connected line segment, 'auger' removes everything within a marked circle on the screen and 'undo p' recovers from a mistake, back for ten steps. If, for example, side chains are being shown in a focus around the active site, one could prune away those that don't interact at all, and then move the second-shell side chains to a separate list with the word 'off' in its first line. 'Text Editable' (Edit menu) enables writing explanations in the text and caption windows, while the graphics window is still active for reference. 'Save As' (File menu) will save the whole edited kinemage file and reload to show the revised kinemage in its startup view. As well as a bitmap screen capture or files for rendering, a PostScript file can also be written to print out a 2D picture of the current graphics window, either in colour or 'black on white'.

At this stage, a word processor can be used to look at the plain ASCII kinemage file, with its text, its views and the hierarchy of group, subgroup and list display objects in human-readable and clearly identified forms. Lists (*e.g.* @vectorlist {name}) can be of vectors, dots, labels, words, balls, spheres, triangles, or ribbons. Any part of the file can be edited, using its existing format as a guide or looking at another kinemage file that provides a desired template. Among the few operations that currently must be edited outside rather than inside *MAGE* are moving things between different lists or groups (for instance, setting up a new list of just active-site side chains in a different colour and controlled by their own button) and adding 'master' buttons that control object display independent of the group heirarchy (*e.g.* side chains can be turned off and on together for all subunits or models if 'master = {side ch}' is added to the first line of each of those lists). The kinemage should be saved without formatting, as a plain text file.

More complex modifications are possible in *MAGE*, using advanced on-screen editing and construction features from the Edit menu. 'Draw new' activates tools that can add labels, draw hydrogen bonds (with shortened, unselectable lines) and make a variety of geometrical constructs by building out from the original atoms (*e.g.* add a Cβ to a Gly, or draw helix axes and measure their distance and angle). 'Show Object Properties' lets one see, and edit, the names and parameters of the object hierarchy for any point picked, which allows renaming buttons, simplifying the button panel, adding animation, editing labels, or deleting entire display objects. 'Remote Update', on the Tools menu, can call *PREKIN* to set up rotations for the last-picked side chain or a mutation of it, and can then call *PROBE* to update all-atom contacts interactively as the angles are changed. On the kinemage web site (Richardson Laboratory, 2000), Demo5_4a.kin includes an introduction to the drawing tools and Demo5_4b.kin to the format and to editing. Make_kin.txt is a more complete tutorial on the process of constructing kinemages. Mage5_4.txt and Pkin5_4.txt document the features of the *MAGE* and *PREKIN* programs. File KinFmt54.txt (which also constitutes the MIME standard chemical/x-kinemage) is a formal description of the kinemage format for 3D display.

All in all, making a simple kinemage is trivial, but making really good ones for use by others is much like making a good web page. There are tools that make the individual steps easy, but one needs to exercise restraint to keep it simple enough to be both fast and comprehensible, patience to keep looking at the result and modifying it where needed, and judgment about both content and aesthetics.

25.2.8.7. *Software notes*

MAGE and *PREKIN* were written in C for Macintosh, PC, Linux, SGI and other UNIX platforms by David C. Richardson, who also maintains and extends them (with the help of Brent K. Presley for the Windows 95/98/NT port). *PROBE* (in C) and *REDUCE* (in C++) were written by J. Michael Word for SGI UNIX, Linux and PC Windows, but can be compiled on other platforms. The contact-dot additions to *O* and *XtalView* were written by Simon C. Lovell, J. Michael Word and Duncan E. McRee. For the modified *XtalView* (version 4.0), see http://www.scripps.edu/pub/dem-web; for *O* scripts and files, see http://origo.imsb.au.dk/~mok/o; the rest of the software, plus source and documentation files, is available free from the kinemage web or ftp site (Richardson Laboratory, 2000).

