averaging. Initial operator refinement can proceed with spherical masks of an appropriate radius centred on the centre of mass of each molecule in the aggregate. As in the 'proper' NC symmetry case, one can also use masks created by hand or generated from atomic coordinates for operator refinement, but hand-traced masks will be desired for the actual averaging. The NC symmetry operator refinement is carried out within the program *LSQROTGEN* (batch).

25.2.1.10.3. *Averaging mask construction*

Masks encompassing the region(s) to be averaged are usually created by hand in the interactive program *MAPVIEW*. Here, one reads in a submap comprising the desired region of whatever type of map is available, usually an MIR map. An appropriate contour level and initial section are selected and the contoured electron density for that section appears on the screen. One then selects the 'add next section' menu item two or three times to create a projection over several sections of the map, since in the projection the molecular boundary is usually more obvious. Selecting the 'trace mask' menu item then allows the user to hand-contour the molecular envelope by directing the cursor tied to a mouse or other pointing device. One then moves to an adjacent section and repeats the process until the complete 3D mask is obtained. To simplify matters and speed up the process, there are 'copy next mask' and 'copy previous mask' menu items allowing one to take advantage of the fact that the mask is a slowly changing function, particularly when near the centre of the molecule. One can use this feature to copy a mask from the previous or following section and apply it to the current section. Up to twelve distinct 3D masks can be selected. Each mask is colour-coded and can be simultaneously displayed superimposed on the contoured electron-density section. Once the mask is completed, the 'make asu' menu item is selected to apply crystallographic symmetry operations to all points within the generated envelope masks. If these operations generate a point also within the envelope masks, the point is flagged in red to indicate that it is redundant, indicating that when tracing the mask, one inadvertently strayed into a symmetry-related molecule. After this check for redundancy, all points within the submap distinct from but related to points within the molecular envelopes by crystal symmetry are flagged in green. This enables one to detect packing contacts and also to ensure that all significant electron density has been assigned to some envelope. Upon completion, the mask is written to a file suitable for use either in averaging, solvent flattening (after expansion by *BLDCEL*), or operator refinement. In cases of 'proper' NC symmetry, it is often desirable to trace the averaging envelope mask in a 'skewed' map, such that one is looking directly down the NC rotation axis. In this case, it is usually very obvious where the NC symmetry breaks down, simplifying identification of the averaging envelope. If the averaging mask is created in a skewed submap, then the batch program *TRNMSK* can be used to transform it so as to correspond to the original, unskewed submap for use in averaging calculations (which do not require skewing).

25.2.1.10.4. *Map averaging*

All averaging calculations are carried out by the program *MAPAVG* (batch), which requires the submap to be averaged along with the envelope masks and NC symmetry operators. A copy of the input submap is made and each grid point in the mask is examined in turn. If the grid point lies within any averaging envelope, then all points related to it by NC symmetry are generated from the operators and examined. If the generated points also lie within the appropriate envelope mask, the electron density there is interpolated, as described earlier, and the density values for all related points are summed. The average value of the electron density is then inserted at the original point in the submap copy. Upon completion, the averaged version of the submap is written to a file and correlation coefficients for regions related by the various NC symmetry operations are output. The averaged submap is then passed to the program *BLDCEL* along with the averaging mask and the original unaveraged *FSFOUR* map from which the submap was created. For all points within the averaging envelope(s), their electron-density values and those at points related by crystallographic symmetry are inserted into the full-cell map, and it is written to a file. This file then contains the NC symmetry averaged electron density expanded to a full-cell map that obeys space-group symmetry. As an option, the averaging mask can also be expanded in *BLDCEL* to a full-cell mask, which could then be used for solvent flattening.

25.2.1.10.4.1. *Single-crystal averaging*

For NC symmetry averaging within a single crystal, the calculations are exactly as described above. One refines the NC symmetry operators with *LSQROT* or *LSQROTGEN*, creates the appropriate envelope mask(s) with *MAPVIEW*, averages with *MAPAVG* and expands the averaged submap to a full cell with *BLDCEL*.