25.2.9. *XDS* (W. KABSCH)

25.2.9.1. *Functional specification*

The program package *XDS* (Kabsch, 1988*a*,*b*, 1993) has been developed for the reduction of single-crystal diffraction data recorded on a planar detector by the rotation method using monochromatic X-rays. It includes a set of five programs:

(1) *XDS* accepts a sequence of adjacent, non-overlapping rotation images from a variety of imaging plate, CCD and multiwire area detectors and produces a list of corrected integrated intensities of the reflections occurring in the images. The program assumes that each image covers the same positive amount of crystal rotation and that rotation axis, incident beam and crystal intersect at one point, but otherwise imposes no limitations on detector position, or directions of rotation axis and incident beam, or on the oscillation range covered by each image.

(2) *XPLAN* provides information for identifying the optimal rotation range for collecting data. Based on detector position and unit-cell orientation obtained from evaluating one or a few rotation images using *XDS*, it reports the expected completeness of the data by simulating measurements at various rotation ranges specified by the user, thereby taking into account already-measured reflections.

(3) *XSCALE* places several data sets on a common scale, optionally merges them into one or several sets of unique reflections, and reports their completeness and quality of integrated intensities.

(4) *VIEW* displays rotation-data images as well as control images produced by *XDS*. It is used for checking the correctness of data processing and for deriving suitable values for some of the input parameters required by *XDS*. This program was coded in the computer language C by Werner Gebhard at the Max-Planck-Institut für medizinische Forschung in Heidelberg. The other programs are written in Fortran77, with the exception of a few C subroutines provided by Abrahams (1993) for handling compressed images.

(5) *XDSCONV* converts reflection data files as obtained from *XDS* or *XSCALE* into various formats required by software packages for crystal structure determination. Test reflections previously selected for monitoring the progress of structure refinement may be inherited by the new output file, which simplifies the use of new data or switching between different structure-determination packages.

25.2.9.2. *Components of the package*

25.2.9.2.1. *XDS*

XDS is organized into eight steps (major subroutines) which are called in succession by the main program. Information is exchanged between the steps by files (see Table 25.2.9.1), which allows repetition of selected steps with a different set of input parameters without rerunning the whole program. ASCII files can be inspected and modified using a text editor, whereas types DIR and BIN indicate binary random access and unformatted sequential access files, respectively. All files have a fixed name defined by *XDS*, which makes it mandatory to process each data set in a newly created directory. Clearly, one should not run more than one *XDS* job at a time in any given directory. Output files affected by

Table 25.2.9.1. *Information exchange between program steps of XDS*

	Input files		Output files	
Program step	Name	Type	Name	Type
XYCORR	XDS.INP	ASCII	XYCORR.LP	ASCII
			XYCORR.TBL	DIR
			FRAME.pck	BIN
INIT	XDS.INP	ASCII	INIT.LP	ASCII
	XYCORR.TBL	DIR	BKGPIX.TBL	DIR
			BLANK.TBL	DIR
			BKGPIX.IMG	DIR
COLSPOT	XDS.INP	ASCII	COLSPOT.LP	ASCII
	BKGPIX.TBL	DIR	SPOT.XDS	ASCII
	BLANK.TBL	DIR	BKGPIX.IMG	DIR
	XYCORR.TBL	DIR	FRAME.pck	BIN
IDXREF	XDS.INP	ASCII	IDXREF.LP	ASCII
	SPOT.XDS	ASCII	SPOT.XDS	ASCII
			XPARM.XDS	ASCII
COLPROF	XDS.INP	ASCII	COLPROF.LP	ASCII
	XPARM.XDS	ASCII	XREC.XDS	BIN
	BKGPIX.TBL	DIR	BKGPIX.IMG	DIR
	BLANK.TBL	DIR	FRAME.pck	BIN
	XYCORR.TBL	DIR		
PROFIT	XDS.INP	ASCII	PROFIT.LP	ASCII
	XREC.XDS	BIN	PROFIT.HKL	DIR
CORRECT	XDS.INP	ASCII	CORRECT.LP	ASCII
	PROFIT.HKL	DIR	NORMAL.HKL	ASCII
			ANOMAL.HKL	ASCII
			XDS.HKL	DIR
			MISFITS	ASCII
GLOREF	XDS.INP	ASCII	GLOREF.LP	ASCII
	PROFIT.HKL	DIR	GXPARM.XDS	ASCII

rerunning selected steps (see Table 25.2.9.1) should also first be given another name if their original contents are meant to be saved.

Data processing begins by copying an appropriate input file into the new directory. Input-file templates are provided with the *XDS* package for a number of frequently used data-collection facilities. The copied input file must be renamed XDS.INP and edited to provide the correct parameter values for the actual data-collection experiment. All parameters in XDS.INP are named by keywords containing an equal sign as the last character, and many of them will be mentioned here in context to clarify their meaning. Execution of *XDS* (JOB = XDS) invokes each of the eight program steps as described below. Results and diagnostics from each step are saved in files with the extension LP attached to the program step name. These files should always be studied carefully to see whether processing was satisfactory or – in case of failure – to find out what could have gone wrong.

XYCORR calculates a lookup table of additive spatial corrections at each detector pixel and stores it in the file XYCORR.TBL. The data images are often already corrected for geometrical distortions, in which case *XYCORR* produces a table of zeros or – as for spiral read-out imaging plate detectors – computes the small corrections resulting from radial (ROFF=) and tangential (TOFF=) offset errors of the scanner. For some multiwire and CCD detectors that deliver geometrically distorted images, corrections are derived from a calibration image (BRASS_PLATE_IMAGE= file name). This image displays the response to a brass plate containing a regular grid of holes which is mounted in front of the detector and illuminated by an X-ray point source, *e.g.* ^{55}Fe. Clearly, the source must be placed exactly at the location to be occupied by the crystal during the actual data collection, as photons emanating from the calibration source are meant to simulate all possible diffracted beam directions. For visual control using the *VIEW* program, spots that have been located and accepted from the brass-plate image by *XYCORR* are marked in the file FRAME.pck.

Problems: (*a*) A misplaced calibration source leads to an incorrect lookup table, impairing the correct prediction of the observed diffraction pattern in subsequent program steps. (*b*) Underexposure of the calibration image results in an incomplete and unreliable list of calibration spots.

INIT estimates the initial background at each pixel and determines the trusted region of the detector surface. The total background at each pixel is the sum of the detector noise and the X-ray background. The detector noise, saved in the lookup table BLANK.TBL, is determined from a specific image recorded in the absence of X-rays (DARK_CURRENT_IMAGE=) or is assumed to be a constant derived from the mean recorded value in each corner of the data images. A lookup table of the X-ray background, saved on the file BKGPIX.TBL, is obtained from the first few data images by the following two-pass procedure. To exclude diffraction spots in the data image, the minimum of the five values at (x, y), $(x \pm \mathrm{d}x, y)(x, y \pm \mathrm{d}y)$ is used as a lower background estimate at pixel (x, y) in the first pass. In the second pass, the background is taken as the maximum of the lower estimates at these five locations. Ideally, the parameters SPOT_WIDTH_ALONG_X $= 2 * \mathrm{d}x + 1$, SPOT_WIDTH_ALONG_Y $= 2 * \mathrm{d}y + 1$ are chosen to match the extent of a spot. The lookup table is obtained by adding the X-ray background from each image. Shaded regions on the detector (*i.e.* from the beam stop), pixels outside a user-defined circular region (RMAX=) or pixels with an undefined spatial correction value are classified as untrustworthy and marked by -3. The table should be inspected using the *VIEW* program.

Problems: (*a*) The addition of background from too many data images may exceed 262 144 at some pixels, which are removed from the trusted detector region due to internal number overflow. (*b*) Some detectors with insufficient protection from electromagnetic pulses may generate badly spoiled images whose inclusion leads to a completely wrong X-ray background table. These images can be identified in INIT.LP by their unexpected high mean pixel contents, and this step should be repeated with a different set of images.