25.2.1.10.4.2. *Multiple-crystal averaging*

If multiple crystal forms are available and one has a source of phase information for each crystal form, then averaging over the independent molecular copies within all crystal forms is possible. In fact, one may also have NC symmetry *within* some of the crystal forms. One can utilize all of this information during averaging by exactly the same process as previously described. For each form, the appropriate envelope mask(s) must be obtained and any internal NC symmetry operators refined, as described earlier. Then operators relating molecules from one crystal form to another must be obtained and refined. The program *LSQROTGEN* can read in multiple submaps, allowing refinement of the additional operators. The program *MAPAVG* accepts submaps from up to six different crystal forms. Averaging over all copies then proceeds exactly as described above, except that prior to averaging, density in all submaps is placed on a common scale, and upon completion averaged submap files are written for each crystal form.

25.2.1.10.5. *Phase combination and extension*

During NC symmetry averaging, phase combination and extension is carried out precisely as described during solvent flattening and negative-density truncation. The only difference is that after generation of each electron-density map, the NC

symmetry averaging is carried out on the appropriate submap region, which is then expanded back to a full-cell map prior to each solvent-flattening calculation.

25.2.1.11. *Automated iterative processing*

The most common iterative processes are carried out by shell scripts or command procedures. These procedures merely direct the flow of map, mask, structure-factor and control-data files between the various programs, while controlling the number of iterations in the process. Generally, one does not have to alter these scripts, although expert users may want to in special circumstances.

25.2.1.11.1. *The DOALL procedure*

A script to carry out a standard solvent-flattening run is provided along with a description of the expected input files, output files and examples. Not surprisingly, this *DOALL* procedure does it all. Execution of the script will create a map from an input 'anchor' set of phases, typically obtained by MIR, SIR, or MAD methods, and will then create a solvent mask from the map after zeroing out density near heavy-atom sites. This solvent mask is used in four cycles of solvent flattening, combining the map-inverted phase information with the anchor phases. A new solvent mask is then generated, starting from a map produced with the phases after the first four cycles. Four cycles of solvent flattening using the second solvent mask are then carried out, restarting from the original map and combining with the anchor phases. These phases are then used to compute a new map from which a third solvent mask is built. The third mask is then used for eight cycles of solvent flattening, again restarting with the original map and combining with the anchor phases. Supplied in the script, but commented out, are instructions to carry out an arbitrary number of additional phase extension cycles, and then an arbitrary number of phase and amplitude extension cycles, all using the third solvent mask. The combined phases and distribution coefficients are written to a file after all cycles with a given mask are completed.

25.2.1.11.2. *The EXTNDAVG and EXTNDAVG_MC procedures*

Additional scripts are provided to carry out phase extension and/or NC symmetry averaging iterations. These scripts are executed after completion of a normal solvent-flattening run with the *DOALL* procedure. With the *EXTNDAVG* script, an input number of additional solvent flattening and/or phase combination cycles are carried out, and phase (and possibly amplitude) extension may be requested. Initial and final *d* spacings are input to the program *SLOEXT* (batch) along with the number of map modification or phase combination iterations per step, where each step represents the extension by one reciprocal-lattice point in each direction if phase extension is to be carried out. The calculations proceed where the *DOALL* script leaves off, starting with a map made from the final phases and using the third solvent mask. If NC symmetry averaging is to be carried out, after each map calculation the appropriate submap is extracted from it and is passed to *MAPAVG* along with the averaging mask. The averaged submap is passed to *BLDCEL*, where it is expanded to a full-cell map, which then is passed to *BNDRY* for solvent flattening. Map inversion and phase combination then proceed normally (although possibly with phase extension). Note that, in general, separate masks are used for solvent flattening and averaging.

The *EXTNDAVG_MC* script carries out the same procedures and options as the *EXTNDAVG* script, except that it is used when carrying out NC symmetry averaging with multiple crystal forms. Starting phase files, anchor phases, solvent masks, averaging masks and control files are provided for each crystal form. For each form, the solvent flattening and phase combination steps are carried out independently with the appropriate data; however, during the averaging step, maps from all crystal forms are involved.