COLSPOT locates, at most, 500 000 strong diffraction spots occurring in a subset of the data images and saves their centroids on the file SPOT.XDS. Up to ten ranges of contiguous images (SPOT_RANGE=) may be specified explicitly; otherwise, spots are taken from the first few data images, covering a total rotation range of 5°. Spots are located automatically by comparing each pixel value with the mean value and standard deviation of surrounding pixels, as described in Chapter 11.3. A lower threshold for accepting pixels and a minimum required number of such pixels within a spot can be defined in XDS.INP by the parameters MINIMUM_SIGNAL_TO_NOISE_FOR_LOCATING_SPOTS= and MINIMUM_NUMBER_OF_PIXELS_IN_A_SPOT=, respectively.

Problem: Sharp edges like ice rings in the images can lead to an excessive number of pixels erroneously classified as contributing to a diffraction spot which extends over many adjacent images, thereby causing a hash-table overflow. The problem can be avoided by specifying non-adjacent images for spot search.

IDXREF uses the initial parameters describing the diffraction experiment as provided by XDS.INP and the observed centroids of the spots occurring in the file SPOT.XDS to find the orientation, metric and symmetry of the crystal lattice, and it refines all or a

specified subset of these parameters. On return, the complete set of parameters are saved in the file XPARM.XDS, and the original file SPOT.XDS is replaced by a file of identical name – now with indices attached to each observed spot. Spots not belonging to the crystal lattice are given indices 0, 0, 0. *XDS* considers the run successful if at least 70% of the given spots can be explained with reasonable accuracy; otherwise, *XDS* will stop with an error message. Alien spots often arise because of the presence of ice or small satellite crystals, and continuation of data processing may still be meaningful. In this case, *XDS* is called again with an explicit list of the subsequent steps specified in XDS.INP.

Using and understanding the results reported in IDXREF.LP requires a knowledge of the concepts employed by this step, as described in Chapter 11.3. First, a reciprocal-lattice vector, referring to the unrotated crystal, is computed from each observed spot centroid. Differences between any two reciprocal-lattice vectors that are above a specified minimal length (SEPMIN=) are accumulated in a three-dimensional histogram. These difference vectors will form clusters in the histogram, since there are many different pairs of reciprocal-lattice vectors of nearly identical vector difference. The clusters are found as maxima in the smoothed histogram (CLUSTER_RADIUS=), and a basis of three linearly independent cluster vectors is selected that allows all other cluster vectors to be expressed as nearly integral multiples of small magnitude with respect to this basis. The basis vectors and the 60 most populated clusters with attached indices are listed in IDXREF.LP. If many of the indices deviate significantly from integral values, the program is unable to find a reasonable lattice basis and all further processing will be meaningless.

If the space group and cell constants are specified, a reduced cell is derived, and the reciprocal-basis vectors found above are reinterpreted accordingly; otherwise, a reduced cell is determined directly from the reciprocal basis. Parameters of the reduced cell, coordinates of the reciprocal-basis vectors and their indices with respect to the reduced cell are reported.

Based on the orientation and metric of the reduced cell now available, *IDXREF* indexes up to 3000 of the strongest spots by the local-indexing method. This method considers each spot as a node of a tree and identifies the largest subtree of nodes which can be assigned reliable indices. The number of reflections in the ten largest subtrees is reported and usually shows a dominant first tree corresponding to a single lattice, whereas alien spots are found in small subtrees. Reflections in the largest subtree are used for initial refinement of the basis vectors of the reduced cell, the incident beam wave vector and the origin of the detector, which is the point in the detector plane nearest to the crystal. Experience has shown that the detector origin and the direction of the incident beam are often specified with insufficient accuracy, which could easily lead to a misindexing of the reflections by a constant offset. For this reason, *IDXREF* considers alternative choices for the index origin and reports their likelihood for being correct. Parameters controlling the local indexing are INDEX_ERROR=, INDEX_MAGNITUDE=, INDEX_QUALITY= (corresponding to ε, δ, $1-\ell_{\min}$ in Chapter 11.3) and INDEX_ORIGIN= h_0, k_0, l_0, which is added to the indices of all reflections in the tree. After initial refinement based on the reflections in the largest subtree, all spots that can now be indexed are included. Usually, the detector distance and the direction of the rotation axis are not refined, but if the spots were extracted from images covering a large range of total crystal rotation, better results are obtained by including these parameters in the refinement (REFINE=).

The refined metric parameters of the reduced cell are used for testing each of the 44 possible lattice types, as described in Chapter 11.3. For each lattice type, *IDXREF* reports the likelihood of being correct, the conventional cell parameters and the linear transformation relating original indices to the new indices with respect to the conventional cell. However, no automatic decisions for space-group assignment are made by *XDS*. If the space group and cell constants are provided by the user, the reduced-cell vectors are reinterpreted accordingly; otherwise, data processing continues with the crystal being described by its reduced-cell basis vectors and triclinic symmetry. On completion, when integrated intensities are available, the user chooses any plausible space group according to the rated list of the 44 possible lattice types and repeats only the *CORRECT* and *GLOREF* steps with the appropriate conventional cell parameters and reindexing transformation (see below).

Problems: (*a*) Indices of many difference-vector clusters deviate significantly from integral values. This can be caused by incorrect input parameters, such as rotation axis, oscillation angle or detector position, by a large fraction of alien spots in SPOT.XDS, by placing the detector too close to the crystal, or by inappropriate choice of parameters SEPMIN= and CLUSTER_RADIUS= in densely populated images. (*b*) Indexing and refinement is unsatisfactory despite well indexed difference-vector clusters. This is probably caused by selection of an incorrect index origin, and *IDXREF* should be rerun with plausible alternatives for INDEX_ORIGIN= after a visual check of a data image with the *VIEW* program.

COLPROF extracts the three-dimensional profile of each reflection predicted to occur in the rotation images within the trusted region of the detector surface and saves all profiles on the file XREC.XDS. A scaling factor is determined for each image, derived by comparing its background region (after subtraction of the detector noise) with the current X-ray background table. This table, initially obtained from the file BKGPIX.TBL, is updated by the background from each data image at a rate defined by the input parameter BFRAC=. For visual control, the contents of the updated X-ray background table are saved on the file BKGPIX.IMG at the end of this program step. Information for predicting reflection positions is initially provided by the file XPARM.XDS. These parameters are either kept constant or refined periodically using centroids of the most recently found strong diffraction spots as data reduction proceeds (REFINE=, NUMBER_OF_FRAMES_BETWEEN_REFINEMENT_IN_COLPROF=, NUMBER_OF_REFLECTIONS_USED_FOR_REFINEMENT_IN_COLPROF=, WEAK=).

In order to include all pixels contributing to the intensity of a spot, approximate values describing their extension and form must be specified, as defined in Chapter 11.3. by the parameters $\delta_M, \sigma_M, \delta_D, \sigma_D$. The value for BEAM_DIVERGENCE= δ_D = arctan(spot diameter/detector distance) is found by measuring the diameter of a strong spot in a data image displayed by the *VIEW* program and should include a few adjacent background pixels. The form of a spot is roughly described as a Gaussian and its standard deviation is specified by the parameter BEAM_DIVERGENCE_E.S.D.= σ_D, which is usually about one-sixth to a tenth of δ_D. Similarly, REFLECTING_RANGE= δ_M is the approximate rotation angle required for a strong spot recorded perpendicular to the rotation axis to pass completely through the Ewald sphere. The standard deviation of the intensity distribution is given by the mosaicity REFLECTING_RANGE_E.S.D.= σ_M. Thus, a three-dimensional domain of pixels belonging to each reflection is defined by the above parameters, and the program automatically removes pixels contaminated by neighbouring reflections. It determines and subtracts the background, corrects for spatial distortions, and maps each pixel content into a reflection-specific coordinate system centred on the Ewald sphere (see Chapter 11.3). The form of these profiles is then similar for all reflections, and their mean obtained from superimposition of strong reflections is reported at regular intervals. On return from this step, the data image last processed with all expected spots encircled is saved in the file FRAME.pck for inspection using the *VIEW* program.