25.2.1.12. *Graphical capabilities*

To facilitate visual evaluation of phasing results and input data, several (mainly interactive) programs are provided within the package. The programs are used to display contoured electron-density or Patterson maps, for interactive editing of solvent or averaging masks, and for visualization of input or difference diffraction data on workstation monitors or terminals. In most instances, hard copies for inclusion in manuscripts are also obtainable. The interactive graphics programs *MAPVIEW*, *PRECESS* and *VIEWPLT* are provided with two versions of each: one for use on Silicon Graphics workstations and the other (indicated by the same program name but ending in _X) for use on any display device supporting the X-Window protocol. The functionality, input and documentation are identical in both versions of each program.

25.2.1.12.1. *Pseudo-precession photographs*

The interactive program *PRECESS* is provided to display diffraction data in the form of pseudo-precession photographs. One can display any zone or step through all zones, with the corresponding intensities mapped to a colour scheme. If a grey scale is selected, the image looks very much like a properly exposed precession photograph taken with Polaroid film. When the cursor is placed near a reciprocal-lattice point, the Miller indices, intensity, standard deviation and *d* spacing are displayed, allowing one to quickly confirm or identify space groups and Laue symmetry. If a scaled file is input containing isomorphous-replacement or anomalous-scattering data, one can display the corresponding intensity differences instead of the native intensities and quickly visualize the distribution of differences to help assess isomorphism.

25.2.1.12.2. *Interactive contouring or mask editing*

The interactive program *MAPVIEW* can be used to examine contoured electron-density or Patterson maps, as well as to examine, create or edit solvent or averaging masks. Either full-cell *FSFOUR* maps or submaps (including skewed submaps) can be used, although only from the former can any arbitrary region be obtained and reordered interactively. The mask creation and editing functions have been described earlier. The program is very useful for Patterson analysis, evaluation of phasing results and to help decide which region is appropriate for isolating a molecule for use in model building. It is usually crucial for construction of averaging masks, but is also useful for examining or editing other masks.

25.2.1.12.3. *Off-line contouring*

While *MAPVIEW* is extremely useful, there are times when it is desirable to have individual plots available either for comparison, stereo viewing of electron density, or incorporation into documents. The program *CTOUR* (batch) handles these functions and accepts an input *FSFOUR* map or submap. The *CTOUR* program can create any number of plot files in a single run, with each consisting of either an individual section, a mono projection, or a stereo projection, with each projection over different multiple sections. If full-cell *FSFOUR* maps are input, any desired region may be selected, whereas if submaps (including skewed maps) are input, the accessible regions are limited by those present in the input map.

25.2.1.12.4. *Generic plot files and drivers*

The plot files created by *CTOUR* are generic in nature and are not directly displayable. One needs a driver program to convert the generic files to the format appropriate for the desired display device. The appropriate drivers for several popular display devices are provided within the package and are described below.

25.2.1.12.4.1. *GL displays*

For display on Silicon Graphics workstations, the interactive program *VIEWPLT* can be used to examine the generic plots created by *CTOUR*. Up to ten plots can be displayed simultaneously. It is particularly useful to display the various contoured Harker sections simultaneously during difference-Patterson interpretation.

25.2.1.12.4.2. *X-Window displays*

For display of *CTOUR* plots on monitors supporting the X-Window protocol, including most workstation monitors and X-terminals, the program *VIEWPLT_X* can be used instead of *VIEWPLT*. The functionality is identical to the GL version.

25.2.1.12.4.3. *PostScript files*

The interactive program *MKPOST* is provided to generate standard PostScript equivalents from the generic plot files produced by *CTOUR*. Multiple plot files can be generated in the same process. The PostScript files can be printed, viewed with a PostScript previewer, or incorporated into other documents.

25.2.1.12.4.4. *Tektronix output*

The interactive program *PLTTEK* can be used to display the generic plots created by *CTOUR* on any device supporting Tektronix 4010 emulation. While slow, this enables visualization of the plots on many 'dumb' terminals.