Problems: (*a*) Off-centred profiles indicate incorrectly predicted reflection positions by using the parameters provided by the file XPARM.XDS (*i.e.* misindexing by using a wrong origin of the indices), crystal slippage, or change in the incident beam direction. (*b*) Profiles extending to the borders of the box indicate too-small values for BEAM_DIVERGENCE= or REFLECTING_RANGE=. This leads to incorrect integrated intensities because of truncated reflection profiles and unreliable background determination. (*c*) Display of the file FRAME.pck showing spots which are not encircled. If these unexpected reflections are not close to the spindle and are not ice reflections, it is likely that the parameters provided by the file XPARM.XDS are wrong.

PROFIT estimates intensities from the three-dimensional profiles of the reflections stored in the input file XREC.XDS and saves the results in the file PROFIT.HKL. In the first pass, templates are generated by superimposing profiles of fully recorded strong reflections, and all grid points with a value above a minimum percentage of the maximum in the template (CUT=) are defined as elements of the integration domain. To allow for variations of their shape, profile templates are generated from reflections located at nine regions of equal size covering the detector surface and additional sets of nine to cover equally sized batches of images. Standard deviations, REFLECTING_RANGE_E.S.D.= and BEAM_DIVERGENCE_E.S.D.=, observed for each profile template are reported and – in the case of large discrepancies – could be used for rerunning *COLPROF* with better values for these parameters. In the second pass, intensities and their standard deviations are estimated by fitting the reflection profile to its template, as described in Chapter 11.3. Overloaded (OVERLOAD=) or incomplete reflections covering less than a minimum percentage of the template volume (MINPK=) or reflections with unreliable background are excluded from further processing.

Problem: The program stops because there are no strong spots for learning profile templates. It is likely that parameters REFLECTING_RANGE=, BEAM_DIVERGENCE= *etc*., which define the box dimensions, have been incorrectly chosen. After correction, both the *COLPROF* and *PROFIT* step should be repeated.

CORRECT applies Lorentz and polarization correction factors as well as factors that partially compensate for radiation damage and absorption effects to intensities and standard deviations of all reflections found in the file PROFIT.HKL, and saves the results on the files XDS.HKL and either NORMAL.HKL or ANOMAL.HKL (if Friedel's law is broken, as specified by a positive value for the input parameter DELFRM=). These factors are determined from many symmetry-equivalent reflections usually found in the data images such that their integrated intensities become as similar as possible. The residual scatter of these intensities is a more realistic measure of their errors and is used to determine a correction factor for the standard deviations previously estimated from profile fitting. An initial guess for this factor (WFAC1=) is provided in XDS.INP and is used to identify outliers, which are collected in the file MISFITS for separate analysis.

Data quality as a function of resolution is described by the agreement of the intensities of symmetry-related reflections and quantified by the *R* factors $R_{\rm sym}$ and the more robust indicator $R_{\rm meas}$ (Diederichs & Karplus, 1997). These *R* factors as well as the intensities of all reflections with indices of type $h00$, $0k0$ and $00l$ and those expected to be systematically absent are important indicators for identification of the correct space group. Clearly, large *R* factors or many rejected reflections (MISFITS) or large observed intensities for systematically absent reflections suggest that the assumed space group or the indexing is incorrect. It is easy to test other possible space groups (SPACE_GROUP_NUMBER=) by simply repeating the *CORRECT* and *GLOREF* steps after copying the appropriate reindexing transformation (REIDX=) and conventional cell constants (UNIT_CELL_CONSTANTS=) found in the rated table of the 44 possible lattice types in IDXREF.LP to XDS.INP. One should remember, however, that the final choice to be kept should be run last, as *XDS* overwrites earlier versions of the output files.

Another useful feature is the possibility of comparing the new data with those from a previously measured crystal (REFERENCE_DATA_SET= file name). For some space groups, like $P4_2$, possessing an ambiguity in the choice of axes, comparison with the reference data set allows one to identify the consistent solution from the complete set of alternatives already listed in IDXREF.LP together with their required index transformation. Reference data are also quite useful for recognizing misindexing or for testing potential heavy-atom derivatives.

Problems: (*a*) Incomplete data sets may lead to wrong conclusions about the space group, as some of its symmetry operators might not be involved in the *R*-factor calculations. (*b*) Conventional cell parameters, as listed in IDXREF.LP, often violate constraints imposed by the space group and must be edited accordingly after copying to XDS.INP.

GLOREF refines the diffraction parameters by using the observed positions of all strong spots contained in the file PROFIT.HKL. It reports the root-mean-square error between calculated and observed positions along with the refined unit-cell constants. Again, for testing possible space groups, the crystallographer consults the table printed by the *IDXREF* step and selects the appropriate reindexing transformation and starting values for the conventional cell constants. The refined diffraction parameters (after possible reindexing) are saved on the file GXPARM.XDS, which is identical in format to XPARM.XDS. Replacing XPARM.XDS with the new file offers a convenient way for repeating *COLPROF*, now with a better set of parameters.

Problem: *GLOREF* will fail if the crystal slips during data collection.

25.2.9.2.2. *XPLAN*

XPLAN supports the planning of data collection. It is based upon information provided by XPLAN.INP and the input files XPARM.XDS and BKGPIX.TBL, both of which become available on processing a few test images with *XDS*. *XPLAN* estimates the completeness of new reflection data, expected to be collected for each given starting angle and total crystal rotation, and reports the results for a number of selected resolution shells in the file XPLAN.LP. To minimize recollection of data, the name of a file containing already measured reflections can be specified in XPLAN.INP.

Problems: (*a*) Incorrect results may occur for some space groups, *e.g.* $P4_2$, if the unit cell determined by *XDS* from processing a few test images implicates reflection indices inconsistent with those from the already-measured data. The correct cell choice can be found, however, by using the old data as a reference and repeating *CORRECT* and *GLOREF* with the appropriate reindexing transformation, followed by copying GXPARM.XDS to XPARM.XDS. The same applies if *IDXREF* was run for an unknown space group and then reindexed in *CORRECT* and *GLOREF*. (*b*) *XPLAN* ignores potential reflection overlap due to the finite oscillation range covered by each image.

25.2.9.2.3. *XSCALE*

XSCALE accepts one or more files of type XDS.HKL, NORMAL.HKL or ANOMAL.HKL, as obtained from data processing with *XDS* or merged output files from *DENZO* (see Chapter 11.4), determines scaling factors for the reflection intensities, merges symmetry-equivalent observations into a unique

set, and reports data completeness and quality in the file XSCALE.LP. The desired program action is specified in the file XSCALE.INP. It consists of a definition of shells used for analysing the resolution dependency of data quality and completeness, space-group number and cell constants, and one line for each set of input reflections. Each set has a file name, type identifier, a resolution window for accepting data, a weighting factor for the standard deviation of the reflection intensity, a decision constant for accepting Bijvoet pairs, a number controlling the degree of smoothness of the scaling function and an optional file name for specifying the output file.

Problem: All reflections beyond the highest shell specified for analysing the resolution dependency of data quality and completeness will be ignored regardless of the resolution window given for each data set.

25.2.9.2.4. *VIEW*

VIEW is used for visualizing data images as well as control images produced by *XDS*. It responds to navigation commands entered by movements of a mouse, and reports the corresponding image coordinates and their pixel contents upon activation of the mouse buttons. *VIEW* also allows magnification of selected image portions and changes in colour.

Problem: For many detector image formats as well as *XDS* produced images, the true pixel value is stored in a coded form which is interpreted by *VIEW* as a signed integer. Numbers less than −4095 displayed by *VIEW* correspond to large positive pixel values.

25.2.9.2.5. *XDSCONV*

XDSCONV accepts reflection-intensity data files as produced by *XSCALE* or *CORRECT* and converts them into a format required by software packages for structure determination. *XDSCONV* estimates structure-factor moduli based on the assumption that the intensity data set obeys Wilson's distribution and uses a Bayesian approach to statistical inference as described by French & Wilson (1978). For anomalous intensity data, both structure-factor amplitudes F_{hkl} and $F_{\bar{h}\bar{k}\bar{l}}$ are simultaneously estimated from the Bijvoet intensity pair by a method similar to that described by Lewis & Rees (1983) – which accounts for the correlation between I_{hkl} and $I_{\bar{h}\bar{k}\bar{l}}$. The output file generated may inherit test reflections previously used for calculating a free R factor (Brünger, 1992*b*) or may contain new test reflections selected by *XDSCONV*.