25.2.1.13. *Auxiliary programs*

In addition to the major programs already described, a number of auxiliary programs (all interactive) are provided in the package to aid the user in porting information to or from external software and to assess phasing methods. These programs are briefly described below.

25.2.1.13.1. *Coordinate conversions*

Within the package, fractional atomic coordinates are used extensively, and the program *PDB_CDS* is provided to convert from PDB (Protein Data Bank) to *PHASES* coordinate files and *vice versa*. The program prompts for input and output file names, the direction of the conversion, chain or residue ranges, and whether to reset occupancies and/or thermal factors to specified values. The coordinate ranges (both fractional and in PDB coordinates) spanned by the model are also listed.

25.2.1.13.2. *NC symmetry operator conversions*

The program *O_TO_SP* is provided to convert NC symmetry operators expressed in terms of a 3×3 rotation matrix and 1×3 translation vector to the *PHASES*-style spherical polar system described earlier. Although originally written to convert the transformation operator as defined in the *O* program (Jones *et al.*, 1991), the procedure works for any rotation or translation operator expressed in this form, provided that the operator is applicable to Cartesian coordinates in Å orthogonalized as in the Protein Data Bank (Bernstein *et al.*, 1977).

25.2.1.13.3. *Binary or formatted file conversions*

For efficiency, structure-factor files used within the package are binary; however, the program *RD*31 is provided to read these binary files and convert them to formatted files that can be examined and possibly edited by the user. The indices, amplitudes, phases, figures of merit, phase probability distribution coefficients, and markers indicating which reflections are centric along with the allowed phase values are thus made readily accessible. A corresponding program, *MK*31*B*, is also provided to reverse the process; it reads the formatted (and possibly edited) versions of the structure-factor files and generates the appropriate binary-file equivalents. Additionally, the program *XPL_PHI* is supplied to convert the binary structure-factor files to a form readable by the *X-PLOR* program (Brünger *et al.*, 1987) in order to facilitate complete model refinement. Phase and figure-of-merit information are also passed to the output file, allowing refinement with phase restraints if desired.

25.2.1.13.4. *Importing phase information*

The program *IMPORT* allows users to 'import' phase information obtained from programs external to the package so it can be used for subsequent calculations within the package. For example, one can use phase and probability distribution information obtained elsewhere to initiate solvent flattening, negative-density truncation and/or NC symmetry averaging within *PHASES*, or simply to generate and display maps with *MAPVIEW* or the other graphics programs. Reflection indices, the observed structure-factor amplitude, figure of merit, phase and phase probability distribution coefficients must be supplied, although free format can be used.

25.2.1.13.5. *Phase set comparisons*

The program *PSTATS* compares phases in two different structure-factor files. It lists mean phase differences as a function of *d* spacing for common reflections. The program is very useful for comparing results from different phasing strategies and for testing new procedures against error-free phases. It can also be used to check for convergence in iterative procedures or to assess the relative contributions of phase sets during phase combination.

25.2.2. *DM/DMMULTI* software for phase improvement by density modification

(K. D. Cowtan, K. Y. J. Zhang and P. Main)

25.2.2.1. *Introduction*

DM is an automated procedure for phase improvement by iterated density modification. It is used to obtain a set of improved phases and figures of merit, using as a starting point the observed diffraction amplitudes and some initial poor estimates for the phases and figures of merit. *DM* improves the phases through an alternate application of two processes: real-space electron-density modification and reciprocal-space phase combination. *DM* can perform solvent flattening, histogram matching, multi-resolution modification, averaging, skeletonization and Sayre refinement, as well as conventional or reflection-omit phase combination. Solvent and averaging masks may be input by the user or calculated automatically. Averaging operators may be refined within the program. Multiple averaging domains may be averaged using different operators.

DMMULTI is a modified version of the *DM* software that can perform density modification simultaneously across multiple crystal forms. The procedure is general, handling an arbitrary number of domains appearing in an arbitrary number of crystal forms. Initial phases may be provided for one or more crystal forms; however, improved phases are calculated in every crystal form.

references