25.2.9.3. *Remarks*

XDS is not an interactive program. It communicates with the input file XDS.INP and during the run accepts only a change in specification of the last image to be included in the data set (DATA_RANGE=) – a useful option when processing overlaps with data collection. To prevent the program from overtaking the measurements, a maximum delay should be set (MINUTE=) to be slightly longer than the time for generating the next image.

Experience has shown that the most frequent obstacle in using the package is the indexing and accurate prediction of the reflections occurring in the images. Usually, the problems arise from incorrect specifications of rotation axis, beam direction or detector position and orientation, oscillation range, or wavelength. The occurrence of gross errors can be reduced by using file templates of XDS.INP specifically tailored to the actual experimental set-up which require only small adjustments to the geometrical parameters.

However, even small errors in the specification of the incident beam direction or the detector position may lead to indices which are all offset by one reciprocal-lattice point, particularly if the initial list of diffraction spots was obtained from a few images covering a small range of crystal rotation. For this reason, *IDXREF* tests a few alternatives for the index origin and reports its results, such as the expected coordinates of the incident beam in the image, which can be checked by looking at a data image with the *VIEW* program. The user may then repeat the *IDXREF* step, thereby forcing the program to use a plausible alternative for the index origin.

It is recommended that all program steps are run on a few images to establish whether the indexing is correct and also to find reasonable values describing crystal mosaicity and spot size. Incorrect indexing may be apparent from large values of symmetry R factors or from comparison with a reference data set reported in the *CORRECT* step. Also, looking at the file FRAME.pck with the *VIEW* program should show the last data image processed with most of the observed diffraction spots circled. More accurate estimates of the parameters describing spot dimensions are reported in PROFIT.LP and should be used for updating these values in XDS.INP before starting data processing for all images.

Refinement of parameters controlling the predicted position of spots is carried out in the *IDXREF*, *COLPROF* and *GLOREF* step, which allows the user to adopt a variety of strategies. If all data images are available, spots should be extracted by *COLSPOT* from images equally distributed in the data set. If *IDXREF* is able to explain most of the spots, the refined parameters will be sufficiently accurate for the complete data processing, and refinements in the *COLPROF* step are unnecessary. In other cases, if processing overlaps with data collection or the first strategy was unsuccessful, *IDXREF* is based on spots extracted from the first few images and provides an initial parameter set which is periodically refined during *COLPROF*. This allows correction for slow crystal slippage or minor changes of the incident beam direction. Finally, if refinement in *GLOREF* was successful, the new values may be used to repeat *COLPROF* (without parameter refinements) and subsequent steps.

25.2.10. Macromolecular applications of *SHELX*

(G. M. SHELDRICK)

25.2.10.1. *Historical introduction to SHELX*

The first version of *SHELX* was written around 1970 for the solution and refinement of small-molecule and inorganic structures. In the meantime, it has become widely distributed and is used at some stage in well over half of current crystal structure determinations. Since small-molecule direct methods and Patterson interpretation algorithms can be used to locate a small number of heavy atoms or anomalous scatterers, the structure-solving program *SHELXS* has been used by macromolecular crystallographers for a number of years. More recently, improvements in cryocrystallography, area detectors and synchrotron data collection have led to a rapid increase in the number of high-resolution (<2 Å) macromolecular data sets. The enormous increase in available computer power makes it feasible to refine these structures using algorithms incorporated in *SHELXL* that were initially designed for small molecules. These algorithms are generally slower but make fewer approximations [*e.g.* conventional structure-factor summation rather than fast Fourier transform (FFT)] and include features, such as anisotropic refinement, modelling of complicated disorder and twinning, estimation of standard uncertainties by inverting the normal matrix *etc.*, that are routine in small-moiety crystallography but, for reasons of efficiency, are not widely implemented in programs written for macromolecular structure refinement. This account will be restricted to features in *SHELX* of potential interest to macromolecular crystallographers.

references